Studies in Fuzziness and Soft Computing

Alireza Sadeghian
Jerry M. Mendel
Hooman Tahayori *Editors*

Advances in Type-2 Fuzzy Sets and Systems

Theory and Applications

Springer

Studies in Fuzziness and Soft Computing

Volume 301

For further volumes:
http://www.springer.com/series/2941

Alireza Sadeghian · Jerry M. Mendel
Hooman Tahayori
Editors

Advances in Type-2 Fuzzy Sets and Systems

Theory and Applications

Springer

Editors
Alireza Sadeghian
Hooman Tahayori
Department of Computer Science
Ryerson University
Toronto, ON
Canada

Jerry M. Mendel
Signal & Image Processing Institute
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA
USA

ISSN 1434-9922 ISSN 1860-0808 (electronic)
ISBN 978-1-4614-6665-9 ISBN 978-1-4614-6666-6 (eBook)
DOI 10.1007/978-1-4614-6666-6
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013938030

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

In 1975, Zadeh proposed Type-2 Fuzzy Sets (T2 FS) as an extension to the previously introduced ordinary fuzzy sets (now called type-1 fuzzy sets). Type-2 fuzzy sets have the ability to capture the uncertainty about membership functions of fuzzy sets through fuzzification of the membership function of type-1 fuzzy sets. Instead of using a single crisp number from the unit interval [0, 1] as the membership value, as is done in a type-1 fuzzy set, in a T2 FS one or more crisp numbers are used as membership values, and with different strengths. More precisely, a membership grade in a T2 FS is a type-1 fuzzy set; this introduces a new third dimension into a fuzzy set which provides more degrees of freedom for handling uncertainties. Unfortunately, practitioners are still cautious to put general T2 FSs to real use due to their computational complexity. Consequently, there has been extensive research toward simplification of the concepts of and operations for T2 FSs, so that Interval Type-2 Fuzzy Sets (IT2 FSs) are often the preferred method of choice.

As a special variation of a general T2 FS, an IT2 FS uses a subinterval of [0, 1] as its membership value. This is in contrast to the membership grades in T2 FSs that are type-1 fuzzy sets. Simplicity of the concept of IT2 FSs in comparison with general T2 FSs, together with the affordable complexity of their operations, has made IT2 FSs a widely used framework for implementation of fuzzy systems.

There has been a recent steady increase of attention and interest in T2 FS theory from the research community. As of 1999 (when intensive research into T2 FSs began), less than 40 publications had anything to do with T2 FSs or logic. As of 2012, there are thousands of articles that have something to do with T2 FSs or logic. Quite a change in less than 15 years.

On one hand, there have been various studies on T2 FS theories with the objectives of providing a uniform set of definitions and terms, a simple introduction of concepts, justification of their existence, and development of efficient algorithms for performing basic operations. On the other hand, there are endeavors related to the applications of T2 FSs. This book is intended to explore recent developments in the theoretical foundations and novel applications of general and

IT2 FSs and systems. Leading researchers in the field of T2 FSs have participated in the preparation of this book through contribution of their most important and recent achievements in theory and applications of T2 FSs. The chapters cover novel theoretical aspects of T2 FSs, methods for generating their membership functions, and promising applications. This book is organized in three parts.

Part I is dedicated to the theoretical foundations of T2 FSs and is composed of eight chapters. In chapter "Interval Type-2 Fuzzy Logic Systems and Perceptual Computers: Their Similarities and Differences", Mendel, compares Interval type-2 fuzzy logic systems and Perceptual Computers and highlights their similarities and differences. By focusing on inputs and membership functions, fuzzifiers versus encoders, rules versus Computing with Words (CWW) engines, inference versus output of CWW engine, output processing versus decoder, and outputs versus recommendation plus data, this chapter shows that the differences outnumber the similarities. In chapter "A Survey of Continuous Karnik–Mendel Algorithms and Their Generalizations", Liu summarizes the extensions of the continuous Karnik–Mendel Algorithms in type-2 fuzzy logic. It provides a general framework for the analysis and design of the Karnik–Mendel algorithms with numerical analysis. In chapter "Two Differences Between Interval Type-2 and Type-1 Fuzzy Logic Controllers: Adaptiveness and Novelty", Wu explores the differences between interval type-2 and type-1 fuzzy logic controllers. This chapter shows that adaptiveness and novelty are two fundamental differences between interval type-2 and type-1 fuzzy logic controllers. In chapter "Interval Type-2 Fuzzy Markov Chains" Figueroa-García presents a framework to use IT2 FSs in Markov chains analysis. This is useful for handling multiple experts' opinions and perceptions, multiple definitions of type-1 fuzzy Markov chains, and uncertain type-1 fuzzy sets. In chapter "zSlices Based General Type-2 Fuzzy Sets and Systems", Wagner and Hagras provide a concise introduction to zSlices based general T2 FSs and their associated set-theoretic operations. In chapter "Geometric Type-2 Fuzzy Sets", Coupland and John give a review and technical overview of the geometric representation of a T2 FS and explore logical operators used to manipulate this representation. In chapter "Type-2 Fuzzy Sets and Bichains", Harding, Walker and Walker study the variety generated by the truth value algebra of T2 FSs. They identify weakly projective bichains for the variety generated by the truth value algebra of T2 FSs with only its two semilattice operations in its type. In chapter "Type-2 Fuzzy Sets and Conceptual Spaces", Aisbett and Rickard extend the conceptual space theory to incorporate T2 FS structures. They study the usefulness of directional overlap (subsethood) as a metric-free notion of similarity. Moreover, they relate the theory of conceptual spaces to conventional multivariate classification and CWW and illustrate its application to land use assessment tasks.

Chapters in Part II, discuss different methodologies for generating membership functions of interval and general T2 FSs. In chapter "Modeling Complex Concepts with Type-2 Fuzzy Sets: The Case of User Satisfaction of Online

Services", Moharrer, Tahayori and Sadeghian propose a two-phase methodology for generating membership functions of general T2 FSs that model complex concepts. As a case study, they extensively discuss modeling of human perceptions of the linguistic terms that are used in evaluating online satisfaction. The chapter is of importance from at least two points of view. First, a decompositional method for implicit calculation of type-1 fuzzy set models of an individual's perception of a complex concept is discussed. Second, a fuzzy approach to the representation of uncertainty in measurement is adopted for constructing the membership functions of general T2 FSs. In chapter "Construction of Interval Type-2 Fuzzy Sets From Fuzzy Sets: Methods and applications", Pagola et al. present three different methods to construct IT2 FS from type-1 fuzzy sets so that the footprint of uncertainties of the IT2 FSs adapt to the model's uncertainty. In chapter "Interval Type-2 Fuzzy Membership Function Generation Methods for Representing Sample Data", Rhee and Choi discuss three methods based on heuristics, histograms, and Interval Type-2 Fuzzy C-Means clustering for automatic generation of interval type-2 fuzzy membership functions from sample data.

Finally, chapters in Part III introduce novel application of T2 FSs. In chapter "Type-2 Fuzzy Logic in Image Analysis and Pattern Recognition", Melin and Castillo show experimental results for several edge detectors that are used to preprocess the same image sets. By way of experiments, they find the better edge detector that can be used to improve the training data of a neural network for an image recognition system. In chapter "Reliable Tool Life Estimation with Multiple Acoustic Emission Signal Feature Selection and Integration Based on Type-2 Fuzzy Logic", Ren, Baron, Balazinski, and Jemielniak present a type-2 fuzzy tool life estimation system. In their proposed system, type-2 fuzzy analysis is used as a powerful tool to model acoustic emission signal features, and also as a very good estimator for the related ambiguities and uncertainties. In chapter "A Review of Cluster Validation with an Example of Type-2 Fuzzy Application in R", Ozkan and Türkşen explain how interval valued type 2 fuzziness can be used to develop a new cluster validation procedure. Their approach identifies the number of clusters based on the stability of cluster centers with respect to the level of fuzziness. In chapter "Type-2 Fuzzy Set and Fuzzy Ontology for Diet Application", Lee, Wang, and Hsu provide a T2 FS and fuzzy ontology for a diet application. They use a type-2 fuzzy markup language to describe the related knowledge base and rule base.

This book outlines notable achievements in the realm of T2 FS to date. The editors hope the materials covered in this book, provided by the leading scholars in the field, motivate and accelerate future progress. Of course, there are still many theoretical and applied issues that need to be addressed before the full potential of Type-2 Fuzzy Systems is realized. The editors encourage the readers to participate in research opportunities that are associated with T2 FSs, e.g., to investigate and demonstrate the applicability, effectiveness, and potential advantages of T2 FSs over type-1 fuzzy sets in a wide range of complex real world problems.

The editors would also like to express their sincere thanks to the distinguished authors for their contributions. The editors would also like to acknowledge the invaluable, continuous assistance, and advice from the Springer editorial team, Brett Kurzman, Elizabeth Dougherty, and Rebecca Hytowitz.

Contents

Part I
Theoretical Foundations

Interval Type-2 Fuzzy Logic Systems and Perceptual Computers: Their Similarities and Differences

Jerry M. Mendel

Abstract In this chapter, we compare the interval type-2 fuzzy logic system and perceptual computer, so as to eliminate confusion among researchers about whether or not there really are differences between them. We show that there are many more differences than similarities between them by focusing on the following six issues: inputs and membership functions, fuzzifier versus encoder, rules versus computing with words (CWW) engines, inference versus output of CWW engine, output processing versus decoder, and outputs versus recommendation plus data.

1 Introduction

This chapter compares two seemingly similar-looking systems that use interval type-2 fuzzy sets[1] (IT2 FSs), an *interval type-2 fuzzy logic system* (IT2 FLS) and a *perceptual computer* (Per-C). We do this because there may be some confusion among researchers as to whether or not there really are differences between the two. We shall demonstrate that there are many more differences than similarities between the two. Our approach will be to focus on the generic architectures of the IT2 FLS and Per-C and six associated issues.

To begin we provide the block diagrams for both the IT2 FLS and Per-C, in Figs. 1 and 2, respectively.

[1] A *type-2 fuzzy set* can be thought of as a type-1 fuzzy set on steroids. Its membership function no longer has a single value at each value of the primary variable, but instead is a blurred version of that function, i.e., at each value of the primary variable the membership is itself a function, called a *secondary membership function* (MF). When the secondary MF is a constant equal to 1,

J. M. Mendel (✉)
Signal and Image Processing Institute, Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2564, USA
e-mail: mendel@sipi.usc.edu

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*, Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_1,

An IT2 FLS (e.g., [1–5]) is the extension of a T1 FLS (e.g., [4, 6–8]) from T1 FSs to IT2 FSs. For an IT2 FLS, the important issues are about its inputs and membership functions, fuzzifier, rules, inference, output processing, and outputs.

A perceptual computer [9–13] is one implementation of Zadeh's paradigm of computing with words (CWW) [14, 15], an implementation that focuses on the broad class of applications that assist people in making subjective judgments. For the perceptual computer, the important issues are about its inputs and membership functions, encoder, CWW engine, output of CWW engine, decoder, and recommendation plus data.

These issues, which will be compared one to one for the two systems, are discussed in the rest of this chapter.

First, however, we wish to remind the reader that an IT2 FLS has been and continues to be applied to *function approximation problems*, e.g., fuzzy logic control (e.g., [1, 16, 17]), signal processing (e.g., [1, 4]), and rule-based classification [4]. In all of these applications, it is numerical values of the output of the FLS that are used and great comfort is taken by the universal approximation property (e.g., [18]) of the FLS (proven for a T1 FLS, yet to be proved for an IT2 FLS, but believed to be true for it, since an IT2 FLS reduces to a T1 FLS when all sources of MF uncertainties disappear). On the other hand, the Per-C has been and continues to be applied to CWW problems, e.g., investment advising, social judgments, distributed decision making and hierarchical, and distributed decision making [13]. In all of these applications, it is linguistic recommendations plus data at the output of the Per-C that are used and great comfort is taken by the ability to interact with the Per-C using words.

2 Inputs and Membership Functions

For an IT2 FLS, the inputs are numbers. How they are modeled is the subject of our next section. For the Per-C, the inputs are a mixture of numbers, uniformly weighted intervals of numbers, nonuniformly weighted intervals of numbers (T1 FSs), or words (IT2 FSs); generally, it will not be numbers alone.

For both an IT2 FLS and the Per-C, the fuzzy sets that are used by rules are modeled using IT2 FSs. It is the order in which these FSs are obtained that is different.

Footnote 1 (continued)
the type-2 fuzzy set is called an *interval type-2 fuzzy set* or an *interval-valued fuzzy set*; otherwise,it is called a *general type-2 fuzzy set*. The MF of a T2 FS is three-dimensional, with *x-axis* called the *primary variable*, *y-axis* called the *secondary variable* (or *primary membership*), and *z-axis* called the *MF value* (or *secondary MF value*). A *vertical slice* is a plane that is parallel to the MF-value *z*-axis. The *footprint of uncertainty* (*FOU*) of a T2 FS lies on the *x–y plane* (i.e., the primary and secondary variable plane) and includes all points on that plane for which the MF value is nonzero; it is the 2D-domain on which sit the secondary membership values. The FOU can be completely covered by T1 FSs that are called *embedded T1 FSs*.

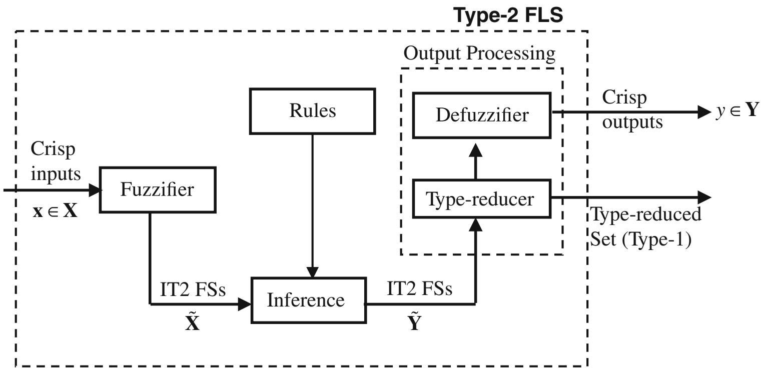


Fig. 1 Interval type-2 fuzzy logic system [4]

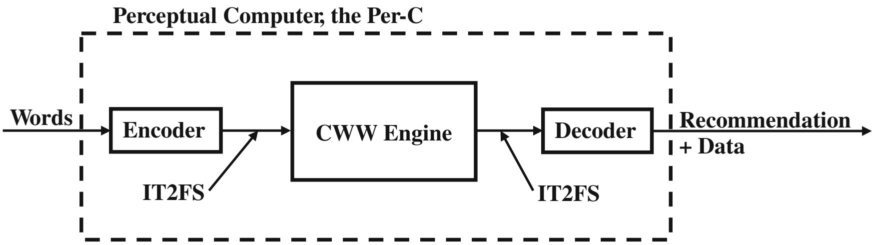


Fig. 2 Perceptual computer [13]

In an IT2 FLS, numerical domains come first; they are then partitioned into overlapping intervals of numbers after which IT2 FSs are assigned to them. It does not matter what these fuzzy sets are called, because they are only used within computer programs that ultimately provide numerical outputs for the FLS. An exception to this is the FLSs where it is also important to interpret the rules; however, for the most part, interpretability is still not so important in most real-world applications of FLSs (e.g., fuzzy-logic control), although it is becoming more important.

In the Per-C words come first, because for each application (A) the very first step in designing a Per-C is to create the *Codebook* that will be used both to design the CWW Engine and decoders. A codebook for an application is the collection of pairs of word and the IT2 FS model for the word, i. e.,

$$\text{Codebook} = \left\{(\tilde{W}_i, FOU(\tilde{W}_i)), i = 1, \ldots, N_A\right\} \tag{1}$$

The words in the codebook must mean something to the end-user; hence, for the Per-C the linguistic labels of the T2 FSs are very important.

3 Fuzzifier Versus Encoder

For an IT2 FS, there can be three kinds of fuzzification [4]:

(1) *Singleton fuzzification* in which numbers are modeled as type-0 fuzzy sets, i.e., they are considered to be perfect. This is by far the most popular kind of fuzzification today, because it leads to enormous simplifications of the inference process.
(2) *Nonsingleton type-1 fuzzification* in which each measured number is modeled as a T1 FS. This kind of fuzzification is more realistic than singleton fuzzification when measurements are corrupted by stationary additive measurement noise.
(3) *Nonsingleton interval type-2 fuzzification* in which each measured number is modeled as an IT2 FS. This kind of fuzzification is more realistic than nonsingleton type-1 fuzzification when measurements are corrupted by nonstationary additive measurement noise.

Inferencing for both nonsingleton T1 and IT2 fuzzifications is more difficult than it is for singleton fuzzification; hence, such fuzzifications are rarely used, although lately nonsingleton type-1 fuzzification is becoming more popular primarily, because people are using nongradient-based optimization procedures (e.g., PSO [19] or QPSO [20, 21]) to optimize FOU parameters during the designs of IT2 FLSs (computing derivatives for such fuzzifications are very complicated [22]).

The encoder of the Per-C models words as IT2 FSs and there is no choice about this (maybe in the future, people will model words using general T2 FSs, but to-date this is not being done). Why? Because *words mean different things to different people*, and the two kinds of uncertainties associated with a word cannot be modeled using T1 FSs [11, 23, 24]. Those uncertainties are: *intra-uncertainty*—the uncertainty that an individual has about a word—and *inter-uncertainty*—the uncertainty that a group of subjects has about the word.

The encoder maps words into IT2 FSs. It uses interval end-point data that are collected from a group of subjects. The subjects are asked a question like: *On a scale of 0–10 where would you locate the end-points of an interval that you associate with word W*? The interval approach [13, 25] [or its enhanced version, the enhanced interval approach (EIA) [26]] maps the interval data into an FOU, and it does not decide what kind of an FOU to choose ahead of time. Instead, it includes a classification step that does this based on the data that are collected from the group of subjects—the data speaks! The result is either a left shoulder, interior, or right shoulder FOU (Fig. 3); but none of these FOUS are usually symmetrical. The IA (EIA) transfers the uncertainties from each subject as well as the group of subjects into the word's FOU.

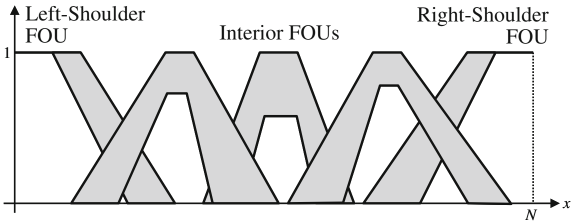


Fig. 3 *Left shoulder*, *right shoulder* and *interior FOUs*, all of whose LMFs and UMFs are piecewise linear [25]

4 Rules Versus CWW Engines

For an IT2 FLS, rules may be obtained from domain experts, extracted from data, or postulated by the designer and then optimized during a training/tuning procedure. Rules are independent of the kind of FSs that are used to model their antecedents and consequent, i.e. a rule is a rule is a rule… . When at least one antecedent or consequent is modeled using an IT2 FS, the resulting FLS is an IT2 FLS.

There are two kinds of rules in an IT2 FLS, *Mamdani* and *TSK*. In a Mamdani rule [27], the rule's consequent is a FS, whereas in the TSK rule its consequent is a linear combination of linear or nonlinear functions of inputs or its FSs.[2] Mamdani rules for a T1 FLS have the structure (e.g., [4, 6, 7])

$$R^l:\ \text{IF } x_1 \text{ is } F_1^l \text{ and } \cdots \text{ and } x_p \text{ is } F_p^l,\ \text{THEN } y \text{ is } G^l, l = 1,\ldots,M;$$

Mamdani rules for an IT2 FLS have the structure (e.g., [3, 4])

$$R^l:\ \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l,\ \text{THEN } y \text{ is } \tilde{G}^l, l = 1,\ldots,M.$$

TSK rules for a T1 FLS have the structure (e.g., [4, 28, 29])

$$R^l: \text{IF } x_1 \text{ is } F_1^l \text{ and } \cdots \text{ and } x_p \text{ is } F_p^l,\ \text{THEN } y^l(\mathbf{x}) = c_0^l + c_1^l x_1 + c_2^l x_2 + \cdots + c_p^l x_p,\quad l = 1,\ldots,M,$$

TSK rules for an IT2 FLS have the structure [4]

$$R^l: \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l,\ \text{THEN } Y^l = C_0^l + C_1^l x_1 + C_2^l x_2 + \cdots + C_p^l x_p,\quad l = 1,\ldots,M,$$

In the Per-C, there can be different kinds of CWW engines. To date, there are two such engines, namely *if–then rules* and *novel weighted averages* (NWA).

[2] TSK rules are also available in which their consequents are dynamical systems, but such rules are outside of the scope of this chapter.

The rules in the Per-C can only have the structure of Mamdani rules because in CWW rules the consequents are words; however, the words must be in the codebook (note that there is no codebook for an IT2 FLS).

A NWA is a weighted average [13] in which at least one weight ($\tilde{W}_i$) or signal ($\tilde{X}_i$) that is being averaged is not just a number. When at least one of them is a uniformly weighted interval of numbers, then the NWA is called an *interval weighted average* (IWA). When at least one of them is a nonuniformly weighted interval of numbers, then the NWA is called a *fuzzy weighted average* (FWA) (e.g., [30]). When at least one of them is a word that is modeled by an IT2 FS, then the NWA is called *linguistic weighted average* (LWA) [31, 32]. The FWA is computed by using IWAs, and the LWA is computed by using FWAs.

A NWA can be *expressed* as:

$$\tilde{Y}_{NWA} = \frac{\sum_{i=1}^{n} \tilde{X}_i \tilde{W}_i}{\sum_{i=1}^{n} \tilde{W}_i} \tag{2}$$

We call this an "expressive equation," because (2) does not mean that the NWA is computed by multiplying, adding and dividing IT2 FSs. How to compute $\tilde{Y}_{NWA}$ is briefly described in Sect. 5.

A recently published *linguistic weighted power mean* (LWPM) [33, 34] generalizes NWAs to the following expressive structure:

$$\tilde{Y}_{LWPM} = \lim_{q \to r} \left(\frac{\sum_{i=1}^{n} \tilde{X}_i^q \tilde{W}_i}{\sum_{i=1}^{n} \tilde{W}_i} \right)^{1/q} \tag{3}$$

According to Rickard et al. [34], as r ranges over the real line, $\tilde{Y}_{LWPM}$ ranges from logical conjunction of the inputs, $\tilde{X}_1 \wedge \ldots \wedge \tilde{X}_n$ (in the limit, as $r \to -\infty$), to logical disjunction of the inputs, $\tilde{X}_1 \vee \ldots \vee \tilde{X}_n$ (in the limit, as $r \to \infty$). $\tilde{Y}_{LWPM}$ is an *orand* operator that can be computed by using the KM algorithms but modified to the LWPM, as explained in [33, 34]. When $r = 1$ it reduces to a NWA.

5 Inference Versus Output of CWW Engine

There are two main kinds of inference procedures for an IT2 FLS, Mamdani, and TSK. Both can be thought of as a two-step procedure: (1) Obtain a firing interval through activating the rule's antecedents by means of the inputs to the FLS, and (2) Blend the firing intervals from the fired rules with each rule's consequent.

In *Mamdani inferencing* one uses the extended sup-star composition [4] to formally compute the firing interval. For an IT2 FLS in which singleton fuzzification is used, the firing interval only involves using the lower and upper MFs for each of the antecedent's IT2 FS, i.e., the lower value of the firing interval is computed as the t-norm between the lower MFs of all of the rule's antecedent MFs, and the upper value of the firing interval is computed as the t-norm between the upper MFs of all of the rule's antecedent MFs. Usually, minimum or product t-norms are used. An example is given in Fig. 4.

Fired rule outputs may be combined or not depending upon the kind of output processing that is used. If they are combined, then this is done using the union operation, the result being one composite IT2 FS (see Fig. 5). It they are not combined, then each of the firing intervals as well as some information about the IT2 consequent FS of each fired rule is sent to output processing, as will be discussed in the next section.

In *TSK inferencing*, one also computes the firing interval as in Mamdani inferencing, but now this is done formulaically, i.e., there is no rigorous justification for doing this. The firing intervals are then usually used to compute a weighed average between each fired rule's consequent expression, in which the firing intervals act as the weights. Sometimes the firing intervals are used only to compute an unnormalized linear combination of each fired rule's consequent expression [4].

Regardless of which kind of inference is used in an IT2 FLS, there is no constraint on it that the resulting IT2 FSs have to resemble the FOUs in a codebook, because, as mentioned above, there is no codebook for an IT2 FLS. The same is not true for the output of the CWW engine of the Per-C.

Because *words must also mean similar things to different people* (or else people will not be communicating effectively), we believe that the output FOUs from the CWW engine must resemble the words that are in the codebook. This is a new kind of constraint for a fuzzy system, but it is one that to date can only be checked after the fact, i.e., it has yet to be used as a constraint during the design of a CWW

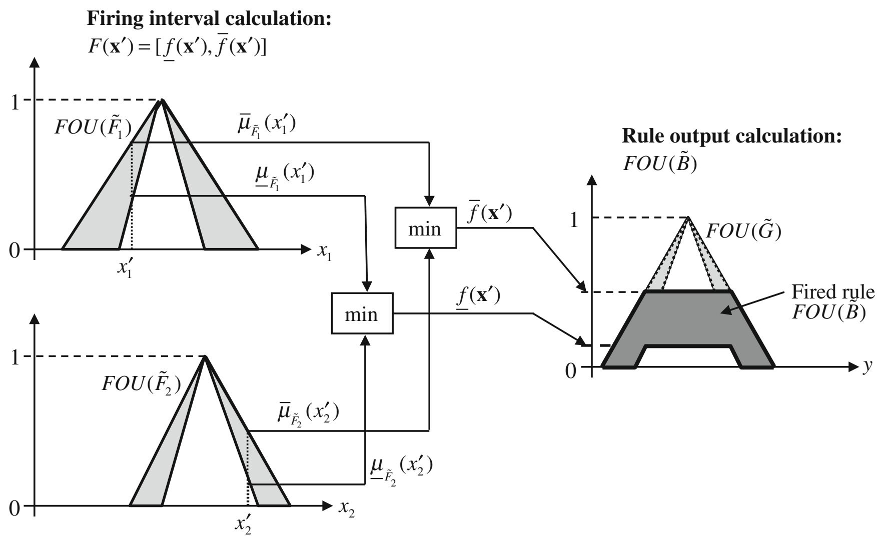


Fig. 4 IT2 FLS inference: from firing interval to fired-rule output FOU [5]

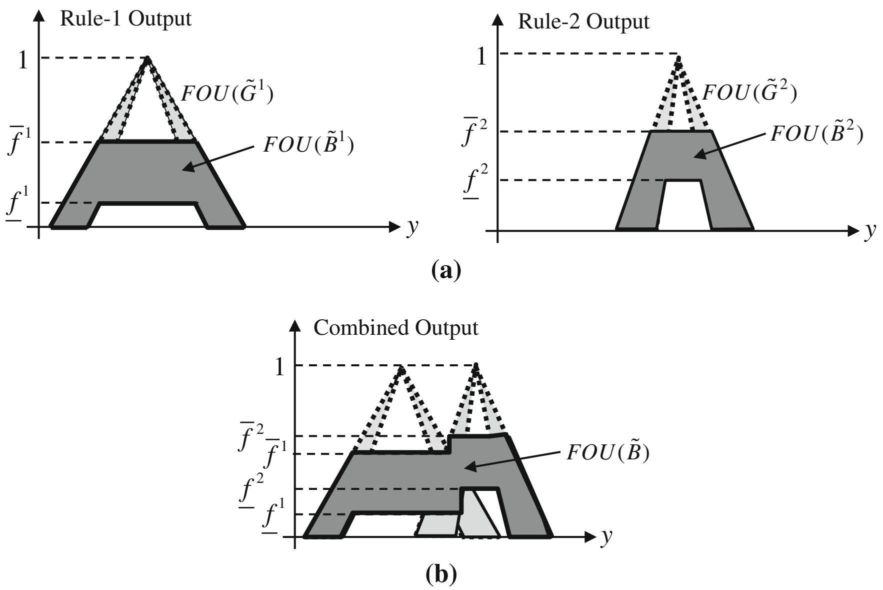


Fig. 5 Pictorial descriptions of (**a**) fired-rule output FOUs for two fired rules, and (**b**) combined fired output FOU for the two fired-rules in (**a**) [13]

engine. Perhaps, in the future, researchers will be able to invoke this constraint at the front end of such a design.

When the Per-C engine is a collection of *if–then rules*, then, as for an IT2 FLS, we: (1) Obtain a firing interval [35] or firing level [36] through activating the rule's antecedents by means of the inputs to the FLS, and (2) Blend the firing intervals or firing levels from the fired rules with each rule's consequent. Because of the constraint that the output FOUs from the if–then rules must resemble the words that are in the codebook, Step 2 is carried out differently from an IT2 FLS. It is performed by means of a special LWA. Steps 1 and 2 together are called *Perceptual Reasoning* [13, 35, 36].

Although it is possible to compute a firing interval for the first step, as in an IT2 FLS [35], we have found that the resulting FOU obtained from the second step does not resemble the FOUs of the words in the codebook as well as when a Jaccard similarity measure is used in the first step to compute a firing level. Because the Jaccard similarity measure for two IT2 FSs is so important to the Per-C, we provide its formula next [13, Chap. 4], [37]:

$$sm_J(\tilde{A},\tilde{B}) = \frac{\sum_{i=1}^{N}\min(\bar{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{B}}(x_i)) + \sum_{i=1}^{N}\min(\underline{\mu}_{\tilde{A}}(x_i),\underline{\mu}_{\tilde{B}}(x_i))}{\sum_{i=1}^{N}\max(\bar{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{B}}(x_i)) + \sum_{i=1}^{N}\max(\underline{\mu}_{\tilde{A}}(x_i),\underline{\mu}_{\tilde{B}}(x_i))} \quad (4)$$

Note that $sm_J(\tilde{A},\tilde{B}) \in [0,1]$.

Table 1 EKM Algorithms for computing the centroid end-points of an IT2 FS, $\tilde{A}$. Note that $x_1 \le x_2 \le \ldots \le x_N$ [13, 45]

Step	EKM algorithm for c_l $c_l = \min_{\forall \theta_i \in \left[\underline{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{A}}(x_i)\right]} \left(\sum_{i=1}^{N} x_i \theta_i \Big/ \sum_{i=1}^{N} \theta_i \right)$	EKM algorithm for c_r $c_r = \max_{\forall \theta_i \in \left[\underline{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{A}}(x_i)\right]} \left(\sum_{i=1}^{N} x_i \theta_i \Big/ \sum_{i=1}^{N} \theta_i \right)$
1	Set $k = [N/2.4]$ (the nearest integer to $N/2.4$) and compute: $a = \sum_{i=1}^{k} x_i \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)$ $b = \sum_{i=1}^{k} \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)$	Set $k = [N/1.7]$ (the nearest integer to $N/1.7$) and compute: $a = \sum_{i=1}^{k} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} x_i \bar{\mu}_{\tilde{A}}(x_i)$ $b = \sum_{i=1}^{k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \bar{\mu}_{\tilde{A}}(x_i)$
	Compute $c' = a/b$	
2	Find $k' \in [1, N-1]$ such that $x_{k'} \le c' \le x_{k'+1}$	
3	Check if $k' = k$. If yes, stop and set $c' = c_l$, and $k = L$. If no, go to Step 4	Check if $k' = k$. If yes, stop and set $c' = c_r$, and $k = R$. If no, go to Step 4
4	Compute $s = sign(k' - k)$ and: $a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} x_i \left[\bar{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)\right]$ $b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} \left[\bar{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)\right]$	Compute $s = sign(k' - k)$ and: $a' = a - s \sum_{i=\min(k,k')+1}^{\max(k,k')} x_i \left[\bar{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)\right]$ $b' = b - s \sum_{i=\min(k,k')+1}^{\max(k,k')} \left[\bar{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)\right]$
	Compute $c''(k') = a'/b'$	
5	Set $c' = c''(k')$, $a = a', b = b'$ and $k = k'$ and go to Step 2	

The special LWA that is used in the second step of perceptual reasoning is one in which the firing levels from the first step are used to weight the IT2 consequent FSs. This way of aggregating fired-rule consequent sets is novel and is quite different from taking the union (using the maximum operation) of such sets. If one does the latter, then the resulting composite FOU does not resemble the FOU of a word in a codebook [e.g., see $FOU(\tilde{B})$ in Fig. 5b; it looks quite different than $FOU(\tilde{G}^1)$ or $FOU(\tilde{G}^2)$, even if there is only one such fired rule, due to the clipping operation of the minimum operation].

It has been proved that the output FOUs from the if–then rules of perceptual reasoning resemble the words that are in the codebook [36, 13].

When the Per-C CWW engine is the IWA, then its output is computed by using two enhanced KM algorithms (see Table 1), one for the left end point and one for the right end point of the IWA. The KM algorithms (and EKM algorithms), which were originally developed in the context of an IT2 FLS for type reduction (see Sect. 6) [4, 38], have turned out to be essential tools for the Per-C.

When the Per-C CWW engine is the FWA, then its output is computed by using the alpha-cut function decomposition theorem[3] [39], because the FWA is a (nonlinear) function of T1 FSs. At each alpha-level the resulting computation reduces to an IWA, because alpha cuts are intervals of real numbers [39].

When the Per-C CWW engine is a LWA, then its output, which is itself an IT2 FS, is computed as two FWAs, one for the LMF and one for the UMF. This comes about by first representing each IT2 FS as the union of all of its embedded T1 FSs [40, 41], so that the LWA can be viewed as a FWA involving a multitude of T1 FSs.

It has been proved that the output FOUs from the LWAs resemble the words that are in the codebook [31]. An example of the LWA is in Fig. 6.

6 Output Processing Versus Decoding

For most IT2 FLSs, output processing consists of two steps: (1) *type reduction* (TR) and (2) *defuzzification*.

Type reduction is a way to project an IT2 FS into a T1 FS [2, 4]. Just as there are many kinds of defuzzifiers for a T1 FLS, there are many comparable type reducers for an IT2 FLS. Center-of-sets TR is most popular. It consists of the following two steps: (1) Compute the centroid of each consequent IT2 FS (it will be an interval-valued set) and put them in storage; and (2) After the firing interval has been computed for each fired rule, compute an IWA in which the firing intervals act as weights and the centroid of the consequents act as the signals. In the IT2 FLS literature, the IWA is called the generalized centroid [4, 38].

---

[3] The MF for a *function of T1 FSs* equals the union (over all values of alpha) of the MFs for the same function applied to the alpha cuts of the T1 FSs.

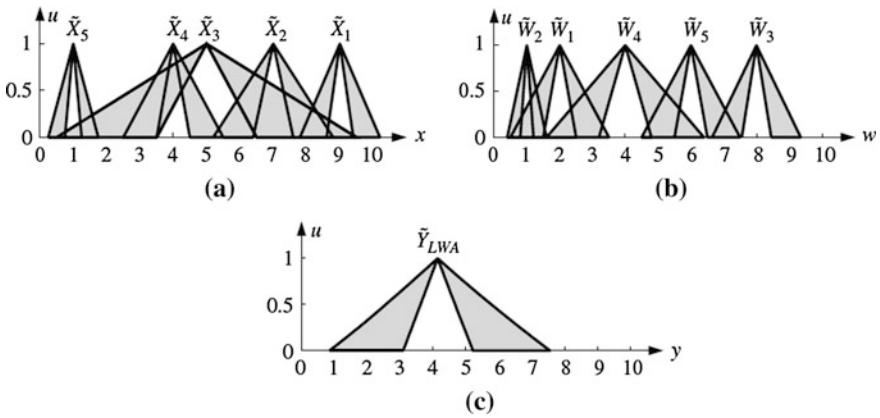


Fig. 6 (**a**) FOUs of five signals, (**b**) FOUs of their corresponding weights, (**c**) FOU of the LWA

No closed-form formulas are available to perform either Steps 1 or 2. Instead, EKM algorithms are used in both of these steps. Step 1 only has to be carried out one time after the design of the IT2 FLS has been completed. The use of EKM algorithms in the second step may sometimes lead to a time-delay, because EKM algorithms are iterative (however, they are quadratically convergent [42]). Such a delay may be unacceptable for real-time applications, such as fuzzy logic control, but does not pose a problem for nonreal-time applications, e. g., classification.

Because the type-reduced set is an interval of real numbers, defuzzification is trivial; it is obtained as the average of the two end-points of the type-reduced set.

The type-reduced set also provides a measure of the MF uncertainties that have flowed through all of the computations within the IT2 FLS. It plays a role that is analogous to standard deviation in probability. When TR is bypassed, as is commonly done in FLC, then no such useful measure is available.

To date, the three decoders for the Per-C are similarity, rank and subsethood [13, 37]. More than 50 similarity measures have been reported for T1 FSs. Additionally, the notion of similarity is very application dependent. Similarity of word FOUs requires a similarity measure that can simultaneously capture the similarities of FOU shapes and FOU proximities. The former is obvious; the latter is because word-FOUs are aligned on a scale. To date, the Jaccard similarity measure in (4) is the preferred one used for the Per-C, because it does an excellent job of simultaneously capturing similarity of both shape and proximity of FOUs.

There is no optimal way to rank FOUs. Our approach for doing this is to first compute the centroid of competing FOUs and then to use the COG of the centroids to provide a numerical ranking [13, 37]. In addition, the centroid provides a *ranking band* that is very useful, because it summarizes the uncertainties about ranking. Let the centroid of $\tilde{A}$ be denoted $C(\tilde{A}) = [c_l(\tilde{A}), c_r(\tilde{A})]$. Then the numerical rank for $\tilde{A}$, $r(\tilde{A})$, is $r(\tilde{A}) = [c_l(\tilde{A}) + c_r(\tilde{A})]/2$. A useful way to summarize the ranking information is as $r(\tilde{A}) \pm [c_r(\tilde{A}) - c_l(\tilde{A})]/2$.

Subsethood is useful when the output of the CWW engine has to be mapped into a class. We use the following subsethood measure, introduced first by Vlachos and Sergiadis [43], and later re-expressed by Wu and Mendel [37], so that it is very clear that it is the extension of Kosko's subsethood measure for T1 FSs [44], namely:

$$ss_{VS}(\tilde{A},\tilde{B}) = \frac{\sum_{i=1}^{N} \min(\bar{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{B}}(x_i)) + \sum_{i=1}^{N} \min(\underline{\mu}_{\tilde{A}}(x_i), \underline{\mu}_{\tilde{B}}(x_i))}{\sum_{i=1}^{N} \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=1}^{N} \underline{\mu}_{\tilde{A}}(x_i)} \quad (5)$$

Although this formula resembles similarity formula (4) it is quite different, because $sm_J(\tilde{A},\tilde{B}) = sm_J(\tilde{B},\tilde{A})$, but $ss_{VS}(\tilde{A},\tilde{B}) \neq ss_{VS}(\tilde{B},\tilde{A})$.

7 Outputs Versus Recommendation + Data

The output of an IT2 FLS is usually just a number. If TR is bypassed, then it is only a number—the defuzzified output value. If TR as well as defuzzification are performed, then the outputs of the IT2 FLS will be both the defuzzified number as well as the type-reduced set, which, as has been mentioned earlier, provides a measure of the MF uncertainties that have flowed through the IT2 FLS.

Observe, in Fig. 2, that the outputs from the Per-C are both a recommendation and data. In our earlier works on the Per-C, its output was only a recommendation or even just a "word." Psychologists have shown that although people want to communicate using words—the recommendation—they also want the recommendation to be backed up by data. For example, if your boss gives you a poor evaluation for the year, you will want to know "Why?" He or she will then provide you with the reasons for this and those will usually involve "data" (e.g., "You did not meet your sales target of $X for the year."). For the Per-C, the centroid, ranking bands, and similarities can be used to provide the data. People seem to understand such measures. On the other hand, we would not use subsethood because people generally do not understand it.

8 Recapitulation and Conclusions

By comparing an IT2 FLS and the Per-C in terms of inputs and membership functions, fuzzifier versus encoder, rules versus CWW engines, inference versus output of CWW engine, output processing versus decoder, and outputs versus recommendation plus data, it should be clear that there are many more differences between these two systems than there are similarities. In the Per-C:

- Words come before their MFs because one must first establish the vocabulary that will be used by Per-C.

- Words mean different things to different peoples, so IT2 FSs are used.
- Words must also mean similar things to different people, so IT2 FS models must also include this requirement.
- Words or a mixture of words and numbers always excite the Per-C.
- CWW engines are constrained; their outputs resemble the FOUs in the codebook.
- Similarity, rank and subsethood are very important in the Per-C.

In an IT2 FLS:

- Words that label a FS are not used by the calculations; hence, the MF can come before the word.
- Uncertainties about the labels of the FSs do not play an important factor in using IT2 FS; it is their additional design degrees of freedom that are important.
- Words do not excite it; numbers (certain or uncertain) do.
- Rules are aggregated in such a way that universal approximation can be appealed to; the shapes of the aggregated FOUs are unimportant.
- Most often, their output is a number that is obtained either directly by defuzzification or by a combination of TR plus defuzzification.

Something that is similar to both an IT2 FLS and the Per-C is they both rely heavily on the EKM algorithms. In an IT2 FLS, TR uses the EKM algorithms, and in the Per-C, NWAs, centroid, and ranking use the EKM algorithms. Although the EKM algorithms were developed in the context of IT2 FLSs, their use has crossed-over into CWW. Perhaps there are other FLS tools that will do the same.

Finally, an IT2 FLS and the Per-C are used for very different applications.

References

1. Castillo, O., Melin, P.: Type-2 Fuzzy Logic Theory and Applications. Springer-Verlag, Berlin (2008)
2. Karnik, N.N., Mendel, J.M., Liang, Q.: Type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **7**, 643–658 (1999)
3. Liang, Q., Mendel, J.M.: Interval type-2 fuzzy logic systems: theory and design. IEEE Trans. Fuzzy Syst. **8**, 535–550 (2000)
4. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Direction*s*. Prentice-Hall, Upper-Saddle River (2001)
5. Mendel, J.M.: Type-2 fuzzy sets and systems: an overview. IEEE Comput. Intell. Mag. **2**, 20–29 (2007)
6. Jang, J.S.R., Sun, C.T.: Neuro-fuzzy modeling and control. IEEE Proc **83**, 378–406 (1995)
7. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-fuzzy and Soft-Computing. Prentice-Hall, Upper Saddle River (1997)
8. Mendel, J.M.: Fuzzy logic systems for engineering: a tutorial. Proc. IEEE **83**, 345–377 (1995)
9. Mendel, J.M: The perceptual computer: an architecture for computing with words. In: Proceedings of Modeling with Words, Workshop in Proceedings of FUZZ-IEEE 35–38, Melbourne, Australia (2001)

10. Mendel, J.M.: An architecture for making judgments using computing with words. Int. J. Appl. Math. Comput. Sci. **12**, 325–335 (2002)
11. Mendel, J.M.: Computing with words and its relationships with fuzzistics. Info. Sci. **177**, 998–1006 (2007)
12. Mendel, J.M.: Computing with words: Zadeh, Turing, Popper and Occam. IEEE Comput. Intell. Mag. **2**, 10–17 (2007)
13. Mendel, J.M., Wu, D.: Perceptual Computing: Aiding People in Making Subjective Judgments. IEEE Press and John Wiley, Hoboken (2010)
14. Zadeh, L.A.: Fuzzy logic = computing with words. IEEE Trans on Fuzzy Syst. **4**, 103–111 (1996)
15. Zadeh, L.A: From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions. IEEE Trans. Circuits Syst.–I: Fundam. Theory Appls **4**, pp. 105–119. (1999)
16. Hagras, H.: Hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Trans. Fuzzy Syst. **12**, 524–539 (2004)
17. Hagras, H.: Type-2 FLCs: a new generation of fuzzy controllers. IEEE Comput. Intell. Mag. **2**, 30–43 (2007)
18. Kreinovich, V., Mouzouris, G.C., Nguyen, H.T: Fuzzy rule-based modeling as a universal approximation tool. In: Nguyen, H.T., Sugeno, M. (eds.) Fuzzy Systems Modeling and Control, pp. 134–195, Kluwer, Boston (1998)
19. Kennedy, J., Eberhart, R: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948. (1995)
20. Xi, M., Sun, J., Xu, W.: An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. Appl. Math. Comput. **205**, 751–759 (2008)
21. Yang, S., Wang, M., Jiao, L: A quantum particle swarm optimization. In: Congress on Evolutionary Computation (CEC2004) vol. 1, pp. 320–324. (2004)
22. Mendel, J.M.: Computing derivatives in interval type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **12**, 84–98 (2004)
23. Mendel, J.M.: Computing with words, when words can mean different things to different People. In: Proceedings of Third International ICSC Symposium on Fuzzy Logic and Applications, Rochester University, Rochester (1999)
24. Mendel, J.M.: Fuzzy sets for words: a new beginning. In: Proceedings of FUZZ-IEEE, pp. 37–42, St. Louis MO (2003)
25. Liu, F., Mendel, J.M.: Encoding words into interval type-2 fuzzy sets using an interval approach. IEEE Trans. Fuzzy Syst. **16**, 1503–1521 (2008)
26. Wu, D., Mendel, J.M., Coupland, S.: Enhanced interval approach for encoding words into interval type-2 fuzzy sets and its convergence analysis. IEEE Trans. Fuzzy Syst. **20**, 499–513 (2012)
27. Mamdani, E.H.: Applications of fuzzy algorithms for simple dynamic plant. Proc. IEEE **121**, 1585–1588 (1974)
28. Sugeno, M., Kang, G.T.: Structure identification of fuzzy model. Fuzzy Sets Syst. **28**, 15–33 (1988)
29. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Syst. Man Cybern. **15**, 116–132 (1985)
30. Liu, F., Mendel, J.M.: Aggregation using the fuzzy weighted average, as computed by the KM algorithms. IEEE Trans. Fuzzy Syst. **16**, 1–12 (2008)
31. Wu, D., Mendel, J.M.: Aggregation using the linguistic weighted average and interval type-2 fuzzy sets. IEEE Trans. Fuzzy Syst. **15**, 1145–1161 (2007)
32. Wu, D., Mendel, J.M.: Corrections to Aggregation using the linguistic weighted average and interval type-2 fuzzy sets. IEEE Trans on Fuzzy Systems **16**, 1664–1666 (2008)
33. Rickard, J.T., Aisbett, J., Yager, R.R., Gibbon, G: Fuzzy weighted power means in evaluating decisions. In: Proceedings of World Conference on Soft Computing, San Francisco (2011)

34. Rickard, J.T., Aisbett, J., Yager, R.R., Gibbon, G.: Linguistic weighted power means: comparison with the linguistic weighted average. In: Proceedings of FUZZ-IEEE, pp. 2185–2192. Taipei, Taiwan (2011)
35. Mendel, J.M., Wu, D.: Perceptual reasoning for perceptual computing. IEEE Trans. Fuzzy Syst. **16**, 1550–1564 (2008)
36. Wu, D., Mendel, J.M.: Perceptual reasoning for perceptual computing: a similarity-based approach. IEEE Trans. Fuzzy Syst. **17**, 1397–1411 (2009)
37. Wu, D., Mendel, J.M.: A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets. Inf. Sci. **179**, 1169–1192 (2009)
38. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Inf. Sci. **132**, 195–220 (2001)
39. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, Upper Saddle River 1995
40. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**, 117–127 (2002)
41. Mendel, J.M., John, R.I., Liu, F.: Interval type-2 fuzzy logic systems made simple. IEEE Trans. Fuzzy Syst. **14**, 808–821 (2006)
42. Liu, X., Mendel, J.M.: Connect Karnik-Mendel algorithms to root-finding for computing the centroid of an interval type-2 fuzzy set. IEEE Trans. Fuzzy Syst. **19**, 652–665 (2011)
43. Vlachos, I., Sergiadis, G.: Subsethood, entropy, and cardinality for interval-valued fuzzy sets—an algebraic derivation. Fuzzy Sets Syst. **158**, 1384–1396 (2007)
44. Kosko, B.: Neural Network and Fuzzy Systems, a Dynamical Systems Approach to Machine Intelligence. Prentice Hall, Englewood Cliffs (1992)
45. Wu, D., Mendel, J.M.: Enhanced Karnik-Mendel algorithms. IEEE Trans. Fuzzy Syst. **17**, 923–934 (2009)

A Survey of Continuous Karnik–Mendel Algorithms and Their Generalizations

Xinwang Liu

Abstract Karnik–Mendel (KM) algorithms are important tools for type-2 fuzzy logic. This survey chapter summarizes some extensions of continuous Karnik–Mendel algorithms. It is shown that the solution of KM algorithms can be transformed into the solution of root-finding problems, and that the iteration formula in KM algorithms is equivalent to the Newton-Raphson root-finding method in numerical analysis. New iteration formulas are summarized that accelerate the convergence speed and it is shown that numerical integration methods can be used to improve computation accuracy. This chapter demonstrates that properties and structures of KM algorithms can be understood and improved with the techniques from numerical analysis.

1 Introduction

A type-2 fuzzy logic system (FLS) allows for better modeling of uncertainty than a type-1 FLS, because a type-2 fuzzy set (T2 FS) has a Footprint of Uncertainty (FOU) that gives it more degrees of freedom than a type-1 fuzzy set (T1 FS) [2, 13, 23]. An interval type-2 fuzzy set (IT2 FS) is a simplified version of a general T2 FS because its membership grade is a crisp interval rather than a function. Most applications of type-2 FLSs involve only IT2 FSs. It has been shown that IT2 FLSs can outperform their type-1 FLSs counterparts in a variety of fields including information processing, fuzzy control, and decision making [2, 3, 7, 16, 23, 30, 31].

The centroid of an IT2 FS, developed originally by Karnik and Mendel [6], which provides a measure of the uncertainty of that FS [32], is also one of the most important computations for that FS, and is a commonly used type reduction

X. Liu (✉)
School of Economics and Management, Southeast University,
Nanjing 210096, People's Republic of China
e-mail: xwliu@seu.edu.cn

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_2,

method in T2 FLSs [8, 14, 21, 22]. Popular and efficient algorithms, called "Karnik-Mendel (KM) algorithms," were developed for centroid type reduction for interval type-2 FLSs [6, 13]. Mendel and Liu [20] proved monotonicity and super-exponential convergence of the algorithms. The iteration number of KM algorithm is usually less than seven for accuracies of 10^{-2}. Wu and Mendel [29] proposed Enhanced KM (EKM) algorithms to reduce the computational cost of the standard KM algorithms. Yeh et al. [33] proposed the Enhanced Karnik–Mendel algorithm with new initialization to compute the generalized centroid of general T2 FSs. Similarly, Zhai and Mendel [34] proposed a new centroid flow (CF) algorithm to compute the generalized centroid of general T2 FSs without having to apply KM/EKM algorithms for every α-plane. Mendel [17] gave reviews on the centroid, the algorithms for centroid computation, and its applications.

KM algorithms and their extensions have been applied to many applications of T2 FSs and play an important role in interval type-2 and general type-2 FLSs [5, 8, 13, 15, 16, 21, 23, 26], computing with words [23, 30, 31] and fuzzy weighted average problems [4, 9, 27]. The latest theoretical development related to KM algorithm is proposed by Liu and Mendel [10], in which the KM algorithm is discovered to be equivalent to the Newton-Raphson method in root finding. KM algorithms can be understood and studied from this novel point of view.

In this survey chapter, new forms of continuous KM algorithms are proposed. Section 2 gives preliminaries about IT2 FSs and the centroid computation of IT2 FSs; Sects. 3–7 provide the extensions of the KM algorithms using our new continuous KM algorithm, and, Sect. 8 summarizes the main results and draws conclusions.

2 Preliminaries

This section provides some background about IT2 FSs, the centroid of such fuzzy sets, the KM algorithms, and continuous KM algorithms.

2.1 Interval Type-2 Fuzzy Sets

An interval type-2 fuzzy set (IT2 FS) $\tilde{A}$ is characterized as [1, 13, 18, 19]:

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

where x, the *primary variable*, has domain X; $u \in U$, the *secondary variable*, has domain J_x at each $x \in X$; J_x is called the *primary membership* of x; and, the *secondary grades* of $\tilde{A}$ all equal 1. Note that (1) means: $\tilde{A}: X \to \{[a,b] : 0 \leqslant a \leqslant b \leqslant 1\}$. Uncertainty about $\tilde{A}$ is conveyed by the union of all the primary memberships, which is called the *footprint of uncertainty* (FOU) of $\tilde{A}$, i.e.,

$$FOU(\tilde{A}) = \{(x,u) : u \in J_x \subseteq [0,1]\}$$

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

An *embedded T1 FS*, A_e, is a function whose range is a subset of [0, 1] determined by $\mu_{\tilde{A}}(x,u)$, i.e.,

$$A_e = \left\{ \left(x, u(x) \mid x \in X, u \in J_x \right) \right\} \tag{2}$$

When the primary variable x is sampled at N values, $x_1, \ldots, x_N$, and at each of these values its primary memberships are sampled at M_i values, $\mu_{i_1}, \ldots, \mu_{i_{M_i}}$, then there will be $n_A = \prod_{i=1}^{N} M_i$ embedded T1 FSs that are contained within $FOU(\tilde{A})$.

2.2 Centroid of an Interval Type-2 Fuzzy Set

Recall that the centroid, $c(A)$, of the T1 FS A ($A \in X = \{x_1, x_2, , x_N\}$) is defined as

$$c(A) = \frac{\sum_{i=1}^{N} x_i \mu_A(x_i)}{\sum_{i=1}^{N} \mu_A(x_i)} \tag{3}$$

The centroid, $C_{\widetilde{A}}$, of an IT2 FS, $\tilde{A}$, which was developed by Karnik and Mendel [6], has turned out to be a very important concept for IT2 FSs and their associated FLSs. The centroid $C_{\widetilde{A}}$ is the union of the centroids of all its embedded T1 FSs A_e, i.e., [28]:

$$C(\widetilde{A}) = [c_l(\widetilde{A}), c_r(\widetilde{A})] \tag{4}$$

where

$$c_l(\widetilde{A}) = \min_{\forall A_e} c(A_e) \tag{5}$$

$$c_r(\widetilde{A}) = \max_{\forall A_e} c(A_e) \tag{6}$$

2.3 KM Algorithms

Let $x_i(i = 1, 2, N)$ represent the discretization of the primary variable of an IT2 FS $\widetilde{A}$. The centroid of IT2 FS $\widetilde{A}$, $c_{\widetilde{A}} = [c_l, c_r]$, can be computed as the optimal solutions of the following interval weighted average problems [6, 20]:

$$c_l = \min_{\forall \theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]} \frac{\sum_{i=1}^N x_i \theta_i}{\sum_{i=1}^N \theta_i} = \frac{\sum_{i=1}^{k_l} x_i \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k_l+1}^N x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k_l} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k_l+1}^N \underline{\mu}_{\tilde{A}}(x_i)} \tag{7}$$

$$c_r = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]} \frac{\sum_{i=1}^N x_i \theta_i}{\sum_{i=1}^N \theta_i} = \frac{\sum_{i=1}^{k_r} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k_r+1}^N x_i \overline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k_r} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k_r+1}^N \underline{\mu}_{\tilde{A}}(x_i)} \tag{8}$$

where k_l and k_r are called "switch points" with $x_{k_l} \leqslant c_l \leqslant x_{k_l+1}$ and $x_{k_r} \leqslant c_r \leqslant x_{k_r+1}$.

The determination of k_l and k_r can be performed by using the KM algorithms and are summarized in Table 1 [6, 13, 23].

Recently, Wu and Mendel [29] proposed Enhanced KM (EKM) algorithms given in Table 2, which improve the KM algorithms with better initializations, computational cost reduction techniques, and stopping rules.

2.4 Continuous KM Algorithms

Continuous KM (CKM) algorithms [24, 25] were proposed for studying the theoretical properties of IT2 FS centroid computations, e. g., they were used to prove that the KM algorithms converge monotonically and super-exponentially fast [20].

We assume all the x_is are different, and they are bounded in $[a, b]$, where $a = \min\limits_{1 \leqslant i \leqslant N} \{x_i\}$ and $b = \max\limits_{1 \leqslant i \leqslant N} \{x_i\}$.[1] Then, the continuous versions of (7) and (8) are

$$c_l = \min_{\xi \in [a,b]} c_l(\xi) \equiv \min_{\xi \in [a,b]} \frac{\int_a^\xi x \overline{\mu}_{\widetilde{A}}(x) dx + \int_\xi^b x \underline{\mu}_{\widetilde{A}}(x) dx}{\int_a^\xi \overline{\mu}_{\widetilde{A}}(x) dx + \int_\xi^b \underline{\mu}_{\widetilde{A}}(x) dx} \tag{9}$$

$$c_r = \max_{\xi \in [a,b]} c_r(\xi) \equiv \max_{\xi \in [a,b]} \frac{\int_a^\xi x \underline{\mu}_{\widetilde{A}}(x) dx + \int_\xi^b x \overline{\mu}_{\widetilde{A}}(x) dx}{\int_a^\xi \underline{\mu}_{\widetilde{A}}(x) dx + \int_\xi^b \overline{\mu}_{\widetilde{A}}(x) dx} \tag{10}$$

Continuous versions of the KM algorithms for c_l and c_r, which give the optimal solutions of (9) and (10), are given in Table 3.

[1] As noted in [25, p. 363], if Gaussian MFs are used, one can extend the theoretical results to $a \to -\infty$, $b \to +\infty$; but, in practice, when truncations are used, a and b are again finite numbers.

Table 1 KM algorithm for computing the centroid end-points of an IT2 FS, $\widetilde{A}$ [6, 13, 23][a]

Step	KM algorithm for c_l	KM algorithm for c_r
	$c_l = \min_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\overline{\mu}_{\tilde{A}}(x_i)]}\left(\sum_{i=1}^N x_i\theta_i / \sum_{i=1}^N \theta_i\right)$	$c_r = \max_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\overline{\mu}_{\tilde{A}}(x_i)]}\left(\sum_{i=1}^N x_i\theta_i / \sum_{i=1}^N \theta_i\right)$
1	Initialize θ_i by setting $\theta_i = [\underline{\mu}_{\tilde{A}}(x_i) + \overline{\mu}_{\tilde{A}}(x_i)]/2, i = 1, 2, N$ and then compute $c' = c(\theta_1, \theta_2, , \theta_N) = \sum_{i=1}^N x_i\theta_i \Big/ \sum_{i=1}^N \theta_i$	
2	Find $k(1 \leqslant k \leqslant N-1)$ such that $x_k \leqslant c' \leqslant x_{k+1}$.	
3	Compute $c_l(k) = \frac{\sum_{i=1}^k x_i\overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^N x_i\underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^k \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^N \underline{\mu}_{\tilde{A}}(x_i)}$	Compute $c_r(k) = \frac{\sum_{i=1}^k x_i\underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^N x_i\overline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^k \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^N \overline{\mu}_{\tilde{A}}(x_i)}$
4	Check if $c_l(k) = c'$. If yes, stop and set $c_l(k) = c_l$ and $k = L$. If no, go to step 5.	Check if $c_r(k) = c'$. If yes, stop and set $c_r(k) = c_r$ and $k = R$. If no, go to step 5.
5	Set $c' = c_l(k)$ and go to Step 2.	Set $c' = c_r(k)$ and go to Step 2.

[a] Note that $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_N$

Table 2 EKM algorithms for computing the centroid end-points of an IT2 FS, $\tilde{A}$ [23, 29][a],

Step	EKM algorithm for c_l	EKM algorithm for c_r
	$c_l = \min_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\overline{\mu}_{\tilde{A}}(x_i)]}\left(\sum_{i=1}^N x_i\theta_i / \sum_{i=1}^N \theta_i\right)$	$c_r = \max_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\overline{\mu}_{\tilde{A}}(x_i)]}\left(\sum_{i=1}^N x_i\theta_i / \sum_{i=1}^N \theta_i\right)$
1	Set $k = [N/2.4]$ (the nearest integer to $N/2.4$) and compute $\alpha = \sum_{i=1}^k x_i\overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^N x_i\underline{\mu}_{\tilde{A}}(x_i)$, $\beta = \sum_{i=1}^k \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^N \underline{\mu}_{\tilde{A}}(x_i)$. Compute $c' = \alpha/\beta$.	Set $k = [N/1.7]$ (the nearest integer to $N/1.7$) and compute $\alpha = \sum_{i=1}^k x_i\underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^N x_i\overline{\mu}_{\tilde{A}}(x_i)$, $\beta = \sum_{i=1}^k \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^N \overline{\mu}_{\tilde{A}}(x_i)$.
2	Find $k' \in [1, N-1]$ such that $x_{k'} \leqslant c' \leqslant x_{k'+1}$.	
3	Check if $k' = k$. If yes, stop and set $c' = c_l$ and $k = L$. If no, go to step 4.	Check if $k' = k$. If yes, stop and set $c' = c_r$ and $k = R$. If no, go to step 4.
4	Compute $s = sign(k'-k)$ and $\alpha' = \alpha + s \sum_{i=\min(k,k')+1}^{\max(k,k')} x_i\left[\overline{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)\right]$, $\beta' = \beta + s \sum_{i=\min(k,k')+1}^{\max(k,k')} \left[\overline{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)\right]$. Compute $c''(k') = \alpha'/\beta'$.	Compute $s = sign(k'-k)$ and $\alpha' = \alpha - s \sum_{i=\min(k,k')+1}^{\max(k,k')} x_i\left[\overline{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)\right]$. $\beta' = \beta - s \sum_{i=\min(k,k')+1}^{\max(k,k')} \left[\overline{\mu}_{\tilde{A}}(x_i) - \underline{\mu}_{\tilde{A}}(x_i)\right]$.
5	Set $c' = c''(k'), \alpha = \alpha', \beta = \beta'$ and $k = k'$ and go to Step 2.	

[a]Note that $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_N$

Table 3 Continuous KM (CKM) algorithms for computing the centroid end-points of an IT2 FS, $\tilde{A}$

Step	CKM algorithm for c_l	CKM algorithm for c_r
	$c_l = \min_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\overline{\mu}_{\tilde{A}}(x)]} \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$	$c_r = \max_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\overline{\mu}_{\tilde{A}}(x)]} \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$
1	Let $\theta(x) = (\underline{\mu}_{\tilde{A}}(x) + \overline{\mu}_{\tilde{A}}(x))/2$, and compute the initial value ξ, as $\xi = \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$	
2	Compute $\xi_l = \frac{\int_a^\xi x\overline{\mu}_{\tilde{A}}(x)dx + \int_\xi^b x\underline{\mu}_{\tilde{A}}(x)dx}{\int_a^\xi \overline{\mu}_{\tilde{A}}(x)dx + \int_\xi^b \underline{\mu}_{\tilde{A}}(x)dx}$	Compute $\xi_r = \frac{\int_a^\xi x\underline{\mu}_{\tilde{A}}(x)dx + \int_\xi^b x\overline{\mu}_{\tilde{A}}(x)dx}{\int_a^\xi \underline{\mu}_{\tilde{A}}(x)dx + \int_\xi^b \overline{\mu}_{\tilde{A}}(x)dx}$
3	Check if $\lvert\xi - \xi_l\rvert \leqslant \varepsilon$ (ε is a given error bound of the algorithms). If yes, stop and set $c_r = \xi_r$. If no, go to step 4.	Check if $\lvert\xi - \xi_r\rvert \leqslant \varepsilon$ (ε is a given error bound of the algorithms). If yes, stop and set $c_l = \xi_l$. If no, go to step 4.
4	Set $\xi = \xi_l$ and go to Step 2.	Set $\xi = \xi_r$ and go to Step 2.

The continuous version of EKM called continuous EKM (CEKM) is given in Table 4.

3 Transforming the Solution of KM Algorithms into Root-Finding Problems

The solutions of (9) and (10) can also be transformed into the solutions of root-finding problems.

Theorem 1 [10] (1) $c_l = c_l(\xi^*)$ *is the unique minimum value of* (9), *and* ξ^* *is the unique simple root of the monotonic increasing convex function*:

$$\varphi(\xi) = \int_a^\xi (\xi - x)\overline{\mu}_{\tilde{A}}(x)dx + \int_\xi^b (\xi - x)\underline{\mu}_{\tilde{A}}(x)dx \tag{11}$$

(2) $c_r = c_r(\xi^*)$ *is the unique maximum value of* (10), *and* ξ^* *is the unique simple root of the monotonic decreasing convex function*:

$$\psi(\xi) = -\int_a^\xi (\xi - x)\underline{\mu}_{\tilde{A}}(x)dx - \int_\xi^b (\xi - x)\overline{\mu}_{\tilde{A}}(x)dx \tag{12}$$

Proof See [10]. □

Table 4 Continuous EKM (CEKM) algorithms for computing the centroid end-points of an IT2 FS, $\tilde{A}$

Step	CEKM algorithm for c_l	CEKM algorithm for c_r
	$c_l = \min_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\overline{\mu}_{\tilde{A}}(x)]} \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$	$c_r = \max_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\overline{\mu}_{\tilde{A}}(x)]} \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$
1[a]	Set $c = a + (b - a)/2.4$, and compute $\alpha = \int_a^c x\overline{\mu}_{\tilde{A}}(x)dx + \int_c^b x\underline{\mu}_{\tilde{A}}(x)dx,$ $\beta = \int_a^c \overline{\mu}_{\tilde{A}}(x)dx + \int_c^b \underline{\mu}_{\tilde{A}}(x)dx.$ Compute $c' = \alpha/\beta$.	Set $c = a + (b - a)/1.7$, and compute $\alpha = \int_a^c x\underline{\mu}_{\tilde{A}}(x)dx + \int_c^b x\overline{\mu}_{\tilde{A}}(x)dx,$ $\beta = \int_a^c x\underline{\mu}_{\tilde{A}}(x)dx + \int_c^b x\overline{\mu}_{\tilde{A}}(x)dx.$
2	Check if $\lvert c' - c\rvert \leqslant \varepsilon$ (ε is a given error bound of the algorithms). If yes, stop and set $c' = c_l$. If no, go to step 4.	Check if $\lvert c' - c\rvert \leqslant \varepsilon$ (ε is a given error bound of the algorithms). If yes, stop and set $c' = c_r$. If no, go to step 4.
3	Compute $s = sign(c' - c)$ and: $\alpha' = \alpha + s\int_{\min(c,c')}^{\max(c,c')} x\left[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)\right]dx,$ $\beta' = \beta + s\int_{\min(c,c')}^{\max(c,c')} \left[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)\right]dx.$ Compute$c'' = \alpha',/\beta'$	Compute $s = sign(c' - c)$ and: $\alpha' = \alpha - s\int_{\min(c,c')}^{\max(c,c')} x\left[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)\right]dx,$ $\beta' = \beta - s\int_{\min(c,c')}^{\max(c,c')} \left[\overline{\mu}_{\tilde{A}}(x) - \underline{\mu}_{\tilde{A}}(x)\right]dx$
4	Set $c = c', c' = c'', \alpha = \alpha', \beta = \beta'$ and go to Step 2	

[a]The initialization step utilizes the shift-invariant property of computing the centroid of an IT2 FS [25], i. e., one can always set $a = 0$, so that the total sample number N corresponds to the integral length $b - a$.

From Theorem 1, the computation of c_l and c_r can be transformed into root-finding problems. Liu and Mendel [10] have also proved that the iteration formula in the CKM algorithms is equivalent to the iteration of the Newton-Raphson root-finding method in numerical analysis, for the root-finding problems (11) and (12). More specifically, the Newton-Raphson iteration formulas can be stated as

$$\xi_l = \xi + \frac{\varphi'(\xi)}{\varphi'(\xi)} \tag{13}$$

and

$$\xi_r = \xi + \frac{\psi'(\xi)}{\psi'(\xi)} \tag{14}$$

When the respective derives are substituted into (13) and (14), then those equations become the same as the iteration formulas in Table 3, respectively. EKM algorithms use the same iteration formula as KM algorithms; hence, convergence speeds of KM and EKM algorithms are quadratic, because Newton-Raphson algorithms are known to be quadratically convergent.

From this viewpoint, the KM algorithm iteration can be regarded as a special form of a root-finding method, which suggests that other techniques from numerical analysis can also be used to further improve the KM algorithms.

4 Halley's Method to Accelerate the Convergence of KM Algorithms

Motivated by (13) and (14), one can use Halley's method from numerical analysis to improve the convergence from quadratic to cubic.

For the root-finding problem (11), it follows from Halley's method [12], that

$$\xi_l = \xi - \frac{\varphi(\xi)}{\varphi'(\xi)}\left(1 - \frac{\varphi(\xi)\varphi''(\xi)}{2(\varphi'(\xi))^2}\right)^{-1} \tag{15}$$

where

$$\varphi(\xi) = \int_a^{\xi}(\xi - x)\overline{\mu}_{\widetilde{A}}(x)dx + \int_{\xi}^{b}(\xi - x)\underline{\mu}_{\widetilde{A}}(x)dx \tag{16}$$

$$\varphi'(\xi) = \int_a^{\xi}\overline{\mu}_{\widetilde{A}}(x)dx + \int_{\xi}^{b}\underline{\mu}_{\widetilde{A}}(x)dx \tag{17}$$

$$\varphi''(\xi) = \overline{\mu}_{\widetilde{A}}(x)(\xi) - \underline{\mu}_{\widetilde{A}}(x)(\xi) \tag{18}$$

Similarly, for the root-finding problem (12), a comparable algorithm for ξ_r is

$$\xi_r = \xi - \frac{\psi(\xi)}{\psi'(\xi)}\left(1 - \frac{\psi(\xi)\psi''(\xi)}{2(\psi'(\xi))^2}\right)^{-1} \tag{19}$$

where

$$\psi(\xi) = -\int_a^{\xi}(\xi - x)\underline{\mu}_{\widetilde{A}}(x)dx - \int_{\xi}^{b}(\xi - x)\overline{\mu}_{\widetilde{A}}(x)dx \tag{20}$$

$$\psi'(\xi) = -\int_a^{\xi}\underline{\mu}_{\widetilde{A}}(x)dx - \int_{\xi}^{b}\overline{\mu}_{\widetilde{A}}(x)dx \tag{21}$$

$$\psi''(\xi) = -\underline{\mu}_{\tilde{A}}(x)(\xi) + \overline{\mu}_{\tilde{A}}(x)(\xi) \tag{22}$$

Although convergence is improved by using (15) and (19), complexity is increased.

5 Another Way to Accelerate the Convergence of KM Algorithms

A Taylor series expansion of $\varphi(\xi_l)$ around ξ, is

$$\varphi(\xi_l) \approx \varphi(\xi) + \varphi'(\xi)(\xi_l - \xi) + \frac{1}{2}\varphi''(\xi)(\xi_l - \xi)^2$$

If $\varphi''(\xi) \neq 0$, then the one root of $\varphi(\xi) = 0$ that is closer to ξ is

$$\xi_l = \xi - \frac{\varphi'(\xi) - \sqrt{\varphi'(\xi)^2 - 2\varphi(\xi)\varphi''(\xi)}}{\varphi''(\xi)} \tag{23}$$

On the other hand, if $\varphi''(\xi) = 0$, then

$$\xi_l = \xi - \frac{\varphi'(\xi)}{\varphi(\xi)} \tag{24}$$

Observe that (24) is the same as a KM algorithm or the Newton-Raphson iteration formula (13), which provides those algorithms with another interesting interpretation.

Considering these two cases together, we have the following new algorithm for computing ξ_l:

$$\xi_l = \begin{cases} \xi - \frac{\varphi'(\xi)}{\varphi(\xi)} & \text{if } \varphi''(\xi) = 0 \\ \xi - \frac{\varphi'(\xi) - \sqrt{\varphi'(\xi)^2 - 2\varphi(\xi)\varphi''(\xi)}}{\varphi''(\xi)} & \text{if } \varphi''(\xi) \neq 0 \end{cases} \tag{25}$$

Similarly, for ξ_r, we have

$$\xi_r = \begin{cases} \xi - \frac{\psi'(\xi)}{\psi(\xi)} & \text{if } \psi''(\xi) = 0 \\ \xi - \frac{\psi'(\xi) - \sqrt{\psi'(\xi)^2 - 2\psi(\xi)\psi''(\xi)}}{\psi''(\xi)} & \text{if } \psi''(\xi) \neq 0 \end{cases} \tag{26}$$

Note that $\varphi(\xi)$ and $\psi(\xi)$ and their derivatives are given in Sect. 4.

These new algorithms can be proved to be cubically convergent (omitted), and they can be used to replace ξ_l and ξ_r in the Table 3 CKM algorithms (Step 2).

6 Extension of EKM Algorithms to Weighted EKM Algorithms

By comparing KM algorithms and CKM algorithms, and also EKM algorithms and CEKM algorithms, respectively, we observe that the former compute the centroid using the sample points of the FOU, whereas the latter compute the centroid using MFs of the FOU. The summing operations in (E) KM algorithms are the approximations of the integral operations in C(E)KM algorithms. Using numerical integration techniques, weighted EKM (WEKM) algorithms can be obtained to improve computational accuracy [11]. Table 5 gives weighted EKM algorithms, where some weight assignments are given in Table 6 for some well-known numerical integration rules.

Ordinary EKM algorithms are special cases of the WEKM algorithms. Like EKM algorithms, WEKM algorithms obtain the centroid values of an IT2 FS

Table 5 Weighted EKM (WEKM) algorithms for computing the centroid end-points of IT2 FS, $\tilde{A}$[a],

Step	WEKM algorithm for c_l	WEKM algorithm for c_r
	$c_l = \min_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\overline{\mu}_{\tilde{A}}(x_i)]} \frac{\sum_{i=1}^N w_i x_i\theta_i}{\sum_{i=1}^N w_i\theta_i}$	$c_r = \max_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\overline{\mu}_{\tilde{A}}(x_i)]} \frac{\sum_{i=1}^N w_i x_i\theta_i}{\sum_{i=1}^N w_i\theta_i}$
1	Set $k=[N/2.4]$ (the nearest integer to $N/2.4$) and compute[b]: $\alpha=\sum_{i=1}^{k} w_i x_i\overline{\mu}_{\tilde{A}}(x_i)+\sum_{i=k+1}^{N} w_i x_i\underline{\mu}_{\tilde{A}}(x_i)$ $\beta=\sum_{i=1}^{k} w_i\overline{\mu}_{\tilde{A}}(x_i)+\sum_{i=k+1}^{N} w_i\underline{\mu}_{\tilde{A}}(x_i)$. Compute $c'=\alpha/\beta$.	Set $k=[N/1.7]$ (the nearest integer to $N/1.7$) and compute[b]: $\alpha=\sum_{i=1}^{k} w_i x_i\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i=k+1}^{N} w_i x_i\overline{\mu}_{\tilde{A}}(x_i)$, $\beta=\sum_{i=1}^{k} w_i\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i=k+1}^{N} w_i\overline{\mu}_{\tilde{A}}(x_i)$.
2	Find $k'\in[1,N-1]$ such that $x_{k'}\leqslant c'\leqslant x_{k'+1}$.	
3	Check if $k'=k$. If yes, stop and set $c'=c_l$ and $k=L$. If no, go to step 4.	Check if $k'=k$. If yes, stop and set $c'=c_r$ and $k=R$. If no, go to step 4.
4	Compute $s=sign(k'-k)$ and: $\alpha'=\alpha+s\sum_{i=\min(k,k')}^{\max(k,k')} w_i x_i\left[\overline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]$, $\beta'=\beta+s\sum_{i=\min(k,k')}^{\max(k,k')} w_i x_i\left[\overline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]$, Compute $c''=\alpha'/\beta'$.	Compute $s=sign(k'-k)$ and: $\alpha'=\alpha-s\sum_{i=\min(k,k')}^{\max(k,k')} w_i\left[\overline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]$. $\beta'=\beta-s\sum_{i=\min(k,k')}^{\max(k,k')} w_i\left[\overline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]$.
5	Set $c'=c'',\alpha=\alpha',\beta=\beta'$ and $k=k'$ and go to Step 2.	

[a]Note that $x_1\leqslant x_2\leqslant\cdots\leqslant x_N$

[b]w_i are given in Table 6

Table 6 Weight assignment methods of WEKM (EKM) algorithms,

Algorithms	Integration rule	Weight value
EKM	Riemann sum	$w_i = 1(i = 1, 2, , N)$
TWEKM	Trapezoidal rule	$w_i = \begin{cases} 1/2 & \text{if } i = 1, N, \\ 1 & \text{if } i \neq 1, N. \end{cases}$
SWEKM	Simpson's rule	$w_i = \begin{cases} 1/2 & \text{if } i = 1, N \\ 1 & \text{if } i = 1 \text{ mod}^{a}(2) \text{ and } i \neq 1, N, \\ 2 & \text{if } i = 0 \text{ mod}(2) \text{and } i \neq N. \end{cases}$
S3/8WEKM	Simpson's 3/8 rule	$w_i = \begin{cases} 1/3 & \text{if } i = 1, N \\ 2/3 & \text{if } i = 1 \text{ mod}(3) \text{and } i \neq 1, N, \\ 1 & \text{if } i = 2 \text{ mod}(3) \text{and } i \neq N, \\ 1 & \text{if } i = 0 \text{ mod}(3) \text{ and } i \neq N. \end{cases}$

[a] mod is modular arithmetic operator. $i = j(d)$ means $i = nd + j$, where n is an integer

approximately. Such approximate values approach the exact values only when the sample size $N \to \infty$.

Numerical examples have shown these new algorithms significantly outperform current KM and EKM algorithms as far as computational accuracy.

7 Relationships Among the Three Extensions of Continuous KM Algorithms

By connecting the transformation of the two KM optimization problems to equivalent rooting-finding problems, demonstrating the equivalence between KM algorithms and Newton-Raphson algorithms, and using different kinds numerical integration methods to obtain WEKM algorithms, three new kinds of methods for computing the centroid of an IT2 FS have been obtained:

1. *Direct root-finding methods*: using Theorem 1, the centroid values of an IT2 FS can be obtained by solving for the roots of (11) and (12) directly. Various root-finding methods can be applied to do this.
2. *CEKM algorithms and their accelerations*: the centroid values of an IT2 FS can also be obtained using the CEKM algorithms in Table 4, which are improvements of the CKM algorithms in Table 3. These algorithms can be accelerated using the new iteration formulas in (25) and (26) from quadratic to cubic.
3. *WEKM algorithms and their accelerations*: WEKM algorithms are the implementations of CEKM algorithms using various numerical integration methods. EKM algorithms are special cases of WEKM algorithms. EKM/WEKM can also be accelerated by using the discrete versions of (25) and (26).

8 Conclusions

This chapter has summarized many extensions of KM/EKM algorithms that are used in the centroid computations of interval type-2 fuzzy sets. Most important is the fact that the KM algorithm iteration is a special form of root finding solved by using a Newton-Raphson formula. Different iteration formulas are given that accelerate the convergence speed. Different numerical integration formulas can be used to improve the accuracy of KM algorithms by means of so-called weighted KM algorithms.

Acknowledgments This work has been supported by the Natural Science Foundation of China Project under Grant Nos. 70771025 and 701171048. Part of the work in this chapter was done jointly with Professor Jerry M. Mendel and Dr. Dongrui Wu in [10, 11]. I would like to thank Professor Jerry M. Mendel and anonymous reviewers for their valuable comments and suggestions.

References

1. Aisbett, J., Rickard, J.T., Morgenthaler, D.G.: Type-2 fuzzy sets as functions on spaces. IEEE Trans. Fuzzy Syst. **18**(4), 841–844 (2010). doi:10.1109/TFUZZ.2010.2046176
2. Castillo, O., Melin, P.: Type-2 Fuzzy Logic Theory and Applications. Springer, Berlin (2008)
3. Hagras, H.: Type-2 FLCs: a new generation of fuzzy controllers. IEEE Comput. Intell. Mag. **2**(1), 30–43 (2007)
4. Hu, H.Z., Gao, M.J., Zhang, H.: A new algorithm for computing the fuzzy weighted average. IEICE Electron. Express **7**(19), 1423–1428 (2010). doi:10.1587/elex.7.1423
5. John, R.I., Coupland, S.: Type-2 fuzzy logic: a historical view. IEEE Comput. Intell. Mag. **2**(1), 57–62 (2007)
6. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Inf. Sci. **132**(1–4), 195–220 (2001)
7. Lee, C.S., Wang, M.H., Hagras, H.: A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation. IEEE Trans. Fuzzy Syst. **18**(2), 374–395 (2010). doi:10.1109/TFUZZ.2010.2042454
8. Liu, F.: An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. Inf. Sci. **178**(9), 2224–2236 (2008)
9. Liu, F., Mendel, J.M.: Aggregation using the fuzzy weighted average as computed by the Karnik-Mendel algorithms. IEEE Trans. Fuzzy Syst. **16**(1), 1–12 (2008)
10. Liu, X., Mendel, J.M.: Connect Karnik-Mendel algorithms to root-finding for computing the centroid of an interval type-2 fuzzy set. IEEE Trans. Fuzzy Syst. **19**(4), 652–665 (2011)
11. Liu, X., Mendel, J.M., Wu, D.: Study on enhanced Karnik-Mendel algorithms: initialization explanations and computation improvements. Inf. Sci. **184**(1), 75–91 (2012)
12. Mathews, J.H., Fink, K.K.: Numerical Methods Using Matlab. Prentice-Hall Inc, Upper Saddle River, NJ (2004)
13. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall Inc, Upper Saddle River, NJ (2001)
14. Mendel, J.M.: Advances in type-2 fuzzy sets and systems. Inf. Sci. **177**(1), 84–110 (2007)
15. Mendel, J.M.: Type-2 fuzzy sets and systems: an overview. IEEE Comput. Intell. Mag. **2**(1), 20–29 (2007)

16. Mendel, J.M.: On answering the question 'where do I start in order to solve a new problem involving interval type-2 fuzzy sets?'. Inf. Sci. **179**(19), 3418–3431 (2009)
17. Mendel, J.M.: On centroid calculations for type-2 fuzzy sets. Appl. Comput. Math. **10**(1), 88–96 (2011)
18. Mendel, J.M., Hagras, H., John, R.I.: Standard background material about interval type-2 fuzzy logic systems that can be used by all authors (2006). http://ieee-cis.org/_files/standards.t2.win.pdf
19. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
20. Mendel, J.M., Liu, F.: Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set. IEEE Trans. Fuzzy Syst. **15**(2), 309–320 (2007)
21. Mendel, J.M., Liu, F., Zhai, D.: α-plane representation for type-2 fuzzy sets: theory and applications. IEEE Trans. Fuzzy Syst. **17**(5), 1189–1207 (2009)
22. Mendel, J.M., Wu, D.: Perceptual reasoning for perceptual computing. IEEE Trans. Fuzzy Syst. **16**(6), 1550–1564 (2008)
23. Mendel, J.M., Wu, D.: Perceptual Computing: Aiding People in Making Subjective Judgments. Wiley-IEEE Press, Hoboken, NJ (2010)
24. Mendel, J.M., Wu, H.: Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: part 1, forward problems. IEEE Trans. Fuzzy Syst. **14**(6), 781–792 (2006)
25. Mendel, J.M., Wu, H.: New results about the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule. Inf. Sci. **177**(2), 360–377 (2007)
26. Wagner, C., Hagras, H.: Toward general type-2 fuzzy logic systems based on zslices. IEEE Trans. Fuzzy Syst. **18**(4), 637–660 (2010)
27. Wu, D., Mendel, J.M.: Aggregation using the linguistic weighted average and interval type-2 fuzzy sets. IEEE Trans. Fuzzy Syst. **15**(6), 1145–1161 (2007)
28. Wu, D., Mendel, J.M.: Uncertainty measures for interval type-2 fuzzy sets. Inf. Sci. **177**(23), 5378–5393 (2007)
29. Wu, D., Mendel, J.M.: Enhanced Karnik-Mendel algorithms. IEEE Trans. Fuzzy Syst. **17**(4), 923–934 (2009)
30. Wu, D., Mendel, J.M.: Perceptual reasoning for perceptual computing: a similarity-based approach. IEEE Trans. Fuzzy Syst. **17**(6), 1397–1411 (2009)
31. Wu, D., Mendel, J.M.: Computing with words for hierarchical decision making applied to evaluating a weapon system. IEEE Trans. Fuzzy Syst. **18**(3), 441–460 (2010). doi:10.1109/TFUZZ.2010.2043439
32. Wu, H., Mendel, J.M.: Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **10**(5), 622–639 (2002)
33. Yeh, C.Y., Jeng, W.H.R., Lee, S.J.: An enhanced type-reduction algorithm for type-2 fuzzy sets. IEEE Trans. Fuzzy Syst. **19**(2), 227–240 (2011)
34. Zhai, D.Y., Mendel, J.M.: Computing the centroid of a general type-2 fuzzy set by means of the centroid-flow algorithm. IEEE Trans. Fuzzy Syst. **19**(3), 401–422 (2011)

Two Differences Between Interval Type-2 and Type-1 Fuzzy Logic Controllers: Adaptiveness and Novelty

Dongrui Wu

Abstract Interval type-2 fuzzy logic controllers (IT2 FLCs) have been attracting great research interests recently. Many reported results have shown that IT2 FLCs are better able to handle uncertainties than their type-1 (T1) counterparts. A challenging question is: *What are the fundamental differences between IT2 and T1 FLCs?* Once the fundamental differences are clear, we can better understand the advantages of IT2 FLCs and hence better make use of them. This chapter explains two fundamental differences between IT2 and T1 FLCs: (1) *Adaptiveness*, meaning that the embedded T1 fuzzy sets used to compute the bounds of the type-reduced interval change as input changes; and, (2) *Novelty*, meaning that the upper and lower membership functions of the same IT2 fuzzy set may be used simultaneously in computing each bound of the type-reduced interval. T1 FLCs do not have these properties; thus, a T1 FLC cannot implement the complex control surface of an IT2 FLC given the same rulebase.

1 Introduction

Interval type-2 fuzzy logic controllers (IT2 FLCs) have been attracting great research interests recently. Many reported results have shown that IT2 FLCs are better able to handle uncertainties than their type-1 (T1) counterparts [1, 5, 10, 22, 23]. A challenging question is: *What are the fundamental differencesbetween IT2 and T1 FLCs?* Once the fundamental differences are clear, we can better understand the advantages of IT2 FLCs and hence better make use of them. In the literature, there has been considerable effort on answering this challenging and fundamental question. Some important arguments are [17]:

D. Wu (✉)
Machine Learning Lab, GE Global Research, Niskayuna, NY, USA
e-mail: wud@ge.com; drwu09@gmail.com

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_3,

1. *An IT2 fuzzy set (FS) can better model intra-personal and inter-personal uncertainties,*[1] which are intrinsic to natural language, because the membership grade of an IT2 FS is an interval instead of a crisp number in a T1 FS. Mendel [11] also showed that IT2 FS is a scientifically correct model for modeling linguistic uncertainties, whereas T1 FS is not.
2. *Using IT2 FSs to represent the FLC inputs and outputs will result in the reduction of the rulebase when compared to using T1 FSs* [5, 10], as the ability of the footprint of uncertainty (FOU) to represent more uncertainties enables one to cover the input/output domains with fewer FSs. This makes it easier to construct the rulebase using expert knowledge and also increases robustness [20, 22, 23].
3. *An IT2 FLC can give a smoother control surfacethan its T1 counterpart, especially in the region around the steady state* (i.e., when both the error and the change of error approach 0) [6, 20, 22, 23]. Wu and Tan [24] showed that when the baseline T1 FLC implements a linear PI control law and the IT2 FSs of an IT2 FLC are obtained from symmetrical perturbations of the T1 FSs, the resulting IT2 FLC implements a variable gain PI controller around the steady state. These gains are smaller than the PI gains of the baseline T1 FLC, which result in a smoother control surface around the steady state. The PI gains of the IT2 FLC also change with the inputs, which cannot be achieved by the baseline T1 FLC.
4. *IT2 FLCs are more adaptive and they can realize more complex input–output relationships which cannot be achieved by T1 FLCs.* Karnik and Mendel [8] pointed out that an IT2 fuzzy logic system can be thought of as a collection of many different embedded T1 fuzzy logic systems. Wu and Tan [21] proposed a systematic method to identify the *equivalent generalized T1 FSs* that can be used to replace the FOU. They showed that the equivalent generalized T1 FSs are significantly different from traditional T1 FSs, and there are different equivalent generalized T1 FSs for different inputs. Du and Ying [3], and Nie and Tan [14], also showed that a symmetrical IT2 fuzzy-PI (or the corresponding PD) controller, obtained from a baseline T1 PI FLC, partitions the input domain into many small regions, and in each region it is equivalent to a nonlinear PI controller with variable gains. The control law of the IT2 FLC in each small region is much more complex than that of the baseline T1 FLC, and

[1] According to Mendel [11], intra-personal uncertainty describes "*the uncertainty a person has about the word.*" It is also explicitly pointed out by psychologists Wallsten and Budescu [15] as "*except in very special cases, all representations are vague to some degree in the minds of the originators and in the minds of the receivers,*" and they suggest to model it by a T1 FS. According to Mendel [11], inter-personal uncertainty describes "*the uncertainty that a group of people have about the word,*" i.e., "*words mean different things to different people.*" It is also explicitly pointed out by psychologists Wallsten and Budescu [15] as "*different individuals use diverse expressions to describe identical situations and understand the same phrases differently when hearing or reading them.*"

hence it can realize more complex input–output relationship that cannot be achieved by a T1 FLC using the same rulebase.

5. *IT2 FLCs have a noveltythat does not exist in traditional T1 FLCs*. Wu [16, 17] showed that in an IT2 FLC *different* membership grades from the same IT2 FS can be used in different rules, whereas for traditional T1 FLC the *same* membership grade from the same T1 FS is always used in different rules. This again implies that an IT2 FLC is more complex than a T1 FLC and it cannot be implemented by a T1 FLC using the same rulebase.

This chapter explains why *adaptiveness* and *novelty* are two fundamental differences between IT2 and T1 FLCs. Methods for visualizing the effects of these two differences can be found in [17].

2 Interval Type-2 Fuzzy Sets and Controllers

This section introduces background materials on IT2 FSs and FLCs, and shows two numerical examples on IT2 FLCs.

2.1 Interval Type-2 Fuzzy Sets (IT2 FSs)

T1 FS theory was first introduced by Zadeh [25] in 1965 and has been successfully applied in many areas.

Definition 1 A *T1 FS X* is comprised of a *domain* D_X of real numbers (also called the *universe of discourse* of *X*) together with a *membership function* (MF) $\mu_X : D_X \to [0,1]$, i.e.,

$$X = \int_{D_X} \mu_X(x)/x. \tag{1}$$

Here $\int$ denotes the collection of all points $x \in D_X$ with associated *membership grade* $\mu_X(x)$. □

Despite having a name which carries the connotation of uncertainty, research has shown that there are limitations in the ability of T1 FSs to model and minimize the effect of uncertainties [4, 5, 10, 22]. This is because a T1 FS is certain in the sense that its membership grades are crisp values. Recently, type-2 FSs [26], characterized by MFs that are themselves fuzzy, have been attracting great interests. IT2 FSs [10], a special case of type-2 FSs, are currently the most widely used for their reduced computational cost.

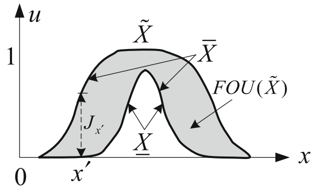


Fig. 1 An IT2 FS. $\underline{X}$ (the LMF) and $\overline{X}$ (the UMF) are two embedded T1 FSs

Definition 2 [10, 12] An IT2 $\widetilde{X}$ is characterized by its MF $\mu_{\widetilde{X}}(x,u)$, i.e.,

$$\tilde{X} = \int_{x\in D_{\tilde{X}}} \int_{u\in J_x\subseteq[0,1]} \mu_{\tilde{X}}(x,u)/(x,u) = \int_{x\in D_{\tilde{X}}} \int_{u\in J_x\subseteq[0,1]} 1/(x,u) = \int_{x\in D_{\tilde{X}}} \left[\int_{u\in J_x\subseteq[0,1]} 1/u \right] \Big/ x \qquad (2)$$

where x, called the *primary variable*, has domain $D_{\tilde{X}}$; $u \in [0,1]$, called the *secondary variable*, has domain $J_x \subseteq [0,1]$ at each $x \in D_{\tilde{X}}$; J_x is also called the *support of the secondary MF*; and the amplitude of $\mu_{\tilde{X}}(x,u)$, called a *secondary grade* of $\tilde{X}$, equals 1 for $\forall x \in D_{\tilde{X}}$ and $\forall u \in J_x \subseteq [0,1]$. □

An example of an IT2 FS, $\widetilde{X}$, is shown in Fig. 1. Observe that unlike a T1 FS, whose membership grade for each x is a number, the membership of an IT2 FS is an interval. Observe also that an IT2 FS is bounded from above and below by two T1 FSs, $\overline{X}$ and $\underline{X}$, which are called *upper membership function* (UMF) and *lower membership function* (LMF), respectively. The area between $\overline{X}$ and $\underline{X}$ is the *footprint of uncertainty* (FOU). An *embedded T1 FS* is any T1 FS within the FOU. $\underline{X}$ and $\overline{X}$ are two such sets.

2.2 Interval Type-2 Fuzzy Logic Controllers (IT2 FLCs)

Figure 2 shows the schematic diagram of an IT2 FLC. It is similar to its T1 counterpart, the major difference being that at least one of the FSs in the rulebase is an IT2 FS. Hence, the outputs of the inference engine are IT2 FSs, and a type-reducer [8, 10] is needed to convert them into a T1 FS before defuzzification can be carried out.

In practice the computations in an IT2 FLC can be significantly simplified. Consider the rulebase of an IT2 FLC consisting of N rules assuming the following form:

$$\widetilde{R}^n : \text{IF } x_1 \text{ is } \widetilde{X}_1^n \text{ and } \cdots \text{ and } x_I \text{ is } \widetilde{X}_I^n, \text{ THEN } y \text{ is } Y^n. \; n = 1, 2, \ldots, N$$

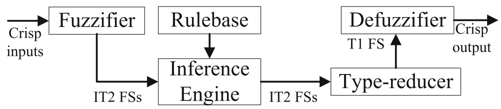


Fig. 2 The schematic diagram of an IT2 FLC

where $\widetilde{X}_i^n$ $(i = 1, \ldots, I)$ are IT2 FSs, and $Y^n = [\underline{y}^n, \overline{y}^n]$ is an interval, which can be understood as the centroid [7, 10] of a consequent IT2 FS,[2] or the simplest TSK model. In many applications [20, 22, 23] we use $\underline{y}^n = \overline{y}^n$, i.e., each rule consequent is represented by a crisp number.

For an input vector $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_I)$, typical computations in an IT2 FLC involve the following steps:

1. Compute the membership interval of x'_i on each X_i^n, $[\mu_{\underline{X}_i^n}(x'_i), \mu_{\overline{X}_i^n}(x'_i)]$, $i = 1, 2, \ldots, I$, $n = 1, 2, \ldots, N$.
2. Compute the firing interval of the n^{th} rule, $F^n(\mathbf{x}')$:

$$F^n(\mathbf{x}') = [\mu_{\underline{X}_1^n}(x'_1) \times \cdots \times \mu_{\underline{X}_I^n}(x'_I), \ \mu_{\overline{X}_1^n}(x'_1) \times \cdots \times \mu_{\overline{X}_I^n}(x'_I)] \equiv [\underline{f}^n, \overline{f}^n] \tag{3}$$

   Note that the *minimum* t-norm may also be used in (3). However, this chapter focuses only on the *product* t-norm.
3. Perform type-reduction to combine $F^n(\mathbf{x}')$ and the corresponding rule consequents. There are many such methods [10]. The most commonly used one is the center-of-sets type-reducer [10], derived from the Extension Principle [25]:

$$Y_{\cos}(\mathbf{x}') = \bigcup_{\substack{f^n \in F^n(\mathbf{x}') \\ y^n \in Y^n}} \frac{\sum_{n=1}^N f^n y^n}{\sum_{n=1}^N f^n} = [y_l, \ y_r] \tag{4}$$

It has been shown that [10, 13, 18]:

$$y_l = \min_{k \in [1, N-1]} \frac{\sum_{n=1}^k \overline{f}^n \underline{y}^n + \sum_{n=k+1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^k \overline{f}^n + \sum_{n=k+1}^N \underline{f}^n} \equiv \frac{\sum_{n=1}^L \overline{f}^n \underline{y}^n + \sum_{n=L+1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^L \overline{f}^n + \sum_{n=L+1}^N \underline{f}^n} \tag{5}$$

$$y_r = \max_{k \in [1, N-1]} \frac{\sum_{n=1}^k \underline{f}^n \overline{y}^n + \sum_{n=k+1}^N \overline{f}^n \overline{y}^n}{\sum_{n=1}^k \underline{f}^n + \sum_{n=k+1}^N \overline{f}^n} \equiv \frac{\sum_{n=1}^R \underline{f}^n \overline{y}^n + \sum_{n=R+1}^N \overline{f}^n \overline{y}^n}{\sum_{n=1}^R \underline{f}^n + \sum_{n=R+1}^N \overline{f}^n} \tag{6}$$

where the *switch points* L and R are determined by

$$\underline{y}^L \leq y_l \leq \underline{y}^{L+1} \tag{7}$$

[2] The rule consequents can be IT2 FSs; however, when the popular center-of-sets type-reduction method [10] is used, these consequent IT2 FSs are replaced by their centroids in the computation; so, it is more convenient to represent the rule consequents as intervals directly.

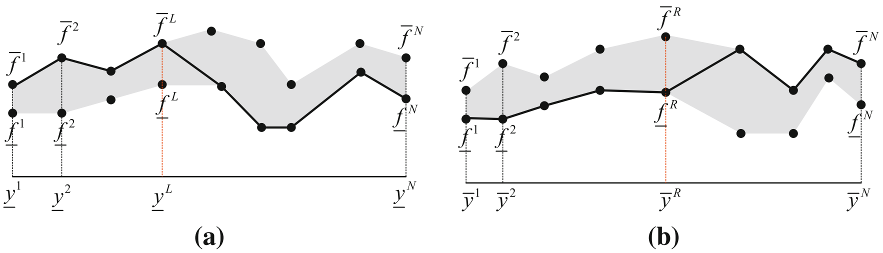


Fig. 3 Illustration of the switch points in computing y_l and y_r. (**a**) Computing y_l switch from the *upper* bounds of the firing intervals to the *lower* bounds. (**b**) Computing y_r switch from the *lower* bounds of the firing intervals to the *upper* bounds

$$\bar{y}^R \leq y_r \leq \bar{y}^{R+1} \tag{8}$$

and $\{\underline{y}^n\}_{n=1,\ldots,N}$ and $\{\bar{y}^n\}_{n=1,\ldots,N}$ have been sorted in ascending order, respectively. y_l and y_r can be computed by the KM algorithms [8, 10] or their many variants [18, 19]. The main idea of the KM algorithms is to find the switch points for y_l and y_r. Take y_l as an example. y_l is the minimum of $Y_{\cos}(\mathbf{x}')$. Since $\underline{y}^n$ increases from the left to the right along the horizontal axis of Fig. 3a, we should choose a large weight (upper bound of the firing interval) for $\underline{y}^n$ on the left and a small weight (lower bound of the firing interval) for $\underline{y}^n$ on the right. The KM algorithm for y_l finds the switch point L. For $n \leq L$, the upper bounds of the firing intervals are used to calculate y_l; for $n > L$, the lower bounds are used. This ensures y_l is the minimum.

4. Compute the defuzzified output as:

$$y = \frac{y_l + y_r}{2}. \tag{9}$$

3 Examples of IT2 FLC

A pair of T1 and IT2 PI FLCs are used in this section to illustrate the differences between them.

3.1 The T1 and IT2 PI FLCs

The MFs of the T1 PI FLC are shown in Fig. 4 as the bold dashed lines, where the standard deviation of all Gaussian MFs is 0.6. Its four rules are:

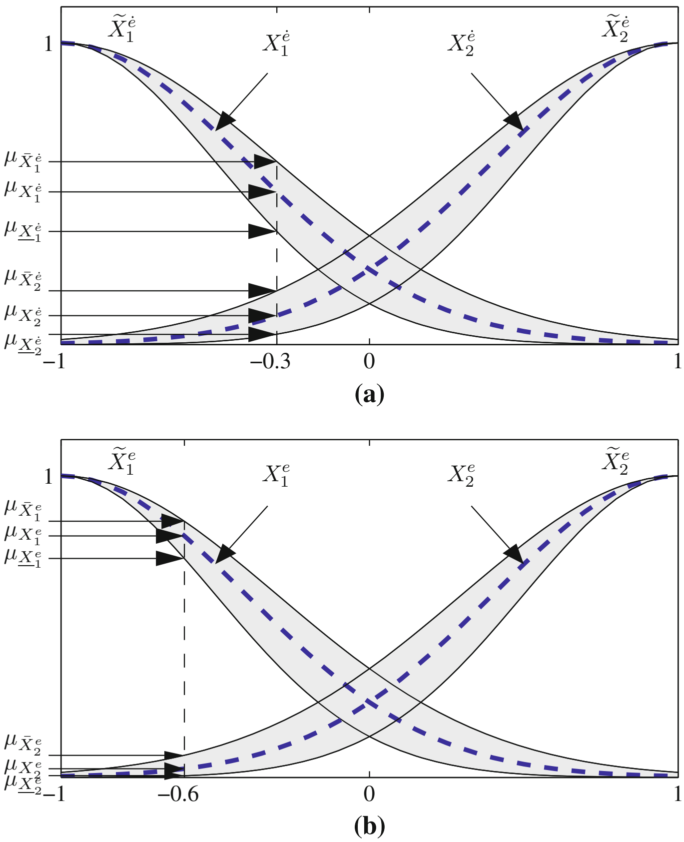


Fig. 4 Firing levels of the T1 FLC, and firing intervals of the IT2 FLC, when $\mathbf{x}' = (\dot{e}', e') = (-0.3, -0.6)$

$$R^1: \quad \text{IF } \dot{e} \text{ is } X_1^{\dot{e}} \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y^1.$$
$$R^2: \quad \text{IF } \dot{e} \text{ is } X_1^{\dot{e}} \text{ and } e \text{ is } X_2^e, \text{ THEN } \dot{u} \text{ is } y^2.$$
$$R^3: \quad \text{IF } \dot{e} \text{ is } X_2^{\dot{e}} \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y^3.$$
$$R^4: \quad \text{IF } \dot{e} \text{ is } X_2^{\dot{e}} \text{ and } e \text{ is } X_2^e, \text{ THEN } \dot{u} \text{ is } y^4.$$

where $\dot{u}$ is the *change of the control signal*, e is the *feedback error*, and $\dot{e}$ is the *change of error*. $y^1 - y^4$ are given in Table 1. For simplicity the rule consequents are crisp numbers instead of intervals. However, they can be arbitrary numbers or intervals and do not affect the conclusions of this chapter.

Table 1 The rule consequents of the T1 and IT2 FLCs

	X_1^e ($\widetilde{X}_1^e$)	X_2^e ($\widetilde{X}_2^e$)
$X_1^{\dot{e}}$ ($\widetilde{X}_1^{\dot{e}}$)	$y^1 = -1$	$y^2 = -0.5$
$X_2^{\dot{e}}$ ($\widetilde{X}_2^{\dot{e}}$)	$y^3 = 0.5$	$y^4 = 1$

An IT2 fuzzy PI controller may be constructed by blurring the T1 FSs of the T1 FLC to IT2 FSs.[3] In this chapter, we blur the standard deviation of the Gaussian MFs from 0.6 to an interval [0.5, 0.7], as shown in Fig. 4. The rulebase of the IT2 FLC is

$$\begin{aligned}
\widetilde{R}^1 &: \text{ IF } \dot{e} \text{ is } \widetilde{X}_1^{\dot{e}} \text{ and } e \text{ is } \widetilde{X}_1^e, \text{ THEN } \dot{u} \text{ is } y^1.\\
\widetilde{R}^2 &: \text{ IF } \dot{e} \text{ is } \widetilde{X}_1^{\dot{e}} \text{ and } e \text{ is } \widetilde{X}_2^e, \text{ THEN } \dot{u} \text{ is } y^2.\\
\widetilde{R}^3 &: \text{ IF } \dot{e} \text{ is } \widetilde{X}_2^{\dot{e}} \text{ and } e \text{ is } \widetilde{X}_1^e, \text{ THEN } \dot{u} \text{ is } y^3.\\
\widetilde{R}^4 &: \text{ IF } \dot{e} \text{ is } \widetilde{X}_2^{\dot{e}} \text{ and } e \text{ is } \widetilde{X}_2^e, \text{ THEN } \dot{u} \text{ is } y^4.
\end{aligned}$$

Next, the mathematical operations in an IT2 FLC, introduced in Sect. 2.2, are illustrated using two numerical examples, which will be revisited in Sect. 4.

3.2 *Example 1*

Consider an input vector $\mathbf{x}' = (\dot{e}',\ e') = (-0.3, -0.6)$, as shown in Fig. 4. The firing levels of the four T1 FSs are:

$$\mu_{X_1^{\dot{e}}}(\dot{e}') = 0.5063, \quad \mu_{X_2^{\dot{e}}}(\dot{e}') = 0.0956, \quad \mu_{X_1^e}(e') = 0.8007, \quad \mu_{X_2^e}(e') = 0.0286$$

The firing levels of their four rules are shown in Table 2. The output of the T1 FLC is

$$\dot{u} = \frac{f^1y^1 + f^2y^2 + f^3y^3 + f^4y^4}{f^1 + f^2 + f^3 + f^4} = -0.3886.$$

For the IT2 FLC, the firing intervals of the four IT2 FSs are:

$$\begin{aligned}
[\mu_{\underline{X}_1^{\dot{e}}}(\dot{e}'), \mu_{\overline{X}_1^{\dot{e}}}(\dot{e}')] &= [0.3753, 0.6065], \quad [\mu_{\underline{X}_2^{\dot{e}}}(\dot{e}'), \mu_{\overline{X}_2^{\dot{e}}}(\dot{e}')] = [0.0340, 0.1783]\\
[\mu_{\underline{X}_1^e}(e'), \mu_{\overline{X}_1^e}(e')] &= [0.7261, 0.8494], \quad [\mu_{\underline{X}_2^e}(e'), \mu_{\overline{X}_2^e}(e')] = [0.0060, 0.0734]
\end{aligned}$$

The firing intervals of the four rules are shown in Table 3. From the KM algorithms, we find that $L = 1$ and $R = 2$. So,

[3] An IT2 FLC can also be constructed from scratch without using a baseline T1 FLC [22, 23]. This chapter uses a baseline T1 FLC for comparison purposes.

Table 2 Firing levels of the four rules of the T1 FLC in Example 1

Rule no.:	Firing level	→	Rule consequent
R^1 :	$f^1 = \mu_{X_1^{\dot{e}}}(\dot{e}') \cdot \mu_{X_1^e}(e') = 0.5063 \times 0.8007 = 0.4054$	→	$y^1 = -1$
R^2 :	$f^2 = \mu_{X_1^{\dot{e}}}(\dot{e}') \cdot \mu_{X_2^e}(e') = 0.5063 \times 0.0286 = 0.0484$	→	$y^2 = -0.5$
R^3 :	$f^3 = \mu_{X_2^{\dot{e}}}(\dot{e}') \cdot \mu_{X_1^e}(e') = 0.0956 \times 0.8007 = 0.0766$	→	$y^3 = 0.5$
R^4 :	$f^4 = \mu_{X_2^{\dot{e}}}(\dot{e}') \cdot \mu_{X_2^e}(e') = 0.0956 \times 0.0286 = 0.0027$	→	$y^4 = 1$

Table 3 Firing intervals of the four rules of the IT2 FLC in Example 1

Rule no.:	Firing interval	→	Rule consequent
$\widetilde{R}^1$:	$[\underline{f}^1, \overline{f}^1] = [\mu_{\underline{X}_1^{\dot{e}}}(\dot{e}') \cdot \mu_{\underline{X}_1^e}(e'), \mu_{\overline{X}_1^{\dot{e}}}(\dot{e}') \cdot \mu_{\overline{X}_1^e}(e')]$ $= [0.3753 \times 0.7261, 0.6065 \times 0.8494] = [0.2725,\ 0.5152]$	→	$y^1 = -1$
$\widetilde{R}^2$:	$[\underline{f}^2, \overline{f}^2] = [\mu_{\underline{X}_1^{\dot{e}}}(\dot{e}') \cdot \mu_{\underline{X}_2^e}(e'), \mu_{\overline{X}_1^{\dot{e}}}(\dot{e}') \cdot \mu_{\overline{X}_2^e}(e')]$ $= [0.3753 \times 0.0060, 0.6065 \times 0.0734] = [0.0022,\ 0.0445]$	→	$y^2 = -0.5$
$\widetilde{R}^3$:	$[\underline{f}^3, \overline{f}^3] = [\mu_{\underline{X}_2^{\dot{e}}}(\dot{e}') \cdot \mu_{\underline{X}_1^e}(e'), \mu_{\overline{X}_2^{\dot{e}}}(\dot{e}') \cdot \mu_{\overline{X}_1^e}(e')]$ $= [0.0340 \times 0.7261, 0.1783 \times 0.8494] = [0.0247,\ 0.1514]$	→	$y^3 = 0.5$
$\widetilde{R}^4$:	$[\underline{f}^4, \overline{f}^4] = [\mu_{\underline{X}_2^{\dot{e}}}(\dot{e}') \cdot \mu_{\underline{X}_2^e}(e'), \mu_{\overline{X}_2^{\dot{e}}}(\dot{e}') \cdot \mu_{\overline{X}_2^e}(e')]$ $= [0.0340 \times 0.0060, 0.1783 \times 0.0734] = [0.0002,\ 0.0131]$	→	$y^4 = 1$

$$
\begin{aligned}
y_l &= \frac{\overline{f}^1 y^1 + \underline{f}^2 y^2 + \underline{f}^3 y^3 + \underline{f}^4 y^4}{\overline{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4} \\
&= \frac{0.5152 \times (-1) + 0.0022 \times (-0.5) + 0.0247 \times 0.5 + 0.0002 \times 1}{0.5152 + 0.0022 + 0.0247 + 0.0002} \\
&= -0.9288 \\
y_r &= \frac{\underline{f}^1 y^1 + \underline{f}^2 y^2 + \overline{f}^3 y^3 + \overline{f}^4 y^4}{\underline{f}^1 + \underline{f}^2 + \overline{f}^3 + \overline{f}^4} \\
&= \frac{0.2725 \times (-1) + 0.0022 \times (-0.5) + 0.1514 \times 0.5 + 0.0131 \times 1}{0.2725 + 0.0022 + 0.1514 + 0.0131} \\
&= -0.4209
\end{aligned}
\tag{10}
$$

Finally, the crisp output of the IT2 FLC is:

$$
\dot{u} = \frac{y_l + y_r}{2} = \frac{-0.9288 - 0.4209}{2} = -0.6748.
$$

Observe from the above example that for the same input, IT2 and T1 FLCs give quite different outputs.

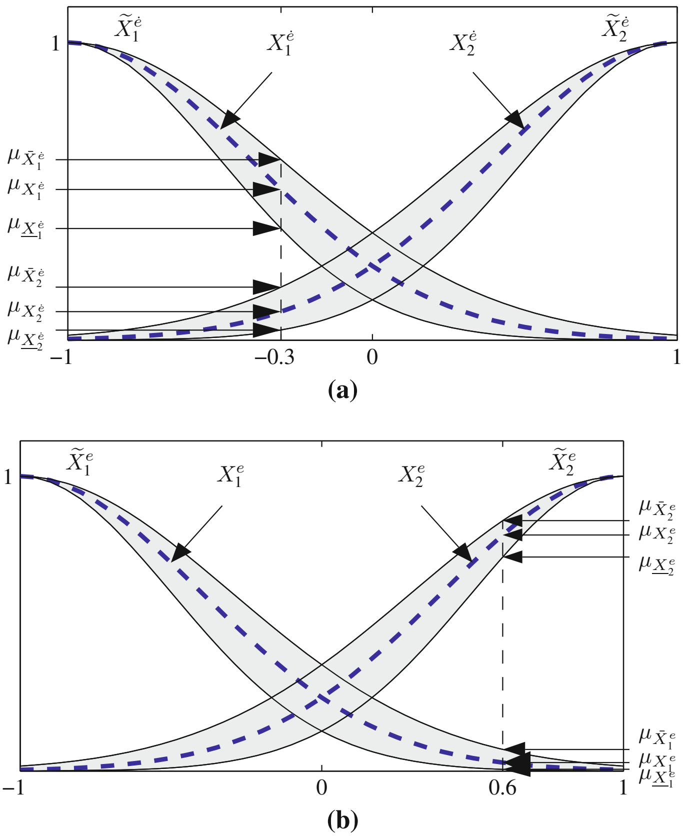


Fig. 5 Firing levels of the T1 FLC, and firing intervals of the IT2 FLC, when $\mathbf{x}' = (\dot{e}', e') = (-0.3, 0.6)$

3.3 Example 2

Consider another input vector $\mathbf{x}' = (\dot{e}',\ e') = (-0.3, 0.6)$, as shown in Fig. 5. The firing levels of the four T1 FSs are:

$$\mu_{X_1^{\dot{e}}}(\dot{e}') = 0.5063, \quad \mu_{X_2^{\dot{e}}}(\dot{e}') = 0.0956, \quad \mu_{X_1^{e}}(e') = 0.0286, \quad \mu_{X_2^{e}}(e') = 0.8007$$

The firing levels of its four rules are shown in Table 4. The output of the T1 FLC is

$$\dot{u} = \frac{f^1y^1 + f^2y^2 + f^3y^3 + f^4y^4}{f^1 + f^2 + f^3 + f^4} = 0.0393.$$

Table 4 Firing levels of the four rules of the T1 FLC in Example 2

Rule no.:	Firing level	→	Rule consequent
R^1 :	$f^1 = \mu_{X_1^{\dot{e}}}(\dot{e}') \cdot \mu_{X_1^e}(e') = 0.5063 \times 0.0286 = 0.0145$	→	$y^1 = -1$
R^2 :	$f^2 = \mu_{X_1^{\dot{e}}}(\dot{e}') \cdot \mu_{X_2^e}(e') = 0.5063 \times 0.8007 = 0.0484$	→	$y^2 = -0.5$
R^3 :	$f^3 = \mu_{X_2^{\dot{e}}}(\dot{e}') \cdot \mu_{X_1^e}(e') = 0.0956 \times 0.0286 = 0.0027$	→	$y^3 = 0.5$
R^4 :	$f^4 = \mu_{X_2^{\dot{e}}}(\dot{e}') \cdot \mu_{X_2^e}(e') = 0.0956 \times 0.8007 = 0.0766$	→	$y^4 = 1$

Table 5 Firing intervals of the four rules of the IT2 FLC in Example 2

Rule no.:	Firing interval	→	Rule consequent
$\widetilde{R}^1$:	$[\underline{f}^1, \overline{f}^1] = [\mu_{\underline{X}_1^{\dot{e}}}(\dot{e}') \cdot \mu_{\underline{X}_1^e}(e'), \mu_{\overline{X}_1^{\dot{e}}}(\dot{e}') \cdot \mu_{\overline{X}_1^e}(e')]$ $= [0.3753 \times 0.0060, 0.6065 \times 0.0734] = [0.0022, 0.0445]$	→	$y^1 = -1$
$\widetilde{R}^2$:	$[\underline{f}^2, \overline{f}^2] = [\mu_{\underline{X}_1^{\dot{e}}}(\dot{e}') \cdot \mu_{\underline{X}_2^e}(e'), \mu_{\overline{X}_1^{\dot{e}}}(\dot{e}') \cdot \mu_{\overline{X}_2^e}(e')]$ $= [0.3753 \times 0.7261, 0.6065 \times 0.8494] = [0.2725, 0.5152]$	→	$y^2 = -0.5$
$\widetilde{R}^3$:	$[\underline{f}^3, \overline{f}^3] = [\mu_{\underline{X}_2^{\dot{e}}}(\dot{e}') \cdot \mu_{\underline{X}_1^e}(e'), \mu_{\overline{X}_2^{\dot{e}}}(\dot{e}') \cdot \mu_{\overline{X}_1^e}(e')]$ $= [0.0340 \times 0.0060, 0.1783 \times 0.0734] = [0.0002, 0.0131]$	→	$y^3 = 0.5$
$\widetilde{R}^4$:	$[\underline{f}^4, \overline{f}^4] = [\mu_{\underline{X}_2^{\dot{e}}}(\dot{e}') \cdot \mu_{\underline{X}_2^e}(e'), \mu_{\overline{X}_2^{\dot{e}}}(\dot{e}') \cdot \mu_{\overline{X}_2^e}(e')]$ $= [0.0340 \times 0.7261, 0.1783 \times 0.8494] = [0.0247, 0.1514]$	→	$y^4 = 1$

For the IT2 FLC, the firing intervals of the four IT2 FSs are:

$$[\mu_{\underline{X}_1^{\dot{e}}}(\dot{e}'),\ \mu_{\overline{X}_1^{\dot{e}}}(\dot{e}')] = [0.3753,\ 0.6065], \quad [\mu_{\underline{X}_2^{\dot{e}}}(\dot{e}'),\ \mu_{\overline{X}_2^{\dot{e}}}(\dot{e}')] = [0.0340,\ 0.1783]$$
$$[\mu_{\underline{X}_1^{e}}(e'),\ \mu_{\overline{X}_1^{e}}(e')] = [0.0060,\ 0.0734], \quad [\mu_{\underline{X}_2^{e}}(e'),\ \mu_{\overline{X}_2^{e}}(e')] = [0.7261,\ 0.8494]$$

The firing intervals of the four rules are show in Table 5. From the KM algorithms, we find that $L = 2$ and $R = 2$. So,

$$\begin{aligned} y_l &= \frac{\overline{f}^1 y^1 + \overline{f}^2 y^2 + \underline{f}^3 y^3 + \underline{f}^4 y^4}{\overline{f}^1 + \overline{f}^2 + \underline{f}^3 + \underline{f}^4} \\ &= \frac{0.0445 \times (-1) + 0.5152 \times (-0.5) + 0.0002 \times 0.5 + 0.0247 \times 1}{0.0445 + 0.5152 + 0.0002 + 0.0247} \\ &= -0.4743 \\ y_r &= \frac{\underline{f}^1 y^1 + \underline{f}^2 y^2 + \overline{f}^3 y^3 + \overline{f}^4 y^4}{\underline{f}^1 + \underline{f}^2 + \overline{f}^3 + \overline{f}^4} \\ &= \frac{0.0022 \times (-1) + 0.2725 \times (-0.5) + 0.0131 \times 0.5 + 0.1514 \times 1}{0.0022 + 0.2725 + 0.0131 + 0.1514} \\ &= 0.0443 \end{aligned} \tag{11}$$

Finally, the crisp output of the IT2 FLC is:

$$\dot{u} = \frac{y_l + y_r}{2} = \frac{-0.4743 + 0.0443}{2} = -0.2150.$$

Observe again from the above example that for the same input, IT2 and T1 FLCs give quite different outputs. The next section explains the fundamental reasons behind this difference.

4 Fundamental Differences Between IT2 and T1 FLCs

Observe from (9), and also Examples 1 and 2, that the output of an IT2 FLC is the average of two "T1 FLCs". However, these two "T1 FLCs" are *fundamentally* different from traditional T1 FLCs, for the following reasons [16, 17]:

1. **Adaptiveness**, meaning that the embedded T1 FSs used to compute the bounds of the type-reduced interval change as input changes. Take y_l in (10) and (11) as an example. The firing levels of the four rules in (10) are $\bar{f}^1$, $\underline{f}^2$, $\underline{f}^3$, and $\underline{f}^4$, respectively, which are computed from different lower and upper MFs, as shown in the first part of Table 6 and Fig. 6a. The firing levels of the four rules in (11) are shown in the second part of Table 6 and Fig. 6b. Comparing the two parts of Table 6, and the two sub-figures in Fig. 6, we can observe that when the input $(\dot{e}', e')$ changes from $(-0.3, -0.6)$ to $(-0.3, 0.6)$, different embedded

Table 6 The embedded T1 FSs from which the four firing levels in (10) and (11) are obtained.

		$\widetilde{X}_1^{\dot{e}}$		$\widetilde{X}_2^{\dot{e}}$		$\widetilde{X}_1^{e}$		$\widetilde{X}_2^{e}$	
		UMF	LMF	UMF	LMF	UMF	LMF	UMF	LMF
Equation (10) $(\dot{e}', e') = (-0.3, -0.6)$	$\bar{f}^1$	√				√			
	$\underline{f}^2$		√						√
	$\underline{f}^3$				√		√		
	$\underline{f}^4$				√				√
Equation (11) $(\dot{e}', e') = (-0.3, 0.6)$	$\bar{f}^1$	√				√			
	$\bar{f}^2$	√						√	
	$\underline{f}^3$				√		√		
	$\underline{f}^4$				√				√

Observe that $\underline{f}^2$ is used when $(\dot{e}', e') = (-0.3, -0.6)$ and $\bar{f}^2$ is used when $(\dot{e}', e') = (-0.3, 0.6)$; as a result, different embedded T1 FSs are used in Rule $\widetilde{R}^2$ when the input changes. Observe also that when $(\dot{e}', e') = (-0.3, -0.6)$ both the UMFs and LMFs of $\widetilde{X}_1^{\dot{e}}$ and $\widetilde{X}_1^{e}$ are used in computing y_l, and when $(\dot{e}', e') = (-0.3, 0.6)$ both the UMFs and LMFs of $\widetilde{X}_1^{e}$ and $\widetilde{X}_2^{e}$ are used in computing y_l

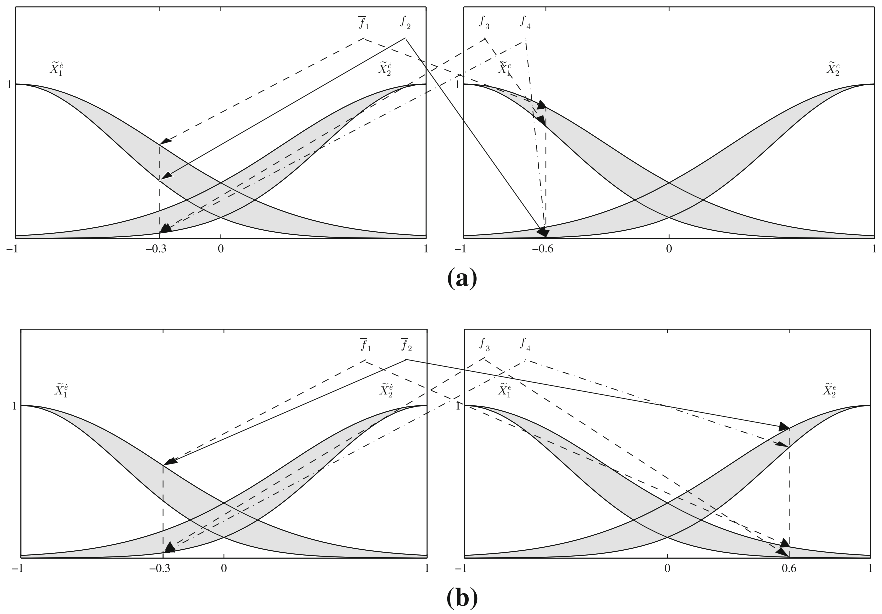


Fig. 6 The embedded T1 FSs used in (**a**) Eq. (10) for computing y_l, where $(\dot{e}', e') = (-0.3, -0.6)$ and the LMFs of $\tilde{X}_1^{\dot{e}}$ and $\tilde{X}_2^{e}$ are used in computing the firing level $\underline{f}_2$ of Rule $\tilde{R}^2$; and, (**b**) Eq. (11) for computing y_l, where $(\dot{e}', e') = (-0.3, 0.6)$ and the UMFs of $\tilde{X}_1^{\dot{e}}$ and $\tilde{X}_2^{e}$ are used in computing the firing level $\overline{f}_2$ of Rule $\tilde{R}^2$. Observe that in (**a**) both the UMFs and LMFs of $\tilde{X}_1^{\dot{e}}$ and $\tilde{X}_1^{e}$ are used in computing y_l, and in (**b**) both the UMFs and LMFs of $\tilde{X}_1^{e}$ and $\tilde{X}_2^{e}$ are used in computing y_l

T1 FSs of $\tilde{X}_1^{\dot{e}}$ and $\tilde{X}_2^{e}$ are used in computing the firing levels for Rule $\tilde{R}^2$ and hence y_l. This adaptiveness is impossible for a T1 FLC since it does not have such embedded T1 FSs.

2. **Novelty**, meaning that the UMF and LMF of the same IT2 FS may be used simultaneously in computing each bound of the type-reduced interval. Observe from the first part of Table 6, and also Fig. 6a, that both the upper and lower MFs of $\tilde{X}_1^{\dot{e}}$ are used in computing y_l, and they are used in different rules: The *UMF* of $\tilde{X}_1^{\dot{e}}$ is used in computing $\overline{f}^1$, the firing level of Rule $\tilde{R}^1$, whereas the *LMF* of $\tilde{X}_1^{\dot{e}}$ is used in computing $\underline{f}^2$, the firing level of Rule $\tilde{R}^2$. Similarly, the upper and lower MFs of $\tilde{X}_1^{e}$ are used simultaneously in different rules for computing y_l. Observe also from the second part of Table 6 and Fig. 6b that the upper and lower MFs of $\tilde{X}_1^{e}$ and $\tilde{X}_2^{e}$ are used simultaneously in different rules for computing y_l. This novelty is again impossible for a T1 FLC because it does not have embedded T1 FSs and the same MFs are always used in computing the firing levels of all rules.

We consider adaptiveness and novelty as two fundamental differences between IT2 and T1 FLCs. Though they are illustrated by specific numerical examples, they are fundamental to an arbitrary IT2 FLC. Furthermore, [17] presents several methods for visualizing and analyzing the effects of these two fundamental differences, including the control surface, the P-map, the equivalent generalized T1 fuzzy sets, and the equivalent PI gains. It also examines five alternative type-reducers for IT2 FLCs and explain why they do not capture the fundamentals of IT2 FLCs.

Theorem 1 [17] *y_l in (5) cannot be implemented by a T1 FLC using the same rulebase.*

Proof In this proof we make use of the following two facts:

- *Fact 1*: The rule firing levels used in the KM algorithms are the bounds of the firing intervals. For an upper bound, all involved embedded T1 FSs must be UMFs, and for a lower bound, all involved embedded T1 FSs must be LMFs. There is no mixture of UMFs and LMFs in computing the firing level of any rule.
- *Fact 2*: $\overline{f}^1$ and $\underline{f}^N$ are always used for computing y_l in (5), though we are not sure about whether the upper or lower firing levels should be used for the rest of the rules. For $\overline{f}^1$, all involved embedded T1 FSs must be UMFs. For $\underline{f}^N$, all involved embedded T1 FSs must be LMFs.

We consider two cases separately:

1. *Rules $\widetilde{R}^1$ and $\widetilde{R}^N$ share at least one IT2 FS $\widetilde{X}_i$*. In this case, according to Fact 2, for Rule $\widetilde{R}^1$ $\overline{X}_i$ must be used, whereas for Rule $\widetilde{R}^N$, $\underline{X}_i$ must be used. This novelty cannot be implemented by a T1 FLC using the same rulebase.
2. *Rules $\widetilde{R}^1$ and $\widetilde{R}^N$ do not have any IT2 FS in common,* (e.g., for y_l in (11), $\widetilde{R}^1$ involves $\widetilde{X}_1^{\dot{e}}$ and $\widetilde{X}_1^{e}$, whereas $\widetilde{R}^4$ involves $\widetilde{X}_2^{\dot{e}}$ and $\widetilde{X}_2^{e}$). This case is more complicated than the previous one. We prove it by contradiction. Assume y_l in (5) can be implemented by a T1 FLC, where the same T1 MFs are used in computing all firing levels, e.g., if the UMF of $\widetilde{X}_1^{\dot{e}}$ is used in computing the firing level of Rule $\widetilde{R}^1$, it must also be used in computing the firing levels of all other rules involving $\widetilde{X}_1^{\dot{e}}$. In the second case, it is always possible to find a Rule $\widetilde{R}^k$ such that Rules $\widetilde{R}^1$ and $\widetilde{R}^k$ share at least one common IT2 FS $\widetilde{X}_i$, and Rules $\widetilde{R}^k$ and $\widetilde{R}^N$ share at least one common IT2 FS $\widetilde{X}_j$ (e.g., for y_l in (11), Rules $\widetilde{R}^1$ and $\widetilde{R}^2$ share $\widetilde{X}_1^{\dot{e}}$, and Rules $\widetilde{R}^2$ and $\widetilde{R}^4$ share $\widetilde{X}_2^{e}$). According to Fact 2, $\overline{X}_i$ must be used in Rule $\widetilde{R}^1$ for computing $\overline{f}^1$. If y_l can be implemented by a T1 FLC using the same rulebase, then $\overline{X}_i$ must also be used in Rule $\widetilde{R}^k$. According to Fact 1, $\overline{X}_j$ must also be used for Rule $\widetilde{R}^k$. For a T1 FLC, this means $\overline{X}_j$ must also be used in Rule $\widetilde{R}^N$, which is a contradiction with Fact 2. So, again y_l in (5) cannot be implemented by a T1 FLC using the same rulebase. □

Theorem 2 [17] *y_r in* (6) *cannot be implemented by a T1 FLC using the same rulebase.*

The proof is very similar to that for Theorem 1, so it is omitted.

Based on Theorems 1 and 2, we can easily reach the following conclusion:

Theorem 3 [17] *An IT2 FLC using the KM type-reducer cannot be implemented by a T1 FLC using the same rulebase.*

Theorem 3 is very helpful in understanding why IT2 FLCs may outperform T1 FLCs. It suggests that an IT2 FLC can implement a more complex control surface than a T1 FLC: when there is no FOU, an IT2 FLC collapses to a T1 FLC; with FOU, an IT2 FLC can implement a control surface that cannot be obtained from a T1 FLC using the same rulebase. Note that Theorem 3 does not conflict with the fact that T1 fuzzy logic systems are universal approximators [2, 9]: Being a universal approximator requires a T1 fuzzy logic system to have an arbitrarily large number of MFs, whereas in this chapter we only consider IT2 and T1 FLCs with the same rulebase and a fixed (small) number of MFs.

5 Conclusions

IT2 FLCs have been widely used and demonstrated better ability to handle uncertainties than their T1 counterparts. A challenging question is what the fundamental differences are between IT2 and T1 FLCs. Once the fundamental differences are clear, we can better understand the advantages of IT2 FLCs and hence better make use of them. In this chapter, we have explained two fundamental differences between IT2 and T1 FLCs: 1) *Adaptiveness*, meaning that the embedded T1 FSs used to compute the bounds of the type-reduced interval change as input changes; and, 2) *Novelty*, meaning that the UMF and LMF of the same IT2 FS may be used simultaneously in computing each bound of the type-reduced interval. As a result, an IT2 FLC can implement a complex control surface that cannot be achieved by a T1 FLC using the same rulebase.

References

1. Castillo, O., Melin, P.: Type-2 Fuzzy Logic Theory and Applications. Springer-Verlag, Berlin (2008)
2. Castro, J.L.: Fuzzy logic controllers are universal approximators. IEEE Trans. Syst. Man Cybern. **25**(4), 629–635 (1995)
3. Du, X., Ying, H.: Derivation and analysis of the analytical structures of the interval type-2 fuzzy-PI and PD controllers. IEEE Trans. Fuzzy Syst. **18**(4), 802–814 (2010)

4. Hagras, H.: A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Trans. Fuzzy Syst. **12**, 524–539 (2004)
5. Hagras, H.: Type-2 FLCs: a new generation of fuzzy controllers. IEEE Comput. Intell. Mag. **2**(1), 30–43 (2007)
6. Jammeh, E.A., Fleury, M., Wagner, C., Hagras, H., Ghanbari, M.: Interval type-2 fuzzy logic congestion control for video streaming across IP networks. IEEE Trans. Fuzzy Syst. **17**(5), 1123–1142 (2009)
7. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Inf. Sci. **132**, 195–220 (2001)
8. Karnik, N.N., Mendel, J.M., Liang, Q.: Type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **7**, 643–658 (1999)
9. Kosko, B.: Fuzzy systems as universal approximators. IEEE Trans. Comput. **43**(11), 1329–1333 (1994)
10. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
11. Mendel, J.M.: Computing with words: Zadeh, Turing, Popper and Occam. IEEE Comput. Intell. Mag. **2**, 10–17 (2007)
12. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
13. Mendel, J.M., Wu, D.: Perceptual Computing: Aiding People in Making Subjective Judgments. Wiley-IEEE Press, Hoboken (2010)
14. Nie, M., Tan, W.: Analytical structure and characteristics of symmetric centroid type-reduced interval type-2 fuzzy PI and PD controllers. IEEE Trans. Fuzzy Syst. **20**(3), 416–430 (2012)
15. Wallsten, T.S., Budescu, D.V.: A review of human linguistic probability processing: general principles and empirical evidence. Knowl. Eng. Rev. **10**(1), 43–62 (1995)
16. Wu, D.: An interval type-2 fuzzy logic system cannot be implemented by traditional type-1 fuzzy logic systems. In: Proceedings of World Conference on Soft Computing, San Francisco (2011)
17. Wu, D.: On the fundamental differences between interval type-2 and type-1 fuzzy logic controllers. IEEE Trans. Fuzzy Syst. **20**(5), 832–848 (2012)
18. Wu, D., Mendel, J.M.: Enhanced Karnik-Mendel algorithms. IEEE Trans. Fuzzy Syst. **17**(4), 923–934 (2009)
19. Wu, D., Nie, M.: Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems. In: Proceedings of IEEE International Conference on Fuzzy Systems, pp. 2131–2138. Taipei, Taiwan (2011)
20. Wu, D., Tan, W.W.: A type-2 fuzzy logic controller for the liquid-level process. In: Proceedings of IEEE International Conference on Fuzzy Systems, vol. 2, pp. 953–958. Budapest, Hungary (2004)
21. Wu, D., Tan, W.W.: Type-2 FLS modeling capability analysis. In: Proceedings of IEEE International Conference on Fuzzy Systems, pp. 242–247. Reno, NV (2005)
22. Wu, D., Tan, W.W.: Genetic learning and performance evaluation of type-2 fuzzy logic controllers. Eng. Appl. Artif. Intell. **19**(8), 829–841 (2006)
23. Wu, D., Tan, W.W.: A simplified type-2 fuzzy controller for real-time control. ISA Trans. **15**(4), 503–516 (2006)
24. Wu, D., Tan, W.W.: Interval type-2 fuzzy PI controllers: why they are more robust. In: Proceedings of IEEE International Conference on Granular Computing, pp. 802–807. San Jose, CA (2010)
25. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
26. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning-1. Inf. Sci. **8**, 199–249 (1975)

Interval Type-2 Fuzzy Markov Chains

Juan Carlos Figueroa-García

Abstract Uncertainties in fuzzy Markov chains can be treated in different ways. The use of interval type-2 fuzzy sets (IT2FS) allows describing the distributional behavior of an uncertain discrete-time Markov process through infinite type-1 fuzzy sets embedded in its *Footprint of Uncertainty*. In this way, a finite state fuzzy Markov chain process is defined in an interval type-2 fuzzy environment. To do so, its limiting properties and its type-reduced behavior are defined and applied to two explanatory examples.

1 Introduction and Motivation

Markov chains are processes well studied through probabilistic measures (For additional information see Grimmet and Strizaker [12], Ross [30], Gordon [11], and Stewart [34]). A first approach to type-1 fuzzy Markov chains (T1FM) was given by Avrachenkov and Sanchez [2, 3] where they used the max-min operator for finding its stationary behavior.

The use of interval-valued probabilities has been suggested by Araiza et al. [1], Campos et al. [5] and Skulj [33]. In these approaches, the transition probabilities of a Markov chain are defined by a bounded interval. Buckley and Eslami [4], Kurano [17, 18], and Symeonakia [35] applied fuzzy theory to interval-valued probabilities, finding complementary results.

Some approaches to IT2FS were provided by Zeng and Liu [39]. In this study, they used a hybrid of Markov random fields (MRFs) and type-2 fuzzy sets (T2FS) to solve a character recognition problem characterized by ETL-9B and KAIST databases. Zeng and Liu [38] defined an IT2 fuzzy hidden Markov model where

J. C. Figueroa-García (✉)
Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
e-mail: jcfigueroag@udistrital.edu.co

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_4,

the secondary membership function of the IT2FS $f_x(u)$ was defined as a probability distribution. Figueroa [6] defined a pure interval type-2 fuzzy Markov process (IT2FM), which is the starting point for defining some necessary conditions for uncertain T1FS through an IT2FS, and also for defining its primary and its secondary membership functions, namely J_x and $f_x(u)/u = 1/u$.

The following chapter is organized as follows. First, some concepts about type-1 fuzzy Markov chains (T1FM) are given. A brief discussion about uncertainty in fuzzy sets is provided in order to give some definitions about IT2FM. Later, a general procedure for computing the stationary distribution of a T1FM is presented followed by a section where an algorithm to compute the IT2FM stationary distribution is presented and a type-reduction method is proposed. Finally, two application examples are provided, and the concluding remarks of the chapter are presented.

2 Basic Definitions of Fuzzy Markov Chains

As in classical Markov processes analysis, the definition of a fuzzy Markov chain is based on a square relational matrix that represents the conditional possibility that a discrete state at an instant t changes into any state at the next instant $t+1$.

Avrachenkov and Sanchez [3] defined some concepts of fuzzy Markov chains, which are shown next:

Definition 2.1 Let $S = \{1, 2, \ldots, n\}$. A finite fuzzy set for a fuzzy distribution on S is defined by a mapping x from S to $[0, 1]$ represented by a vector $x = \{x_1, x_2, \ldots, x_n\}$, where $0 \leqslant x_i \leqslant 1$, $i \in S$.

In this way, x_i is the membership degree[1] that a state i has regarding a fuzzy set S, $i \in S$ with cardinality m, $\mathscr{C}(S) = m$. In this approach, all operations and relations are defined for fuzzy sets, so their properties and computations are preserved.

Now, a fuzzy relational matrix P is defined in a metric space $S \times S$ by a matrix $\{p_{ij}\}_{i,j=1}^{m}$ where $0 \leqslant p_{ij} \leqslant 1$, $i, j \in S$. The complete set of all fuzzy sets is denoted by $\mathscr{F}(s)$ where $\mathscr{C}(S) = m$.

Definition 2.2 At each instant t, $t = 1, 2, \ldots, n$, the state of the system is described by the fuzzy set[2] $x^{(t)} \in \mathscr{F}(S)$. The transition law of a fuzzy Markov chain is given by the fuzzy relational matrix P at an instant t, $T = 1, 2, \ldots, n$, as follows:

$$x_j^{(t+1)} = \max_{i \in S} \{x_j^{(t)} \wedge p_{ij}\}, \, j \in S. \tag{1}$$

[1] As a function of the i_{th} state e.g. $x(i)$.

[2] This matrix is also known as the *Fuzzy Distribution* of x.

$$x^{(t+1)} = x^{(t)} \circ P \tag{2}$$

where i and j, $i,j = 1,2,\ldots,m$ are the initial and final states of the transition and $x^{(0)}$ is its initial distribution.

In probabilistic Markov chains, it is possible to obtain the steady state of a Marko-vian process based on convergence laws of random variables and/or matrix powers. Now, the powers of a fuzzy transition matrix P can be obtained as follows:

$$p_{ij}^{t} \triangleq \max_{k \in S} \left\{ p_{ik} \wedge p_{kj}^{t-1} \right\} \tag{3}$$

Note that $p_{ij}^{1} = p_{ij}$ and $p_{ij}^{0} = \delta_{ij}$ where δ_{ij} is a *Kronecker Delta*. In matrix form:

$$P^{t} \triangleq P \circ P^{t-1} \tag{4}$$

Specifically, the state $x^{(t)}$ at the instant $t = 1,2,\ldots,n$ can be computed as:

$$x_j^{(t)} = \max_{i \in S} \left\{ x_j^{(0)} \wedge p_{ij}^{t} \right\},\ j \in S \tag{5}$$

Which can be shown in a matrix form:

$$x^{(t)} = x^{(0)} \circ P^{t} \tag{6}$$

In a fuzzy environment, it is also posible that P^{t} never reach a steady state (see Figueroa et al. [7]). On the other hand, Thomason [36] shows that the powers of a fuzzy matrix exhibit a stable behavior if the operator used is the max-min. Moreover, Chin-Tzong Pang [29] used these results to obtain the powers of a fuzzy matrix using the max-archimedean operator. For further information about powers of a fuzzy matrix, see Sánchez [31, 32]. In this way, the following theorem is useful to find the stationary distribution of a fuzzy matrix by using a time convergence criterion.

Theorem 2.1 (Thomason [36]) *The powers of the fuzzy transition matrix* $\{p_{ij}\}_{i,j=1}^{m}$ *either converge to idempotent* $\{p_{ij}^{\tau}\}_{i,j=1}^{m}$ *with elements* π_j *where* $\tau \leqslant n$, *or oscillate with a finite period* υ*starting from some finite power.*

Consequently, the *Stationary Distribution* of a fuzzy markov chain is defined as follows:

Definition 2.3 *(Stationary Distribution)* Let the powers of the fuzzy transition matrix P converge in τ steps to a nonperiodic solution, then the associated fuzzy Markov chain is called aperiodic fuzzy Markov chain and $P^{*} = P^{\tau}$ is its stationary fuzzy transition matrix.

Figueroa et al. [7] found that the use of the max-min operator could lead to a periodical behavior, so they defined two useful properties of a T1FM, Definitions 2.4 and 2.5 below:

Definition 2.4 *(Strong Ergodicity for FM)* A fuzzy Markov chain is called **Strong Ergodic** if it is aperiodic and its stationary transition matrix has identical rows.

Definition 2.5 *(Weak Ergodicity for FM)* A fuzzy Markov chain is called **Weakly Ergodic** if it is aperiodic and its stationary transition matrix is stable with no identical rows.

These definitions imply that any fuzzy markov chain with a stationary distribution given by both an idempotent and aperiodic matrix P^{τ} with *no identical rows* is an *Ergodic* Markov chain on a *weak* sense. In other words:

Proposition 2.2 *Denote P_i^{τ} as the i_{th} row of the stationary distribution of P obtained from its τ powers. If P is strong ergodic then*:

$$P_{i_1}^{\tau} = P_{i_2}^{\tau} \quad \text{for all} \quad i_1 \neq i_2,\ i_1, i_2 \in m, \tag{7}$$

Now, P is Weak Ergodic iff:

$$P_{i_1}^{\tau} \neq P_{i_2}^{\tau} \quad \text{for any} \quad i_1 \neq i_2,\ i_1, i_2 \in m, \tag{8}$$

So any fuzzy markov chain fulfills only one of these two statements.

Now, if the stationary distribution of P is given by $P^* = P^{\tau}$ where $\lim_{n \to \tau} P^n = P^*$, then P becomes an idempotent matrix as described in Theorem 2.1.

Remark 2.3 (Periodical FM) The Definition 2.2 refers to an ergodic FM that has a steady state with the possibility of having nonidentical rows. Another case is a periodical FM, where P^{τ} has a stable but periodic behavior (see Theorem 2.1). This case has been recently treated by Martin Gavalec [8–10] so his results are useful for identifying the period of a fuzzy markov chain.

2.1 From Classical to Uncertain FM

Type-1 fuzzy sets are measures of imprecision based on the perception about a variable through its linguistic label. Uncertain-based models use multiple perceptions about the same linguistic variable, where uncertainty can be treated in two ways: *Linguistic* and *Random* uncertainty. Mendel [24, 25] extends the concept of linguistic uncertainty through type-2 fuzzy sets and its *Footprint of Uncertainty*. In this way, an uncertain FM can be defined with uncertain type-1 fuzzy sets, in other words, an uncertain FM can be defined with type-2 fuzzy sets.

An IT2FS approach involves uncertainty in the sense that it covers different opinions of the experts around $p = \mu_{S_i}(x)$ by using a secondary grade, namely $f_x(u)/u$. This uncertainty is modeled by means of a three-dimensional membership function with point value $(x, u, \mu_{\tilde{S}_i}(x, u))$ where $x \in X$ and $u \in J_x$. Note that J_x is the primary membership function of x defined for an interval boundary of membership values e.g. $J_x = \left[\underline{\mu}_{S_i}(x), \overline{\mu}_{S_i}(x)\right]$.

This bounded interval J_x which encloses an infinite number of type-1 fuzzy sets, represents a large number of possible choices that an expert can use for x, moreover, J_x has embedded the perception of many experts about p and together with their estimates. On the other hand, another source of uncertainty is given by the numerous techniques and methods available to obtain p, which lead to the following question: what is the correct method to define p? In this sense, an IT2 fuzzy sets approach can handle this kind of uncertainty.

3 Definitions for IT2FM

In the last section, a discussion about how to come from classical to interval type-2 fuzzy sets was provided, so we need an appropriate framework for using IT2FS in Markov chains. To do so, all the theory used in this chapter is based on the results of Mendel [14, 16, 19, 23, 26, 27], and Melgarejo [20, 22]. Now, some basic definitions about interval type-2 fuzzy Markov chains are presented next.

Definition 3.1 *(IT2 fuzzy conditional state)* Let $\tilde{P}$ be an Interval Type-2 fuzzy relational matrix defined in $\mathscr{C}(S) \times \mathscr{C}(S)$ with elements $\{\tilde{p}_{ij}\}_{i,j=1}^{m}$, composed by n_a embedded values of $p_{ij}^{J_x}$ in the closed interval $\left[\underline{p}_{ij}, \overline{p}_{ij}\right]$ characterized by a secondary membership function $f_x(u)/u = 1/u$, $J_x \subseteq [0,1]\ \forall\ x \in S, j \in S$. Denote the conditional state $\left\{x^t = j \,|\, x^{t-1} = i\right\}$ as x_{ij}, then we have:

$$\tilde{S}_i = \sum_{x_{ij=1}}^{m} \left[\sum_{u \in J_{x_{ij}}} 1/u\right] \Big/ x_{ij}\ \forall\ i \in S \tag{9}$$

and consequently:

$$J_{x_{ij}} = \left[\sum_{k=1}^{M_j} 1/u_{jk}\right] \Big/ x_{ij}, \quad i,j \in S \tag{10}$$

Note that each $\tilde{S}_i$ is composed by an infinite amount of fuzzy sets, n_a. The union of all those n_a embedded fuzzy sets namely $(e(i))$, is:

$$\tilde{S}_i = \sum_{l=1}^{n_a} \tilde{S}_{e(i)}^{l} \tag{11}$$

where $\tilde{S}_{e(i)}^{l}$ is defined as follows

$$\tilde{S}^l_{e(i)} = \sum_{x \in S} \left[1/u^l_{jk} \right] \Big/ x_{ij}, \quad u_{jk} \in J_{x_{ij}} [0,1] \tag{12}$$

Now, an IT2FS called $\tilde{P}$ can be composed by m individual IT2FS $\tilde{S}_i$, obtaining the following result:

$$\tilde{S}_i = \left\{ (x_{ij}, \mu_{(\tilde{S}_i)}(x_{ij})) \mid x_{ij} = i \right\} \quad i,j \in S \tag{13}$$

This leads us to define a compact form of an IT2FM, whose elements are conditional statements according to the Markovian property, as we present as follows.

Definition 3.2 *(IT2 fuzzy conditional matrix)* Let $\tilde{S}_i$ be a row conditional state of an IT2FS. A squared matrix $\tilde{P}$ of m states is defined as follows:

$$\tilde{P} = \begin{bmatrix} \mu_{(\tilde{S}_1)}(x_{1j}) \\ \mu_{(\tilde{S}_2)}(x_{2j}) \\ \vdots \\ \mu_{(\tilde{S}_m)}(x_{mj}) \end{bmatrix} = \begin{bmatrix} \tilde{p}_{11} & \tilde{p}_{12} & \cdots & \tilde{p}_{1m} \\ \tilde{p}_{21} & \tilde{p}_{22} & \cdots & \tilde{p}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{p}_{m1} & \tilde{p}_{m2} & \cdots & \tilde{p}_{mm} \end{bmatrix} \tag{14}$$

where $\tilde{P}$ can be decomposed into the following two matrices:

$$\tilde{P} = \left[\begin{bmatrix} \underline{p}_{11} & \underline{p}_{12} & \cdots & \underline{p}_{1m} \\ \underline{p}_{21} & \underline{p}_{22} & \cdots & \underline{p}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{p}_{m1} & \underline{p}_{m2} & \cdots & \underline{p}_{mm} \end{bmatrix}, \begin{bmatrix} \overline{p}_{11} & \overline{p}_{12} & \cdots & \overline{p}_{1m} \\ \overline{p}_{21} & \overline{p}_{22} & \cdots & \overline{p}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{p}_{m1} & \overline{p}_{m2} & \cdots & \overline{p}_{mm} \end{bmatrix} \right] \tag{15}$$

Here $\tilde{p}_{ij}$ is the IT2 membership degree that any $x^{(t)} = j$ state reaches regarding the fuzzy set S conditioned to an initial state i, $i,j \in S$ where $\mathscr{C}(S) = m$. Thus, the $\tilde{P}$ IT2 fuzzy transition matrix can be defined by two matrices $\underline{P}$ and $\overline{P}$:

$$\tilde{P} = \left[\underline{P}, \overline{P} \right] \tag{16}$$

The footprint of uncertainty namely *FOU* of an specific set S_i is displayed in Fig. 1. The continuous line shows the FOU where n_a sets $\tilde{S}^l_{e(i)}$ are bounded by two values: ∇ which represents $\overline{p}_{ij}$ and $\triangle$ which represents $\underline{p}_{ij}$.

Now, some useful theorems about the distribution of $\tilde{P}$ are:

Theorem 3.1 Mendel [23]. *The Join $\sqcup_{i=1}^{n} F_i$ of n IT2FS $F_1, F_2, \ldots, F_n$ having domains $[l_1, r_1], [l_2, r_2], \ldots, [l_n, r_n]$, respectively, is an IT2FS with domain $[(l_1 \vee l_2 \vee \ldots \vee l_n), (r_1 \vee r_2 \vee \ldots \vee r_n)]$ where $\vee$ denotes maximum.*

Theorem 3.2 Mendel [23]. *The Meet* $\sqcap_{i=1}^{n} F_i$ *of n IT2FS* $F_1, F_2, \ldots, F_n$ *having domains* $[l_1, r_1], [l_2, r_2], \ldots, [l_n, r_n]$ *, respectively, where* $l_i \geqslant 0$ *and* $r_i \geqslant 0$ *, is an IT2FS with domain* $[(l_1 \star l_2 \star \ldots \star l_n), (r_1 \star r_2 \star \ldots \star r_n)]$ *where* $\star$ *denotes either minimum or product t-norm.*

There exists an infinite amount of T1FM enclosed in the FOU of an IT2FM, which is a natural condition of the process. Moreover, the powers of $\tilde{P}$ namely $\tilde{P}^t$ are bounded by the interval $\underline{P}^t$, $\overline{P}^t$ and any T1FM namely $\breve{P}^t$ embedded in its FOU is enclosed in $\left[\underline{P}^t, \overline{P}^t\right]$, that is $\breve{P}^t \subseteq \left[\underline{P}^t, \overline{P}^t\right]$.

This allows us to define the transition law of an IT2FM as follows.

Theorem 3.3 (Transition law for an IT2FM) *The transition law for an IT2FM is given by fuzzy operations on its IT2 fuzzy matrix* $\tilde{P}$ *at an instant* t, $T = 1, 2, \ldots, n$ *, as follows:*

$$\tilde{x}_j^{(t+1)} = \bigsqcup_{i \in S} \left\{ \tilde{x}_j^{(t)} \sqcap \tilde{p}_{ij} \right\}, j \in S \tag{17}$$

By using the max *t-conorm as the union operator, and the* min *t-norm as the intersection operator, we have*

$$\tilde{x}_j^{(t+1)} = \max_{i \in S} \{ \tilde{x}_j^{(t)} \sqcap \tilde{p}_{ij} \}, j \in S \tag{18}$$

$$\tilde{x}^{(t+1)} = \tilde{x}^{(t)} \circ \tilde{P}^t \tag{19}$$

$$\tilde{p}_{ij}^t \triangleq \max_{k \in S} \left\{ \tilde{p}_{ik} \wedge \tilde{p}_{kj}^{t-1} \right\} \tag{20}$$

where $\tilde{x}^{(0)}$ *is the IT2 fuzzy initial distribution of* $\{x_t\}$.

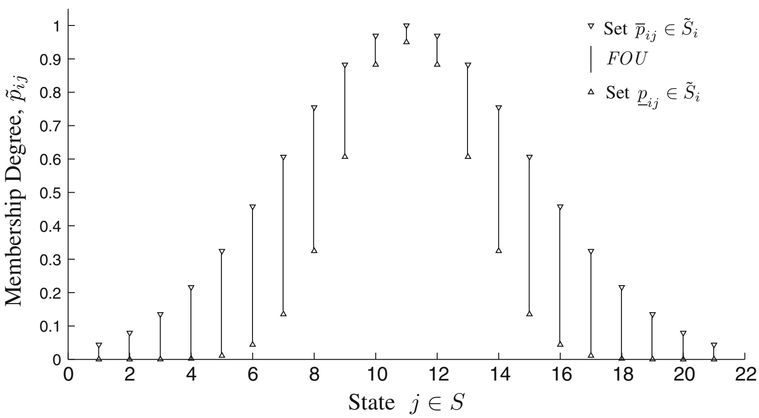


Fig. 1 Footprint of uncertainty of an IT2FM

Remark 3.4 Usually $\tilde{x}^{(0)}$ is unknown, so a common approach is to use the following supposition:

$$\tilde{x}^{(0)} = \underline{p}_{ij} = \overline{p}_{ij} = 1 \text{ for } i = j, \text{ and } 0 \text{ for } i \neq j$$

By extension of Definition 2.3, it is possible to define the following statements:

Proposition 3.5 *The powers of an IT2FM transition matrix* $\{\tilde{p}_{ij}\}_{i,j=1}^{m}$ *either converge to an idempotent* $\{\tilde{p}_{ij}^{\tau}\}_{i,j=1}^{m}$ *where* $\tau \leqslant n$ *with elements* $\tilde{\pi}_j$ *, or oscillate with a finite period* υ *starting from some finite power.*

Remark 3.6 Note that it is possible that only one of the primary transition matrices oscillates with a finite period.

Therefore, it is suitable to define the *Time Limiting Distribution* of a fuzzy matrix.

Definition 3.3 Let the powers of the fuzzy transition matrix $\tilde{P}$ converge in τ steps to a nonperiodic solution, then such solution is called **Aperiodic Type-2 Fuzzy Markov Chain** and it is divided into two matrices: $\underline{P}^{*} = \underline{P}^{\tau}$ and $\overline{P}^{*} = \overline{P}^{\tau}$ namely its **Limiting Type-2 Fuzzy Transition Matrix**.

The transition law of an IT2FM can be divided in two steps: (a) computations for the lower fuzzy transition matrix $\underline{P}$ and (b) computations for the upper fuzzy transition matrix $\overline{P}$, where there exist n_a embedded T1FS $\tilde{S}_i = \sum_{l=1}^{n_a} \tilde{S}_{e(i)}^{l}$.

Remark 3.7 It is important to emphasize that an IT2FM is completely ergodic if and only if both $\underline{P}$ and $\overline{P}$ are aperiodic and meet the condition given in Theorem 2.1.

4 Computing the T1 Fuzzy Limiting Distribution

Several algorithms and methods can be used to compute the limiting fuzzy distribution of a T1FM process (see [2, 3, 31, 32]). A first method consists in computing the max-min operations for a fuzzy matrix P, as follows:

$$P^{n} = P \circ P^{n-1} = \cdots = \underbrace{P \circ P \circ \cdots \circ P}_{n \text{ times}} \tag{21}$$

This method converges to $\lim_{n \to \tau} P^{n} = P^{*}$, turning P into an idempotent matrix. Sánchez [2, 31, 32] has created three useful algorithms in order to reduce the computing complexity for obtaining the limiting fuzzy distribution of P. His results are based on the definition of an *Eigen Fuzzy Set* similar to the classical concept of an *Eigenvector* and *Eigenvalue*. These two definitions are given below:

Definition 4.1 Let P be a fuzzy relation in a given matrix format. Then x is called an Eigen Fuzzy set of P, iff:

$$x \circ P = x \tag{22}$$

Definition 4.2 The fuzzy set $x \in \mathscr{F}(S)$ is contained in the fuzzy set $y \in \mathscr{F}(S)$ written $(x \subseteq y)$, iff $x_i \leqslant y_i$ for all $i \in S$.

Definition 4.3 Let $\mathscr{X}$ be the complete set of eigen fuzzy sets for P, namely:

$$\mathscr{X} \triangleq \{x \in \mathscr{F}(S) \mid x \circ P = x\} \tag{23}$$

The elements of (X) are invariants of P according to the $\circ-(\max-\min)$ composition. Then, if there exists any $\overset{\vee}{x} \in \mathscr{F}(S)$ such that $x \subseteq \overset{\vee}{x}$ for any $x \in (X)$, then $\overset{\vee}{x}$ is the Greatest Eigen Fuzzy Set of P.

Sánchez uses these concepts to find a maximum eigen fuzzy set, idempotent and stable; in other words:

$$\overset{\vee}{x} = \max_{i \in S} P_{ij}^{n} \tag{24}$$

Recall that if P is a *Strong Ergodic* fuzzy Markov chain, then its greatest eigen fuzzy set converges to an idempotent matrix P^{τ} with equal rows. This implies that all rows of the ergodic projection and the greatest eigen fuzzy set are equal.

Sánchez then designed three algorithms to compute the greatest eigen fuzzy set, namely *Method I, Method II and Method III*. In this chapter, Method II will be used as described below:

(i) Determine $\overset{1}{x}$ first with the elements corresponding to the greatest element in each column of P.

(ii) Compute $P^2 = P \circ P$ and determine the greatest elements in each column of P^2. They yield $\overset{2}{x}$ where $\max_{i \in S} P_{ij}^{k} = (\overset{k}{x} \circ P^{k-1})_j = \overset{k}{x}_j, j = \overline{1, n},$ for all $k \succ 0$. Here $k = 2$ and $j = \overline{1, m}$.

(iii) Compare $\overset{2}{x}$ with $\overset{1}{x}$: If they are different, compute $P^3 = P^2 \circ P$ to get $\overset{3}{x}$ where $\max_{i \in S} P_{ij}^{3} = (\overset{3}{x} \circ P^2)_j = \overset{3}{x}_j, j = \overline{1, m}$.

(iv) Compare $\overset{3}{x}$ with $\overset{2}{x}$: If they are different, compute $P^4 = P^3 \circ P$ to get $\overset{4}{x}$ where $\max_{i \in S} P_{ij}^{4} = (\overset{4}{x} \circ P^3)_j = \overset{4}{x}_j, j = \overline{1, m}$. And so on. Stop when $\vee$ is found such that $\overset{n+1}{x} = \overset{n}{x}$, that is $\overset{\vee}{x} = \overset{n}{x} \circ P$.

It is important to recall that if P is a *Strong Ergodic* fuzzy Markov chain, for any $\tau \leqslant n$, then the greatest eigen fuzzy set converges to an idempotent matrix P^{τ} with equal rows. This result implies that the rows of the strong ergodic matrix are equal to the greatest eigen fuzzy set of P. Now, for the *Weak Ergodic* case, the following proposition is given:

Proposition 4.1 *Let P be a weak ergodic fuzzy transition matrix. Since P^τ has no equal rows, then a proper estimation of its steady state is its eigen fuzzy set, $\overset{\vee}{x}$.*

5 Computing the IT2 Fuzzy Limiting Distribution

There are two ways for computing the steady state $\tilde{P}^\tau$ of $\tilde{P}$: either by computing the fuzzy powers of $\tilde{P}$ or by using an adaptation of the above algorithm to compute its steady state efficiently. For this purpose, the following Lemma is given.

Lemma 5.1 *Consider an IT2FM chain namely $\tilde{P}$, decomposable into two matrices: $\underline{P}$ and $\overline{P}$ where $\underline{P} \preccurlyeq \overline{P}$. If all powers of $\tilde{P}$ can be obtained by operations on $\underline{P}$ and $\overline{P}$ separately, as presented in Proposition 4.1, then each one of its stationary distributions converges to its greatest eigen fuzzy set $\overset{\vee}{x}$ only if each one of them is strong ergodic, that is $\overset{\vee}{x} \rightarrow P^\tau$ and the IT2FM has a steady state composed by the two stationary distributions $\underline{P}^\tau$ and $\overline{P}^\tau$.*

Therefore, the steady state of $\tilde{P}$ can be decomposable into two stages: $\underline{P}^\tau$ and $\overline{P}^\tau$, and each one has a greatest eigen fuzzy set which represents the point of major inertia of the matrix. Based on the above Lemma and on the Proposition 4.1, the following algorithm is proposed for finding the steady state of $\tilde{P}$:

1. Compute the eigen fuzzy set $\underline{\overset{\vee}{x}}$ for the lower fuzzy transition matrix $\underline{P}$, called $\underline{P}^\tau$ by using any of the referred methods in Sect. 4.
2. Compute the eigen fuzzy set $\overline{\overset{\vee}{x}}$ for the lower fuzzy transition matrix $\overline{P}$, called $\overline{P}^\tau$ by using any of the referred methods in Sect. 4.
3. Compute the type-reduced steady state of the process called P_r^τ by using any IT2FS type-reducer.
4. Compose the uncertain steady-state fuzzy distribution $\tilde{P}^\tau$ whose elements are they type-2 stationary possibilities called $\tilde{\pi}_j$ by using the three distributions mentioned above ($\underline{P}^\tau$, $\overline{P}^\tau$ and P_r^τ) as follows:

$$\tilde{P}^\tau = \langle \underline{P}^\tau, P_r^\tau, \overline{P}^\tau \rangle \tag{25}$$

   where $\underline{P}^\tau \preccurlyeq P_r^\tau \preccurlyeq \overline{P}^\tau$. $\tilde{P}^\tau$ has elements called $\tilde{\pi}_j$, $\underline{P}^\tau$ has elements called $\underline{\pi}_j$, $\overline{P}^\tau$ has elements called $\overline{\pi}_j$ and P_r^τ has elements called π_{rj}.

In practice, the analyst should make decisions using a real-value criteria, because it reduces the complexity of the problem, increasing its interpretability. The use of a type-reducer improves the inference process by reducing its computing cost. In this way, the following definitions about type-reduction are proposed.

6 Type-Reduction of an Interval Type-2 Fuzzy Markov Chain

Most decision-making applications use real-value criteria to reduce the complexity of the problem, increasing its interpretability. These measures can be computed using a type-reduction strategy that employs well-known algorithms.

The type-reduction of an IT2FM can be performed in two ways: Type-reduction of its stationary distribution and the type-reduction of its expected value (Centroid); the first one is defined as follows:

Theorem 6.1 (Type-reduced stationary distribution) *The IT2FM type-reduced distribution of the steady state of an ergodic IT2FM, namely P_r^{τ}, is the average between its lower and upper fuzzy distributions, element by element as follows:*

$$\pi_{rj} = \frac{\underline{\pi}_j + \overline{\pi}_j}{2} \quad \forall \ j \in S \tag{26}$$

Proof First, it is assumed that a stationary distribution for $\breve{P}$ exists, and by extension, the existence of an ergodic IT2FM is ensured. Now for an IT2FM, its secondary membership function is defined as one, that is $f_x(u)/u = 1/u$, so dividing the FOU of any jth state by M_j parts and using (10) leads to:

$$J_{\tilde{\pi}_j} = \left[\sum_{k=1}^{M_j} 1/u_{jk} \right] \bigg/ \tilde{\pi}_j, \quad j \in S \tag{27}$$

The centroid of $\sum_{k=1}^{M_j} 1/u_{jk} = \pi_{rj}$ yields[3]:

$$\pi_{rj} = \frac{\oint_{\underline{\pi}_j}^{\overline{\pi}_j} x \, dx}{\oint_{\underline{\pi}_j}^{\overline{\pi}_j} 1 \, dx}$$

Which finally is the centroid of any jth state of the IT2FS:

$$C_{\tilde{\pi}_j} = \left[\frac{\underline{\pi}_j + \overline{\pi}_j}{2} \right] / \tilde{\pi}_j, \ j \in S$$

where the type-reduced stationary distribution π_{rj} of each $\tilde{\pi}_j$ is $\frac{\underline{\pi}_j + \overline{\pi}_j}{2}$ and the proof of the theorem is complete.

The type-reduced expected value of an IT2FS can be obtained by using any type-reduction strategy. In this work, a centroid-based defuzzification method is implemented since it is a well-known measure of central tendency of an IT2FS. This leads us to the following statement

[3] Here, $\oint$ denotes crisp integration.

Proposition 6.2 *Given $\tilde{P}^{\tau}$ in (25) and its finite projections onto x_j, it is possible to compute its centroid, called $C(\tilde{P}^{\tau})$, through any centroid-based type-reduction method, which yields the following interval:*

$$C(\tilde{P}^{\tau}) = 1/\left[C_l(\tilde{P}^{\tau});\ C_u(\tilde{P}^{\tau}) \right] \tag{28}$$

where $C_l(\tilde{P}^{\tau})$ and $C_u(\tilde{P}^{\tau})$ are the lower and upper centroids of an IT2FS.

In the same way, an IT2FM is defined by a finite amount of states, the computation of the centroid of $\tilde{P}$ is bounded by m. The best known type-reduction algorithms were proposed by Mendel, Karnik, Melgarejo, and Liu [15, 20, 21, 23, 28, 37]. Unfortunately, both the IASCO and the EKM algorithms have nonclosed forms, so (29) and (30) are expressed as general forms of $C_l(\tilde{P}^{\tau})$ and $C_u(\tilde{P}^{\tau})$, where m is the cardinality of the markovian process.

$$C_l(\tilde{P}^{\tau}) = \frac{\sum_{j=1}^{L} x_j\, \overline{\pi}_j + \sum_{i=L+1}^{m} x_j\, \underline{\pi}_j}{\sum_{i=1}^{L} \overline{\pi}_j + \sum_{i=L+1}^{m} \underline{\pi}_j} \tag{29}$$

$$C_u(\tilde{P}^{\tau}) = \frac{\sum_{j=1}^{U} x_j\, \underline{\pi}_j + \sum_{i=U+1}^{m} x_j\, \overline{\pi}_j}{\sum_{i=1}^{U} \underline{\pi}_j + \sum_{i=U+1}^{m} \overline{\pi}_j} \tag{30}$$

As usual, a crisp measure is a desirable output of the model, so the most used crisp output of an IT2FS is the expected value of its centroid

$$C(\tilde{P}^{\tau}) = \frac{C_l(\tilde{P}^{\tau}) + C_u(\tilde{P}^{\tau})}{2} \tag{31}$$

The uncertain fuzzy expected value of $\tilde{P}$ is defined by (29–31) where L and U are computed using either the IASCO or the EKM algorithm.

Remark 6.3 The type-reduced centroid $C(\tilde{P})$ of $\tilde{P}$ can be computed by either the IASCO or the EKM algorithm. The main focus of both the IASCO and the EKM algorithms is to compute the values of L and U iteratively, which minimizes and maximizes $C(\tilde{P}^{\tau})$, finding $C_l(\tilde{P}^{\tau})$, and $C_u(\tilde{P}^{\tau})$ respectively. For all technical details about their initialization points, their recursive iterations and computations, see Melgarejo [21] and Mendel [15, 28].

7 Application Examples

Two introductory examples are presented. First, a five states matrix and afterwards an inventory control example.

7.1 A 5×5 IT2FM Example

Let $\tilde{P}$ have the following transition matrices $\underline{P}$ and $\overline{P}$:

$$\underline{P} = \begin{bmatrix} 0.721 & 0.569 & 0.438 & 0.025 & 0.241 \\ 0.342 & 0.020 & 0.452 & 0.824 & 0.915 \\ 0.529 & 0.060 & 0.289 & 0.774 & 0.057 \\ 0.746 & 0.013 & 0.385 & 0.015 & 0.008 \\ 0.746 & 0.490 & 0.459 & 0.356 & 0.521 \end{bmatrix} \quad \overline{P} = \begin{bmatrix} 0.856 & 0.773 & 0.529 & 0.160 & 0.734 \\ 0.563 & 0.061 & 0.626 & 0.924 & 0.974 \\ 0.734 & 0.080 & 0.404 & 0.871 & 0.348 \\ 0.880 & 0.154 & 0.449 & 0.108 & 0.067 \\ 0.870 & 0.690 & 0.763 & 0.732 & 0.582 \end{bmatrix}$$

By using Method II, exposed by Sánchez [31, 32], the following vectors $\underline{P}^{\tau}$ and $\overline{P}^{\tau}$ are obtained:

$$\underline{P}^{\tau} = [\, 0.721 \;\; 0.569 \;\; 0.459 \;\; 0.569 \;\; 0.569 \,]$$

$$\overline{P}^{\tau} = [\, 0.856 \;\; 0.773 \;\; 0.763 \;\; 0.773 \;\; 0.773 \,]$$

The type-reduced fuzzy stationary distribution P_r^{τ} is obtained from (26), achieving the following results:

$$P_r^{\tau} = [\, 0.789 \;\; 0.671 \;\; 0.611 \;\; 0.671 \;\; 0.671 \,]$$

The type-reduced expected values $C_l(\tilde{P}^{\tau}), C_u(\tilde{P}^{\tau})$ and $C(\tilde{P}^{\tau})$ are obtained from (29)–(31), achieving the following results:

$$C_l(\tilde{P}^{\tau}) = 2.759; \; C_u(\tilde{P}^{\tau}) = 3.093; \; C(\tilde{P}^{\tau}) = 2.926$$

7.2 An Inventory Control Example

This example is taken from Lieberman [13]. The case consists in a seller who can only hold three cameras in his store with the following policy of weekly ordering: If there are no cameras in stock at the end of the Saturday, the store must order of three cameras. However, if there is one or more cameras in stock, no order is placed. The stochastic process is defined as amount of cameras are sold in week t as a function of the demand of cameras and the stock of the store in week t, namely X_t. For our purposes, we assume that the demand of cameras is a random variable and there are multiple experts who try to define a transition fuzzy matrix that represents having X_t cameras in stock regarding the previous week X_{t-1}. For instance, if $X_t = 1$ and $X_{t-1} = 0$ means that the store ordered three cameras in week X_{t-1} and only two of those were sold in week X_t, and so on.

Now, the opinion of all experts is collected and embedded into an IT2FM where $\tilde{P}$ is composed by the matrices $\underline{P}$ and $\overline{P}$.

$$\underline{P} = \begin{bmatrix} 0.318 & 0.268 & 0.539 & 0.392 \\ 0.472 & 0.375 & 0 & 0 \\ 0.204 & 0.426 & 0.241 & 0 \\ 0.314 & 0.312 & 0.490 & 0.415 \end{bmatrix} \quad \overline{P} = \begin{bmatrix} 0.479 & 0.289 & 0.641 & 0.549 \\ 0.766 & 0.489 & 0 & 0 \\ 0.267 & 0.485 & 0.380 & 0 \\ 0.534 & 0.367 & 0.654 & 0.477 \end{bmatrix}$$

where the vectors $\underline{P}^\tau$ and $\overline{P}^\tau$ are:

$$\underline{P}^\tau = [\, 0.426 \;\; 0.426 \;\; 0.426 \;\; 0.415 \,]$$

$$\overline{P}^\tau = [\, 0.534 \;\; 0.489 \;\; 0.534 \;\; 0.534 \,]$$

Finally, the type-reduced stationary distribution of $\tilde{P}$ is:

$$P_r^\tau = [\, 0.480 \;\; 0.458 \;\; 0.480 \;\; 0.475 \,]$$

By using (29)–(31) its type-reduced expected values $C_l(\tilde{P}^\tau), C_u(\tilde{P}^\tau)$ and $C(\tilde{P}^\tau)$, are obtained as follows:

$$C_l(\tilde{P}^\tau) = 1.388; \;\; C_u(\tilde{P}^\tau) = 1.612; \;\; C(\tilde{P}^\tau) = 1.5$$

Now, the camera seller should have an average of 1.5 cameras in store per week, so the demand of the market is less than 1.5 cameras per week. In fact, the demand of cameras is less than $[1.388, 1.612]$ on average; this means that the possibility of having more than two cameras sold per week decreases as their demand decreases.

8 Concluding Remarks

A theoretical framework of IT2FS applied to Markov chains is provided assuming an effect of linguistic uncertainty contained in the opinion of multiple experts about the same conditional relational matrix $\tilde{P}$. The computation of its stationary behavior, its type-reduced centroid and its properties are presented through two examples.

This study presents some interesting definitions about uncertain fuzzy markov chains, treated as IT2FS where the type-reduced behavior is computed through well-known type-reduction methods such as the IASCO and the EKM algorithms.

There are different methods to handle uncertain probabilities (see Sect. 1), but the focus of this work is to define some necessary conditions of a pure fuzzy approach through the use of IT2FS.

The results of Avrachenkov and Sanchez can be extended to a type-2 fuzzy sets approach. While these authors use type-1 fuzzy Markov chains processes, this work extends its scope of their findings to uncertain FM by means of IT2 fuzzy transition matrices.

Finally, an uncertain fuzzy Markov chain can deal either with the opinion of different experts or inference methods to get an interval of solutions, and therefore yield a crisp stationary solution using a type-reduction algorithm. Linguistic uncertainty embedded into the FOU of an IT2FS is involved in the problem, and its crisp solution can be found by using any type-reduction strategy.

9 Future Work

The use of general type-2 fuzzy sets (GT2FS) and quasi type-2 fuzzy sets (QT2FS) emerges as a new subject in the field of Markov chains analysis. These approaches use the secondary membership function $f_x(u)/u$ of an IT2FS, inducing researchers to take new directions.

References

1. Araiza, R., Xiang, G., Kosheleva, O., Skulj, D.: Under interval and fuzzy uncertainty, symmetric markov chains are more difficult to predict. In: 2007 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), vol. 26. IEEE, Los Alamitos, pp. 526–531 (2007)
2. Avrachenkov, K.E., Sanchez, E.: Fuzzy markov chains: specifities and properties. In: IEEE (ed.) 8th IPMU'2000 Conference, Madrid, Spain, pp. 1851–1856 (2000)
3. Avrachenkov, K.E., Sanchez, E.: Fuzzy markov chains and decision-making. Fuzzy Optim. Decis. Making **1**(2), 143–159 (2002)
4. Buckley, J., Eslami, E.: Fuzzy markov chains: uncertain probabilities. Mathware Soft Comput. **9**(1), 33–41 (2002)
5. Campos, M.A., Dimuro, G.P., da Rocha Costa, A.C., Kreinovich, V.: Computing 2-step predictions for interval-valued finite stationary markov chains. Technical Report UTEP-CS-03-20 (2003)
6. Figueroa, J.C.: Interval type-2 fuzzy markov chains: an approach. In: 2010 Annual Meeting of the IEEE North American Fuzzy Information Processing Society (NAFIPS), IEEE, pp. 1–6 (2010)
7. Figueroa, J.C., Kalenatic, D., Lopéz, C.A.: A simulation study on fuzzy markov chains. Commun. Comput. Inf. Sci. **15**(1), 109–117 (2008)
8. Gavalec, M.: Reaching matrix period is NP-complete. Tatra Mt. Math. Publ. **12**(1), 81–88 (1997)
9. Gavalec, M.: Periods of special fuzzy matrices. Tatra Mt. Math. Publ. **16**(1), 47–60 (1999)
10. Gavalec, M.: Computing orbit period in max-min algebra. Discrete Appl. Math. **100**(1), 49–65 (2000)
11. Gordon, P.: Cadenas de Markov Finitas y sus Aplicaciones. Editorial Hispano Europea, Barcelona (1967)
12. Grimmet, G., Stirzaker, D.: Probability and Random Processes. Oxford University Press, UK (2001)
13. Hillier, F., Lieberman, G.: Introduction to Operations Research. Mc Graw Hill, New York (2001)
14. Karnik, N.N., Mendel, J.M.: Type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **7**(6), 643–658 (1999)
15. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Inf. Sci. **132**(1), 195–220 (2001)
16. Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets. Fuzzy Sets Syst. **122**, 327–348 (2001)

17. Kurano, M., Yasuda, M., Nakagami, J.: Interval methods for uncertain markov decision processes. In: International Workshop on Markov Processes and Controlled Markov Chains (1999)
18. Kurano, M., Yasuda, M., Nakagami, J., Yoshida, Y.: A fuzzy approach to markov decision processes with uncertain transition probabilities. Fuzzy Sets Syst. **157**(1), 2674–2682 (2006)
19. Liang, Q., Mendel, J.M.: Interval type-2 fuzzy logic systems: theory and design. IEEE Trans. Fuzzy Syst. **8**(5), 535–550 (2000)
20. Melgarejo, M.: Implementing interval type-2 fuzzy processors. IEEE Comput. Intell. Mag. **2**(1), 63–71 (2007)
21. Melgarejo, M., Bernal, H., Duran, K.: Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set. In: 2008 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), vol. 27. IEEE, pp. 1–6 (2008)
22. Melgarejo, M.A.: A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set. In: Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), IEEE, pp. 190–194 (2007)
23. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall, New Jersey (2001)
24. Mendel, J.: Fuzzy sets for words: a new beginning. In: The IEEE International Conference on Fuzzy Systems, pp. 37–42 (2003)
25. Mendel, J.M.: Type-2 Fuzzy Sets: Some Questions and Answers. IEEE coNNectionS. A publication of the IEEE Neural Networks. Society. **8**, 10–13 (2003)
26. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
27. Mendel, J.M., John, R.I., Liu, F.: Interval type-2 fuzzy logic systems made simple. IEEE Trans. Fuzzy Syst. **14**(6), 808–821 (2006)
28. Mendel, J.M., Liu, F.: Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set. IEEE Trans. Fuzzy Syst. **15**(2), 309–320 (2007)
29. Pang, C.T.: On the sequence of consecutive powers of a fuzzy matrix with max-archimedean t-norms. Fuzzy Sets Syst. **138**(3), 643–656 (2003)
30. Ross, S.M.: Stochastic Processes. Wiley, New York (1996)
31. Sanchez, E.: Resolution of eigen fuzzy sets equations. Fuzzy Sets Syst. **1**(1), 69–74 (1978)
32. Sanchez, E.: Eigen fuzzy sets and fuzzy relations. J. Math. Anal. Appl. **81**(1), 399–421 (1981)
33. Skulj, D.: Regular finite markov chains with interval probabilities. In: 5th International Symposium on Imprecise Probability: Theories and Applications, Prague, Czech Republic (2007)
34. Stewart, W.J.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Princeton (1994)
35. Symeonaki, M., Stamou, G.: Theory of Markov systems with fuzzy states. Fuzzy Sets Syst. **143**(1), 427–445 (2004)
36. Thomason, M.: Convergence of powers of a fuzzy matrix. J. Math. Anal. Appl. **57**(1), 476–480 (1977)
37. Wu, D., Mendel, J.M.: Enhanced Karnik-Mendel algorithms for interval type-2 fuzzy sets and systems. In: Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), vol. 26. IEEE, pp. 184–189 (2007)
38. Zeng, J., Liu, Z.Q.: Interval type-2 fuzzy hidden Markov models. In: IEEE 2004 International Conference on Fuzzy Systems, vol. 2. IEEE, pp. 1123–1128 (2004)
39. Zeng, J., Liu, Z.Q.: Type-2 fuzzy markov random fields to handwritten character recognition. In: Proceedings of pattern recognition, 2006. ICPR 2006. 18th international conference on ICPR 2006, vol 1. pp. 1162–1165 (2006)

zSlices Based General Type-2 Fuzzy Sets and Systems

Christian Wagner and Hani Hagras

Abstract This chapter provides a concise introduction to zSlices based general type-2 fuzzy sets and their associated set-theoretic operations. zSlices based general type-2 fuzzy sets allow the representation of and computation with general type-2 fuzzy sets by modeling each fuzzy set as a series of zSlices, i.e., modified interval type-2 fuzzy sets, thus greatly reducing computational as well as design and implementation complexity. The chapter proceeds to illustrate the role and application of zSlices based general type-2 fuzzy sets as part of general type-2 fuzzy systems and reviews their utility as part of both traditional, control style, as well as more recent applications such as fuzzy set based agreement modeling.

1 Introduction

In 1975, Lotfi Zadeh first introduced the concept of type-2 fuzzy logic in the context of linguistic variables [1]. While the concept was clear and the utility of type-2 fuzzy sets and systems (FSSs) seemed obvious, the complexity (specifically in computation terms) prevented the wider application of type-2 FSSs until the late 1990s. Since then, the vast majority of research and application have focused on a simplified version of type-2 fuzzy logic, generally referred to as interval type-2 fuzzy logic. While (general) type-2 fuzzy sets are an extension of type-1 fuzzy sets in the sense that rather than the membership function of the set associating each given point with a crisp degree of membership as in the type-1 case, in the (general) type-2 case, each

C. Wagner (✉)
School of Computer Science, University of Nottingham, Wollaton Road,
Nottingham NG8 1BB, UK
e-mail: christian.wagner@nottingham.ac.uk

H. Hagras
School of Computer Science, University of Essex, Wivenhoe Park,
Colchester CO4 3SQ, UK

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_5,

point is associated itself with a type-1 set as its membership. Interval type-2 fuzzy sets finally are a simplification of the general case of (general) type-2 fuzzy sets in the sense that each point is associated with an interval type-1 fuzzy set, effectively modeling the membership of each point through an interval rather than a distribution (type-1 set), as in the (general) type-2 case.

Since about the year 2000, the interest in type-2 fuzzy logic has been growing rapidly driven by a several landmark publications such as [2, 3]. While the majority of the work focused on interval type-2 fuzzy logic (e.g. [4–10],), an increasing level of interest in "taming the complexity" of general type-2 FSSs in order to leverage their potential as part of applications led to the development of new approaches for the implementation of general type-2 FSSs, most notably the development of novel representations of general type-2 fuzzy sets and associated set-theoretic operations [11–16]. The first complete alternative representation reducing the computational complexity of general type-2 FSSs was proposed by Coupland et al. in 2007. This was followed in 2008 by the introduction of the alpha-plane and zSlices representations which were independently developed by Liu et al. and Wagner et al., respectively. The full detail of a complete implementation of a zSlices based general type-2 fuzzy system (FS) was published for the first time in 2010 in [17]. The alpha-planes and zSlices representations allow the representation of general type-2 fuzzy sets as a series of modified interval type-2 fuzzy sets and thus not only greatly reduce computational complexity but also allow the re-use of most of the theoretical results developed for interval type-2 fuzzy FSSs.

This chapter focuses on an overview of the zSlices based representation for general type-2 fuzzy sets and its utility for general type-2 fuzzy systems. It provides insight into the theory behind and the applications of general type-2 FSSs and briefly introduces some of the recent areas of application, specifically the notion of agreement as modeled based on general type-2 fuzzy sets.

Parts of the material provided here have been adopted from previous publications by the authors.

2 General Type-2 Fuzzy Sets

General type-2 fuzzy sets [1] are an extension of type-1 fuzzy sets. While a type-1 fuzzy set F is characterized by a type-1 membership function (MF) $\mu_F(x)$, where $x \in X$ and $\mu_F(x) \in [0,1]$, a general type-2 set $\tilde{F}$ is characterized by a general type-2 MF $\mu_{\tilde{F}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,

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

in which $\mu_{\tilde{F}}(x,u) \in [0,1]$. $\tilde{F}$ can also be expressed as follows [2]:

$$\tilde{F} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{F}}(x, u)/(x, u) \quad J_x \subseteq [0, 1] \tag{2}$$

where $\int\int$ denotes union over all admissible x and u. An example of a general type-2 fuzzy set is depicted in Fig. 1a, b. J_x is called the primary membership of x in $\tilde{F}$. At each value of x say $x = x'$, the 2-D u and $\mu_{\tilde{F}}(x', u)$ is called a vertical slice of $\tilde{F}$ [3]. A secondary membership function is a vertical slice of $\tilde{F}$. It is $\mu_{\tilde{F}}(x = x', u)$, for $x' \in X$ and $\forall\ u \in J_{x'} \subseteq [0,1]$, [2], i.e.,

$$\mu_{\tilde{F}}(x = x', u) \equiv \mu_{\tilde{F}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u \quad J_{x'} \subseteq [0, 1] \tag{3}$$

in which $0 \leq f_{x'}(u) \leq 1$. Because $\forall x' \in X$, the prime notation on $\mu_{\tilde{F}}(x')$ is dropped and $\mu_{\tilde{F}}(x)$ is referred to as a secondary membership function [3]; it is a type-1 fuzzy set which is also referred to as a secondary set [3]. If $\forall x \in X$, the secondary membership function is an interval type-1 set where $f_x(u) = 1$, the type-2 set $\tilde{F}$ is referred to as an interval type-2 fuzzy set.

Besides the vertical slice representation mentioned above, a general type-2 fuzzy set can also be represented as a series of wavy slices where for discrete

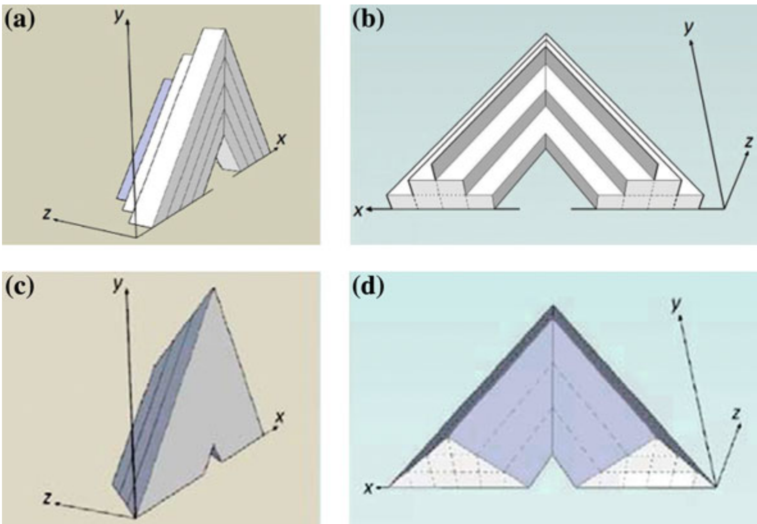


Fig. 1 (**a**) Side view of a general type-2 fuzzy set, indicating three zLevels on the third dimension. (**b**) Tilted rear/below view of the same set, indicating the position of the three zSlices (*dashed lines*). (**c**) Side view of the zSlices version of the set in (a), with $I = 3$. (**d**) Tilted rear/below view of the same set, showing the zSlices. *Note* to improve the accessibility of the complex 3D nature of general type-2 fuzzy set, we are referring to the three dimensions in the traditional mathematical notation of x, y, and z. These designations are equivalent to the respective traditional designations in the fuzzy logic field of x, u, and $\mu(x, u)(\text{or} f_x(u))$

universes of discourse X and U, Mendel and John [3] have shown that a type-2 fuzzy set $\tilde{F}$ can be represented as follows:

$$\tilde{F} = \sum_{j=1}^{n} \tilde{F}_e^j, \tag{4}$$

where $\tilde{F}_e^j$ is an embedded type-2 fuzzy set which can be written as follows:

$$\tilde{F}_e^j = \sum_{d=1}^{N} \left[f_{x_d}\left(u_d^j\right) / u_d^j \right] / x_d, \tag{5}$$

$$\text{where } u_d^j \in J_{x_d} \subseteq U = [0, 1].$$

$\tilde{F}_e^j$ has N elements, as it contains exactly one element from $J_{x_1}, J_{x_2}, \ldots, J_{x_N}$, namely u_1^j, u_2^j, …, u_N^j, each with its associated secondary grade namely, $f_{x_1}\left(u_1^j\right)$, $f_{x_2}\left(u_2^j\right)$, …, $f_{x_N}\left(u_N^j\right)$ [2]. $\tilde{F}_e^j$ is embedded in $\tilde{F}$ and there is a total of $n = \prod_{d=1}^{N} M_d$ embedded sets $\tilde{F}_e^j$ [3]. Where M_d is the discretization levels of u_d^j at each x_d [2, 3].

While the vertical and wavy-slice representations have proved highly useful for theoretical developments of general type-2 fuzzy sets, they have proved less useful for the practical implementation of general type-2 fuzzy sets. In the following section we will review the concept of zSlices introduced in [16] and [17] and show how zSlices are related to interval type-2 fuzzy sets before proceeding to the zSlices representation of general type-2 fuzzy sets and systems.

3 From Interval Type-2 Fuzzy Sets to zSlices

As noted above, an interval type-2 fuzzy set is a general type-2 fuzzy set where the secondary membership is 1 for all primary memberships. As discussed in [2], this simplification allows for a drastic reduction in the complexity of the computation with the respective (interval type-2) fuzzy sets. zSlices aim to capture this benefit of interval type-2 fuzzy sets for the subsequent modeling of general type-2 fuzzy sets.

Conceptually, a zSlice is formed by “slicing” a general type-2 fuzzy set in the third dimension (z) at level z_i. This slicing action will result in an interval set in the third dimension with height z_i. As such, a zSlice $\tilde{Z}_i$ is equivalent to an interval type-2 fuzzy set with the exception that its membership grade $\mu_{\tilde{Z}_i(x,u)}$ in the third dimension is not fixed to 1 but is equal to z_i where $0 \leq z_i \leq 1$. Thus, the zSlice $\tilde{Z}_i$ can be written as follows:

$$\tilde{Z}_i = \int_{x \in X} \int_{u_i \in J_{i_x}} z_i / (x, u_i) \tag{6}$$

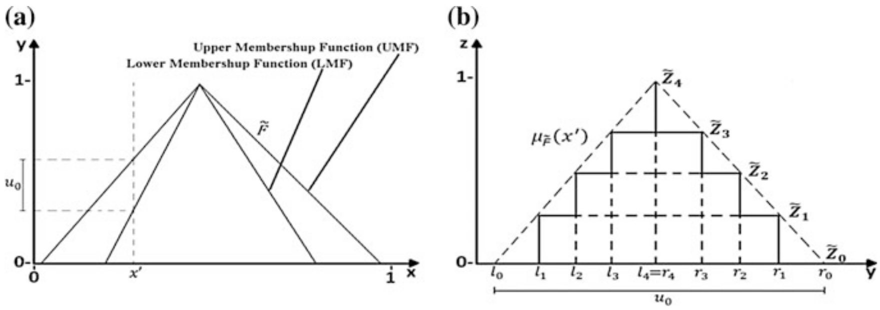


Fig. 2 (**a**) Front view of a general type-2 set $\tilde{F}$. (**b**) Third dimension at *x'* of a zSlices-based type-2 fuzzy set with $I = 4$

Where at each x value (as shown in Fig. 2a), zSlicing creates an interval set with height z_i and domain J_{i_x} which ranges from l_i to r_i as shown in Fig. 2b, $1 \leq i \leq I$, I is number of zSlices (excluding $\tilde{Z}_0$) and $z_i = i/I$.

Thus Eq. (6) can be written as follows:

$$\tilde{Z}_i = \int_{x \in X} \int_{u_i \in [l_i, r_i]} z_i/(x, u_i) \tag{7}$$

Additionally,

$$\tilde{Z}_0 = \int_{x \in X} \int_{u \in J_x} 0/(x, u) \tag{8}$$

Where $\tilde{Z}_0$ is considered as a special case with $z = 0$. In applications of zSlices as part of zSlices based general type-2 fuzzy sets (and systems), zSlice $\tilde{Z}_0$ can generally be disregarded with no effect (its secondary membership is 0) as shown in [17].

Finally, a zSlice can also be expressed as follows:

$$\tilde{Z}_i = \{((x, u_i), z_i) | \forall x \in X, \forall u_i \in [l_i, r_i]\} \tag{9}$$

Having defined the basic concept of zSlices, we proceed by reviewing the concept of zSlices based general type-2 fuzzy sets (zFSs) in the following section.

4 zSlices Based General Type-2 Fuzzy Sets

A general type-2 fuzzy set $\tilde{F}$ can be seen equivalent to the collection of an infinite number of zSlices:

$$\tilde{F} = \int_{0 \le i \le I} \tilde{Z}_i \quad I \to \infty \tag{10}$$

In a discrete universe of discourse Eq. (10) can be rewritten as follows:

$$\tilde{F} = \sum_{i=0}^{I} \tilde{Z}_i \tag{11}$$

We will be referring to the discrete version in Eq. (11) throughout the chapter. It should be noted that the summation signs in Eqs. (11) and (12) do not denote arithmetic addition but they denote the union set theoretic operation [2]. We have employed the max operation to represent the union, hence whenever a u value is attached to more than one z_i values, the maximum z_i is chosen and attached to the given u value. Hence, the MF $\mu_{\tilde{F}}(x')$ at x′ of the zFS $\tilde{F}$ shown in Fig. 2b can be expressed as:

$$\begin{aligned} \mu_{\tilde{F}}(x') &= \sum_{i=0}^{I} \sum_{u_i \in [l_i, r_i]} z_i / u_i \\ &= \sum_{u \in J_{x'}} \max(z_i)/u, J_{x'} \subseteq [0, 1] \end{aligned} \tag{12}$$

where $0 \le i \le I$. It is worth noting that at x′, $\mu_{\tilde{F}}(x')$ is a type-1 fuzzy set.

Figure 1 shows a three dimensional diagram for a general type-2 fuzzy set (shown in Figs. 1a, b) that is represented as a zFS (Fig. 1c, d) with $I = 3$.

5 Operations on zSlices Based General Type-2 Fuzzy Sets

zFSs provide a straightforward representation framework for general type-2 fuzzy sets. In order to employ the sets for logical inference, for example as part of zFSs, extensions of the set theoretical operations for union and intersection as well as centroid calculation and defuzzification operations have been developed. We briefly review the operations before addressing the mechanics of a complete zSlices based general type-2 fuzzy system in the following subsection.

5.1 Set-Theoretic Operations

The set-theoretic operations of union and intersection for zFSs were initially described in [16] and are implemented through the join and meet operations on the vertical slices of the respective sets.

Theorem 1 *The join operation between two zSlices-based general type-2 fuzzy sets reduces to the computation of the join operation (which employs the maximum t-conorm) between each corresponding zSlice in both sets and can be computed as follows:*

$$\begin{aligned}\tilde{A} \sqcup \tilde{B} \Leftrightarrow \mu_{\tilde{A}\sqcup\tilde{B}} &= \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x)\\ &= \sum_{i=0}^{I} \sum_{k\in[\max(\mathrm{l_{Ai}},\mathrm{l_{Bi}}),\max(\mathrm{r_{Ai}},\mathrm{r_{Bi}})]} z_i/k, \forall x \in X\end{aligned} \tag{13}$$

Theorem 2 *The meet operation between two zSlices-based general type-2 fuzzy sets reduces to the computation of the meet operation (which employs the minimum t-norm) between each corresponding zSlices in both sets and can be written as follows:*

$$\begin{aligned}\tilde{A} \sqcap \tilde{B} \Leftrightarrow \mu_{\tilde{A}\sqcap\tilde{B}} &= \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x)\\ &= \sum_{i=0}^{I} \sum_{k\in[\min(\mathrm{l_{Ai}},\mathrm{l_{Bi}}),\min(\mathrm{r_{Ai}},\mathrm{r_{Bi}})]} z_i/k, \forall x \in X\end{aligned} \tag{14}$$

The proofs for Theorems 1 and 2 can be found in [16]. It is worth noting that a zFS $\tilde{Z}$ where $I = 1$ is a general type-2 fuzzy set with a zSlice $\tilde{Z}_0$ at zLevel 0 which does not contribute to the fuzzy set (points with a secondary membership of 0 are not actually part of the set) and a zSlice $\tilde{Z}_1$ at zLevel 1. As such, a zFS with $I = 1$ is equivalent to a standard interval type-2 fuzzy set and consequently—standard interval type-2 operations are applicable.

5.2 Type Reduction

Type reduction has generally been the main stumbling block for the application of general type-2 FSSs. The standard type-reduction method for general type-2 sets is the centroid type reduction [2] which is based on computing the centroid of every wavy slice within the output set [3]. This is usually not possible in real-time control as it leads to exponentially growing computational requirements as the number of discretizations increases along the x and y axis [17]. zSlices based general type-2 fuzzy sets eliminate the need for a “brute force” calculation of the centroid by leveraging the zSlices based structure of the zFS and applying well-known techniques designed for interval type-2 fuzzy sets to compute the centroid. The actual slices based approach to computing the centroid was first published (in the context of alpha-planes) by Liu [13]. It was independently developed and published in the context of zSlices later in [17].

Theorem 3 *The centroid* $C_{\tilde{Z}}$ *for a zSlices-based general type-2 fuzzy set* $\tilde{Z}$ *is equivalent to the combination of the centroids of its zSlices* $\tilde{Z}_i$. *The centroid of each individual zSlice can be calculated in exactly the same fashion as the centroid for interval type-2 fuzzy sets while maintaining the zLevel of each individual zSlice. As such,* $C_{\tilde{Z}}$ *can be written as the combination of the centroids of its zSlices* $C_{\tilde{Z}_i}$ *each associated to their respective zLevel* z_i:

$$C_{\tilde{Z}} = \sum_{i=1}^{I} z_i / C_{\tilde{Z}_i} \qquad (15)$$

$C_{\tilde{Z}}$ represents the centroid of the zSlices based general type-2 fuzzy set formed by the zSlices $\tilde{Z}_i$. As $C_{\tilde{Z}_i}$ are bounded by two endpoints, we can write $C_{\tilde{Z}_i} = [c_{lz_i}, c_{rz_i}]$, where c_{lz_i} and c_{rz_i} are the left and right endpoints of the interval, respectively. These endpoints are calculated using standard interval type-2 algorithms (like the iterative KM procedure [2]) applied to every zSlice $\tilde{Z}_i$.

The proof of Theorem 3 as well as a comparison between it and the standard centroid calculation as well as numeric examples can be found in [17].

5.3 Defuzzification

The defuzzification of zFSs employs the centroid defuzzifier on the type-1 fuzzy sets that are generated using the type-reducer described in the previous subsection. As such, the standard centroid defuzzifier can be applied to the type-reduced set $C_{\tilde{Z}}$ by discretizing the type-reduced set and applying the normal centroid defuzzifier equation $\sum_{t=0}^{N-1} \mathrm{z}(g_t) * g_t / \sum_{t=0}^{N-1} \mathrm{z}(g_t)$, where $t = (yr_o - yl_o)/(N-1)$, N is the number of discretization points and $g_0 = c_{lz_0}$ and $g_{N-1} = c_{rz_0}$, $\mathrm{z}(g_t)$ is the maximum z_i level corresponding to any g_t according to Eq. (12). Note that all values associated with $z_0 = 0$ will vanish from the numerator and denominator of the equation, and hence all values associated with z_0 will not affect the total defuzzified output of the zFS. As such, as previously mentioned, the processing of $\tilde{Z}_0$ can be omitted both for individual zFSs as well as throughout complete zFLSs as they will not affect the FLS output. This is intuitive as at z_0, the certainty about the secondary membership is 0, thus it can be considered to not be part of the fuzzy set.

A new way particular to zFSs introduced in [17] is that their structure allows defuzzification by leveraging the fact that each zLevel is associated with a zSlice $\tilde{Z}_i$. Hence, the resulting defuzzified value of the zFS (at a given zLevel) will be the average of the type reduced—set as in interval type-2 FLSs. This will result in a discrete set where we will have the average of the type-reduced set for each zSlice, associated with the relevant zLevel z_i. Hence, by applying the centroid defuzzifier for this discrete set, we can write the defuzzified $D_{\tilde{Z}}$ of the zFS $\tilde{Z}$ as:

$$D_{\tilde{Z}} = \frac{\left(z_1 \frac{(c_{lz_1}+c_{rz_1})}{2} + z_2 \frac{(c_{lz_2}+c_{rz_2})}{2} + \cdots + z_I \frac{(c_{lz_I}+c_{rz_I})}{2}\right)}{(z_1 + z_2 + \cdots + z_I)} = \frac{\sum_{i=1}^{I}\left(z_i \frac{(c_{lz_i}+c_{rz_i})}{2}\right)}{\sum_{i=1}^{I} z_i} \quad (16)$$

Note that we have excluded the values associated the zSlice $\tilde{Z}_0$ as they will not have any impact as previously noted. From Eq. (16), it can be easily seen that the defuzzified value of the zFS is the weighted average of the outputs of the different zSlices.

The use of zFSs provides a series of advantages for the straightforward application of (zSlices based) general type-2 fuzzy sets and systems, ranging from the reduction in computational complexity to the possibility of re-using the existing interval type-2 fuzzy approaches and implementations. We briefly review zSlices based general type-2 fuzzy systems in the following section, a detailed introduction can be found in [17].

6 zSlices Based General Type-2 Fuzzy Systems

zFS based fuzzy systems are identical in structure to standard type-2 FLSs, with the obvious difference that throughout the FLS, zSlices based general type-2 fuzzy sets are employed. As part of the current chapter, we will mainly focus on the nature of zFS without proceeding in detail to their application as part of zFLSs which we only briefly review in this section. The main components of a zFLS are depicted in Fig. 3. For a detailed description of all parts of the FLS, see [17].

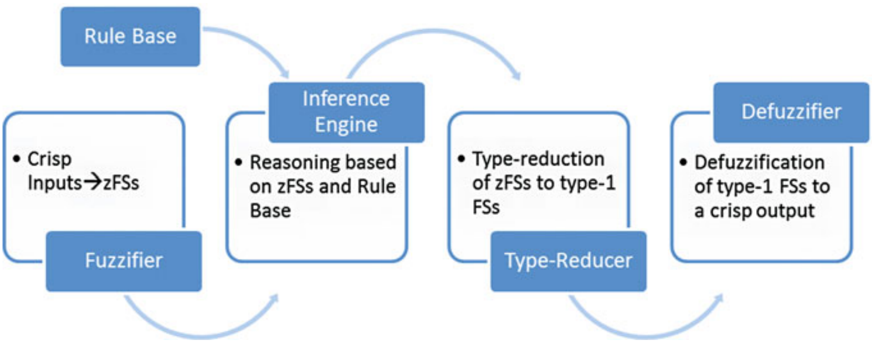


Fig. 3 The structure of a zSlices based general type-2 FLS

6.1 Fuzzifier

In zSlices based FLSs (zFLSs), the fuzzification step is identical to standard general type-2 FLSs (while employing zSlices based general type-2 fuzzy sets). As in most current applications of FLSs, singleton fuzzification provides the simplest form of fuzzification. Non-singleton fuzzification can be employed in cases where the specific uncertainty related to the input of the fuzzy system is to be modeled directly. It should be noted that a non-singleton fuzzifier will map a given input modeled as a (zSlices based) general type-2 fuzzy set to the given zSlices based antecedent fuzzy sets, thus, in the general case, a zSlices based general type-2 fuzzy set model of the input with an identical number of zSlices (at the same zLevels) as the rest of the zFLS should be employed.

6.2 Rule Base

The rule structure within zSlices based general type-2 FLSs is the standard Mamdani type FLS rule structure employed in standard type-1 and type-2 FLSs.

As such, a rule R_s from a zFLS can be written as:

$$\begin{aligned} R_S:\ &\text{IF}\, x_1 \text{ is } \tilde{F}_1 \text{ AND}\ldots\text{AND } x_{\mathrm{P}} \text{ is } \tilde{F}_{\mathrm{P}} \\ &\text{THEN } g_1 \text{ is } \tilde{G}_1, \ldots, g_{\mathrm{Q}} \text{ is } \tilde{G}_{\mathrm{Q}},\ s \in \{1, \ldots, S\} \end{aligned} \tag{17}$$

where P is the number of FLS inputs, Q the number of FLS outputs, and S is the number of rules in the rule base. The fuzzy sets employed in the rules are zFSs and all zFSs within the zFLS employ the same number of zSlices at the same zLevels. The latter enables the different zLevels within the zFLS to be computed in parallel, greatly improving performance when parallel computing resources are available.

6.3 Fuzzy Inference Engine

At a high level, the inference engine within a zFLS is similar in concept to that in all FLSs, i.e., from a given set of inputs, the firing strength of the antecedent membership functions (MFs) is determined for each rule. Note that, in this case, the firing strengths are zSlices induced type-1 fuzzy sets (see [17]). In order to proceed with the actual inference, the firing strengths (respectively their cylindrical extensions) are combined with the consequent sets applicable in the individual rules. The resulting outputs are combined through the union operation to produce the overall output (a zFS) for the given set of inputs. Full details of the inference process in zFLSs can be found in [17].

6.4 Type Reduction

Type reduction for zSlicesbased general type-2 sets also employs the nature of zSlices which can be seen as standard interval type-2 fuzzy sets with a specific zLevel $z_i \in [0, 1]$. Type reduction relies on the principle of a centroid calculation on the set in question as addressed in Theorem 3. In the practical application of zFLSs (as in interval type-2 FLSs) we do not need to find the fuzzy outputs of each rule, to combine the outputs of each fired rule and finally to compute the centroid of this combined set in order to find the type-reduced set. We can employ the centre-of-sets (COS) type reducer for each individual zLevel z_{i}:

$$y_{\cos} = \sum_{i=1}^{I} z_i / y_{\mathrm{cosi}}, \tag{18}$$

where $y_{\cos}$ refers to the overall type-reduced set (a type-1 set) of the zSlicesbased FLS. $y_{\cos_{\mathrm{i}}}$ is bounded by the interval $[y_{l_i}, y_{r_i}]$. $y_{\cos}$ is the combination of the type-reduced sets at each individual zLevel referred to as $y_{\cos_{\mathrm{i}}}$ which are each associated with their respective zLevel z_i. The type-reduced sets $y_{\cos_{\mathrm{i}}}$ can be found using standard interval type-reduction methods such as the KM iterative procedure [2] or alternatively the type-reduced sets $y_{\cos_{\mathrm{i}}}$ can be approximated using the Wu–Mendel uncertainty bounds method [18]. We should note that as in standard fuzzy logic theory, the summation sign (union operation) in Eq. (18) implies that for any point that is associated with more than one membership, we choose for this point the maximum of the associated membership values.

Thus, in zFLSs, we take the firing interval of each fired rule (which is calculated as in interval type-2 FLSs) and the associated centroid interval, and then,over all the fired rules we calculate $y_{\cos_{\mathrm{i}}}$ for that specific zLevel using the KM iterative procedure or using the Wu–Mendel uncertainty bounds method. Each $y_{\cos_{\mathrm{i}}}$ is associated to its respective zLevel z_i. Hence, it can be seen that zFLS is aggregating the outputs of several interval type-2 FLSs, each associated with a given zLevel (i.e., zSlices). This allows for a parallel implementation which results in a significantly faster computation which, in turn, makes it possible for us to use the zFLS for real-time real-world applications such as the robotic control example shown in [17].

Full details on the centroid calculation and the COS type reduction in zFLSs can be found in [17].

6.5 Defuzzification

The defuzzification step in a zFLS employs the centroid defuzzifier on the type-1 fuzzy set that was generated using the type-reducer described in the previous subsection. The details of the defuzzification and in particular the rapid defuzzification harnessing the zSlices induced structure of the output set have been introduced in the previous section and are available in [17].

7 Utility of zSlices Based General Type-2 Fuzzy Sets

After establishing the nature and properties of both zSlices based general type-2 fuzzy sets and systems, we proceed to review their applications and general utility. Fundamentally, (zSlices based) general type-2 fuzzy systems can be applied to the same problems as other FLSs (type-1, etc.), for example, in [17], we demonstrated their applicability to real world, real-time robotic control. However, a series of mostly recent applications benefit and in some cases depend on the more powerful and more complex modeling capabilities of zSlices based fuzzy sets. Examples here include applications where system interpretability (i.e., low number of rules) is important and recent advances such as fuzzy set based agreement modeling [19]. We briefly expand on both the application of zSlices based general type-2 fuzzy sets in more traditional, "control-style" applications in Sect. 7.1 and on more recent application areas "beyond control" in Sect. 7.2.

7.1 *zSlices Based General Type-2 Fuzzy Sets in Control-Style Applications*

In control and similar applications, the potential for general type-2 FLSSs lies with their greater potential to model uncertainty precisely—compared to type-1 and interval type-2 fuzzy systems. In order for general type-2 FLSSs to be able to exploit this potential, a significant effort is required to design the appropriate (zSlices based) general type-2 fuzzy sets for each application. This is not possible as part of all applications and in combination with the computational overhead of general type-2 fuzzy systems which,—even though much reduced—is still present (when comparing type-1 to type-2 fuzzy systems), a decision on whether or not to employ general type-2 fuzzy systems as part of traditional applications needs to be made very carefully. A number of guidelines and standard questions can be helpful to make this decision.

1. Does the system we are trying to model encompass varying levels of uncertainty and is the uncertainty significant?
2. If 1 is true, can we capture information on the uncertainty distribution in order to specify the secondary membership of general type-2 fuzzy sets (either from data, through learning, etc.)?
3. If 2 is true, employing general type-2 FLSSs may provide benefits. If not, interval type-2 FLSSs may be a more interesting option.

It is clear that the guidelines above are "fuzzy" and the design of an appropriate FLS is still dependent on a high familiarity with the problem and significant experience with FLSSs. Further, context and domain constraints, such as for example very low levels of available processing power are an intrinsic part of the design and decision process and crucially impact the viability of employing general type-2 fuzzy sets.

7.2 zSlices Based General Type-2 Fuzzy Sets Beyond Control

While the choice in terms of type of FLS is clearly not trivial for the standard set of FLS applications, there are applications which can directly benefit from higher order type-2 FLSs in ways which specifically are not possible by using for example type-1 FLSs. A known example of this is the interpretability of FLSs and the fact that interval type-2 fuzzy systems can provide similar performance to type-1 FLSs with a lower number of rules [20]. In cases where the number of rules is relevant, for example in intelligent inhabited environment (IIE) applications, this is a crucial advantage facilitating the interpretability of the system by the human user. zSlices based general type-2 fuzzy FLSs provide a unique set of features not available in type-1 or interval type-2 FLSs, namely the possibility to employ the secondary membership for the advanced modeling of concepts or values. A recent application of this potential is the concept of general type-2 based agreement modeling [19]. Agreement modeling harnesses the secondary membership of zSlices based general type-2 fuzzy sets to model agreement between multiple nodes, where the latter can be people, sensors, actuators, etc. The following section provides a brief introduction to the principles behind agreement-based modeling.

8 Agreement Modeling

We refer to the notion of "agreement" as agreement between sets. In other words, the agreement of two sets A and B is the set constituted by the overlap of both sets. In set terms, this overlap is referred to as the intersection of A and B, denoted as $A \cap B$.

Further, consider a specific concept (such as size, weight, beauty, strength, light levels, temperature, etc.). The agreement (i.e., the intersection) between multiple sets describes the "common ground" expressed by the sets. Practically speaking, if for example several people provide an interval of medium temperature on a temperature scale, the intersection (an interval) of the provided intervals describes the least common denominator of the provided interpretations (in the form of intervals) by the individuals, in other words: their agreement on the meaning of the concept of "medium" temperature.

While "agreement" could be considered merely an interpretation of the logical intersection operation, it is the modeling of levels of agreement (i.e., different degrees of agreement over a number of sets) which cannot be captured by standard intersections of sets and for which the notion of "Multi-Leveled Agreement" (MLA) has been established [19]. MLA models agreement of multiple sets in such a way that the resulting agreement set expresses the proportional level of agreement of its constituting sets, i.e., areas where multiple sets overlap are considered as more significant than areas where few sets overlap or even just one set exists.

For example, if three people define a fuzzy set for the linguistic label "comfortable indoor temperature", the three resulting sets will not be identical. The MLA of the three sets is itself a set which gives the most significance to the areas of the provided sets that are common to all three sets, less significance to areas which are common to two of the three sets, and finally low significance to the areas which belong only to one set.

This notion of MLA, while very intuitive, cannot be modeled using classical sets or indeed type-1 or interval type-2 fuzzy sets as was shown in [19]. However, the additional degree of freedom provided by the third dimension (secondary membership) of (zSlices based) general type-2 fuzzy sets allows the accurate modeling of multiple levels of agreement. As such, in MLA, the modeling of the uncertainty encompassed in the fuzzy set is identical to that of interval type-2 fuzzy sets in the sense that it is expressed in the FOU of each zSlice. It should be noted that the uncertainty encompassed in the FOU relates to the uncertainty about the primary membership, i.e., the sensor value, the actual variable like temperature, tallness, etc.

As noted, the secondary membership i.e., the third dimension is employed to model the level of agreement. A higher secondary membership as such reflects a higher degree of agreement. As has been shown, a zSlices based general type-2 fuzzy set is based on a series of zSlices. As part of MLA modeling, the total number of zLevels I is equal to the number of constituting (or input) interval type-2 fuzzy sets and the agreement is modeled as follows (full details are available in [19]):

- Areas which belong to only **one** interval type-2 fuzzy set are associated with a zLevel equal to $1/I$.
- Areas which belong to areas where at least **two** interval type-2 fuzzy sets intersect, i.e., "agree", are associated with the zLevel $2 * 1/\mathrm{I}$.
- Areas where **all** interval type-2 fuzzy sets intersect, i.e., "agree", are associated with the zLevel $I * 1/\mathrm{I} = I$.

It should be noted that the number of zLevels can be reduced as for all zSlices based general type-2 fuzzy sets by relying on interpolation. However, the MLA agreement model will deteriorate in accuracy as a result. Further, the actual application of the resulting MLA sets as part of FLSs is currently constrained to zLevel by zLevel processing and subsequent recombination of centroids and is a current topic of research.

9 Conclusions

For many years, general type-2 fuzzy sets and systems have been little more than objects of theory, concepts constructed to account for obvious limitations in type-1 fuzzy sets and systems. Many advances have contributed to the progress of general type-2 FLSSs from the conceptual stage to real-world application. The “in-between” stage (between type-1 and general type-2) of intensive research into interval type-2 FLSS has been crucial not only to develop vital contributions such as the KM algorithm(s) but also to develop a deeper understanding of the nature of (type-2) fuzzy sets and their modeling of uncertainty. The most recent advances in representation of general type-2 fuzzy sets [11, 15, 17] have leveraged the existing knowledge and today provide a realistically applicable platform for general type-2 FLSSs. As we have shown in this book and in more detail in [16, 17] with the first complete description of a slices based representation of zSlices based FLSSs, zSlices based sets, and the zSlices based FLSs that employ them are based on a direct expansion of interval type-2 fuzzy logic theory.

This shows a great strength of zFLSs which enables a series of advantages:

- Complex operations on general type-2 sets can be reduced to common interval type-2 operations, significantly reducing the design and implementation complexity, and thus facilitating the use of general type-2 FLSs.
- The property of zFLSs that allows the computation of each zLevel independently allows for a high degree of parallel computation. In fact, all zSlices levels can be computed simultaneously on separate processors followed only by the very simple defuzzification stage which is done centrally and the output of which is fed to the system. This offers great potential with minimal implementation effort and should allow the use of general type-2 FLSs in a far wider set of applications.
- In zFLSs, current interval type-2 theory can be re-used and only very small modifications are necessary to use current interval type-2 implementations to compute zFLSs.
- When computing the centroid of a zSlices based general type-2 set as done during the type-reduction stage, the resulting type-reduced type-1 set still (as for standard general type-2 FLSs) gives an indicative model of the amount of uncertainty contained within the current iteration of the zFLS.
- The use of zFLSs allows achieving real-time performance for general type-2 FLSs as a result of significantly simplifying the computational complexity associated with the deployment of general type-2 FLSs.

In this chapter, we have detailed zSlices based general type-2 fuzzy sets and provided a brief review of zSlices based general type-2 fuzzy systems. Specifically we have focused on fuzzy sets themselves and the most common operations required to employ them. Finally, we have given a brief outlook on applications of general type-2 fuzzy logic sets and systems in the existing and particularly in new areas such as zSlices based agreement modeling.

The area of general type-2 fuzzy logic is still in its infancy and it is without doubt that the coming years will see significant developments both in theory and application. It is an exciting time to be part of the fuzzy logic community and we are excited to be part of it.

References

1. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning-I. Inf. Sci. **8**(3), 199–249 (1975)
2. Mendel, J.: Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice hall, Upper Saddle River (2001)
3. Mendel, J., John, R.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**, 117–127 (2002)
4. Figueroa, J., Posada, J., Soriano, J., Rojas, S., Melgarejo, M.: A Type-2 Fuzzy controller for tracking mobile objects in the context of robotic soccer games. Proceedings of the International Conference of Fuzzy Systems, Reno, USA, May 2005, pp. 359–364
5. Greenfield, S., John, R., Coupland, S.: A novel sampling method for type-2 defuzzification. In: Proceedings of the UKCI, pp. 120–127, 05 Sept (2005)
6. Lynch., Hagras, H., Callaghan, V.: Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines. In: Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 1446–1453. Vancouver, Canada, July (2006)
7. Melgarejo, M.C., Reyes, A.P., and Garcia, A.: Computational model and architectural proposal for a hardware type-2 fuzzy system. In: Proceedings of the IEEE FUZZ Conference, Budapest, Hungary, July (2004)
8. Melin, P., Castillo, O.: A new method for adaptive control of nonlinear plants using type-2 fuzzy logic and neural networks. Int. J. Gen. Syst. **33**, 289–304 (2004)
9. Wagner, C., Hagras, H.: A genetic algorithm based architecture for evolving type-2 fuzzy controllers for real world autonomous mobile robots. In: Proceedings of the IEEE International Conference of Fuzzy Systems, pp. 193–198. London, UK, July 2007
10. Wu, D., Tan, W.: A type-2 fuzzy logic controller for the liquid-level process. In: Proceedings of IEEE International Conference of Fuzzy Systems, Budapest, pp. 953–958 July 2004
11. Coupland, S., John, R.: Geometric type-1 and type-2 fuzzy logic systems. In: IEEE Transactions on Fuzzy Systems, vol. 15, pp. 3–15, Feb (2007)
12. Greenfield, S., John, R.: Optimised generalised type-2 join and meet operations. In: Proceedings of the IEEE International Conference of Fuzzy Systems, pp. 141–146. London, UK, July (2007)
13. Liu, F.: An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. Inf. Sci. **178**(9), 2224–2236 (2008)
14. Lucas, L., Centeno, T., Delgado, M.: General type-2 fuzzy inference systems: analysis, design and computational aspects. In: Proceedings of the IEEE International Conference of Fuzzy Systems, pp. 1107–1112. London, UK, July (2007)
15. Mendel, J., Liu, F.: On new quasi-type-2 fuzzy logic systems. In: Proceedings of the IEEE International Conference of Fuzzy Systems, pp. 354–360. Hong Kong, China, June (2008)
16. Wagner, C., Hagras, H.: zSlices—towards bridging the gap between interval and general type-2 fuzzy logic. In: Proceedings of the IEEE International Conference of Fuzzy Systems, pp. 489–497. Hong Kong, China, June (2008)
17. Wagner, C., Hagras, H.: Toward general type-2 fuzzy logic systems based on zslices. IEEE Trans. Fuzzy Syst. **18**(4), 637–660 (2010)
18. Wu, H., Mendel, J.: Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **10**, 622–639 (2002)
19. Wagner, C., Hagras, H.: Employing zSlices based general type-2 fuzzy sets to model multi level agreement. In: Proceedings of the 2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems, pp. 50–57, Paris, France, July (2011)
20. Hagras, H.: A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Trans. Fuzzy Syst. **12**(4), 524–539 (2004)

Geometric Type-2 Fuzzy Sets

Simon Coupland and Robert John

Abstract This chapter gives a review and technical overview of the geometric representation of a type-2 fuzzy set and explores logical operators used to manipulate this representation. Geometric fuzzy logic provides a distinct way of understanding a fuzzy system, where fuzzy sets and fuzzy logic operators are seen purely as geometric objects which are manipulated only using knowledge of geometry. This approach is simple and intuitive, ideal for those who are not well versed in discrete mathematics. For researchers working with fuzzy systems regularly, this approach can raise some interesting questions about how fuzzy sets and systems are constructed.

1 Geometry and Fuzzy Logic

Fuzzy logic, based around the fuzzy set, is an extension of classical set and Boolean logic. It may seem odd to attempt to bring together this methodology rooted in discrete mathematics with the distinct paradigm of geometry. However, fuzzy logic is already reliant of aspects of geometry for modelling membership functions of fuzzy sets. The vast majority of fuzzy sets used take the form of continuous (Gaussian) of piecewise linear functions (triangle, trapezoid and shoulder). The notion of a geometric fuzzy set [6] came out of modelling a membership function as a piecewise linear function. A geometric type-1 fuzzy set is a piecewise linear approximation of a continuous membership function, which is

S. Coupland (✉)
De Montfort University, The Gateway, Leicester LE1 9BH, UK
e-mail: simonc@dmu.ac.uk

R. John
University of Nottingham, Wollaton Road, Nottingham NG8 1BB, UK
e-mail: robert.john@nottingham.ac.uk

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_6,

of course totally accurate if that function happens to be piecewise linear in nature anyway, i.e. triangular. This notion of approximating a continuous function with discrete geometric objects is continued in geometric type-2 fuzzy sets described in Sect. 2. A more formal definition of a type-1 geometric fuzzy set is given below.

Definition 1 A geometric type-1 fuzzy set is a series of ordered vertices that are connected by line segments to form a function over a continuous domain. This function is linear in all but a finite set of points. A geometric fuzzy set A over the domain X consists of pairs of vertices (x, y) where the $x \in X$ and the y component of all the vertices are in the interval $[0, 1]$ i.e.,

$$\mu_A : X \rightarrow [0, 1] \tag{1}$$

The membership grade μ_A for any particular value of x is given by

$$\mu_A(x) = \begin{cases} 0; & x \leq x_1 \text{ or } x_n \leq x \\ y_i; & x = x_i \\ y_i + \frac{x - x_i}{x_{i+1} - x_i}(y_{i+1} - y_i); & x_i < x < x_{i+1} \end{cases} \tag{2}$$

where x_0 and x_n are, respectively, the x-component of the first and last vertices of A. For convenience, a geometric type-1 fuzzy set can also be denoted by a set of vertices, i.e.

$$A = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) | x_i \in X, y_i \in [0, 1], x_i < x_{i+1}, \forall i\} \tag{3}$$

where x_i is the x component or domain value of the ith vertex and y_i is the y component or range value of the ith vertex.

Figure 1 depicts a Gaussian membership function and one possible geometric representation of this set.

Geometric type-1 fuzzy sets provide a practically useful representation of a fuzzy set; however, the most interesting and important aspect of geometric fuzzy logic is the geometric inference process. Logical operations, namely AND, OR and IMPLIES, may be defined using geometry operators. To define geometric logic

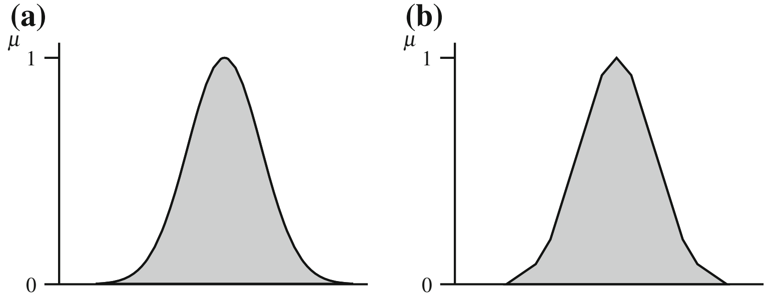


Fig. 1 (**a**) A Gaussian fuzzy set and (**b**) a geometric Gaussian fuzzy set

operators, it is necessary to identify all points where the line segments which make up a geometric membership function intersect with another geometric membership function. This can be done efficiently with the Bentley–Ottmann plane sweep algorithm [2]. Simple algorithms given in [6] describe how AND, OR and IMPLIES can be performed simply once all intersection points have been identified. Figure 2 depicts the OR operation on two geometric type-1 one fuzzy sets. The geometric type-1 fuzzy set A OR B = $\{v, i_1, i_2, z\}$ and A AND B = $\{a, i_1, x, y, i_2, d\}$.

To summarise, geometric type-1 fuzzy sets are made up of a discrete set of connected line segments. In order to perform logical operations, all points where the line segments from two sets intersect must be identified and processed. In the next section, we take this notion forward from 2-dimensional type-1 fuzzy sets to 3-dimensional type-2 fuzzy sets.

2 Geometric Type-2 Fuzzy Sets

We saw in the last section that geometric type-1 fuzzy sets are a discrete set of connected line segments. Each line segment fits the equation:

$$Ax + By + C = 0 \tag{4}$$

and is constrained by a start and end point on this line. This is the simplest geometric primitive which could be used to model a type-1 membership function. Type-2 fuzzy sets exist in 3-dimensions and as such the natural extension from the geometric type-1 model is to add a third dimension to the geometry. In 3D, we need a geometric primitive which fits the equation of a plane:

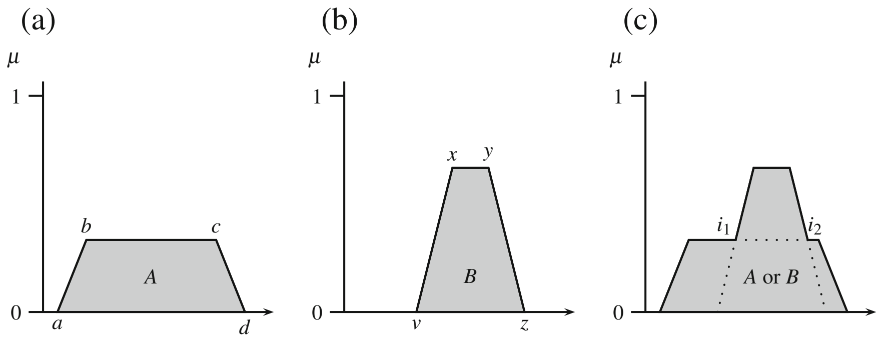


Fig. 2 (**a**) The geometric fuzzy set A. (**b**) The geometric fuzzy set B. (**c**) The geometric fuzzy set A OR B

$$Ax + By + Cz + D = 0 \tag{5}$$

Although any 3-dimensional polygon can fit the equation of plane, there is only one primitive which by definition must fit the equation of a plane and that is a 3D triangle. A 3D triangle is a constrained area on a plane which is analogous to how a line segment is a constrained length on a linear function. Geometric type-2 fuzzy sets are defined as discrete set of connected 3D triangles which approximate the membership function of a type-2 fuzzy set, a formal definition is given below.

Definition 2 A geometric type-2 fuzzy set is defined as a collection of n triangles in 3D space where the edges of these triangles connect to form a 3D polyhedron, i.e.

$$\widetilde{A} = \bigcup_{i=1..n} t^i \text{ where } t^i = \begin{bmatrix} x_1^i & y_1^i & z_1^i \\ x_2^i & y_2^i & z_2^i \\ x_3^i & y_3^i & z_3^i \end{bmatrix} \tag{6}$$

where x_1^i, x_2^i and $x_3^i \in X$ and y_1^i, y_2^i, y_3^i, z_1^i, z_2^i and $z_3^i \in [0, 1]$. In this geometric model, values on the y axis represent primary membership grades and values on the z axis represent secondary membership grades.

An example geometric type-2 fuzzy set $\widetilde{Moderate}$ is depicted in Fig. 3. The membership function of $\widetilde{Moderate}$ is a polyhedron, in this case made up of eight triangles. These eight triangles approximate the membership function of $\widetilde{Moderate}$ over a continuous domain X. The polyhedron provides an approximation of the actual membership function of $\widetilde{Moderate}$ as a surface modelled by triangles. The set $\widetilde{Moderate}$ give a good illustrative example of a geometric type-2 fuzzy membership function; however, it is possible to model more complex membership functions.

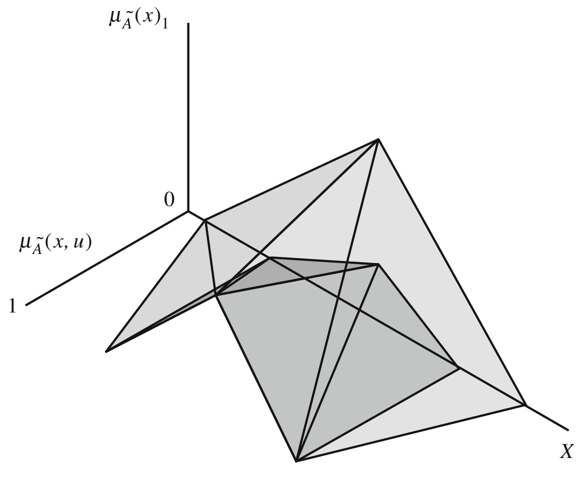


Fig. 3 The geometric type-2 fuzzy set $\widetilde{Moderate}$

In the previous section, we saw how line segments can approximate a Gaussian type-1 membership function. The same concept is now demonstrated for a Gaussian type-2 membership function where triangles are used to approximate the membership function. The equation of $\widetilde{A}$, a Gaussian-like set with an uncertain standard deviation, is given in Eq. (7) and depicted in Fig. 4.

$$\mu_{\tilde{A}}(x,u) = \exp\left(-\frac{u - c(x)^2}{2\sigma(x)^2}\right) \tag{7}$$

where $c(x)$ is given by Eq. (8) and $\sigma(x)$ is given by Eq. (9).

$$c(x) = \exp\left(-\frac{x - c^2}{(\sigma_1 + \sigma_2)^2}\right) \tag{8}$$

$$\sigma(x) = \exp\left(-\frac{\left(\frac{x-c^2}{2\sigma_1^2}\right) - \left(\frac{x-c^2}{2\sigma_2^2}\right)}{5}\right) \tag{9}$$

where $x \in X$, $u \in [0,1]$, σ_1 and σ_2 give the range of values for standard deviation. Figure 4 depicts 11,600 points of this continuous function. Any values of $\mu_{\tilde{A}}(x,u) < 0.001$ are not included, this prevents $\widetilde{A}$ from having an infinite FOU, hence it is Gaussian like and not Gaussian. This continuous function is contrasted to two other type-2 fuzzy set models, a discrete model and geometric model.

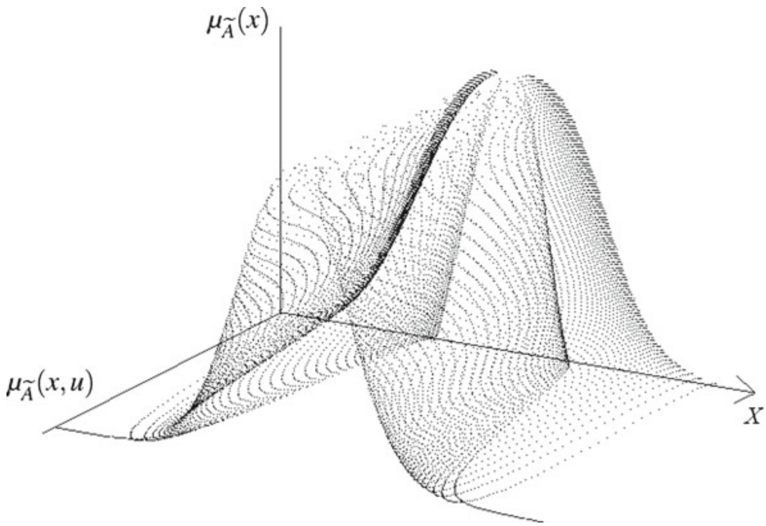


Fig. 4 A type-2 Gaussian-like fuzzy set with an uncertain standard deviation

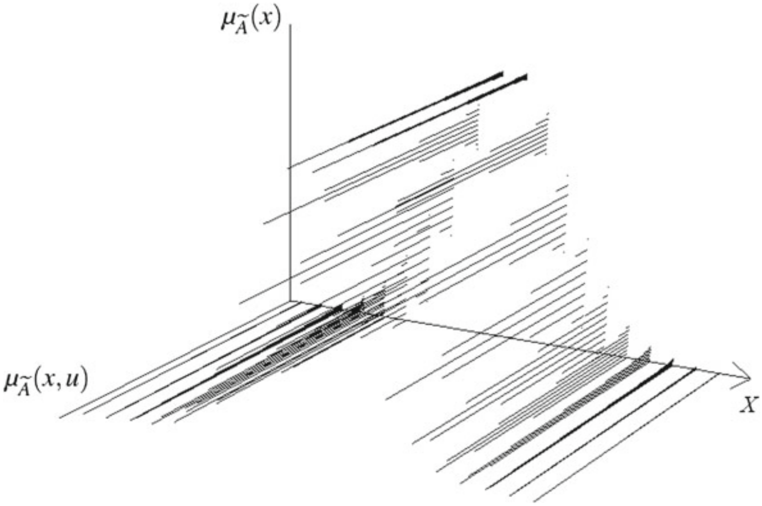


Fig. 5 A discrete type-2 Gaussian-like fuzzy set with an uncertain standard deviation

The discrete model (see Fig. 5) has 20 discrete points in the primary domain and the secondary domain ([0,1]) has been discretised into 11 points. The geometric model used in this comparison was constructed from this discrete model using the method presented in [7]. The geometric model consists of 365 triangles and is depicted in Fig. 6. In each of these figures, the domain X of $\widetilde{A}$ runs along the x-axis, the

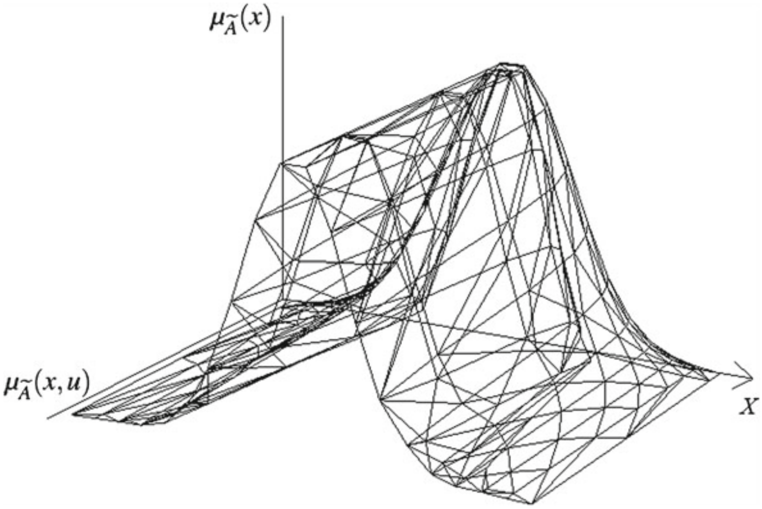


Fig. 6 A geometric type-2 Gaussian-like fuzzy set with an uncertain standard deviation

co-domain $\mu_{\tilde{A}}(x)$ runs vertically along the y-axis and the secondary membership grades $\mu_{\tilde{A}}(x, u)$ are depicted as if coming out of the page, on the z-axis.

This example demonstrates the complexity inherent in type-2 fuzzy sets and the modelling of these sets. However, conceptual and mathematically geometric type-2 fuzzy sets are simply geometric type-1 fuzzy sets with a third dimension added. Type-2 fuzzy membership functions exist in 3D and geometric type-2 fuzzy sets simply approximate this 3D surface with triangles. The next section defines algorithms for manipulating such models to implement logical operators.

3 Geometric Type-2 Fuzzy Logic Operators

We saw in Sect. 1 that in order to manipulate a geometric type-1 fuzzy set, it was necessary to identify all points where the line segments which made up two fuzzy sets intersected. For type-2 fuzzy sets, we need to take this 2D idea into 3D, that is we need to identify all line segments where two triangles intersect and the construct new triangles at these intersection lines. Guigue and Devilliers [9] provide an extension of Möller's triangle–triangle intersection test [13] which provides this functionality for a pair of triangles. Figure 7 depicts the intersection of two such triangles, where the line segment at the intersection of the two triangles is the line $\{K, J\}$. Once this line of intersection has been identified, a new set of triangles is needed to model this intersection. With type-1 inference, we identified the minimum and maximum lines to produce the AND and OR of the sets. A similar operation is required to produce a set of triangles which model the minimum and maximum of the surface modelled by these triangles. In [8], the authors defined the

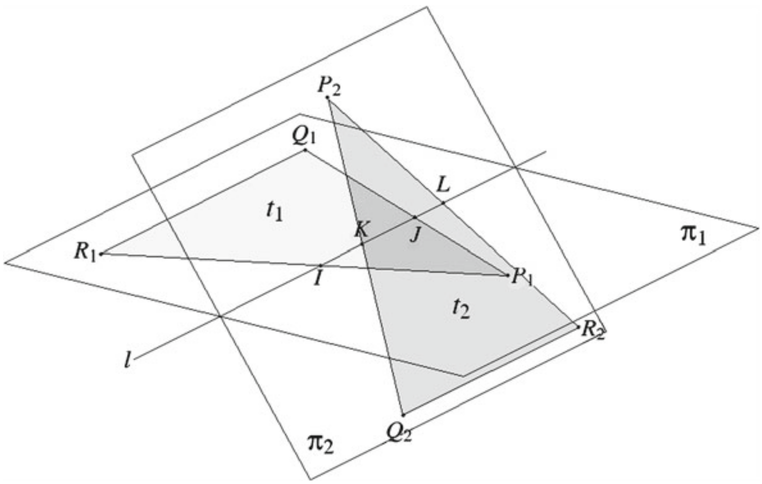


Fig. 7 Intersecting *triangles* and the planes in which they lie. Adapted from [9]

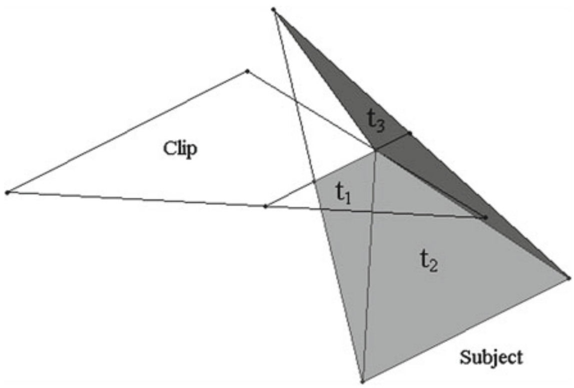


Fig. 8 The minimum surface calculated from two *triangles*

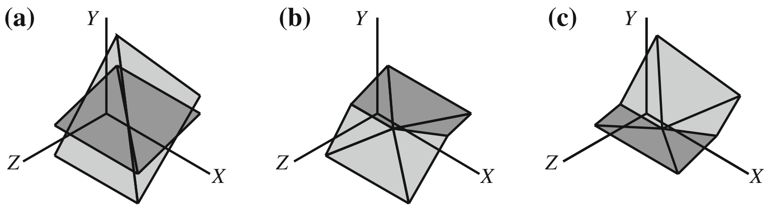


Fig. 9 (**a**) Two surfaces. (**b**) The minimum of those two surfaces. (**c**) The maximum of those two surfaces

surface clipping algorithm which performs this operation across a surface made up of a set of triangles. This algorithm works by performing an ordered search of two surfaces made up of triangles, identifying all pairs of intersecting triangles and the line on which they intersect. It is then trivial to obtain the minimum or maximum surface as a new collection of triangles.

Figure 8 shows an example of the intersection of two triangles and how the minimum surface modelled by this pair of triangles can be created. Triangles t_1 and t_2 where originally contained within the subject triangle. These two triangles make up the minimum surface, they are area contained with in the clipping triangle which lie below the intersection. The other triangle t_3 lies outside of the intersection of the two triangles and so is not clipped. The remainder of the subject triangle (within and above the intersection of the two triangles) is clipped. Figure 9 shows how this operation may be used to find either the minimum or maximum of two surfaces made up of a discrete set of triangles using the surface clipping algorithm.

So, the membership function of a type-2 fuzzy set can be modelled by a collection of 3D triangles. These triangles form a surface and using the surface clipping algorithm, it is possible to calculate a new set of triangles which form the minimum and maximum of two such surfaces. These are all the tools we need to

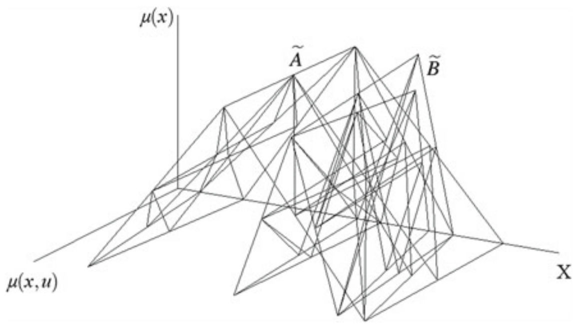


Fig. 10 The geometric type-2 fuzzy sets $\widetilde{A}$ and $\widetilde{B}$

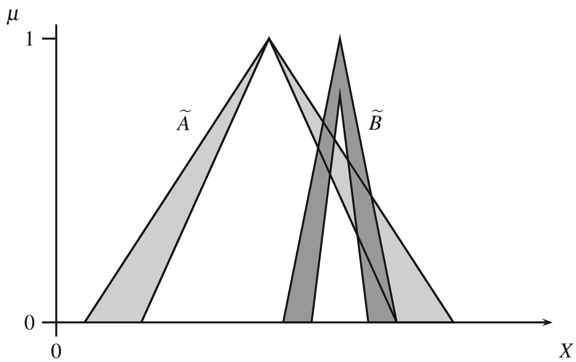


Fig. 11 The FOU of the geometric type-2 fuzzy sets $\widetilde{A}$ and $\widetilde{B}$

implement the AND, OR and IMPLIES operators for type-2 fuzzy sets. The first step is to separate the triangles which make up the membership function of a type-2 fuzzy set $\widetilde{A}$ into two new sets of triangles which form the upper and lower surface of the membership function $\overline{\widetilde{A}}$ and $\underline{\widetilde{A}}$. It is easy to identify which set an individual triangle belongs to. Simply take the normal of the triangle; if the y component of the normal is positive, then it belongs to the upper surface, if negative then if belongs to the lower surface. We can now define the AND and OR operations for geometric type-2 fuzzy sets. The AND and OR will now be defined using as examples the geometric type-2 fuzzy sets $\widetilde{A}$ and $\widetilde{B}$ depicted in full in Fig. 10 and the associated FOUs in Fig. 11.

3.1 The Geometric AND Operator

The AND of two type-2 fuzzy sets is defined as the result of taking the meet [5, 11] of the secondary membership functions of the two sets at each point in the domain of the sets. The surface clipping operation is used to give the meet, not at every

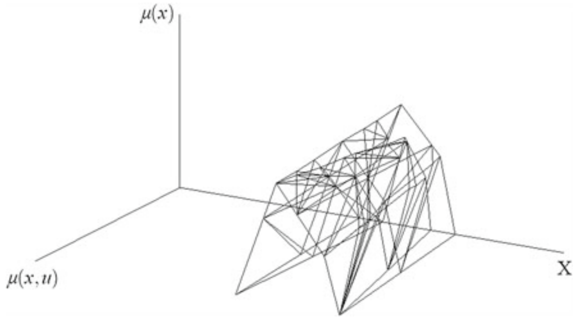


Fig. 12 The geometric type-2 fuzzy set $\widetilde{C} = \widetilde{A} \cap \widetilde{B}$

discrete point, but a every point along the continuous domain of the two geometric type-2 fuzzy sets.

Definition 3 Let $\widetilde{A}$ and $\widetilde{B}$ be two geometric type-2 fuzzy sets, each with membership functions defined by lower and upper surface over the continuous domain X. Let the logical AND of $\widetilde{A}$ and $\widetilde{B}$ be a third geometric type-2 fuzzy set $\widetilde{C}$.

- The lower surface of $\widetilde{C} =$ the minimum, as given by the surface clipping algorithm, of the lower surfaces of $\widetilde{A}$ and $\widetilde{B}$.
- The upper surface of $\widetilde{C} =$ the minimum, as given by the surface clipping algorithm, of the upper surfaces of $\widetilde{A}$ and $\widetilde{B}$.

This performs the meet across the entire domain of $\widetilde{A}$ and $\widetilde{B}$ giving the logical AND.

The logical AND of the example geometric type-2 fuzzy sets $\widetilde{A}$ and $\widetilde{B}$ is depicted in full in Fig. 12 and just the FOU in Fig. 13.

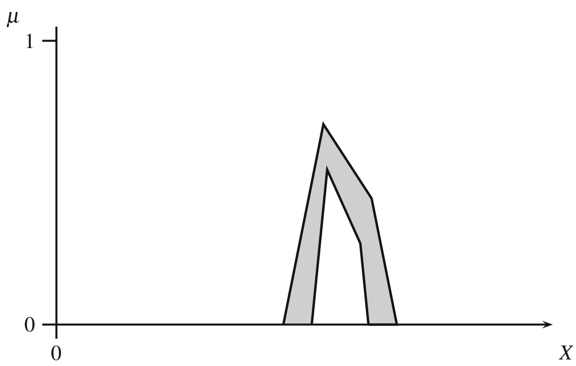


Fig. 13 The FOU of the geometric type-2 fuzzy set $\widetilde{C} = \widetilde{A} \cap \widetilde{B}$

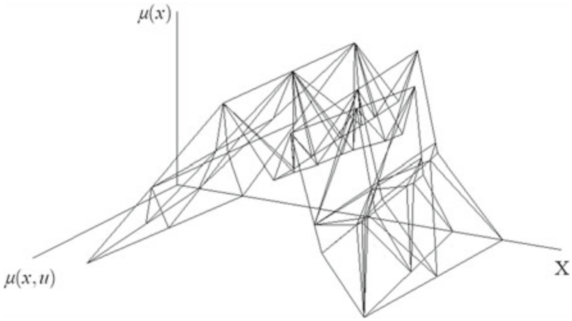


Fig. 14 The geometric type-2 fuzzy set $\widetilde{C} = \widetilde{A} \cup \widetilde{B}$

3.2 The Geometric OR Operator

The OR of two type-2 fuzzy sets is defined as the result of taking the join [5, 11] of the secondary membership functions of the two sets at each point in the domain of the sets. Again, the surface clipping operation is used to give the join, not at every discrete point, but every point along the continuous domain of the two geometric type-2 fuzzy sets.

Definition 4 Let $\widetilde{A}$ and $\widetilde{B}$ be two geometric type-2 fuzzy sets, each with membership functions defined by lower and upper surface over the continuous domain X. Let the logical OR of $\widetilde{A}$ and $\widetilde{B}$ be a third geometric type-2 fuzzy set $\widetilde{C}$.

- The lower surface of $\widetilde{C} =$ the maximum, as given by the surface clipping algorithm, of the lower surfaces of $\widetilde{A}$ and $\widetilde{B}$.
- The upper surface of $\widetilde{C} =$ the maximum, as given by the surface clipping algorithm, of the upper surfaces of $\widetilde{A}$ and $\widetilde{B}$.

This performs the join across the entire domain of $\widetilde{A}$ and $\widetilde{B}$ giving the logical OR.

The logical OR of the example geometric type-2 fuzzy sets $\widetilde{A}$ and $\widetilde{B}$ is depicted in full in Fig. 14 and just the FOU in Fig. 15

We have now seen how the concepts of geometric primitives and their intersections used in type-1 geometric fuzzy operations are taken forward with the addition of a third dimension to give type-2 geometric fuzzy operators. Negation and implementation may be implemented in a similar fashion [8]. The next section presents a method for defuzzifying a geometric type-2 fuzzy set.

4 Defuzzification of Geometric Type-2 Fuzzy Sets

Only one defuzzifier has been defined for geometric fuzzy sets based on the centre of area defuzzifier [4, 7]. For a discrete type-1 fuzzy set, the centre of area

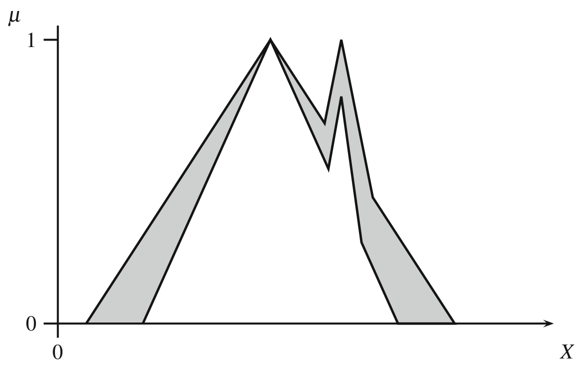


Fig. 15 The FOU of the geometric type-2 fuzzy set $\widetilde{C} = \widetilde{A} \cup \widetilde{B}$

defuzzifier calculates the weighted midpoint of the membership function typically expressed as:

$$C_A = \frac{\sum_{i=1}^{n} \mu_A(x_i)x_i}{\sum_{i=1}^{n} \mu_A(x_i)} \tag{10}$$

where A is a discrete type-1 fuzzy set made up of n discrete points. The centre of area defuzzifier does exactly what is says, it identifies the geometric centre of the membership function that defines that fuzzy set.

For a geometric type-1 fuzzy set, the centre of the area can be calculated by treating the membership function as a closed polygon. The centroid of a polygon [1, 3] can be calculated by deconstructing the polygon into a collection of triangles. Each of these triangles has one vertex at $(0, 0)$ with the other two vertices taking values in order from one of the line segments that forms the polygon. This means a polygon with n vertices can be broken down into n triangles. The centroid of the polygon is the weighted average of the area and centre of these triangles. Consider the polygon P_1 depicted in Fig. 16a. This polygon has four vertices and can therefore be broken down into four triangles t_1 to t_4 as depicted in Fig. 16b–e where the dotted lines depict P_1. Note that the triangles t_1, t_2 and t_3 all encompass an area that lies outside the polygon P_1. The sum of these areas from t_1 to t_3 is equal to the entire area of the triangle t_4. Since, a signed value for the area is taken for each triangle these overlapping areas will cancel out. This is because the sign of area of t_4 will be the opposite to all the other triangles.

The signed area of a triangle is given by the half of the cross product of two of the edge vectors. The centre of a triangle is the sum of the vertices divided by three. The area of t_1 is therefore

$$\text{Area } t_1 = \frac{(x_1 - 0)(y_2 - 0) - (x_2 - 0)(y_1 - 0)}{2} = \frac{x_1 y_2 - x_2 y_1}{2} \tag{11}$$

Since all the triangles from the polygon P_1 contain the vertex $(0, 0)$ the area A of any triangle t_i is given by

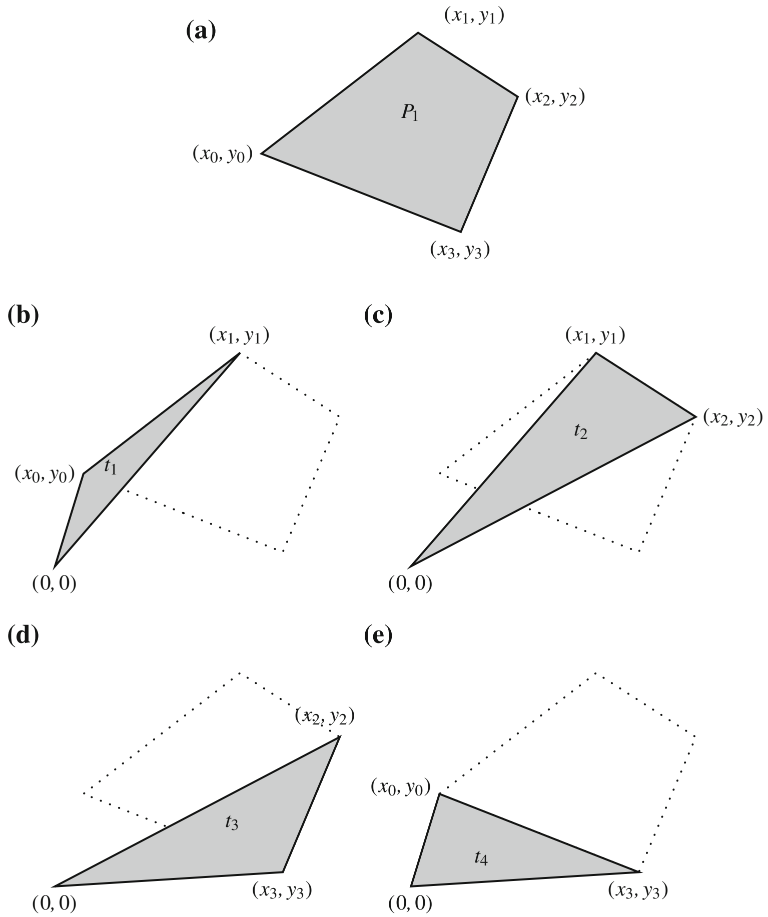


Fig. 16 Calculating the centroid of *Polygon* P_1 using constituent triangles t_1 to t_4

$$A(t_i) = \frac{x_i y_{i+1} - x_{i+1} y_i}{2} \tag{12}$$

An assumption is made that the polygon starts and ends at the same vertex (x_0, y_0). The x-component of the centre or centroid C of the triangle t_i is given by

$$C(t_i) = \frac{x_i + x_{i+1}}{3} \tag{13}$$

The x-component of the centroid C of a polygon P is given by

$$C = \frac{\sum_{i=0}^{n-1} A(t_i) C(t_i)}{\sum_{i=0}^{n-1} A(t_i)} \tag{14}$$

where n is the number of vertices that make up P, $A(t_i)$ and $C(t_1)$ are given by Eqs. (12) and (13), respectively. Substituting Eqs. (12) and (13) into Eq. (14) gives

$$C = \frac{\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)}{3(\sum_{i=0}^{n-1} x_i y_{i+1} - x_{i+1} y_i)} \tag{15}$$

which is used to calculate the centroid of a geometric type-1 fuzzy set.

The centroid of geometric type-2 fuzzy set is calculated in much the same way. The geometric membership function is already made up triangles, so there is no need to do any further decomposition. All that needs to be done is that this area and centroid of each triangle must be calculated in 3D and the weighted average taken [7]. This calculation requires the following notational definitions:

The ith triangle t in the polyhedron $t^i = \begin{bmatrix} x_1^i & y_1^i & z_1^i \\ x_2^i & y_2^i & z_2^i \\ x_3^i & y_3^i & z_3^i \end{bmatrix}$

The x-value of the centroid of $t_i = C^i$

The area of $t_i = A^i$

The centroid of a geometric type-2 fuzzy set is given in Eq. (16) where $C_{\tilde{A}}$ is the centroid of a type-2 fuzzy set $\widetilde{A}$ made up of n triangles.

$$C_{\tilde{A}} = \frac{\sum_{i=1}^{n} C^i A^i}{\sum_{i=1}^{n} A^i} \tag{16}$$

Since we are only interested in the x component of a triangles centroid we only need to work out the arithmetic mean of the x components of three vertices that make up that triangle as given by Eq. (17).

$$C^i = \frac{x_1^i + x_2^i + x_3^i}{3} \tag{17}$$

The area on a single triangle in 3D is calculated by

$$A^i = 0.5\sqrt{\begin{array}{c} ((y_2^i - y_1^i)(z_3^i - z_1^i) - (y_3^i - y_1^i)(z_2^i - z_1^i))^2 + \\ ((x_2^i - x_1^i)(z_3^i - z_1^i) - (x_3^i - x_1^i)(z_2^i - z_1^i))^2 + \\ ((x_2^i - x_1^i)(y_3^i - y_1^i) - (x_3^i - x_1^i)(y_2^i - y_1^i))^2 \end{array}} \tag{18}$$

So, the centroid of a geometric type-2 fuzzy set is calculated as the centre of the volume that makes up that sets membership function. Although this is analogous to the centroid of a geometric type-1 fuzzy set there is one important difference. The membership function of a geometric type-1 fuzzy set is a piecewise-linear function, which is closed to form a polygon for defuzzification. For a geometric type-2 fuzzy set the membership function is already a closed polyhedron. So, why is this important? It means a type-1 geometric membership function can only ever have on line of symmetry parallel to the y axis. A type-2 geometric membership

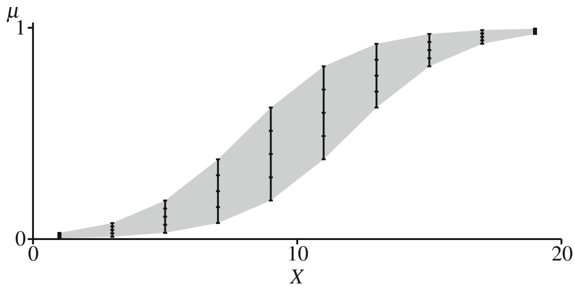


Fig. 17 The FOU of the geometric type-2 fuzzy set $\widetilde{S}$

function can also have a line of rotational symmetry. A geometric defuzzifier will identify this line of symmetry, which may not be the required answer. Consider the FOU of a geometric type-2 fuzzy set $\widetilde{S}$ depicted in Fig. 17. Most people would think the centroid of this set should be around 14. Indeed if we discretise $\widetilde{S}$ and calculate the type-reduced centroid we get an answer of 14.16. The geometric defuzzifier calculate the centroid as 10.00. Clearly, sets with rotational symmetry are problematic to geometric type-2 fuzzy logic.

5 Conclusion

This chapter has presented a review of geometric fuzzy systems of type-1 and type-2. Geometric systems offer advantages over conventional fuzzy systems. The models of the membership functions are more accurate. Furthermore, the operators maintain this accuracy throughout the inference process. Geometric fuzzy sets offer models over a truly continuous domain. For type-1 systems discrete models are quicker to process, for type-2 systems the performance of the geometric model far exceeds the discrete equivalent. Although new highly efficient representations [10, 12, 14] are yet to be benchmarked against the geometric approach. For type-2 systems, the problem of defuzzifing type-2 fuzzy sets with rotational symmetry remains unsolved.

References

1. Bashein, G., Detmer, P.R.: Centroid of a Polygon. In: Heckbert, P.S. (ed.) Graphics Gems IV, pp. 3–6. Academic Press, Massachusetts (1994)
2. Bentley, J.L., Ottmann, T.A.: Algorithms for reporting and counting geometric intersections. IEEE Trans. Comput. **28**(9), 643–647 (1979)
3. Bourke, P.: Calculating the area and centroid of a polygon). http://local.wasp.uwa.edu.au/~pbourke/geometry/lineline/polyarea/ or on CDROM from http://local.wasp.uwa.edu.au/~pbourke/. July 1988

4. Coupland, S.: Type-2 fuzzy sets: geometric defuzzification and type-reduction. In: Proceedings of the IEEE Symposium on Foundations of Computational Intelligence, Honolulu, Hawaii, pp. 622–629, April 2007
5. Coupland, S., John, R.: A new and efficient method for the type-2 meet operation. In: Proceedings of FUZZ-IEEE 2004, Budapest, Hungary, pp. 959–964, July 2004
6. Coupland, S., John, R.: Fuzzy logic and computational geometry. In: Proceedings of RASC 2004, Nottingham, England, pp. 3–8, Dec 2004
7. Coupland, S., John, R.: A fast geometric method for defuzzification of type-2 fuzzy sets. IEEE Trans. Fuzzy Syst. **16**(4), 929–941 (2008)
8. Coupland, S., John, R.: New geometric inference techniques for type-2 fuzzy sets. Int. J. Approx. Reason. **49**(1), 198–211 (2008)
9. Guigue, P., Devillers, O.: Fast and robust triangle-triangle overlap test using orientation predicates. J. Graph. Tools **8**(1), 25–32 (2003)
10. Hamrawi, H., Coupland, S., John, R.: A novel alpha-cut representation for type-2 fuzzy sets. In: FUZZ IEEE 2010 (WCCI 2010), IEEE, Barcelona, Spain, IEEE, pp. 1–8, July 2010
11. Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets. Fuzzy Sets Syst. **122**, 327–348 (2001)
12. Mendel, J.M., Liu, F., Zhai, D.: Alpha-plane representation for type-2 fuzzy sets: theory and applications. IEEE Trans. Fuzzy Syst. **17**, 1189–1207 (October 2009)
13. Möller, T.: A fast triangle-triangle intersection test. J. Graph. Tools **2**(2), 25–30 (1997)
14. Wagner, C., Hagras, H.: Toward general type-2 fuzzy logic systems based on zslices. IEEE Trans. Fuzzy Syst. **18**(4), 637–660 (Aug. 2010)

Type-2 Fuzzy Sets and Bichains

John Harding, Carol L. Walker and Elbert Walker

Abstract This chapter is a continuation of the study of the variety generated by the truth value algebra of type-2 fuzzy sets. That variety and some of its reducts were shown to be generated by finite algebras, and in particular to be locally finite. A basic question remaining is whether or not these algebras have finite equational bases, and that is our principal concern in this chapter. The variety generated by the truth value algebra of type-2 fuzzy sets with only its two semilattice operations in its type is generated by a four element algebra that is a bichain. Our initial goal is to understand the equational properties of this particular bichain, and in particular whether or not the variety generated by it has a finite equational basis.

1 Introduction

The underlying set of the algebra of truth values of type-2 fuzzy sets is the set $M = Map([0,1],[0,1])$ of all functions from the unit interval into itself. This set is equipped with the binary operations $+$ and $\cdot$, the unary operation *, and the nullary operations $\bar{1}$ and $\bar{0}$ as spelled out below, where $\vee$ and $\wedge$ denote maximum and minimum, respectively.

J. Harding · C. L. Walker (✉) · E. Walker
New Mexico State University, Las Cruces 88003 NM, USA
e-mail: hardy@nmsu.edu

J. Harding
e-mail: jharding@nmsu.edu

E. Walker
e-mail: elbert@nmsu.edu

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_7,

$$(f + g)(x) = \sup\{f(y) \wedge g(z) : y \vee z = x\}$$
$$(f \cdot g)(x) = \sup\{f(y) \wedge g(z) : y \wedge z = x\}$$
$$f^*(x) = \sup\{f(y) : 1 - y = x\} = f(1 - x)$$

$$\bar{1}(x) = \begin{cases} 0 & \text{if } x \neq 1 \\ 1 & \text{if } x = 1 \end{cases} \quad \text{and} \quad \bar{0}(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}$$

The algebra of truth values of type-2 fuzzy sets was introduced by Zadeh in 1975, generalizing the truth value algebras of ordinary fuzzy sets, and of interval-valued fuzzy sets. (Sometimes in the fuzzy literature, the operations $+$ and $\cdot$ are denoted $\sqcup$ and $\sqcap$, respectively, but we choose to use the less cumbersome notations $+$ and $\cdot$. We also frequently write fg instead of $f \cdot g$.) The definitions of the convolutions $+$, $\cdot$, and $*$ are sometimes referred to as Zadeh's extension principle.

Definition 1 The algebra $\mathbb{M} = (M, +, \cdot, *, \bar{1}, \bar{0})$ is the *algebra of truth values* for fuzzy sets of type-2.

Type-2 fuzzy sets, that is, fuzzy sets with this algebra $\mathbb{M}$ of truth values, play an increasingly important role in applications, making $\mathbb{M}$ of some theoretical interest. See, for example, [1–6].

We are concerned here with the equational properties of this algebra, much as one is concerned with the equational properties of the Boolean algebras used in classical logic. The main question we are interested in is whether there is a finite equational basis for the variety $\mathscr{V}(\mathbb{M})$ generated by $\mathbb{M}$. We have made some progress toward this, and other questions, but it remains open.

An important step in understanding the equational theory of $\mathbb{M}$ was taken in [7, 8] where the operations $+$ and $\cdot$ were written in a tractable way using the auxiliary operations L and R, where f^L and f^R are the least increasing and decreasing functions, respectively, above f. Using this, it was shown that $\mathbb{M}$ satisfies the following equations.

Proposition 1 *Let* $f,\ g,\ h \in M$.

1. $f + f = f; f \cdot f = f$
2. $f + g = g + f; f \cdot g = g \cdot f$
3. $f + (g + h) = (f + g) + h; f \cdot (g \cdot h) = (f \cdot g) \cdot h$
4. $f + (f \cdot g) = f \cdot (f + g)$
5. $\bar{1} \cdot f = f; \bar{0} + f = f$
6. $f^{**} = f$
7. $(f + g)^* = f^* \cdot g^*; (f \cdot g)^* = f^* + g^*$

Algebras, such as $\mathbb{M}$, that satisfy the above equations have been studied in the literature under the name *De Morgan bisemilattices* [9–11].

Definition 2 A *variety* of algebras is the class of all algebras of a given type satisfying a given set of identities (a basis for the variety). Equivalently (by a famous theorem of Birkhoff), a variety is a class of algebras of the same type which is closed under the taking of homomorphic images, subalgebras and (direct) products.

Definition 3 For an algebra $\mathbb{A}$, the variety $\mathscr{V}(\mathbb{A})$ generated by $\mathbb{A}$ is the class of all algebras with the same type as $\mathbb{A}$ that satisfy all the equations satisfied by $\mathbb{A}$. An algebra $\mathbb{A}$ is locally finite if each finite subset of $\mathbb{A}$ generates a finite subalgebra of $\mathbb{A}$, and a variety is locally finite if each algebra in the variety is locally finite.

An advance in understanding $\mathbb{M}$ and its equational properties came in [12], where it was shown that the variety $\mathscr{V}(\mathbb{M})$ is finitely generated, meaning it is generated by a single finite algebra. In fact, it is generated by the complex algebra (algebra of subsets) of a 5-element bounded chain with involution. In this same paper, it was shown $\mathscr{V}(\mathbb{M})$ is generated by a smaller 12-element De Morgan bisemilattice, but this algebra is not so easily described. An important consequence of this result is an algorithm to determine whether an equation holds in $\mathbb{M}$. One simply checks to see if the equation holds in the finite algebra generating $\mathscr{V}(\mathbb{M})$. In this same paper, a normal form for terms in $\mathscr{V}(\mathbb{M})$ was given, and used to develop a syntactic algorithm to determine when an equation holds in $\mathscr{V}(\mathbb{M})$.

It is natural to consider whether the equations in Proposition 1 could be a basis for the variety $\mathscr{V}(\mathbb{M})$; that is, whether or not every equation satisfied by the algebras in $\mathscr{V}(\mathbb{M})$ is a consequence of those equations in Proposition 1. This is not the case as $\mathscr{V}(\mathbb{M})$ is locally finite, and there are De Morgan bisemilattices that are not locally finite, such as certain ortholattices. So to find a basis for the variety $\mathscr{V}(\mathbb{M})$ one must add equations to this list. We will exhibit later some equations that hold in $\mathbb{M}$ that are not consequences of the equations above. Whether there is a finite basis for $\mathscr{V}(\mathbb{M})$ remains open.

The observant reader at this point will have considered Baker's Theorem [13] that says a finitely generated congruence distributive variety has a finite basis. Unfortunately we cannot apply this result as $\mathscr{V}(\mathbb{M})$ is not congruence distributive, as is noted in a later section.

We decided to simplify the problem, and restrict attention to equations involving only the operations $+$ and $\cdot$ and not using the negation $*$ or constants $\bar{1}$ and $\bar{0}$.

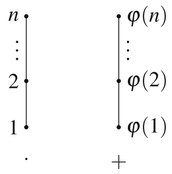
Definition 4 An algebra $(A, \cdot, +)$ with two binary operations is called a *bisemilattice* if it satisfies Eqs. 1–3 of Proposition 1, and a Birkhoff system if it satisfies Eqs. 1–4 of Proposition 1.

Of course the reduct $(M, +, \cdot)$ of $\mathbb{M}$ to this type satisfies Eqs. 1–4 of Proposition 1, so is a Birkhoff system.

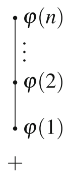
In any bisemilattice$(A, \cdot, +)$, the binary operations $\cdot$ and $+$ induce partial orders by $x \leq {}_{\cdot}y$ if $x = xy$ and $x \leq {}_{+}y$ if $x + y = y$. It is not difficult to show that these two partial orders are the same if and only if the bisemilattice is a lattice.

Definition 5 A bisemilattice $(A, \cdot, +)$ is a *bichain* if the two partial orders $\leq_{\cdot}$ and $\leq_{+}$ are chains.

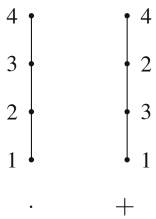
A bichain is thus given by a set and two linear orderings on it. This is the same as giving an ordering on a set, and a permutation on that set. Of particular importance here will be finite bichains. Here we often assume the underlying set is $\{1, \ldots, n\}$, that the $\cdot$-ordering is $1 <_{\cdot} 2 <_{\cdot} \cdots <_{\cdot} n$, and that the $+$-ordering is given by some permutation φ of $\{1, 2, \ldots, n\}$. The situation is shown in below.



Any permutation φ gives an ordering of $1, 2, \ldots, n$ for the $+$-order, so up to isomorphism there are $n!$ n-element bichains. We assume the $\cdot$-order is $1 < 2 < \cdots < n$ and then just give the $+$-order. So we may depict bichains in the following manner:

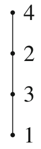


Our reduct $(M, \cdot, +)$ is a Birkhoff system. Of course, the variety generated by this algebra is generated by the reduct of the 12-element De Morgan system that generates $\mathcal{V}(\mathbb{M})$, but one can do better. In [12] it was shown that the variety generated by $(M, +, \cdot)$ is generated by the 4-element bichain we call $\mathbb{B}$, shown in below



.

Of course, this bichain can be depicted simply by



$+$

While there is considerable literature on bisemilattices (see, for example, [11, 14, 15]), there seems to be relatively little known about the quite natural case of bichains. Our efforts here are largely devoted to studying bichains and the varieties they generate. We believe this is of interest for its own sake, as well as for its application to understanding equational properties of $\mathbb{M}$. One thing it enables us to do is to produce equations satisfied by $\mathbb{M}$ that are not a consequence of the Eqs. 1–4 of Proposition 1. We list four such equations below. The names come from their donations by Fred (L)inton, Peter (J)ipsen, Keith (K)earnes, and a key equation (S) that is a splitting equation of a certain variety.

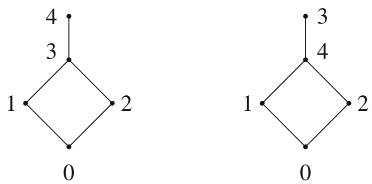
$$xz + y(x + z) = (x + z)(y + xz) \tag{L}$$

$$y(x + xz) = y(x + y)(x + z) \tag{J}$$

$$x(y + z)(xy + xz) = x(y + z) + (xy + xz) \tag{S}$$

$$x(xy + xz) = xy + xz \tag{K}$$

These equations hold in $\mathbb{B}$ as is easily checked. However, they do not hold in the variety of Birkhoff systems, so are not consequences of Eqs. 1–4 of Proposition 1. The first three equations fail in the 3-element bichain denoted $\mathbb{A}_5$ in the following section. The fourth is valid in all six 3-element bichains. Each subset of a bichain is a subalgebra, and it follows that this fourth equation (K) is valid in all bichains; however, it fails in the Birkhoff system depicted below.



$\cdot$ $+$

We further remark that using the third equation (S) and several equations valid in all bichains, such as (K), we can prove the first two equations (L) and (J). Rather, a software package called Prover9 [16] can prove them. We conjecture that any equation valid in $\mathbb{B}$ can be proved from (S) and equations valid in *BiCh*, or equivalently, that $\mathscr{V}(\mathbb{B})$ is defined by the equations defining *BiCh* and the equation (S).

2 Subvarieties of $\mathscr{V}(\mathbb{B})$

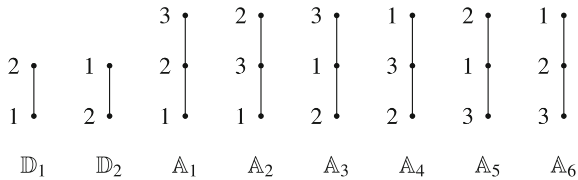
Let *BiSemi* be the variety of all bisemilattices, *Birk* be the variety of all Birkhoff systems, *BiCh* be the variety generated by all bichains, *DL* be the variety of all distributive lattices, and *SL* be the variety of all bisemilattices satisfying $x \cdot y = x + y$, which is called the variety of semilattices. For any bisemilattice $\mathbb{S}$ we let $\mathscr{V}(\mathbb{S})$ be the variety generated by $\mathbb{S}$.

Proposition 2 *Every bichain is a Birkhoff system, so BiCh* $\subseteq$ *Birk.*

Proof Suppose x, y are elements of a bichain. Then each of xy and $x + y$ is either x or y, and we check that in the four possible cases $x(x + y) = x + xy$. □

The inclusion *BiCh* $\subseteq$ *Birk* is proper, since (K) is valid in all bichains, but not in all Birkhoff systems.

Below we describe and name all bichains with two or three elements.



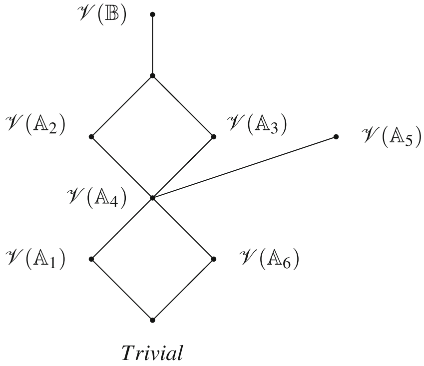
Note that $\mathbb{D}_1$ and $\mathbb{A}_1$ are distributive lattices so generate the variety *DL*, and $\mathbb{D}_2$ and $\mathbb{A}_6$ are semilattices so generate *SL* [13]. By [17] the join of *DL* and *SL* is the variety of distributive bisemilattices; that is, bisemilattices satisfying both distributive laws. As $\mathbb{D}_1$ and $\mathbb{D}_2$ are subalgebras of $\mathbb{A}_4$, and $\mathbb{A}_4$ is a quotient of their product, $\mathbb{A}_4$ generates $DL \vee SL$. By [11] the variety of bisemilattices satisfying the meet-distributive law $x(y + z) = xy + xz$ covers the distributive bisemilattices, as does the variety of bisemilattices satisfying the join-distributive law $x + yz = (x + y)(x + z)$. As $\mathbb{A}_2$ satisfies meet-distributivity but not join distributivity, and $\mathbb{A}_3$ satisfies join distributivity but not meet distributivity, $\mathscr{V}(\mathbb{A}_2)$ and $\mathscr{V}(\mathbb{A}_3)$ cover $\mathscr{V}(\mathbb{A}_4)$. As $\mathbb{A}_2$ and $\mathbb{A}_3$ are subalgebras of $\mathbb{B}$, we have $\mathscr{V}(\mathbb{A}_2) \vee \mathscr{V}(\mathbb{A}_3)$ is contained in $\mathscr{V}(\mathbb{B})$. Using the Universal Algebra calculator [18] we can find an equation to show this containment is strict. The algebras $\mathbb{A}_2$ and $\mathbb{A}_3$ satisfy

$$(x+z)(wx+w+y) = (x+z)(xy+w+y)$$

and this equation fails in $\mathbb{B}$. The program also provides equations to show neither $\mathscr{V}(\mathbb{A}_2)$ nor $\mathscr{V}(\mathbb{A}_3)$ is contained in $\mathscr{V}(\mathbb{A}_5)$, and $\mathscr{V}(\mathbb{A}_5)$ is not contained in $\mathscr{V}(\mathbb{B})$.

$$z(x+z)(y+z) = z(z+xy)$$
$$z+xz+yz = z+z(x+y)$$
$$x(y+z)(xy+xz) = x(y+z)+(xy+xz)$$

The first holds in $\mathbb{A}_5$ and fails in $\mathbb{A}_2$, the second holds in $\mathbb{A}_5$ and fails in $\mathbb{A}_3$, and the third holds in $\mathbb{B}$ and fails in $\mathbb{A}_5$. A diagram of the containments between these varieties follows.



Our conjecture is that $\mathscr{V}(\mathbb{B})$ is the largest subvariety of *BiCh* not containing $\mathbb{A}_5$, a situation known as a **splitting**. If this is indeed the case, $\mathscr{V}(\mathbb{B})$ is defined by a single equation called a **splitting equation**, together with equations defining *BiCh*. In this case, a splitting equation is

$$x(y+z)(xy+xz) = x(y+z)+(xy+xz) \tag{S}$$

That (S) is the splitting equation of $\mathbb{A}_5$ in *BiCh* comes through the fact that $\mathbb{A}_5$ is weakly projective in this variety, a topic we shall return to later. We remark that (S) is a type of generalized distributive law, with the left side of (S) being the meet of the two sides of the usual distributive law, and the right side of (S) being their join. We have not yet determined an equational basis of *BiCh*, and indeed do not even know if this variety is finitely based.

3 Bichains in the variety $\mathscr{V}(\mathbb{B})$

To lend some credence to our conjecture that $\mathscr{V}(\mathbb{B})$ is the largest subvariety of *BiCh* not containing $\mathbb{A}_5$, we use this section to show that a bichain belongs to $\mathscr{V}(\mathbb{B})$ if and only if it does not contain $\mathbb{A}_5$ as a subalgebra. We remark that if the variety *BiCh* were congruence distributive, our conjecture would follow from this using Jónsson's Lemma and Łoś's Theorem.

Theorem 1 *For a bichain* $\mathbb{C}$*, the following are equivalent.*

1. $\mathbb{C} \in \mathscr{V}(\mathbb{B})$.
2. $\mathbb{A}_5$ *is not a subalgebra of* $\mathbb{C}$.
3. $\mathbb{C}$ *satisfies (S).*

Proof ($1 \Rightarrow 3$) This is of course simply a matter of checking that the equation (S) holds in $\mathbb{B}$, but the situation is a bit more interesting than this. Note there is a congruence on $\mathbb{B}$ that collapses only the two middle elements $\{2,3\}$, and the resulting quotient is a distributive lattice. Take any equation $s = t$ that holds in all distributive lattices. If this equation is to fail in $\mathbb{B}$ for some choice of elements, it must be that s and t evaluate to 2 and 3. As $\{2,3\}$ is a subalgebra of $\mathbb{B}$ isomorphic to the 2-element semilattice, it then follows that $st = s + t$ holds in $\mathbb{B}$. The equation (S) is an instance of this, taking $s = t$ to be the meet distributive law.

($3 \Rightarrow 2$) Take $x = 2$, $y = 1$, and $z = 3$ to see that $\mathbb{A}_5$ does not satisfy (S).

($2 \Rightarrow 1$) To show $\mathbb{C} \in \mathscr{V}(\mathbb{B})$, it is sufficient to show every finite sub-bichain of $\mathbb{C}$ belongs to $\mathscr{V}(\mathbb{B})$. Indeed, if $\mathbb{C} \notin \mathscr{V}(\mathbb{B})$, there is some equation valid in $\mathbb{B}$ that fails in $\mathbb{C}$. This equation involves only finitely many variables, so there is some finitely generated subalgebra of $\mathbb{C}$ that does not belong to $\mathscr{V}(\mathbb{B})$. But as $\mathbb{C}$ is a bichain, every subset of $\mathbb{C}$ is in fact a subalgebra of $\mathbb{C}$. So to show $2 \Rightarrow 1$, it is enough to show this for $\mathbb{C}$ a finite bichain.

We show by induction on $n = |\mathbb{C}|$ that if $\mathbb{A}_5$ is not isomorphic to a subalgebra of $\mathbb{C}$, then $\mathbb{C} \in \mathscr{V}(\mathbb{B})$. For $n \leq 3$ all n-element bichains are given in the figure in the previous section, and all but $\mathbb{A}_5$ are shown to belong to $\mathscr{V}(\mathbb{B})$. Suppose $\mathbb{C}$ has $n \geq 4$ elements. We first establish a lemma that handles several cases.

Lemma 1 *For a finite bichain* $\mathbb{C}$*, let* $\mathbb{C} \cup \{\infty\}$ *be the bichain formed from* $\mathbb{C}$ *by adding a new element to the bottom of the* $\cdot$*-order and the top of the* $+$*-order; let* $\mathbb{C} \cup \{b\}$ *be formed from* $\mathbb{C}$ *by adding a new element to the bottom of both orders; and let* $\mathbb{C} \cup \{t\}$ *be formed from* $\mathbb{C}$ *by adding a new element to the top of both orders. Then if* $\mathbb{C} \in \mathscr{V}(\mathbb{B})$*, so are* $\mathbb{C} \cup \{\infty\}$*,* $\mathbb{C} \cup \{b\}$ *and* $\mathbb{C} \cup \{t\}$*.*

Proof (*Proof of Lemma*). We first show $\mathbb{B} \cup \{\infty\}$, $\mathbb{B} \cup \{b\}$ and $\mathbb{B} \cup \{t\}$ belong to $\mathscr{V}(\mathbb{B})$. Note $\mathbb{B} \cup \{\infty\}$ is the quotient of $\mathbb{B} \times \mathbb{D}_2$ by the congruence θ that has one non-trivial block consisting of $\mathbb{B} \times \{1\}$; $\mathbb{B} \cup \{b\}$ is the subalgebra of $\mathbb{B} \times \mathbb{D}_1$ consisting of $\mathbb{B} \times \{2\}$ and $(1,1)$; and $\mathbb{B} \cup \{t\}$ is the subalgebra of $\mathbb{B} \times \mathbb{D}_2$ consisting of $\mathbb{B} \times \{1\}$ and $(4,2)$. As $\mathbb{D}_1$ and $\mathbb{D}_2$ belong to $\mathscr{V}(\mathbb{B})$, so do these algebras.

Assume $\mathbb{C}$ belongs to $\mathscr{V}(\mathbb{B})$. Then there is a set I, a subalgebra $\mathbb{S} \leq \mathbb{B}^I$, and an onto homomorphism $\varphi : \mathbb{S} \to \mathbb{C}$. Consider the constant function $\overline{\infty}$ in $(\mathbb{B} \cup \{\infty\})^I$ whose constant value is the new element ∞ added to $\mathbb{B}$. In $\mathbb{B}$, $x \cdot \infty = \infty$ and $x + \infty = \infty$. It follows that $\mathbb{S} \cup \{\infty\}$is a subalgebra of this power, and φ extends to a homomorphism from $\mathbb{S} \cup \{\infty\}$ onto $\mathbb{C} \cup \{\infty\}$. The arguments for $\mathbb{C} \cup \{b\}$ and $\mathbb{C} \cup \{t\}$ are similar, using powers of $\mathbb{B} \cup \{b\}$ and $\mathbb{B} \cup \{t\}$. □

(Proof of Theorem continued) Assume the $\cdot$-order of $\mathbb{C}$ is $1 < 2 < \cdots < n$. If the bottom element of the $+$-order of $\mathbb{C}$ is 1, then $\mathbb{C}$ is isomorphic to $\mathbb{C}' \cup \{b\}$ where $\mathbb{C}'$ is the sub-bichain$\{2, \ldots, n\}$ of $\mathbb{C}$. Then by the inductive hypothesis and the above lemma, $\mathbb{C} \in \mathscr{V}(\mathbb{B})$. A similar argument handles the cases where either 1 or n is the top element of the $+$-order of $\mathbb{C}$. Set

$$U = \{k : 2 \leq k \leq n \text{ and } k \text{ precedes } 1 \text{ in the } +\text{-order}\}$$

$$V = \{k : 2 \leq k \leq n \text{ and } 1 \text{ precedes } k \text{ in the } +\text{-order}\}$$

As 1 is not the bottom or top of the $+$-order, U and V are non-empty. Also, as $\mathbb{A}_5$ is not a subalgebra of $\mathbb{C}$, if $u \in U$ and $v \in V$, then $u < v$. Also, as n is not the top element of the $+$-order, V must have at least two elements. So there is some $2 \leq k \leq n-2$ with $U = \{2, \ldots, k\}$ and $V = \{k+1, \ldots, n\}$.

There are congruences θ and ϕ on $\mathbb{C}$ with θ collapsing $\{1, \ldots, k\}$ and nothing else, and ϕ collapsing V and nothing else. Note $\mathbb{C}/\theta$ is isomorphic to the sub-bichain$\{1, k+1, \ldots, n\}$ of $\mathbb{C}$, and $\mathbb{C}/\phi$ is isomorphic to the sub-bichain$\{1, \ldots, k, k+1\}$ of $\mathbb{C}$. It follows from the inductive hypothesis that $\mathbb{C}/\theta$ and $\mathbb{C}/\phi$ belong to $\mathscr{V}(\mathbb{B})$. As θ and ϕ intersect to the diagonal, $\mathbb{C}$ is a subalgebra of their product, so belongs to $\mathscr{V}(\mathbb{B})$. □

At this point, if we had congruence distributivity, it would follow that every subdirectly irreducible in the variety *BiCh* is a bichain, and then the above theorem would imply $\mathscr{V}(\mathbb{B})$ is defined, relative to the equations defining *BiCh*, by the single equation (S). However we do not have congruence distributivity [14].

4 Splitting

In this section we investigate projectivity and splitting for various bichains, and in particular for $\mathbb{A}_5$. Our main result here shows there is a largest subvariety of *BiCh* not containing $\mathbb{A}_5$, and the theorem of the previous section leads us to believe this may be the variety $\mathscr{V}(\mathbb{B})$.

Definition 6 An algebra $\mathbb{P}$ is *weakly projective* in a variety $\mathscr{V}$ if for any two algebras $\mathbb{E}$ and $\mathbb{A}$ in $\mathscr{V}$, for every homomorphism $f : \mathbb{P} \to \mathbb{E}$, and for every onto homomorphism $g : \mathbb{A} \twoheadrightarrow \mathbb{E}$, there is a homomorphism $h : \mathbb{P} \to \mathbb{A}$ with $gh = f$.

The usual definition of projective uses the categorical notion of an epimorphism in place of the onto homomorphism g. In a variety $\mathscr{V}$, there may be more epimorphisms than onto homomorphisms, so an algebra that is weakly projective may not be projective. However, we do not know whether epimorphisms must be onto in either of the varieties *Birk* or *BiCh*.

The following well-known result [19] is a convenient reformulation.

Proposition 3 *An algebra $\mathbb{P}$ is weakly projective in $\mathscr{V}$ if and only if for every onto homomorphism $u : \mathbb{A} \twoheadrightarrow \mathbb{P}$, there is an embedding $r : \mathbb{P} \to \mathbb{A}$ with $u \circ r = id_{\mathbb{P}}$.*

Weak projectives are of interest for several reasons, but our primary one lies in Proposition 4 below. Before stating this, we define for an algebra $\mathbb{P}$ in a variety $\mathscr{V}$,

$$\mathscr{W}(\mathbb{P}) = \{\mathbb{A} \in \mathscr{V} : \mathbb{P} \not\hookrightarrow \mathbb{A}\}$$

Here $\mathbb{P} \not\hookrightarrow \mathbb{A}$ means $\mathbb{P}$ is not isomorphic to a subalgebra of $\mathbb{A}$.

Proposition 4 *If $\mathbb{P}$ is weakly projective in $\mathscr{V}$ and subdirectly irreducible, then $\mathscr{W}(\mathbb{P})$ is a variety, and is the largest subvariety of $\mathscr{V}$ that does not contain $\mathbb{P}$.*

This is a well-known result [19] and not difficult to prove. The situation is sometimes referred to as a splitting, as it splits the lattice of subvarieties of $\mathscr{V}$ into two parts, those that contain the variety $\mathscr{V}(\mathbb{P})$, and those that are contained in $\mathscr{W}(\mathbb{P})$. Further, such a splitting yields an equation, called the splitting equation, defining the variety $\mathscr{W}(\mathbb{P})$ relative to the equations defining $\mathscr{V}$. We now apply these results in our setting.

Proposition 5 *The 2-element distributive lattice $\mathbb{D}_1$ is subdirectly irreducible and weakly projective in BiCh. Its splitting variety $\mathscr{W}(\mathbb{D}_1)$ is the variety SL of semilattices.*

Proof Clearly $\mathbb{D}_1$ is subdirectly irreducible. Let $\mathbb{A}$ be a bichain and $f : \mathbb{A} \twoheadrightarrow \mathbb{D}_1$ be an onto homomorphism. Then there are x and y in $\mathbb{A}$ with $f(x) = 1$ and $f(y) = 2$. Then $f(xy) = 1$ and $f(x+y) = 2$, so xy is different from $x+y$. In any Birkhoff system we have $xy(x+y) = xy$ and $x+y+xy = x+y$. So there is a homomorphism $r : \mathbb{D}_1 \to \mathbb{A}$ defined by $r(1) = xy$ and $r(2) = x+y$, and this homomorphism satisfies $f \circ r = id_{\mathbb{D}_1}$. So $\mathbb{D}_1$ is weakly projective.

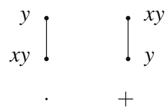
To see that $\mathscr{W}(\mathbb{D}_1) = SL$, note that the two-element semilattice $\mathbb{D}_2$ belongs to $\mathscr{W}(\mathbb{D}_1)$, so one containment is trivial. For the other, suppose $\mathbb{A}$ does not belong to *SL*. Then there are $x, y \in \mathbb{A}$ with xy not equal to $x+y$, giving $\{xy, x+y\}$ is a subalgebra of $\mathbb{A}$ isomorphic to $\mathbb{D}_1$, so $\mathbb{A} \notin \mathscr{W}(\mathbb{D}_1)$. □

Note that for $\mathbb{D}_1$, these results hold also in the larger variety *Birk*.

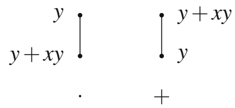
Proposition 6 *The 2-element semilattice $\mathbb{D}_2$ is subdirectly irreducible and weakly projective in BiCh. Its splitting variety $\mathscr{W}(\mathbb{D}_2)$ in BiCh is the variety DL of distributive lattices.*

Proof Clearly $\mathbb{D}_2$ is subdirectly irreducible. Let $\mathbb{A}$ be a bichain and $f : \mathbb{A} \twoheadrightarrow \mathbb{D}_2$ be an onto homomorphism. Then there are x and y in $\mathbb{A}$ with $f(x) = 1$ and $f(y) = 2$.

While we could now just jump to the answer, we build it step at a time to demonstrate an idea that will be used in a later proof. This same idea would have worked above. We first patch up the meet operation and consider the following:



We have $f(xy) = f(x)f(y) = (1)(2) = 1$ and $f(y) = 2$. Also $(xy)y = xy$. So $\{x, xy\}$ is a 2-element subset of $\mathbb{A}$ that works well with respect to meet. But it doesn't work well with respect to join since we would like that $y + xy = xy$ and there is no reason for this to be true. We work with what we have now and get it to work with respect to join.



Now by construction, this works well with respect to join, as $y + y + xy = y + xy$. It also works with respect to meet, as $y + xy = y(x + y)$, so $y(y + xy) = y + xy$. So $\{y + xy, y\}$ is a subalgebra of $\mathbb{A}$, $f(y + xy) = 2 + 1 = 1$ and $f(y) = 2$. So there is $r : \mathbb{D}_2 \to A$ with $r(1) = y + xy$ and $r(2) = y$, so $\mathbb{D}_2$ is weakly projective.

We next show that $\mathscr{W}(\mathbb{D}_2) = DL$. Surely $\mathscr{W}(\mathbb{D}_2) \supseteq DL$. To show $\mathscr{W}(\mathbb{D}_2) \subseteq DL$, suppose $\mathbb{A} \in BiCh$ and $\mathbb{A}$ has no subalgebra isomorphic to $\mathbb{D}_2$. Note that for any $x, y \in \mathbb{A}$ we have $x[x(x + y)] = x(x + y)$, and Birkhoff's equation $a(a + b) = a + ab$ gives $x + x(x + y) = x(x + x + y) = x(x + y)$. As $\mathbb{A}$ has no subalgebra isomorphic to $\mathbb{D}_2$, it follows that $x(x + y) = x$ for each $x, y \in \mathbb{A}$, and then by Birkhoff's equation that $x + xy = x$ for each $x, y \in \mathbb{A}$. So $\mathbb{A}$ is a lattice.

Consider the equations

$$x(x + y)(xz + y) = x(x + y)(xz + y + z)$$

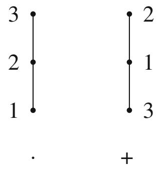
$$z(x + y)(y + xz) = z(x + y)(y + z + xz)$$

Both hold in every bichain. To see this, as these equations involve three variables it is enough to check them in each 3-element bichain, and this is not difficult. So these equations hold in the variety *BiCh*, hence also in $\mathbb{A}$. The first does not hold in the 5-element modular, non-distributive lattice $\mathbb{M}_3$, and the second does not hold in the 5-element non-modular lattice $\mathbb{N}_5$. So $\mathbb{A}$ is a lattice containing neither $\mathbb{M}_3$ nor $\mathbb{N}_5$ as a subalgebra, showing $\mathbb{A}$ is a distributive lattice [13]. □

Note that our proof shows more. The algebra $\mathbb{D}_2$ is weakly projective in the larger variety *Birk*. It therefore has a splitting variety in *Birk*, but this is not *DL*, but the variety *Lat* of all lattices. This proof also shows $Lat \cap BiCh = DL$. In particular,

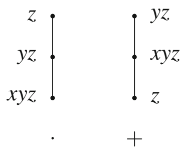
Corollary 1 *Any lattice in* $\mathscr{V}(\mathbb{B}) = \mathscr{V}(\mathbb{M})$ *is distributive.*

Now to the result most pertinent to our variety $\mathscr{V}(\mathbb{B})$. For convenience, we recall what $\mathbb{A}_5$ looks like.

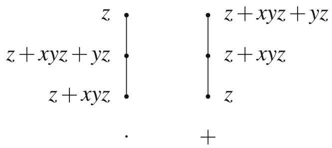


Proposition 7 $\mathbb{A}_5$ *is subdirectly irreducible and weakly projective in BiCh.*

Proof The bichain $\mathbb{A}_5$ is subdirectly irreducible with its minimal congruence being the one collapsing 1 and 2. To see that it is weakly projective, assume $\mathbb{A} \in BiCh$ and $f : \mathbb{A} \twoheadrightarrow \mathbb{A}_5$. Then there are x, y, and z in $\mathbb{A}$ with $f(x) = 1, f(y) = 2$, and $f(z) = 3$. We follow the process in the previous proof to try to build a subalgebra of $\mathbb{A}$ that is isomorphic to $\mathbb{A}_5$. As our first step, we fix meets.



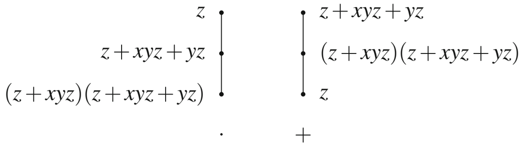
So now meets are okay, but joins are a problem. We fix them, bearing in mind we may wreck our meets when we do so.



So now we have fixed joins, but have troubles with the meets again. Before we continue further, note that Birkhoff's identity $a(a+b) = a+ab$ gives the following

$$
\begin{aligned}
z(z+xyz+yz) &= z+z(yz+xyz)\\
&= z+zyz(yz+x)\\
&= z+yz(yz+x)
\end{aligned}
$$

So in fixing meets again, we may leave intact the top two elements of the $\cdot$-order to obtain the following.



Birkhoff's identity gives $(z+xyz)(z+xyz+yz) = z+xyz+yz(z+xyz)$. So the join of the bottom two elements of the $+$-order are correct since z will be absorbed when added to this element. To see the join of the top two elements of the $+$-order are correct, we again use Birkhoff's identity.

$$
\begin{aligned}
z+xyz+yz+(z+xyz+yz)*z+xyz) &= (z+xyz+yz)(z+xyz+yz+z+xyz)\\
&= (z+xyz+yz)(z+xyz+yz)\\
&= z+xyz+yz
\end{aligned}
$$

So after the last round of fixing meets, joins also are fixed.

We then get that $\{z, z+xyz+yz, (z+xyz)(z+xyz+yz)\}$ is a subalgebra of $\mathbb{A}$. One easily sees that $f((z+xyz)(z+xyz+yz)) = 1, f(z+xyz+yz) = 2$, and $f(z) = 3$. So $\mathbb{A}_5$ is weakly projective. □

We have shown somewhat more, that $\mathbb{A}_5$ is weakly projective in the larger variety *Birk*. In [20] we are able to extend this result significantly and show any finite bichain not containing the algebra $\mathbb{A}_4$ is weakly projective in the variety *Birk*. However, it is the specific instance given above that is applicable to our study of Type-2 fuzzy sets. The main points are summarized below.

Theorem 2 *The algebra $\mathbb{A}_5$ is subdirectly irreducible and weakly projective in the variety BiCh. Its splitting variety $\mathscr{W}(\mathbb{A}_5)$ in BiCh contains $\mathscr{V}(\mathbb{B})$ and these two varieties contain exactly the same bichains. Equations defining $\mathscr{W}(\mathbb{A}_5)$ are given by the equations defining the variety BiCh and the splitting equation below, which is a generalized form of the distributive law.*

$$x(y+z)(xy+xz) = x(y+z)+(xy+xz). \tag{S}$$

Proof That $\mathbb{A}_5$ is subdirectly irreducible and weakly projective in the variety *BiCh* is the content of Proposition 7. By Proposition 4, $\mathscr{W}(\mathbb{A}_5)$ is a variety and is the largest subvariety of *BiCh* not containing $\mathbb{A}_5$. From its definition, $\mathbb{B}$ belongs to

$\mathscr{W}(\mathbb{A}_5)$, so $\mathscr{W}(\mathbb{A}_5)$ contains $\mathscr{V}(\mathbb{B})$. That $\mathscr{W}(\mathbb{A}_5)$ and $\mathscr{V}(\mathbb{B})$ contain the same bichains is provided by Theorem 1.

It remains to find the splitting equation for $\mathbb{A}_5$ in *BiCh*. Let F be the free Birkhoff system on the generators x, y, and z and let $\varphi : F \to \mathbb{A}_5$ be the homomorphism mapping x, y and z to $1, 2$ and 3, respectively. In the proof of the previous result, we found that $\{(z + yz + xyz)(z + xyz), z + yz + xyz, z\}$ is a subalgebra of F mapping isomorphically onto $\mathbb{A}_5$. Since $\{1, 2\}$ generates the smallest non-trivial congruence on the subdirectly irreducible algebra $\mathbb{A}_5$, it follows from general considerations that the elements of this subalgebra mapped to 1 and 2 give the splitting equation for $\mathbb{A}_5$ in the variety of Birkhoff systems:

$$(z + yz + xyz)(z + xyz) = z + yz + xyz \tag{T}$$

Using the software packages Prover9 and Mace4 [16], we can find an example to show equation (T) is not equivalent to (S) in the variety of Birkhoff systems. However, consider the equations

$$x(x + y)(xz + y) = x(x + y)(xz + y + z) \tag{1}$$

$$x(xy + xz) = xy + xz \tag{2}$$

Considering cases, one checks that these equations are valid in every bichain, so are valid in the variety *BiCh*. Prover9 shows that in the presence of the identities for Birkhoff systems, equations (T), (1), and (2) together imply (S), and (S), (1), and (2) together imply (T). So in the variety*BiCh* we have (T) and (S) are equivalent, showing (S) is the splitting equation for S in the variety *BiCh*. □

We remark that as (S) is not equivalent to the splitting equation for $\mathbb{A}_5$ in the variety *Birk*, the splitting variety for $\mathbb{A}_5$ in *Birk* is strictly larger than the splitting variety for $\mathbb{A}_5$ in *BiCh*, and therefore strictly larger than $\mathscr{V}(\mathbb{B})$. So $\mathscr{V}(\mathbb{B})$ is not simply defined by the equations for Birkhoff systems plus (S), thus we do need some additional equations.

5 Conclusions and Remarks

From a previous paper [12], we know that the variety generated by the truth value algebra of type-2 fuzzy sets with only its two semilattice operations in its type is generated by a 4-element algebra $\mathbb{B}$ that is a bichain and, in particular, a Birkhoff system.

Our aim is to find an equational basis for the variety generated by $\mathbb{B}$. This problem is difficult, but we have some progress. Our technique is to consider a particular 3-element bichain $\mathbb{A}_5$, show it is subdirectly irreducible and weakly projective, hence splitting, and that its splitting variety $\mathscr{W}(\mathbb{A}_5)$ in *BiCh* contains $\mathscr{V}(\mathbb{B})$.

$$\begin{array}{c} Birk \\ | \\ BiCh \\ | \\ \mathscr{W}(\mathbb{A}_5) \\ | \\ \mathscr{V}(\mathbb{B}) \end{array}$$

We conjecture that $\mathscr{W}(\mathbb{A}_5) = \mathscr{V}(\mathbb{B})$. If so, this will show the splitting equation (S) for $\mathbb{A}_5$ then defines $\mathscr{V}(\mathbb{B})$ within *BiCh*. The results of Sect. 3 lend credence to this as we have shown a bichain belongs to $\mathscr{W}(\mathbb{A}_5)$ if and only if it belongs to $\mathscr{V}(\mathbb{B})$.

There remain a number of open problems in connection with this work. These include determining whether or not $\mathscr{W}(\mathbb{A}_5) = \mathscr{V}(\mathbb{B})$, and finding an equational basis for *BiCh*. Together, these will provide an equational basis for $\mathscr{V}(\mathbb{B})$, and hence for the $\cdot, +$ fragment of the truth value algebra $\mathbb{M}$ of type-2 fuzzy sets. One could conjecture that an equational basis for M with all its operations is one for $\mathbb{B}$ plus the equations for negation and the constants.

We have determined [20] that a bichain is weakly projective in the variety *Birk* if and only if it does not contain a copy of the bichain $\mathbb{A}_4$. As each weakly projective subdirectly irreducible algebra gives a splitting of the lattice of subvarieties, this adds to our knowledge of the lattice of subvarieties of Birkhoff systems, and in particular, of subvarieties of *BiCh*. We believe this variety *BiCh* is natural and of interest independent of its connection to fuzzy logic.

Finally, we remark that in preparing this still incomplete work, we made use of Universal Algebra Calculator [18], as well as the programs Prover9 and Mace4 [16] to find and work with equations. After finding equations with these programs we further verified all properties by hand. We are grateful to several people for providing equations of help to us, including Peter Jipsen, Keith Kearnes, and Fred Linton, and also to Anna Romanowska for several communications.

References

1. Dubois, D., Prade, H.: Operations in a fuzzy-valued logic. Inf. Control **43**, 224–240 (1979)
2. John, R.: Type-2 fuzzy sets: an appraisal of theory and applications. Int. J. Uncertainty Fuzziness Knowl. Based Syst. 6(6), 563–576 (1998)
3. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems. Prentice Hall PTR, Upper Saddle River, NJ (2001)
4. Mizumoto, M., Tanaka, K.: Some properties of fuzzy sets of type-2. Inf. Control **31**, 312–340 (1976)
5. Mukaidono, M.: Algebraic structures of truth values in fuzzy logic. Fuzzy Logic and Fuzzy Control, Lecture Notes in Artificial Intelligence, vol. 833, pp. 15–21. Springer-Verlag (1994)
6. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning. Inform Sci. **8**, 199–249 (1975)

7. Emoto, M., Mukaidono, M.: Necessary and sufficient conditions for fuzzy truth values to form a De Morgan algebra. Int. J. Uncertainty Fuzziness Knowl. Based Syst. 7(4), 309–316 (1999)
8. Walker, C., Walker, E.: The algebra of fuzzy truth values. Fuzzy Sets Syst. **149**, 309–347 (2005)
9. Brzozowski, J.A.: De Morgan bisemilattices. In: Proceedings of the 30th IEEE International Symposium on Multiple-Valued Logic (ISMVL 2000) pp. 173–178 (2000)
10. Kikuchi, H., Takagi, N.: De Morgan bisemilattice of fuzzy truth value. In: Proceedings of 32nd IEEE International Symposium on Multiple-Valued Logic (ISMVL 2002) vol. 180 (2002)
11. McKenzie, R., Romanowska, A.: Varieties of ○-distributive bisemilattices. In: Kautschitsch, H., Muller, W.B., Nobauer, W. (eds.) Contributions to General Algebra, (Proc. Klagenfurt Conf., Klagenfurt, Austria, 1978) pp. 213–218 (1979)
12. Harding, J., Walker, C., Walker, E.: The variety generated by the truth value algebra of type-2 fuzzy sets. Fuzzy Sets Syst. **161**(5), 735–749 (2010)
13. Burris, S.N.: Sankappanavar. A Course in Universal Algebra. Springer-Verlag, H.P. (1981)
14. Romanowska, A.: On bisemilattices with one distributive law. Alg. Univ. **10**, 36–47 (1980)
15. Romanowska, A.: On distributivity of bisemilattices with one distributive law. In; Proceedings of the Colloquium on Universal Algebra, Esztergom (1977)
16. McCune, W.: Prover9 and Mace4. http://www.cs.unm.edu/~mccune/Prover9 (2005–2010)
17. Padmanabhan, R.: Regular identities in lattices. Trans. Amer. Soc. **158**, 179–188 (1981)
18. Freese, R., Kiss, E.: UACalc, A universal algebra calculator. http://www.uacalc.org/
19. Freese, R., Ježek, J.: Free lattices, mathematical surveys and monographs. Amer. Math. Soc. (Providence) **42**, 293 (1995)
20. Harding, J., Walker, C., Walker, E.: Projective Bichains. Alg. Univ. **67**, 347–374 (2012)

Type-2 Fuzzy Sets and Conceptual Spaces

Janet Aisbett and John T. Rickard

Abstract Conceptual spaces provide a rich interpretation for computing with words, offering additional structure to that provided by fuzzy set models alone. In fuzzy conceptual spaces, properties are type-2 fuzzy sets on domains, concepts are type-2 fuzzy sets on pairs of properties and an observation is a family of fuzzy sets on domains relevant to a context. These type-2 fuzzy set structures are derived and manipulated using subsethood. This chapter relates such a theory of conceptual spaces to conventional multivariate classification and computing with words (CWW), and illustrates its application to land use assessment tasks.

1 Introduction

This chapter describes an approach to classification and computing with words (CWW) based on conceptual spaces [1]. Conceptual spaces are in turn based on the feature spaces of conventional multivariate classification. Fuzzy conceptual spaces enhance the knowledge structure of conceptual spaces with the ability to model imprecision through type-1 and type-2 fuzzy sets.

Conventional classification assigns an observation to a concept (class) according to some notion of the match between values observed on various domains and the properties that define the concept on those domains [2]. Typically, classes are defined as conjunctions of properties. Reference [1] distinguishes properties defined on domains such as sensory domains as *natural properties*, in

J. Aisbett
Faculty of Science and IT, The University of Newcastle, Newcastle, NSW, Australia
e-mail: janet.aisbett@newcastle.edu.au

J. T. Rickard (✉)
Distributed Infinity, Inc., Larkspur, CO, USA
e-mail: terry.rickard@reagan.com

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*, Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_8,

contrast to abstract or constructed properties. Nevertheless, most domains are assumed to be equipped with a metric.

In the terminology of CWW, a property is a value of a linguistic variable, such as *young* as a value of the variable *age*, together with a fuzzy set model of the value [3, 4]. More complex linguistic descriptors are usually formed as a conjunction of linguistic values, although negation and disjunction may also be invoked. The match between observations (inputs) and a linguistic descriptor (the antecedent) is the firing level of a fuzzy system rule.

The departure of conceptual space modeling [1] from these approaches is to define concepts as labeled sets of properties *and* the associations between these properties. This co-occurrence of properties is a potentially important discriminator. In order to formalize concepts and to compare observations made about an entity with a concept, Rickard represents concepts using square co-occurrence matrices, with dimension equal to the number of properties used to define the concept [5]. He models observed entities using symmetric matrices with entries the minimum of the degrees to which a the entity possesses the two associated properties. Subsethood, the directional measure of overlap of fuzzy sets, is used to define directional similarity, rather than some inverse function of distances.

Table 1 translates terminology from conceptual spaces [6] into their mathematical forms and into the nearest equivalents in CWW and multivariate classification. This table points to another distinguishing feature of conceptual spaces as presented in [6], i.e., the explicit modeling of context.

Table 1 Approximate translations of standard conceptual space (CS) terminology used in [6]

CS	Mathematical	CWW	Multivariate modeling
Label	Member of index set	Word	
Domain	Set, possibly with structure, e.g., topology	Universe—a set	Domain—usually a categorical set, or a subset of reals R or R^n
Property	Indexed function mapping a set into the unit interval	Linguistic value and a fuzzy model of the value	Properties (or features, attributes, characteristics)—usually modeled as elements of categories or connected regions in R^n
Observation	Element in a set, or a vector of elements[a]	Perception—a linguistic value and its fuzzy model	Value on a domain, or a set of such values over multiple domains
Concept	Square matrix representing associations between properties	Linguistic descriptor or conjunction of linguistic values	Class—a set of properties, one on each domain
Context	Indexed set of functions into [0,1]	(Implicitly defined through linguistic variables)	(Implicitly defined through the domains considered)

[a] Converted to symmetric square matrix when compared with a concept

Imprecision has been introduced into conceptual space representations used in land use assessments [7–9]. This is an important application area. Land use involves a rich set of domains, usually real-valued or categorical. Observations indexed by location are recorded directly, as ground-registered observations, or by processing raw data such as satellite imagery. The data are uncertain due to a wide range of factors including processing errors, averaging over heterogeneous regions, changes since raw data were recorded, spatial registration errors, and interpolation errors [10]. Concepts are labeled with a set of linguistic terms such as *mixed forest* or *developed open space* [8, 11, 12]. A land use class description is typically a conjunction of propositions built from descriptors such as *ice cover*, or *impervious surface*, relations such as *greater than 25 %* or *dominated by*, and vague qualifiers such as *generally, most commonly*, as in “generally vegetation accounts for less than 15 % of total cover”.

This chapter extends the representation of imprecision in conceptual spaces, building on [13]. This chapter is organized as follows: the remainder of this section presents notation, conventions, and the key definition of subsethood. Section 2 formally defines type-1 and type-2 fuzzy constructs and operations required for an extended theory of conceptual spaces. Section 3 illustrates their application to land use management. The final section briefly discusses benefits and overheads of fuzzifying conceptual spaces, and suggests further research.

1.1 Notation and Conventions and the Definition of Subsethood

Symbols V, X, Y, Z denote domains and $M(V)$ denotes the set of normal (i.e., unity height) membership functions on V. Symbols F, G, H denote type-1 fuzzy sets on a domain with respective membership functions f, g, h and $\tilde{F}, \tilde{G}, \tilde{H}$ denote type-2 fuzzy sets with respective membership functions $\tilde{f}, \tilde{g}, \tilde{h}$. As usual $\tilde{f}_v : [0, 1] \to [0, 1]$ denotes the secondary membership function at v, i.e., the restriction of $\tilde{f}$ to fixed $v \in V$. We assume all secondary membership functions are normal.

A fuzzy set G is at times identified with its membership function g, and likewise for type-2 fuzzy sets and membership functions. A type-2 fuzzy set $\tilde{G}$ on V is also identified with the fuzzy set on $M(V)$ with membership function

$$m_{\tilde{G}}(F) = \inf\{\tilde{g}(v, f(v)) : v \in V\}, \tag{1}$$

where $\tilde{g} : V \times [0, 1] \to [0, 1]$ and inf is the infimum [14].

Subsethood. Subsethood S_V is a membership function on pairs of fuzzy sets defined on the same domain V. It directionally measures the extent to which the fuzzy sets overlap. If V is finite,

$$S_V(F, G) \equiv S_V(f, g) = \sum\nolimits_{v_i \in V} \min\{f(v_i), g(v_i)\} / \sum\nolimits_{v_i \in V} f(v_i). \tag{2}$$

In general, if V has a measure [15] with respect to which f and g are measurable, the subsethood of F, G is defined as in (2) with Lesbegue integral replacing summation.

Subsethood is extended to pairs of type-2 fuzzy sets $\tilde{G}$ and $\tilde{H}$ on domain V by considering them to be fuzzy sets on $M(V)$ via (1) and applying the Extension Principle. The subsethood of $\tilde{G}$ and $\tilde{H}$ is thus a type-2 fuzzy set with membership

$$\tilde{S}_V\big((\tilde{G},\tilde{H}),y\big) = \sup_{c,d \in M(V)} \big\{\inf\big\{\tilde{g}(v,c(v)),\tilde{h}(v,d(v)) : v \in V\big\} : S_V(c,d) = y\big\} \quad (3)$$

for $y \in [0,1]$. When $\tilde{G}$ and $\tilde{H}$ are interval type-2 fuzzy sets on V the subsethood secondary membership function is unity on a subinterval whose limits can be specified in terms of the bounding functions; [16] presents an algorithm to compute the limits.

2 Fuzzy Conceptual Spaces

Real world situations involve fuzzy observations and fuzzy properties. Observations are imprecise for many reasons besides measurement error. A characteristic is often subjectively assessed or partially satisfied, e.g., does the goldfish show *lassitude*? Even the definition of a property such as *yellow* may need to account for the observer's visual processing, the effects of lighting conditions, and so on.

The formal definition of fuzzy observations and properties that follows is straightforward. However, the expression for the degree that an observed entity possesses a (fuzzy) property is important. Proposition 1 shows this expression generalizes the usual formulation when observations are crisp.

Definition 1 (*Properties and observations* [13])

1. A (fuzzy) property on V is a labeled (type-2) fuzzy set on V. Thus a fuzzy property is defined by a name and a type-2 membership function $\tilde{g} : V \times [0,1] \to [0,1]$.
2. A fuzzy observation on domain V is a labeled fuzzy set on V, where the label denotes the entity observed. A fuzzy observation on a set of domains is a labeled set of fuzzy sets, one for each domain. This set records the observations made about a single entity on each of the domains.
3. The membership of an entity observed as F in a fuzzy property $\tilde{G}$ is

$$\tilde{g}(f)(y) \triangleq \sup\{\min\{f(v),\tilde{g}(v,y)\} : v \in V\},\ y \in [0,1] \quad (4)$$

   where $\tilde{g}$ (resp. f) is the membership function of the property (resp. observation). In the case that the property is representable by a type-1 fuzzy membership function g, (4) reduces to

$$\tilde{g}(f)(y) = \sup\{f(v) : v \in V, g(v) = y\}. \tag{5}$$

When $\tilde{g}$ is interval type-2, (4) reduces to $\tilde{g}(f)(y) = \sup\{f(v) : v \in V, \tilde{g}(v, y) = 1\}$, and when the property collapses to a type-1 fuzzy set this becomes the expression (5).

Proposition 1 *If* $\theta : V \times M(V) \to [0, 1]$ *is defined by* $\theta(v, g) = g(v)$ *for all* $(v, g) \in V \times M(V)$ *then* $\tilde{g}(f)(y) = \theta(F, \tilde{G})$ *for* $\tilde{g}(f)(y)$ *as in Eq.* (4) *and* $f \in M(V)$ *and* $\tilde{g}_v \in M([0, 1]), v \in V$.

Proof Applying the Extension Principle to θ gives $\theta(F, \tilde{G})(y) = \sup\{\min\{f(v), m_{\tilde{G}}(h)\} : (v, h) \in V \times M(V), h(v) = y\}$ where $m_{\tilde{G}}(h) = \inf\{\tilde{g}(v, h(v)) : v \in V\}$ as in (1). By construction, $m_{\tilde{G}}(h) \leq \tilde{g}(v, h(v))$ for all $v \in V$. Therefore, since the supremum in (4) is computed over all $v \in V$ satisfying $h(v) = y, \theta(F, \tilde{G})(y) \leq \sup\{\min\{f(v), \tilde{g}(v, y)\} : v \in V\}$.

However, we can define $h_{\tilde{g},v} \in M(V)$ with $h_{\tilde{g},v}(v) = y$ and $h_{\tilde{g},v}(u) = u_{\tilde{g},v}$ where $\tilde{g}(u, u_{\tilde{g},v}) = 1$ when $u \neq v$. Then $m_{\tilde{G}}(h_{\tilde{g},v}) = \tilde{g}(v, y)$. So $\theta(F, \tilde{G})(y) \geq \sup\{\min\{f(v), \tilde{g}(v, y)\} : v \in V\}$. This completes the proof.

In the context of finite fuzzy sets, Bellman and Zadeh called $\theta(F, G)$ the *fuzzy compatibility* of the reference F with the linguistic value G (see [14] for a discussion). We interpret $\theta(F, G)$ as the membership of an entity observed as F in property G, and extend the interpretation to fuzzy properties $\tilde{G}$. Whether or not the property is a type-2 function on V, membership of entities in the property is a type-2 fuzzy set on the set $M(V)$ of possible observations.

Suppose X consists of all the labels of a set of fuzzy observations with membership functions $f_x, x \in X$ on V (for example, X consists of the identifiers of people whose heights have been measured, or the pixel locations of a satellite image). Then the observations are represented by a membership function $f : X \times V \to [0, 1]$ with $f(x, y) = f_x(v)$. Using (4), the membership of the entities in X in property $\tilde{G}$ can be collected into a type-2 fuzzy set $\tilde{g}(f) : X \times [0, 1] \to [0, 1]$ with

$$\tilde{g}(f)(x, y) = \sup\{\min\{f(x, v), \tilde{g}(v, y)\} : v \in V\}. \tag{6}$$

This type-2 fuzzy set can be interpreted as the degree to which the collection X has property $\tilde{G}$.

Example of membership of fuzzy observation in a fuzzy property. Figure 1 illustrates two examples in which a fuzzy observation has triangular fuzzy membership function and a fuzzy property is an interval type-2 fuzzy set with membership function of the form $\tilde{g}(v, y) = 1$ for $1 - \exp(-cv) \leq y \leq 1 - \exp(-dv)$ and $\tilde{g}(v, y) = 0$ elsewhere. The entity property membership is readily computed from (4) to be $\tilde{g}(f_x)(y) = \sup\{f_x(v) : v \in V_0\}$ where $V_0 = [d^{-1}\ln(1 - y), c^{-1}\ln(1 - y)]$.

The centroid of the fuzzy set in Fig 1a representing the membership of the observed entity in the property is 0.90. If the observation is defuzzified to the value at which its membership function peaks, and if the property is defuzzified by taking the centroid of the secondary membership intervals, the membership in the property is 0.83. These values are shown as asterisks in Fig 1a. The corresponding values in Fig 1b are 0.64 and 0.59. In both cases the fuller modeling increases the defuzzified membership value by about 8%.

We next present the key constructions of conceptual spaces [6] in fuzzy form.

Definition 2 (*Context and concepts*)

1. A *context* is a set I of (possibly fuzzy) properties, or equivalently, words with associated (type-2) fuzzy sets. The properties may or may not be defined on the same domains.
2. A *concept* C in the context I is a named membership function (fuzzy set) on $I \times I$. A concept can be presented as an n-square matrix $[C_{ij}]$ where $\{1, 2, \ldots n\}$ indexes the elements of I and $C_{ij} = C(\tilde{g}_i, \tilde{g}_j)$ is interpreted as the degree to which the occurrence of the property indexed by i coincides with the occurrence of the property indexed by j.
3. A *fuzzy concept* $\tilde{C}$ in the context of a set of (fuzzy) properties I is a named type-2 membership function on $I \times I$, equivalently, a mapping $\tilde{C} : I \times I \times [0, 1] \to [0, 1]$.

The values $C(\tilde{g}_i, \tilde{g}_j)$ and $\tilde{C}(\tilde{g}_i, \tilde{g}_j)$ are interpreted as the way in which property $\tilde{g}_j$ is associated with property $\tilde{g}_i$ in understanding the concept. For example, are fluffy patches near the mouth and a yellowish fin always observed together in a diseased fish, or are these two symptoms most likely to be observed at different stage of the disease and hence unlikely to co-occur? $\tilde{C}(\tilde{g}_i, \tilde{g}_i)$ and $\tilde{C}(\tilde{g}_i, \tilde{g}_i)$ are interpreted as the *salience* of property $\tilde{g}_i$ for the concept. For example, are the

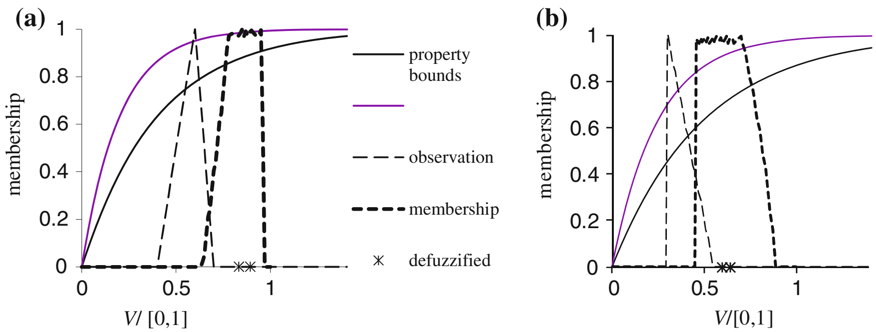


Fig. 1 Degree to which an entity has a fuzzy property when the observation is fuzzy. *Solid lines* are bounding functions of the fuzzy property on V. For these bounding functions and for the membership function of the observation, the horizontal axis is the domain V. The horizontal axis is the unit interval for the property membership (shown as *coarse dashes*) and for the values at the *asterisks* (see text). The vertical axis is the membership interval $[0, 1]$

fluffy patches a reliably observed symptom? As in this example, properties in a context are generally defined on different domains. Reference [6] discusses estimation of these entries.

Definition 2 makes explicit the context in which concepts are modeled (and in which observations are made). Context is important in defining both concepts and similarity, as human judgments of similarity are well known to be context dependent [17]. The same concept has different representation in different contexts, and different problems require different contexts, e.g., a *wealth* concept may require the ownership of cattle in some countries.

When words not in a context overlap words in a context. According to Definition 2, properties are either in a context or are not. However, at times it may be necessary to translate a description into words used in a context. For example, an observation may have been made of a *red* apple when the context only contains property *crimson*. A context I induces a (type-2) fuzzy indicator membership function μ_I on the set of all (fuzzy) properties involved in a concept, namely,

$$\mu_I(\tilde{g}) = \max\{\tilde{S}_V(\tilde{g}, \tilde{g}') : \tilde{g}' \in I\} \tag{7}$$

for subsethood $\tilde{S}_V$ defined as in (3) and for V the domain of $\tilde{g}$. Properties in I obviously have membership unity, and fuzzy properties in I have peak membership $\mu_I(\tilde{g}, y) = \inf_{v \in V}\{\tilde{g}(v, y_v) : \tilde{g}(v, y_v) \geq \tilde{g}(v, s), s \in [0, 1]\}$ at $y = 1$. On the other hand, $\mu_I(\tilde{g}) = 0$ if the support of $\tilde{G}$ is disjoint from that of each property in the context (in particular, if $\tilde{G}$ is defined on a different domain to those on which the properties in I are defined).

The indicator is used to translate concepts between contexts as follows. If $\tilde{C}$ is defined in context I and $\tilde{g}_i, \tilde{g}_j \in I'$ then

$$\tilde{C}(\tilde{g}_i, \tilde{g}_j) = \mu_I(\tilde{g}_i)\mu_I(\tilde{g}_j)\tilde{C}(\tilde{h}_i, \tilde{h}_j) \tag{8}$$

where $\tilde{h}_i, \tilde{h}_j \in I$ and $\mu_I(\tilde{g}_k) = \tilde{S}_{V_k}(\tilde{g}_k, \tilde{h}_k), k = i, j$. (Note that this is a product of fuzzy numbers, and that the choice $\tilde{h}_i, \tilde{h}_j$ may not be unique so the estimate in (8) may not be uniquely defined.) Applying (8) to all pairs in $I' \times I'$ redefines $\tilde{C}$ in context I'.

The computational implications of modeling additional vagueness, even with crisp context, become apparent when we fuzzify the definition of concept similarity.

Definition 3 (*Similarity and pseudo concepts*)

1. The *similarity* of a fuzzy concept $\tilde{C}$ to concept $\tilde{C}'$ in context I is their subsethood $\tilde{S}_{I \times I}(\tilde{C}, \tilde{C}')$ computed using (3) on domain $I \times I$.
2. Suppose fuzzy observations $f = \{f_k\}$ are made of an entity on each domain V_k in a context $I = \{\tilde{g}_k : V_k \times [0, 1] \to [0, 1], k = 1, .., \mathrm{K}\}$. Then the *pseudo-concept* $\tilde{C}_f$ representation of the observed entity in context I is the fuzzy concept with

$$\tilde{C}_f(\tilde{g}_i, \tilde{g}_j) = \min\{\tilde{g}_i(f_i), \tilde{g}_j(f_j)\}. \tag{9}$$

Thus, a pseudo-concept of an entity can be thought of as a matrix with entries the pairwise conjunction of the membership of the entity in the context properties.

3. The membership of the entity in concept $\tilde{C}$ is

$$\tilde{S}_{I\times I}(\tilde{C}, \tilde{C}_f). \tag{10}$$

Equation (9) is the minimum of two fuzzy membership values of the form (4), so entries in $\tilde{C}_f$ are likewise fuzzy membership values which the Extension Principle gives as

$$\tilde{C}_f((\tilde{g}_i, \tilde{g}_j), y) = \sup\{\min\{\tilde{g}_i(f_i)(z), \tilde{g}_j(f_j)(z')\} : \min\{z', z\} = y\} \tag{11}$$

The pseudo-concept construction was first provided in [5] in the case of crisp observations, to allow similarity of an observation to a concept to be defined using subsethood. In standard classification schemes, the similarity of an observation to a number of concepts or classes is computed and the observation is classed as belonging to the concept to which it has greatest similarity. When the similarities are fuzzy numbers, a method of ordering them must be chosen, e.g. by their centroids.

3 Examples of Land Use Suitability and Change Assessments

This section illustrates how land use assessment tasks are conducted within the conceptual space framework. We first look at land suitability assessment, and then at land use change assessment. We show that performance can be improved even with simple models of vagueness, but that this cannot replace more sophisticated modeling.

3.1 Land Suitability Assessment and Classification

In land use suitability assessments, the concepts (classes) represent the suitability of location for a potential use, such as growing a commercial crop or building public housing. Some schemes combine *suites* of concepts, for example, concepts of climate, water resources, environmental hazards, and soil suitability [18], from which complex suitability concepts are constructed. In most schemes, disjoint classes and crisp membership are assumed [8, 11, 12].

Linguistic descriptors identify a set of characteristics such as *propensity to flooding* and *slope* [19] taking values in a small ordinal domain, say {0,1,2,3}. The ordinals reflect the impact of an underlying domain rating on the land suitability. The values might be an expert's assessment with respect to a linguistic descriptor such as *perfect drainage*, or might be a range, such as *stoniness* value of 2 when 10–20% percent of the subsoil is stones, or slope value of 1 for <2°, 2 for 2–8° etc.

The values (sometimes called *indexes* [20]) are combined in various ways to form a land use suitability rating or land use classification. A class in most land use definitional schemes is a set of crisp properties, with a location classified as belonging to a class only if it satisfies all the properties. The FAO land suitability scheme [18] also uses the least favorable factor to determine land suitability. (In our terminology, characteristics and factors are properties $g_i : V_i \to \{n_{i1}, \ldots, n_{iL(i)} : n_{1i} \in [0, 1]\}$ and the rating of the location as suitable under the FAO strategy is the membership value $\min\{g_i(f_i) : i = 1, \ldots, J\}$ for f_i the observation on domain i.)

Concept membership is also sometimes calculated as a weighted average$\sum_i w_i g_i(f_i)$, possibly converted to a categorical rating by partitioning the range of the aggregate score and assigning a rating to each partition. Reference [10] discusses other algorithms including ordered weighted averages in which an additional set of location-dependent weights are assigned according to the size-based rank of the factor value at that location. Alternatively, a continuous land suitability membership function can be defined, for example as $\exp\left(-\alpha \sum_i (1 - w_i g_i(f_i))\right)$ [12] or the jth root of the product of property memberships, $\left(\prod_i g_i(f_i)\right)^{\frac{1}{j}}$ [20].

Translated to our terminology, suitability is determined by a context $I = \{g_i : V_i \to [0, 1] : i = 1, \ldots, J\}$. Each property takes membership values in a set of $L(i)$ ratings between 0 and 1, with higher ratings indicating greater suitability for that crop. Thus g_i has membership function of the form $g_i(v) = n_{il}, v \in V_{il}, l = 1, \ldots L(i)$, where $V_{il} = g_i^{-1}(n_{il})$ and $V_i = \cup_{l=1,\ldots,L(i)} V_{il}$. (For example, Ref. [12] partitions the possible attribute values on each continuous domain into a range of values, then assigns a rating on different domains according to the partition element to which an observation belongs. The partition element is mapped to a score.)

Suppose the overall suitability of a location is determined by a finite partition $\{i_r\}_{r=1,\ldots,L}$ of the range of the aggregate suitability score. After normalization, this partition can be modeled as a property g on the unit interval, taking L membership values, with $i_r \equiv g^{-1}(r)$. If the observation f at location x is crisp on all domains, under this standard approach x has suitability rating r when $\sum_{(i,l_i):f_i \in V_{il_i}} n_{il_i}/J \in i_r$.

In contrast, we define suitability using subsethood as in (10). A suitability concept C of the form *suitable for crop b* is modeled as the up-fuzzification of the concept in which all associations are unity. (Like current models, this assumes all properties are satisfied in an ideally suitable location.) Thus for $(i,j) \in I \times I$, $\tilde{C}\left(\left(g_i, g_j\right), 1\right) = 1$ and $\tilde{C}\left(\left(g_i, g_j\right), y\right) = 0$ for $y \in [0, 1]$. Suitability of x is the subsethood of the fuzzy observation f_x in $\tilde{C}$, which, from (9), (10), (3) and (2), is

$$\tilde{S}_{I \times I}\left(\left(\tilde{C}, \tilde{C}_{f_x}\right), y\right) = \inf_{i,j \in I}\left\{\tilde{C}_{f_x}\left(\left(g_i, g_j\right), y_{ij}\right) : y_{ij} \in [0, 1], y = \sum_{i,j=1}^{J} y_{ij}/J^2\right\}. \quad (12)$$

Next, suppose the membership of an observation $f_x = \{f_{xi} : V_i \to [0, 1)\}$ in a property G_i is a truncated Gaussian, that is, $g_i(f_{xi})(y) = \exp\left(-\alpha_{xi}(y - z_{xi})^2\right)$ for some parameters $\alpha_{xi} > 0, z_{xi} \in [0, 1]$. Such a form is approximated for the expression $g_i(f_{xi})(y)$ given in (5) when properties and fuzzy observations both have Gaussian membership functions. Such a form could also be estimated by an expert observer, for example, assessing the degree to which topsoil at a location x is *stony* in a method analogous to experiments in [21].

By (9), $\tilde{C}_{f_x}\left(g_i, g_j\right)$ is the minimum of two fuzzy sets. Therefore if $z_{xi} < z_{xj}$, by a well-known result generalized in [22], $\tilde{C}_{f_x}\left(\left(g_i, g_j\right), y\right) = \exp\left(-\alpha_{xi}(y - z_{xi})^2\right)$ when $y \in [0, 1]$ is less than the largest crossover point (if any) of the Gaussians, after which $\tilde{C}_{f_x}\left(\left(g_i, g_j\right), y\right) = \exp\left(-\alpha_{xj}\left(y - z_{xj}\right)^2\right)$. When the variances of the Gaussians are similar, $\tilde{C}_{f_x}\left(\left(g_i, g_j\right), y\right) \approx \exp\left(-\alpha_{xi}(y - z_{xi})^2\right)$ over the unit interval. Using this approximation and, with parameters z_{xi} sorted to increase with i, (12) becomes

$$\tilde{S}_{I \times I}\left(\left(\tilde{C}, \tilde{C}_{f_x}\right), y\right) = \inf_{i \in I}\left\{\exp\left(-\alpha_{xi}(y_i - z_{xi})^2\right) : y = \sum_{i=1}^{J}(2n - 2i + 1)y_i/J^2\right\}. \quad (13)$$

This is solved by forcing $\alpha_{xi}(y_i - z_{xi})^2 = d$ for each i and using the constraint to solve for d as a function of $y \in [0, 1]$. Recall that the z_{xi} help describe the membership value that location x has the property G_i, and so the sorting and hence the weights $(2n - 2i + 1)$ in (13) are observation dependent.

Figure 2a depicts $g_i(f_{xi})(y)$ as a function of $y \in [0, 1]$ for five properties and one location x (referred to as location 3). We will also consider two other locations (locations 1 and 2) whose membership in each of the properties has similar shape but whose Gaussian means congregate respectively in the lower and upper regions of the unit interval, being (0.06, 0.15, 0.06, 0.06, 0.08) and (1.0, 0.8, 0.9, 0.8, 0.9). This positioning is reflected in the overall location suitability, Fig 2(b), computed in the context of the 5 properties using (13). The peak of the suitability membership function is, respectively, 0.07, 0.84, and 0.66 for the three locations. The simple average of the means of the membership of the location in each of the five properties is 0.08, 0.88, and 0.59 for locations 1, 2, and 3 respectively.

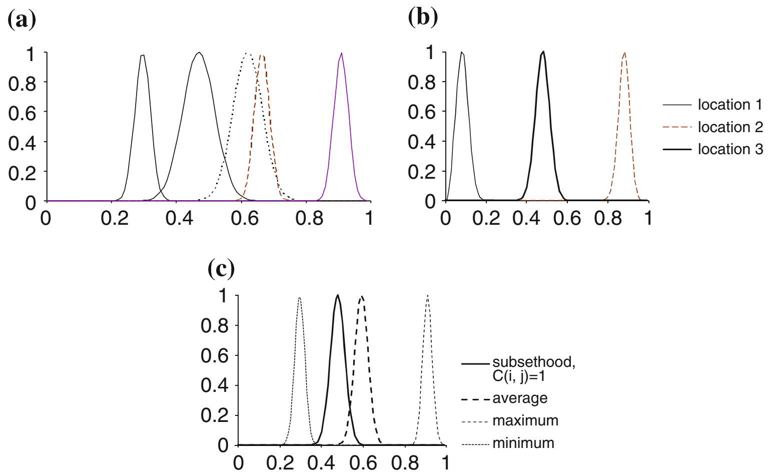


Fig. 2 (**a**) Membership of location 3 in five properties. See text for description of the membership of locations 1 and 2. (**b**) Membership of the three locations in the concept *suitability for crop* based on the five properties. (**c**) Suitability membership functions for location 3 computed using minimum, maximum, and the average of property membership functions at that location

Figure 2c compares suitability of location 3 computed using (13) with the minimum, maximum, and average of the location's membership in the five properties. Minimum and maximum were computed using the methods described in [22]. The fuzzy average is obtained using (10) if the concept is treated as a diagonal matrix, i.e., with nontrivial membership values $\tilde{C}\left(\left(g_i, g_j\right), 0\right) = 1, i \neq j$, and $\tilde{C}((g_i, g_i), 1) = 1$. The constraint in the expression (13) is then replaced by $y = \sum_{i=1,\ldots,J} y_i/_J$, i.e., the fuzzy average. The fuzzy average is more optimistic than the subsethood measure of suitability.

3.2 Land Use Change

An important application of land use maps is monitoring change. Standard approaches aggregate the area whose use has changed from one class to a second class in the period between mappings. The resulting value is entered into a land use change matrix. Not all land use class changes are equally significant [8]; two classes may be more similar to each other than they are to a third class. Reference [7] proposes using two type of similarity measure, based on overlap of properties on a domain and on distance between fuzzy membership functions. Overlap is a directional measure, related to the subsethood of properties on a domain. In our notation, the expression proposed by [7] for the overlap of two concepts defined respectively by property g_i, g_i' on each of J domains V_i is:

$$O(C,C') = \sqrt{\sum_{i=1}^{J} w_i S_{V_i}(g'_i, g_i)^2}. \tag{14}$$

Generally, in land use change modeling each class/concept under consideration is described in terms of a set of properties, one per domain. The classes can be assumed to involve the same set of domains; if the definition of a concept does not involve a domain used in defining some other concept, then a dummy property consisting of the entire domain can be used.

So suppose the modeling concerns properties $\tilde{g}_{mi}$, one per domain V_i for $i = 1, \ldots, J$, say, and M classes. Represent each class $\tilde{C}^{(m)}, m = 1, \ldots, M$ as a concept in a context $I^{(m)}$ in which all entries are unity, i.e. for $1 \le i,j \le J$, $\tilde{C}^{(m)}\left((\tilde{g}_{mi}, \tilde{g}_{mj}), 1\right) = 1$ and $\tilde{C}^{(m)}\left((\tilde{g}_{mi}, \tilde{g}_{mj}), y\right) = 0, y < 1$. In the context $I^{(n)}$, concept $\tilde{C}^{(n)}$ would also have all entries of unity. However, in the context $I^{(m)}$ the associations in $\tilde{C}^{(n)}$ have to be estimated from the overlap of properties, using (8). Since there is only one property per domain in each context, (8) becomes

$$\tilde{C}^{(n)}(\tilde{g}_{mi}, \tilde{g}_{mj}) = \tilde{S}_{V_i}(\tilde{g}_{mi}, \tilde{g}_{ni})\tilde{S}_{V_j}(\tilde{g}_{mj}, \tilde{g}_{nj})\tilde{C}^{(n)}(\tilde{g}_{ni}, \tilde{g}_{nj}) \tag{15}$$

This reduces to $\tilde{S}_{V_i}(\tilde{g}_{mi}, \tilde{g}_{ni})\tilde{S}_{V_j}(\tilde{g}_{mj}, \tilde{g}_{nj})$, which in general is a product of fuzzy numbers.

Now suppose $\tilde{C}^{(m)}_{x(t)}$ is the pseudo concept formed from observations $f_{xi}(t) \in M(V_i)$ made at location x at time t in the context of $I(\tilde{C}^m)$. As in (12), the membership of location x in class m at time t is

$$\mathbf{C}^{\mathbf{t}}_{\mathbf{x}}(m)(y) = \inf_{i,j}\left\{\left(\tilde{C}^{(m)}_{x(t)}\left((g_{mi}, g_{mj}), y_{ij}\right) : y_{ij} \in [0,1], y = \sum_{i,j=1}^{J} y_{ij}/J^2\right)\right\} \tag{16}$$

where $\tilde{C}^{(m)}_{x(t)}\left((g_{mi}, g_{mj}), y_{ij}\right)$ is as in (11).

A directional measure of change at location x between times $t = 1$ and $t = 2$ is $1 - S_P\left(\mathbf{C}^{\mathbf{1}}_{\mathbf{x}}, \mathbf{C}^{\mathbf{2}}_{\mathbf{x}}\right)$, where $\mathbf{C}^{\mathbf{t}}_{\mathbf{x}}$ are interpreted as fuzzy sets on the universe P consisting of the classes $m = 1, \ldots, M$ with membership functions $\mathbf{C}^{\mathbf{t}}_{\mathbf{x}}(m)$. Thus from (2), $1 - S_P\left(\mathbf{C}^{\mathbf{1}}_{\mathbf{x}}, \mathbf{C}^{\mathbf{2}}_{\mathbf{x}}\right) = 1 - \sum_{m=1}^{M} \min\left\{\mathbf{C}^{\mathbf{1}}_{\mathbf{x}}(m), \mathbf{C}^{\mathbf{2}}_{\mathbf{x}}(m)\right\} / \sum_{m=1}^{M} \mathbf{C}^{\mathbf{1}}_{\mathbf{x}}(m)$. This subsethood change measure is based on a fuzzy notion of whether the location is in a class, rather than crisp classification as in the standard approach. If the only non-zero membership at time 1 is $\mathbf{C}^{\mathbf{1}}_{\mathbf{x}}(m)$ but $\mathbf{C}^{\mathbf{2}}_{\mathbf{x}}(m) = 0$ then the change is 1 regardless of the actual value of $\mathbf{C}^{\mathbf{1}}_{\mathbf{x}}(m)$.

A related symmetric subsethood measure of change is

$$1 - \frac{1}{2}\left(S_P\left(\mathbf{C}^{\mathbf{1}}_{\mathbf{x}}, \mathbf{C}^{\mathbf{2}}_{\mathbf{x}}\right) + S_P\left(\mathbf{C}^{\mathbf{2}}_{\mathbf{x}}, \mathbf{C}^{\mathbf{1}}_{\mathbf{x}}\right)\right). \tag{17}$$

Aggregation of the local change measure gives a directed subsethood measure of change over the region of interest X, namely $|X| - \sum_{x \in X} |x| S_P\left(\mathbf{C}_\mathbf{x}^\mathbf{1}, \mathbf{C}_\mathbf{x}^\mathbf{2}\right)$ where $|X|$ is the total area of X and $|x|$ denotes the subarea of X to which the observations at x refer. A symmetric subsethood measures of change over a region X is

$$|X| - \frac{1}{2} \sum_{x \in X} |x| \left(S_P\left(\mathbf{C}_\mathbf{x}^\mathbf{1}, \mathbf{C}_\mathbf{x}^\mathbf{2}\right) + S_P\left(\mathbf{C}_\mathbf{x}^\mathbf{2}, \mathbf{C}_\mathbf{x}^\mathbf{1}\right) \right) \tag{18}$$

Typically x is a pixel location in an image and $|x| = 1$. If class membership is crisp, then this measure is the count of pixels that have changed class.

When Properties are Fuzzy Sets. On properties which are fuzzy sets, as we henceforth assume, subsethood is a number. Computing concept similarity as subsethood (3) using (15) with $C^{(m)}\left(g_{mi}, g_{mj}\right) = 1 = C^{(n)}\left(g_{ni}, g_{nj}\right)$, we see that $S_{I^{(n)} \times I^{(n)}}\left(C^{(n)}, C^{(m)}\right)$ is

$$\sum_{i,j=1,\ldots,J} S_{V_i}(g_{mi}, g_{ni}) S_{V_j}\left(g_{mj}, g_{nj}\right) / J^2 = \left(\sum_{i=1,\ldots,J} S_{V_i}(g_{mi}, g_{ni}) / J \right)^2. \tag{19}$$

This can be written in terms of concept overlap $O(C', C)$ as defined in (13):

$$S_{I^{(n)} \times I^{(n)}}\left(C^{(n)}, C^{(m)}\right) = \sum_{i=1,\ldots,J} S_{V_i}(g_{mi}, g_{ni})^2 \frac{\sum_{j=1,\ldots,J} S_{V_j}\left(g_{mj}, g_{nj}\right)}{J^2 S_{V_i}(g_{mi}, g_{ni})} \tag{20}$$

$$\triangleq O\left(C^{(n)}, C^{(m)}\right)^2$$

with $w_i = \frac{\sum_{j=1,\ldots,J} S_{V_j}\left(g_{mj}, g_{nj}\right)}{J^2 S_{V_i}(g_{mi}, g_{ni})}$ when $S_{V_i}(g_{mi}, g_{ni}) \neq 0$.

Thus, subsethood as concept overlap gives relatively greater weight to the contribution of domains on which the respective concept properties have small overlaps. Put another way, a small property overlap diminishes the concept overlap more than in the case when concept overlap is computed using uniform weights.

Land Use Change Example when Properties and Observations are Fuzzy Sets. We now present a simple example of land use change measurement that compares the subsethood change measure with the standard measure when observations are interval valued. Three land use classes are considered: open grassland, forest, and urban development. There are three real-valued domains, respectively, percentage of canopies, of grassland, and of impervious (man-made) surfaces. The entities observed are locations (pixels) in region X. Full membership of each of the classes under investigation requires that the location satisfy a set of properties which are the crisp intervals listed in Table 2. When a class definition does not involve a domain we use a dummy property over the full range of that domain. Note from the table that even a crisp observation could be classified as both class 1, open grassland, and class 3, urban development. The classes also do not cover all possible observations, so some locations may not belong to any of the studied classes at one or both of the observed times.

In such a scenario, under standard land use classification a location x would be assigned to a class m if and only if on each domain V_i the observation interval overlaps the class property interval. Because the studied classes are not all mutually exclusive and do not cover all possible observations, the standard absolute measure of change can take one of three values: it is 1 when the location changes class between observations or changes from satisfying two of the studied classes to satisfying none; it is 0.5 if it changes to or from one of the studied classes or between satisfying two to just one of the studied classes; and it is 0 if it does not change.

By (9), $\tilde{C}^{(m)}_{x(t)}\big((g_{mi}, g_{mj}), 1\big)$ is the minimum of two fuzzy sets. Inspection of (11) shows if properties and observations are crisp subsets, $\tilde{C}^{(m)}_{x(t)}\big((g_{mi}, g_{mj}), 1\big) = 1$ when the observation overlaps the class m properties on the ith and jth domains and is otherwise zero.

It then follows from (16) that $\mathbf{C}^{\mathbf{t}}_{\mathbf{x}}(m)$ is the number $i^t_x(m)^2/J^2$ for $i^t_x(m)$ the count of domains on which the observation at location x at time t overlaps the property of class m on that domain. Inspecting (17) in conjunction with Table 2 we see that the subsethood measure can take a larger number of values (in fact, it takes ten values). When a location satisfies none of the studied classes at either time period then inspection shows that the location must satisfy two properties for each of the classes and so change must be zero.

Figure 4 is generated with the singleton observations shown in Figs. 3, and 5 with interval observations of width 10% of the width of X and with centroids as depicted in Fig 3. Comparing Fig. 4a with Fig. 4c, Fig. 4b with Fig. 4d, and Fig. 4e with Fig. 4f shows that the subsethood measure in this situation gives similar results to the standard measure. Figure 4g gives the difference between the measures, with darker areas indicating where the subsethood measure (17) has recorded greater change. Aggregated over all pixels, change was detected in 38% of locations with the conventional method, in 47% with the subsethood method, but with aggregated measure of change based on (17) was 13%.

Allowing vagueness in the observations reduces the measured change, as can be seen by looking at the difference between the change measure with vagueness and without, in Figs. 4 and 5. Imprecision effectively acts to broaden the property boundaries by the width of the interval. The conventional change measure reports less change although the difference is reduced when imprecision is modeled.

Table 2 Concepts and their defining properties on domains V_1, V_2 and V_3

	V_1 % grassland	V_2 % canopies	V_3 % *impervious*
$C(1)$ open	[60, 100]	[0, 40]	
$C(2)$ forest	[0, 40]	[60, 100]	[0, 40]
$C(3)$ development			[60, 100]

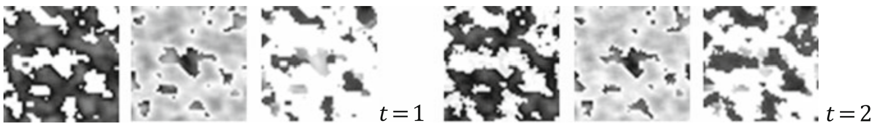


Fig. 3 Observations in the square region X at times $t = 1, 2$ of percentage grassland V_1, canopy V_2 and impervious surface V_3

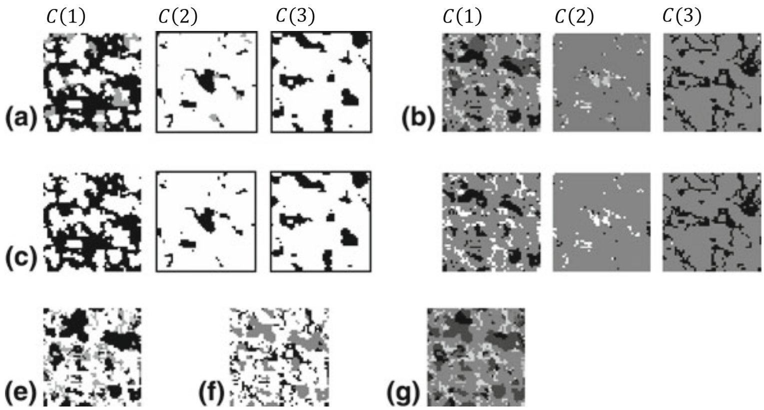


Fig. 4 Land use change over a square region X. (**a**) Count of properties satisfied by each pixel for the 3 classes at time $t = 1$. (**b**) Change in counts by time $t = 2$ (*grey* = no change). (**c**) Membership of pixels in each of the three classes at time 1 (Eq. 16). (**d**) Change in membership for each class by time 2. (**e**) Subsethood measure (17) of overall change. (**f**) Standard absolute change measure. (**g**) Difference between measures (*white* implies greater change registered with standard measure)

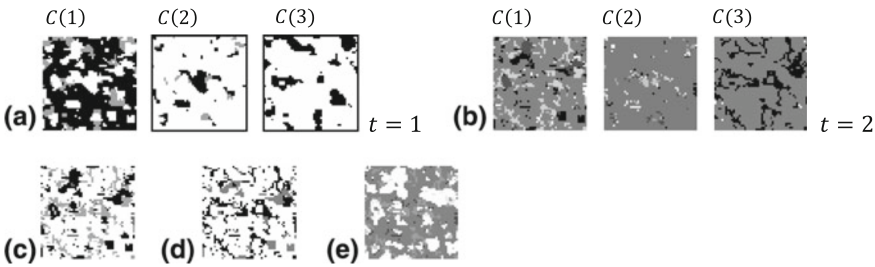


Fig. 5 Land use change over a square region X when centroids of interval observations on each domain are as in Fig 3 and interval length is 10% of breadth of X. (**a**) Membership values (16) for pixels being in each of the three classes at time 1. (**b**) Change in membership values by time $t = 2$ (*grey* = no change). (**c**) Subsethood measure of overall change. (**d**) Standard absolute change measure. (**e**) Difference between subsethood measure (17) of change when observations are fuzzy and when they are crisp (*white* = greater change registered with crisp observations, *grey* = no difference)

4 Discussion

This chapter extended conceptual space theory to incorporate type-2 fuzzy set structures. The need for such an extension is motivated by real world applications, in which observations about a phenomenon are often vague and imprecise. The key operation is subsethood, which is used to aggregate and to compare. Thus, practical implementation of conceptual spaces benefits from recent algorithms to efficiently compute subsethood of interval type-2 fuzzy sets [16].

Type-2 fuzzy sets played a variety of roles in our simulations of land use suitability and land use change assessments. Neglect of imprecision was shown to affect results by the order of 10%. The additional accuracy may warrant the workload required in type-2 modeling for applications such as these involving public impact and sensitivity, or where there is financial interest, as in share market prediction.

A property or a fuzzy property on a domain V induces a type-2 fuzzy set on the set of fuzzy observations $M(V)$, i.e., on the fuzzy sets on V. Vagueness captured in these type-2 fuzzy sets can be attributed to the vagueness inherent in the property membership combined with imprecision of an observation.

Explicit modeling of context is an important aspect of our approach to conceptual spaces. The properties in a context are generally defined on different domains, such as height and weight, and are thus open to the objection that membership functions on different domains are incommensurate (so that, for example, a minimum of two membership functions cannot meaningfully be defined) [23, 24]. Our modeling attributes membership to the object. The fact that we still attribute more heaviness than tallness to Jack is supported by evidence such as that found in classification tasks [25], where people are prepared to make comparative judgments on incommensurate domains such as “this object is more red than it is square”.

A perennial issue which we could not adequately address was how best to compare results presented as type-1 and type-2 fuzzy sets on the unit interval, for example, membership of a location in concepts *suitable for rice* or *suitable for corn*, or membership of two locations in a suitability concept.

Finally, fuzzifying conceptual spaces introduces additional modeling decisions as well as computational overheads. When should we use a (fuzzy) conjunction of properties *tall and heavy* defined on a product of length and weight using a procedure such as that in [21], rather than a concept *tall and heavy* referenced to a defining population (for example, Japanese men, 10-year old girls, ships)? More work is needed on real applications to provide guidance for such decision making.

References

1. Gärdenfors, P.: Conceptual Spaces: the Geometry of Thought. MIT Press, Cambridge (2000)
2. Freund, M.: On the notion of concept. Art. Int. **172**(4–5), 570–590 (2008)
3. Zadeh, L.A.: Fuzzy logic = computing with words. IEEE Trans. Fuzzy Syst. **4**(2), 103–111 (1996)

4. Zadeh, L.A.: Is there a need for fuzzy logic? Inf. Sci. **178**, 2751–2779 (2008)
5. Rickard, J.: A concept geometry for conceptual spaces. J. Fuzzy Optim. Decis. Mak. **5**, 311–329 (2006)
6. Rickard, J., Aisbett, J., Gibbon, G.: Reformulation of the theory of conceptual spaces. Inf. Sci. **177**, 4539–4565 (2007)
7. Ahlqvist, O.: A parameterized representation of uncertain conceptual spaces. Trans. GIS **8**(4), 493–514 (2004)
8. Ahlqvist, O.: Extending post classification change detection using semantic similarity metrics to overcome class heterogeneity: a study of 1992 and 2001 national land cover database changes. Remote Sens. Environ. **112**(3), 1226–1241 (2008)
9. Fisher, P., Cheng, T., Wood, J.: Higher order vagueness in geographical information. Geoinformatica **11**, 311–330 (2007)
10. Malczewski, J.: GIS-based land-use suitability analysis: a critical overview. Progress Plan. **62**, 3–65 (2004)
11. FAO: land use classification for agri-environmental statistics/indicators. Working paper No.13, Joint ECE/Eurostat work session on methodological issues of environment statistics, Statistics Division FAO (1999)
12. Triantafilis, J., Ward, W.T., Mcbratney, A.B.: Land suitability assessment in the lower Namoi valley of Australia. Aust. J. Soil Res. **39**, 273–290 (2001)
13. Rickard, J., Aisbett, J., Gibbon G.: Type-2 fuzzy conceptual spaces. In: IEEE International Conference on Fuzzy Systems, 1–8 (2010)
14. Aisbett, J., Rickard, J.T., Morgenthaler, D.: Multivariate modeling and type-2 fuzzy sets. Fuzzy Sets Syst. **163**, 78–95 (2011)
15. Halmos, P.: Measure Theory. Springer, New York (1974)
16. Wu, D., Mendel, J.M.: Efficient algorithms for computing a class of subsethood and similarity measures for interval type-2 fuzzy sets. In: IEEE International Conference on Fuzzy Systems, 1–7 (2010)
17. Schyns, P.G., Goldstone, R. L. Thibaut, J-P.: The development of features in object concepts. Behav. Brain Sci. **21**, 1–54 (1998)
18. FAO: agro-ecological land resources assessment for agricultural development planning—a case study of Kenya. Technical report. Natural Resources Management and Environment Department, FAO (1991)
19. Soil Survey Staff NRCS: Natural Resources Conservation Service, United States Department of Agriculture. Web Soil Survey. http://websoilsurvey.nrcs.usda.gov/ (2008). Accessed 20 Jan 2008
20. Ochola, W.O., Kerkides, P.: An integrated indicator-based spatial decision support system for land quality assessment in Kenya. Comput. Electron. Ag **45**, 3–26 (2004)
21. Mendel, J.M.: Computing with words and its relationships with fuzzistics. Inf. Sci. **177**, 985–1006 (2007)
22. Aisbett, J., Rickard, J.T., Morgenthaler, D.: Intersection and union of type-n fuzzy sets. IEEE International Conference on Fuzzy Systems, 1–8 (2010)
23. Finch, P.D.: Characteristics of interest and fuzzy subsets. Inf. Sci. **24**, 121–134 (1981)
24. Lee, J.W.T.: Ordinal decomposability and fuzzy connectives. Fuzzy Sets Syst. **136**, 237–249 (2003)
25. Nosofsky, R.: Attention, similarity and the identification-categorization relationship. J. Exp. Psych. Gen. **115**(1), 39–57 (1986)

Part II
Type-2 Fuzzy Set Membership Function Generation

Modeling Complex Concepts with Type-2 Fuzzy Sets: The Case of User Satisfaction of Online Services

Masoomeh Moharrer, Hooman Tahayori and Alireza Sadeghian

Abstract Specific characteristics of human perception, like context-dependency, imprecision, and diversity, demand capable formal frameworks for modeling the human mind. This chapter discusses a two-phase method for deriving type-2 fuzzy sets that model human perceptions of the linguistic terms used in describing online satisfaction. In the first phase, we describe the identification of the determinants of user satisfaction of online tourism services. We will demonstrate how the decomposition of the satisfaction concept into a set of simpler, albeit covering subconcepts, would be used to calculate a type-1 fuzzy set model of an individual's perception. In the second phase, type-2 fuzzy sets modeling online user satisfaction are derived based on the obtained type-1 fuzzy sets. The construction of type-2 fuzzy sets is based on the exploitation of the fuzzy approach to represent uncertainty and by stacking the α-planes calculated at different levels of confidence around the estimated mean values of the type-1 fuzzy set.

1 Introduction

In daily communications, there are linguistic terms with immediate and trivial interpretations. Generally, for such terms, like *tall man*, *high temperature*, *low speed,* with a well-defined context, people may simply come up with a single number or an interval as its interpretation. On the other hand, there are complex

M. Moharrer (✉) · H. Tahayori · A. Sadeghian
Department of Computer Science, Ryerson University, Toronto, ON M5B 2K3, Canada
e-mail: mmoharrer@ryerson.ca, masoomeh.moharrer@polimi.it

H. Tahayori
e-mail: htahayor@scs.ryerson.ca

A. Sadeghian
e-mail: asadeghi@scs.ryerson.ca

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_9,

concepts like *high satisfaction*, *low performance, good creditworthiness,* whose interpretations are not straightforward. Their complexity arises from the fact that such concepts do not rely on a single determinant, but rather several factors with different weights take part in forming their *meaning*. People are sometimes asked to provide exact interpretations of such concepts, i.e., numbers or interval estimations. In order to determine a number or the endpoints of an interval, different machineries in different parts of the brain should be invoked, but the determination is more associated with emotions [1]. Expecting an individual to be cognizant of all the factors affecting such concepts is very optimistic.

In this chapter, we elaborate on modeling user satisfaction of online services with type-2 fuzzy sets. To determine user satisfaction, users' preferences are usually measured using linguistically labeled rating scales. The use of linguistically labeled rating scales is based on the fact that people naturally demonstrate their perceptions and cognitions with words, expressions, and sentences from natural languages [1] and despite their diversity of perception, they communicate smoothly with each other. However, human perception is not precise and varies over time; one's intuitions and cognition of a concept highly depends on the context, domain knowledge, individual senses, etc. Hence, different individuals may perceive the same concept differently [2]. That is why for the same concept, although individuals use the same linguistic term, their perceptions and hence interpretations are not necessarily similar [3, 4]. Nonetheless, people communicate using linguistic terms, which suggests that the meanings of these terms among individuals should not vary greatly [5].

Zadeh in 1975 [6] introduced the concept of linguistic variables whose key aspect is the use of fuzzy sets to represent the *meaning* of words or terms from natural languages. Deriving fuzzy set models of words usually relies on the data gathered from different subjects. Although the theory of Type-1 Fuzzy Sets (T1 FS) provides a natural framework for modeling intrauncertainty, i.e., the uncertainty of a subject on a concept, (*i*) except from fuzzy set experts, generally we cannot expect people to explicitly express their intuitions using fuzzy sets and (*ii*) using T1 FS to model a word requires all T1 FSs gathered from different individuals to be *reduced* to a single T1 FS which conceals the diversity and uncertainty that exist among a group of people.

Type-2 Fuzzy Sets (T2 FS), fuzzy sets with fuzzy membership functions, are proposed [6] as an extension of type-1 fuzzy sets. T2 FS are said to provide a correct scientific uncertainty model for words and linguistic terms from natural languages [7]. By providing more degrees of freedom than type-1 fuzzy sets, type-2 fuzzy sets have more potential for handling uncertainties [3]. However, the most challenging part of modeling words with T2 FSs is the calculation of their membership functions. This is true to a lesser extent for Interval T2 FSs and Shadowed Fuzzy Sets (SFS) [8, 9] as specific cases of T2 FS. In SFSs, the fuzziness associated with membership functions of T2 FSs are redistributed in shadowed sets [10]. SFSs provide more freedom degrees for handling uncertainties in comparison with IT2 FSs, but with lower computational complexity than general T2 FSs.

To model the linguistic terms uses for describing user satisfaction, we will primarily demonstrate decomposition of the concept into some covering subconcepts. We will calculate the unique effect of each subconcept on the overall user satisfaction. Then we will discuss the indirect derivation of a T1 FS, modeling each individual's satisfaction for whom we do not assume any fuzzy set theory literacy. To capture the diversity of the T1 FSs derived from different subjects, we will detail the construction of a T2 FS that models the concept. The methodology proposed in this chapter which is an extension and enhancement to [11] can be applied to model other *decomposable* complex concepts.

2 Review of Type-2 Fuzzy Sets

Type-2 Fuzzy Set $\tilde{A}$ defined on the universe of discourse X is represented as,

$$\tilde{A} = \left\{ \left(x,\, \mu_{\tilde{A}}(x) \right) \middle| \, x \in X,\, \mu_{\tilde{A}}(x) = \{(u, f_x(u)),\, u \in J_x \subseteq [0,\, 1], f_x(u) \in [0,\, 1]\} \right\} \quad (1)$$

We refer to $\mu_{\tilde{A}}(x)$ as the *fuzzy membership value* of x in $\tilde{A}$—it is also known as *secondary membership function* or *secondary set*. Moreover, in (1), x is called the *primary variable*, J_x represents the *primary membership values* of x and $f_x(u)$is named *secondary grade*. When $\forall u \in J_x, f_x(u) = 1$, the type-2 fuzzy sets would be reduced to interval type-2 fuzzy set.

Given a type-2 fuzzy set $\tilde{A}$, its Footprint of Uncertainty (FOU) is defined as,

$$\mathrm{FOU}(\tilde{A}) = \bigcup_{x \in X} J_x = \bigcup_{x \in X} [\underline{u}(x),\ \overline{u}(x)],\ \underline{u}(x) = \operatorname*{Inf}_{u \in J_x}(u),\ \overline{u}(x) = \operatorname*{Sup}_{u \in J_x}(u) \quad (2)$$

FOU of a type-2 fuzzy set $\tilde{A}$, can be fully characterized by two type-1 fuzzy sets, named *Upper Membership Function* (UMF) and *Lower membership Function* (LMF),

$$\mathrm{UMF}(\tilde{A}) = \overline{\mathrm{FOU}}\left(\tilde{A}\right) = \{(\bar{u}(x),\, x),\ x \in X\} \quad (3)$$

$$\mathrm{LMF}(\tilde{A}) = \underline{\mathrm{FOU}}\left(\tilde{A}\right) = \{(\underline{u}(x),\, x),\ x \in X\} \quad (4)$$

In other words, $\mathrm{FOU}(\tilde{A})$ is bounded by $\mathrm{UMF}(\tilde{A})$ and $\mathrm{LMF}(\tilde{A})$. Note that an interval type-2 fuzzy set is fully characterized by its FOU. More detailed discussion on interval T2 FSs and general T2 FSs and their operations can be found in [12–18].

3 User Satisfaction in Online Tourism

Alluding to the psychologists' efforts in satisfaction-related studies, there is a general agreement that satisfaction in a situation is a multidimensional attitude which is the resultant of one's perceptions, feelings, or attitudes toward a variety of factors that affect the situation [19]. The focus of this chapter is on satisfaction of users of online tourism services in the prestage of a travel, i.e., before they actually start their travel. In the prestage, potential tourists search for different tourist services, airline tickets, hotels, etc., and make purchases in order to schedule their trip.

Unlike durable goods, intangible tourism services cannot be physically displayed or inspected at the point of sale before the trip. Therefore, tourism products are thoroughly dependent on different aspects of their representations and offerings by the tourism firms. Hence, proper and efficient utilization and management of information is essential for tourism organizations to satisfy their customers. Websites, hence, are the main channel which link tourism organizations with potential tourists.

In order to evaluate user satisfaction of online services, we designed a questionnaire inspired by the models proposed by Szymansky and Hise [20], and Servequal [21]. The main idea behind the design was to decompose the concept of online satisfaction into its covering but simpler, more focused subconcepts. Consequently, each question in the questionnaire was designed to shed light on the concept of user satisfaction of online services from a unique aspect.

The questionnaire was comprehensively reviewed by different experts in the field. Notably, the answers to the questions in the questionnaire were designed to be in five-point Likert scale on the bipolar adjective pair *bad*-*good* with qualifiers *much worse than*, *worse than*, *the same*, *better than,* and *much better than*. The qualifiers are, respectively, assigned the values of 1–5. The questionnaire was finalized after conducting a pilot test.

To gather data, a survey was conducted in front of the check-in desk in Beauvais airport, Paris. The airport serves airlines like Ryanair, which sell the majority of their tickets online. The respondents were the people who had experienced e-tourism at least once before. A total of 115 questionnaires were collected. Cases with standard residuals above three were identified as outliers. After omitting outliers and missing valued questionnaires, the number of valid responses reduced to 99.

In order to determine the factors affecting the user's overall satisfaction, a factor analysis was performed on independent variables (the questions). Factor analysis attempts to identify underlying variables or factors that explain the pattern of correlations within a set of observed variables. Factor analysis is often used in data reduction to identify a small number of factors that explain most of the variance observed in a much larger number of manifest variables. Factor analysis can also be used to generate hypotheses regarding causal mechanisms or to screen variables for subsequent analysis (for example, to identify co-linearity prior to performing a linear regression analysis).

The results confirmed that the five-factor measurement model introduced by Szymansky and Hise [20] reflects the underlying respondents' mental model in e-tourism. The five factors observed were site design, convenience, financial security, product information, and product offering. Moreover, the items of the questionnaire belonging to each factor were identified. Importantly, similar to [20] and

Table 1 Factors, Items, factor loading of items, and unique effects of factors in Satisfaction of online Tourism Services

Factors	Items	Factor loading	Factor weight
Factor 1:	Attractive website	0.864	0.377
Site design	Friendliness ease of use	0.682	
	Uncluttered screens	0.645	
Factor 2:	Purchase any time	0.797	0.375
Convenience	Purchase anywhere	0.737	
	Time efficiency	0.711	
Factor 3:	Formal privacy	0.897	0.119
Financial security	Safe feeling in transactions	0.807	
Factor 4:	Quantity of information	0.832	0.144
Product information	Quality of information	0.740	
Factor 5:	Number of tourism services	0.753	0.163
Product offering	Variety of tourism services	0.789	

[22] a regression analysis was performed to estimate the unique effect of each factor. The results of the regression and factor analysis are shown in Table 1.

To ensure that we could use factor analysis on gathered data, Kaiser–Meyer–Olkin Measure of Sampling Adequacy (KMO measurement) and Bartlett's test analysis were conducted. Results showed that KMO is greater than 0.5 (=0.79), which signifies the number of samples, is big enough for using factor analysis [23]. Moreover, the significance of Bartlett's test is 0, which signifies factor analysis was applicable [24].

To test the reliability and internal consistency of each factor, Cronbach's alpha scores [25]—for factors with more than two items—and correlation coefficient—for factors with two items—were calculated. The Cronbach's alpha scores and the correlation coefficient ranged from 0.53 to 0.82 for the five factors. Since 0.5 is the minimum value for accepting the reliability test [26], the results of factor analysis are reliable. More details on the method and results may be found in [27–30].

4 Modeling Individual's Satisfaction with T1 FS

To model a subject's overall satisfaction of online tourism services using T1 FSs, we rely on the decomposition of the concept, the calculated weight of each constituting item, and the answers to each of the items. As discussed, we have identified the set of all factors and their items comprising the domain of satisfaction in online tourism and we have calculated the contributing effect of each factor—Table 1. Moreover, we have also calculated the weight of each item/question of the questionnaire that is loaded into a specific factor—Table 1.

Let R_{ij} be the response of the individual i to the question j with the contributing weight W_j. An individual's overall satisfaction would be calculated through aggregating the responses of the subject to each question with respect to the weight of the item for which the question is asked [19],

$$S_i = \sum_j W_j R_{ij} \tag{5}$$

Given M questions—whose answers are designed on L point-scale—that are loaded into N factors, we expand the Eq. (5) as follows:

$$S_i = \sum_{j=1}^{M} \left(F_{kj} \times Q_{kj}\right) \times R_{ij},\ k = 1, \ldots, N \tag{6}$$

where, Q_{kj} represents the loading factor of question j into the factor k, and F_{kj} denotes the weight of the factor k that the item j is loaded into. Reordering (6) to group the items with equal response values, we have,

$$S_i = \sum_{\substack{j \in J_1 \\ \forall j \in J_1,\, R_{ij} = R_1}} \left(F_{kj} \times Q_{kj}\right) \times R_1 + \ldots + \sum_{\substack{j \in J_L \\ \forall j \in J_L,\, R_{ij} = R_L}} \left(F_{kj} \times Q_{kj}\right) \times R_L\ , \quad k = 1, \ldots, N \tag{7}$$

Since $R_1, \ldots, R_L$ are constant, then

$$S_i = R_1 \times \sum_{\substack{j \in J_1 \\ \forall j \in J_1,\, R_{ij} = R_1}} \left(F_{kj} \times Q_{kj}\right) + \ldots + R_L \times \sum_{\substack{j \in J_L \\ \forall j \in J_L,\, R_{ij} = R_L}} \left(F_{kj} \times Q_{kj}\right)\ , \quad k = 1, \ldots, N \tag{8}$$

In (8), $\sum_{\substack{j \in J_n \\ \forall j \in J_n,\, R_{ij} = R_n}} \left(F_{kj} \times Q_{kj}\right)$ denotes the cumulative belief degree of the individual i for R_n, $n = 1, \ldots, L$ to be his/her satisfaction degree. Dividing the (8) by the maximum possible belief degree, we have,

$$S'_i = \frac{\left(R_1 \times \sum_{\substack{j \in J_1 \\ \forall j \in J_1,\, R_{ij} = R_1}} \left(F_{kj} \times Q_{kj}\right) + \ldots + R_L \times \sum_{\substack{j \in J_L \\ \forall j \in J_L,\, R_{ij} = R_L}} \left(F_{kj} \times Q_{kj}\right)\right)}{\sum_{j=1}^{M} \left(F_{kj} \times Q_{kj}\right)},\quad k = 1, \ldots, N \tag{9}$$

Considering the responses of a subject to all the items that were identified as the building blocks of the satisfaction, Eq. (9) generates the numerical value of the respondent overall satisfaction of tourism online services.

In the survey, based on their experience with tourism online services, each respondent was also asked to provide the evaluation of his/her overall sense of satisfaction in online tourism by choosing from a list of linguistic labels. To capture the respondent's overall judgment, we used the following five-point bipolar scale of *Fully Dissatisfied—Fully Satisfied*, where the points were, respectively, qualified by the linguistic terms *Very poor*, *Poor*, Fair, *Good*, and *Very good*.

We argue that the crisp value calculated by (9) is not easily interpretable. More importantly, the correlation of the crisp number with the label that the subject has chosen as his/her overall rate of satisfaction is not straightforward. Instead, elaboration on (9) reveals that it can be considered as the centroid of the following fuzzy set,

$$F_{S'_i} = \frac{\sum_{\substack{j\in J_1\\ \forall j\in J_1, R_{ij}=R_1}} \left(F_{kj}\times Q_{kj}\right) \Big/ \sum_{j=1}^{M} F_{kj}\times Q_{kj}}{R_1} + \ldots + \frac{\sum_{\substack{j\in J_L\\ \forall j\in J_L, R_{ij}=R_L}} \left(F_{kj}\times Q_{kj}\right) \Big/ \sum_{j=1}^{M} F_{kj}\times Q_{kj}}{R_L}, \quad k = 1,\ldots,N \tag{10}$$

where '+' denotes union operation and horizontal division line shows that its numerator is the membership value of the denominator in the fuzzy set $F_{S'_i}$. We observe that by Eq. (10) we may represent a subject's overall satisfaction as a T1 FS. In effect, we argue that, for a completed questionnaire, the calculated T1 FS by (10) would represent the *meaning* of the label chosen by the respondent from the following scale as his/her overall satisfaction.

	1	2	3	4	5	
Fully Dissatisfied	*Very-poor*	*Poor*	*Fair*	*Good*	*Very-good*	*Fully Satisfied*

Figure 1a represents all fuzzy sets calculated from the questionnaires of respondents who had chosen the linguistic label Fair as their overall satisfaction rate in online tourism. Figure 1b, however, shows a normalized fuzzy set derived by calculating the point-wise average of all Fair fuzzy sets shown in Fig. 1a. Figure 1c–f, respectively, show fuzzy sets associated with *Good* and *Very good* overall satisfaction rates. The number of fuzzy sets associated with *Very poor* and *Poor* were not sufficient for processing.

A notable characteristic of the T1 FSs is observed by calculating the average of all T1 FSs modeling the same overall satisfaction—Fig. 1b. The resulting FSs are all unimodal convex fuzzy sets with their apex on the numeric value that was

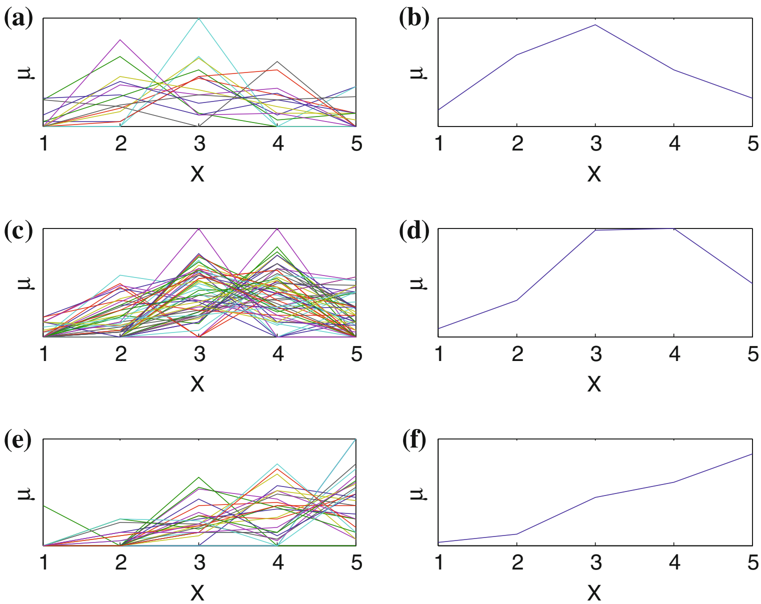


Fig. 1 Overall satisfaction rate: (**a**, **c**, **e**) All T1 FS models of individuals whose overall satisfaction has been, respectively, Fair, *Good*, and *Very good*, (**b**, **d**, **f**) Point-wise average of all T1 FSs of Fair, *Good*, and *Very good* overall satisfaction

associated to the label they represent. The apexes of FSs shown in Fig. 1b, d, and f that are respectively calculated from all T1 FSs modeling Fair, *Good,* and *Very Good* are on 3, 4, and 5.

5 T2 FS Model of User Satisfaction

There was no preassumption on participants in the survey, except having a prior experience in online tourism services. The T1 FS that is calculated based on a valid questionnaire, models the intra-uncertainty of its respondent about his/her overall satisfaction. In effect, the T1 FS is the representative of the respondent's perception of the meaning of the linguistic term he/she has chosen as the overall rate of satisfaction. Figures 1a, c, and e show that the perceptions of different people of the same concept are not essentially equal. Using T1 FS to model a word requires all T1 FSs gathered from different individuals to be *reduced* to a single T1 FS which conceals the diversity and uncertainty that exist among a group of people [31, 32]. In the following, we will demonstrate the exploitation of IT2 FS and T2 FSs to handle the uncertainties that obviously exist among a group of individuals on a single concept. In this section, we will only show the figures related to the Fair satisfaction.

5.1 Union of All T1 FSs

As discussed in Sect. 2, an interval T2 FS (IT2 FS) would be represented by its embedded T1 FSs. Let ${F_{S_i'}}^{l}$ denote the fuzzy satisfaction model of individual i who has chosen linguistic term l as his/her overall satisfaction rate. With respect to the Mendel–John IT2 FS representation theorem [13], in order to derive an IT2 FS $\widetilde{F^l}$ based on the set of T1 FSs we have,

$$\mathrm{UMF}_{\widetilde{F^l}}(x) = \operatorname*{Max}_i\left({F_{S_i'}}^{l}(x)\right), \quad i = 1, \ldots, N_l \tag{11}$$

$$\mathrm{LMF}_{\widetilde{F^l}}(x) = \operatorname*{Min}_i\left({F_{S_i'}}^{l}(x)\right), \quad i = 1, \ldots, N_l \tag{12}$$

The result of applying this method on the T1 FSs of Fair satisfaction rate is shown in Fig. 2. It can be seen that the resulting IT2 FS is almost filled-in, i.e., $\mathrm{LMF}_{\widetilde{F^{\mathrm{Fair}}}}(x) \approx 0$ for most part of the U, which does not exhibit a reasonable IT2 FS [5].

5.2 Mean Confidence Interval

The point-wise averages of the calculated T1 FSs for each of the overall satisfaction rate labels, shown in Fig. 1b, d, and f, are important from at least two aspects. On one hand, from an individual basis, their convexity, apex position, and unimodality and on the other hand, as the frame of cognition, their proper ordering, and distinguishability, make them exhibit considerable signs of interpretability [33]. However, since they are point estimations calculated from sample data, we cannot expect the calculated values to coincide with what they are intended to estimate. Hence, we will calculate the confidence interval $M_{1-\alpha}$ around the mean estimation, such that with the confidence level of $1-\alpha$, it contains the *real* mean value. The interval estimation of mean at the confidence level of $1-\alpha$, for $x \in X$ is,

$$m_{F^{\mathrm{l}}}(x) - t_{\alpha/2}.\frac{s_{F^{\mathrm{l}}}(x)}{\sqrt{N_l}} \le \mu_{F^{\mathrm{l}}}(x) \le m_{F^{\mathrm{l}}}(x) + t_{\alpha/2}.\frac{s_{F^{\mathrm{l}}}(x)}{\sqrt{N_l}} \tag{13}$$

where $m_{F^l}(x)$, $s_{F^l}(x)$ and N_l ,respectively, represent the mean, standard deviation, and the number of samples gathered for the overall satisfaction rate l. In (13), $\mu_{F^l}(x)$ denotes the mean value of the population—all possible valid T1 FSs representing the meaning of the label l—at x, which is actually unknown.

The IT2 FS of Fair satisfaction rate shown in Fig. 3 is derived by calculating the estimation of mean bounds for all $x \in X$ with $1-\alpha = 0.99$, i.e.,

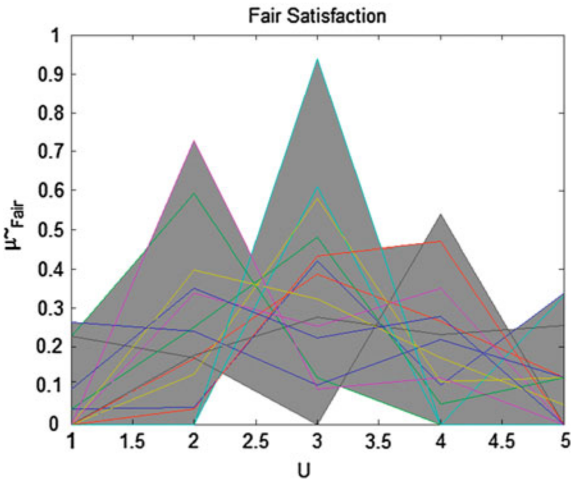


Fig. 2 IT2 FS of 'Fair' satisfaction rate calculated as the union of all individual's perception of 'Fair' satisfaction rate

$$UMF_{F^{Fair}}(x) = m_{F^{Fair}}(x) + t_{\alpha/2}.\frac{s_{F^{Fair}}(x)}{\sqrt{N_{Fair}}} \tag{14}$$

$$LMF_{F^{Fair}}(x) = \mathrm{Max}\left\{0, \boldsymbol{m}_{\boldsymbol{F}^{Fair}}(x) - \mathbf{t}_{\alpha/2}.\frac{s_{F^{Fair}}(x)}{\sqrt{N_{Fair}}}\right\} \tag{15}$$

The resulting IT2 FS demonstrates to be more reasonable model of *Fair* Satisfaction than the IT2 FS shown in Fig. 2.

Based on the fuzzy approach to the representation of uncertainty in measurement [34, 35], a family of nested confidence intervals of point estimation—that are calculated with various confidence levels—would be used to construct the membership function of a fuzzy set that represents an uncertainty in the gathered data

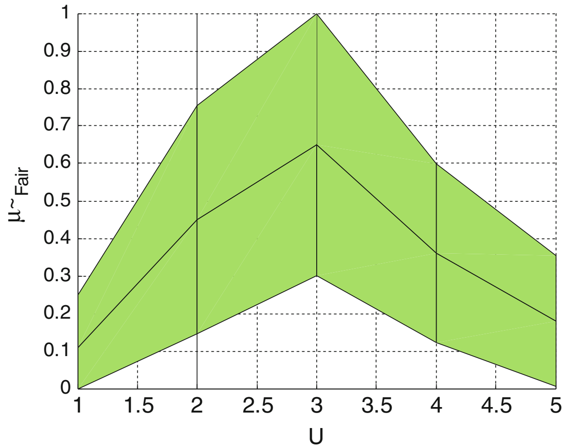


Fig. 3 IT2 FS of 'Fair Satisfaction' derived by calculating the mean estimation intervals at all $x \in X$, $1 - \alpha = 0.99$

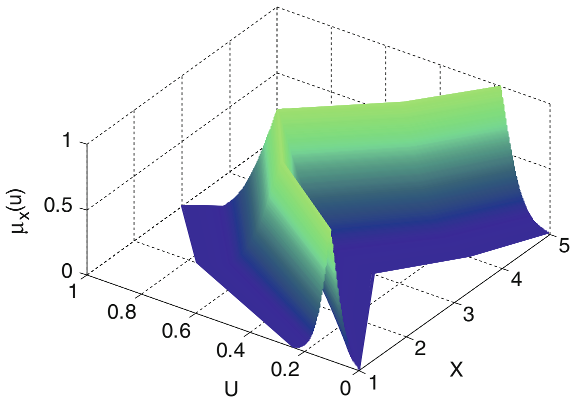


Fig. 4 T2 FS model of Fair *Satisfaction* in the context of online tourism services

for the measurand. In effect, each confidence interval at the level of $1 - \alpha$, corresponds to the α-cut of a fuzzy set that represents the uncertainty about the measurand.

In order to construct a T2 FS representing the *meaning* of l overall satisfaction rate, we will construct FOUs as discussed above, with various levels of confidence. In practice, we refer to each calculated FOU as an α-plane [16, 36] and by stacking nested planes the T2 FS would be constructed. Equivalently, from the vertical standpoint [16], at each $u \in U$, we construct a fuzzy grade by calculating mean confidence intervals at various levels of confidence and pile them up. More precisely, for l at $x \in X$ we calculate,

$$M_{1-\alpha}^{l}(x) = \left[m_{F^l}(x) - t_{\alpha/2}.\frac{s_{F^l}(x)}{\sqrt{N_l}},\ m_{F^l}(x) + t_{\alpha/2}.\frac{s_{F^l}(x)}{\sqrt{N_l}} \right],\ \alpha \in (0,\ 1] \tag{16}$$

Consequently, the fuzzy grade of $u \in U$ in the T2 FS $\boldsymbol{F}_{S_i'}{}^{l}$ is,

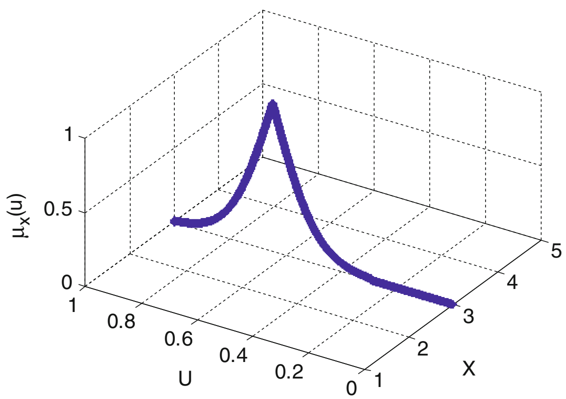


Fig. 5 Fuzzy grade of $x = 3$ in the T2 FS model of Fair *Satisfaction* in the context of online tourism services

$$\mu_{F_{S_i'^l}}(x) = \bigcup_{\alpha} M_{1-\alpha}^{l}(x) \tag{17}$$

Figure 4 shows the T2 FS calculated for *Fair Satisfaction* in the context of online tourism services. Figure 5 depicts the vertical slice of the T2 FS shown in Fig. 4 at $x = 3$.

6 Conclusion

In this chapter, we discussed a practical method for modeling the complex concept of user satisfaction in the context of online tourism services with T2 FSs. The method is composed of two parts, first, identification of each individual's mental model with T1 FS and then integrating all T1 FSs into a reasonable T2 FS. Existing methods for modeling concepts with T1 FSs mainly rely on mapping gathered data from a group of individuals into a parametric T1 FS. These methods require the person to provide endpoints of an interval as the quantification of his/her perception that should be later mapped to a parametric T1 FS. If the individual is knowledgeable enough, he/she might be expected to directly provide a T1 FS. Although these methods may work for simple concepts, they might not be applicable to complex concepts like user satisfaction. This would be explained with respect to the fact that one's quantification of such concepts is rather emotional, and expecting individuals to directly provide T1 FSs, at least highly limits the number of participating individuals. To handle the uncertainty of different individuals on the same concept, we mapped the calculated T1 FSs into interval and general T2 FSs. General T2 FSs were constructed based on the fuzzy approach to represent the uncertainty in measurement and by stacking the α-planes calculated at different levels of confidence around the estimated mean values of T1 FSs.

Acknowledgments The authors would like to sincerely acknowledge Professor G. Degli Antoni from Universita Degli Studi di Milano, and the staff of the Beauvais airport, Paris, without their help the data collection would not have been completed.

References

1. Montero, J., Ruan, D.: Modeling uncertainty. Inf. Sci. **180**, 799–802 (2010)
2. Pedrycz, W.: Human centricity in computing with fuzzy sets: an interpretability quest for higher order granular constructs. J. Ambient Intell Humaniz Comput **1**, 65–74 (2010)
3. Mendel, J.M.: Computing with words and its relationships with fuzzistics. Inf. Sci. **177**, 988–1006 (2007)
4. Mendel, J.M.: Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice-Hall, Upper Saddle River (2001)
5. Liu, F., Mendel, J.M.: Encoding words into interval type-2 fuzzy sets using an interval approach. IEEE Trans. Fuzzy Syst. **16**(6), 1503–1521 (2008)

6. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning-1. Inf. Sci. **8**, 199–249 (1975)
7. Mendel JM (2003) Fuzzy sets for words: a new beginning. Proc FUZZ-IEEE 37–42, St. Louis, MO
8. Tahayori H, Sadeghian A (2012) Handling uncertainties of membership functions with shadowed fuzzy sets. Proceeding of World Automation Congress, pp. 1–5, Puerto Vallarta, Mexico
9. Tahayori H, Sadeghian A.: Shadowed fuzzy sets: a framework with more freedom degrees than interval Type-2 fuzzy sets for handling uncertainties and lower computational complexity than general Type-2 fuzzy sets. In: Balas, V.E. et al. (ed.) New Concepts and Applications in Soft Computing, SCI 417, pp. 97–117, Springer, New York (2013)
10. Tahayori H, Sadeghian A, Pedrycz W.: Induction of shadowed sets based on the Gradual Grade of Fuzziness to be published in IEEE Transactions on Fuzzy Systems, doi:10.1109/TFUZZ.2012.2236843.
11. Moharrer M, Tahayori H, Sadeghian A.: Modeling linguistic label perception in tourism E-Satisfaction with Type-2 fuzzy sets. Proceeding of North American Fuzzy Information Processing Society (NAFIPS) Annual Meeting, pp. 1–6, Toronto, Canada (2010)
12. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
13. Mendel, J.M., John, R.I., Liu, F.: Interval type-2 fuzzy logic systems made simple. IEEE Trans. Fuzzy Syst. **14**(6), 808–821 (2006)
14. Tahayori, H., Degli Antoni, G.: Operations on concavoconvex Type-2 fuzzy sets. Int J.Fuzzy Syst. **10**(4), 276–286 (2008)
15. Tahayori H, Sadeghian A, Visconti A.: Operations on Type-2 Fuzzy Sets based on the set of Pseudo-Highest intersection points of convex Fuzzy Sets. Proceeding of North American Fuzzy Information Processing Society (NAFIPS) Annual Meeting, pp. 1–6, Toronto, Canada (2010)
16. Tahayori H, Tettamanzi AGB, Degli Antoni G.: Approximated Type-2 Fuzzy Set Operations. Proceeding of IEEE World Congress on Computational Intelligence, pp. 1910–1917, Vancouver, Canada (2006)
17. Tahayori, H., Tettamanzi, A.G.B., Degli Antoni, G., Visconti, A.: On the calculation of extended Max and Min operations between convex Fuzzy Sets of the real line. Fuzzy Sets Syst. **160**(21), 3103–3114 (2009)
18. Tahayori, H., Tettamanzi, A.G.B., Degli Antoni, G., Visconti, A., Moharrer, M.: Concave Type-2 fuzzy sets: properties and operations. Soft Comput. J. **14**(7), 749–756 (2010)
19. Bailey, J.E., Pearson, S.W.: Development of a tool for measuring and analyzing computer user satisfaction. Manage. Sci. **29**(5), 530–545 (1983)
20. Szymansky, M.D., Hise, R.: E-satisfaction: an initial examination. J. Retail. **76**(3), 309–322 (2000)
21. Akama, J.S., Kieti, D.M.: Measuring tourist satisfaction with Kenya's safari. Tourism Manag. **24**, 73–81 (2002)
22. Evanschitzky, H., Gopalkrishnan, R.I., Hesse, J., Ahlert, D.: E-satisfaction: a re-examination. J. Retail. **80**, 239–247 (2004)
23. Kaiser, H.F.: An index of factorial simplicity. Pschometrika **39**, 31–36 (1974)
24. Bartlett, MS.: (1937) Properties of sufficiency and statistical tests. Royal Stat. Soc. Ser. A **160**, 268–282
25. Cronbach, L.J., Shavelson, R.J.: My current thoughts on coefficient alpha and successor procedures. Educ. Psychol. Measur. **64**(3), 391–418 (2004)
26. Nunnally, J.C.: Psychometric theory. McGraw-Hill, New York (1967)
27. Moharrer M, Tahayori H.: Clustering E-satisfaction factors in tourism industry. Proceeding of International Conference Information Society, pp. 182–185, Merrillville Indiana, USA (2007)
28. Moharrer M, Tahayori H.: Drivers of customer convenience in electronic tourism industry. Proceeding of IEEE 20th Canadian Conference on Electrical and Computer Engineering, pp. 836–839, Vancouver, Canada (2007)

29. Moharrer M, Tahayori H, Albadvi A, Zegordi SH, Perzon H.: Satisfaction in E-Tourism, a case of European Online Customers. Proceeding of IADIS e-Commerce Conference, pp. 303–307, Barcelona, Spain (2006)
30. Moharrer M, Zegordi SH, Tahayori H.: Drivers of customer satisfaction in Electronic Tourism Industry. Proceeding of 24th EuroCHRIE Congress, Thessaloniki, Greece (2006)
31. Turksen, I.B.: Type 2 representation and reasoning for CWW. Fuzzy Sets Syst. **127**(1), 17–36 (2002)
32. Turksen, I.B., Willson, I.A.: A fuzzy set preference model for consumer choice. Fuzzy Sets Syst. **68**, 253–266 (1994)
33. Mencar, C., Fanelli, A.M.: Interpretability constraints for fuzzy information granulation. Inf. Sci. **178**(24), 4585–4618 (2008)
34. Dubois, D., Foulloy, L., Mauris, G., Prade, H.: Probability-possibility transformations, triangular Fuzzy sets, and probabilistic inequalities. Reliable Comput. **10**, 273–297 (2004)
35. Mauris, G., Lasserre, V., Foulloy, L.: A fuzzy approach for the expression of uncertainty in measurement. Measurement **29**, 165–177 (2001)
36. Mendel, J.M., Liu, F., Zhai, D.: α-Plane representation for Type-2 fuzzy sets: theory and applications. IEEE Trans. Fuzzy Syst. **17**(5), 1189–1207 (2009)

Construction of Interval Type-2 Fuzzy Sets From Fuzzy Sets: Methods and Applications

Miguel Pagola, Edurne Barrenechea, Javier Fernández, Aranzazu Jurio, Mikel Galar, Jose Antonio Sanz, Daniel Paternain, Carlos Lopez-Molina, Juan Cerrón and Humberto Bustince

Abstract In this chapter, we present some methods to construct interval type-2 membership functions from fuzzy membership functions and their applications in image processing, classification, and decision making. First, we review some basic concepts of interval type-2 fuzzy sets (IT2FSs). Next, we analyze three different approaches to construct IT2FSs starting from fuzzy sets and their applications in different fields.

1 Interval Type-2 Fuzzy Sets

From the beginning, it was clear that fuzzy set theory [30] was an extraordinary tool for representing human knowledge. The use of linguistic labels enables the acquisition of interpretable knowledge systems, and in this manner the choice of the membership function plays an essential role in their success. The punctual value set as membership degree is usually defined either by means of expert knowledge or homogeneously over the input space. Nevertheless, Zadeh himself established (see [31]) that sometimes, in decision-making processes, knowledge is better represented by means of some generalizations of fuzzy sets.

Extensions of fuzzy sets are not as specific as their counter-parts of fuzzy sets, but this lack of specificity makes them more realistic for some applications. Their advantage is that they allow us to express our uncertainty in identifying a particular membership function. This uncertainty is involved when extensions of fuzzy sets are processed, making results of the processing less specific but more reliable.

The concept of *type-2 fuzzy set* was suggested by Zadeh in 1975 [31] as a generalization of an ordinary fuzzy set. Type-2 fuzzy sets are characterized by a fuzzy membership function, that is, the membership value for each element of the

M. Pagola (✉) · E. Barrenechea · J. Fernández · A. Jurio · M. Galar · J. A. Sanz · D. Paternain · C. Lopez-Molina · J. Cerrón · H. Bustince
Universidad Pública de Navarra, Campus Arrosadia sn, 31006 Pamplona, Spain
e-mail: miguel.pagola@unavarra.es

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_10,

set is given by a fuzzy set defined in the reference set $[0, 1]$. These sets were first studied and analyzed in [20].

There is still some discussion about the notation for type-2 fuzzy sets. We shall follow the standard mathematical notation in the following definitions (for equivalences with other notations, see [1]). A good review of these sets can be found in [18].

Sometimes, it is appropriate to represent the membership degree of each element to the fuzzy set by means of an interval. Hence, not only vagueness (lack of sharp class boundaries), but also a feature of uncertainty (lack of information) can be addressed intuitively.

A particular case of type-2 fuzzy sets called interval type-2 fuzzy sets (see [19]). In May 1975 Sambuc (see [24]) presented in his doctoral thesis, the concept of an interval-valued fuzzy set named a Φ-fuzzy set. That same year, Zadeh [31] discussed the representation of type 2 fuzzy sets and its potential in approximate reasoning. One year later, Grattan-Guinness [13] established a definition of an interval-valued membership function. In that decade, interval-valued fuzzy sets appeared in the literature in various guises and it was not until the 1980s, that the importance of these sets, as well as their name, was definitely established. In [10, 16, 18], it is proved that interval-valued fuzzy sets are a particular case of IT2FSs. It turns out that interval type-2 fuzzy sets are isomorphic to *interval-valued fuzzy set* [24].

In this chapter, we work with finite, nonempty reference sets. We denote by $L([0, 1])$ the set of all closed subintervals of the unit interval $[0, 1]$ in the following way:

$$L([0,1]) = \{\mathbf{x} = [\underline{x}, \overline{x}] | (\underline{x}, \overline{x}) \in [0,1]^2 \text{ and } \underline{x} \leq \overline{x}\}. \quad (1)$$

We use bold letters to refer the elements $\mathbf{x} \in L([0, 1])$ and we denote with W the length of an interval, that is, $W(\mathbf{x}) = \overline{x} - \underline{x}$.

$L([0, 1])$ is a partially ordered set with respect to the relation $\leq_L$ defined in the following way: given $\mathbf{x}, \mathbf{y} \in L([0, 1])$,

$$\mathbf{x} \leq_L \mathbf{y} \text{ if andonly if } \underline{x} \leq \underline{y} \text{ and } \overline{x} \leq \overline{y}. \quad (2)$$

With this order relation, $(L([0, 1]), \leq_L)$ is a complete lattice, where the smallest element is $0_L = [0, 0]$ and the largest is $1_L = [1, 1]$.

An interval type 2 fuzzy set $\tilde{A}$ on U is defined by

$$\tilde{A} = \{(u, A(u), \mu_u(x)) | u \in U, A(u) \in L([0,1])\},$$

where $A(u) = [\underline{A}(u), \overline{A}(u)]$ is a closed subinterval of $[0, 1]$, and the function $\mu_u(x)$ represents the fuzzy set associated with the element $u \in U$ obtained when x covers the interval $[0, 1]$; $\mu_u(x)$ is given in the following way:

$$(x) = \begin{cases} a & \text{if } \underline{A}(u) \leq x \leq \overline{A}(u) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $0 \leq a \leq 1$. As we have said previously with $a = 1$ an interval type 2 fuzzy set is the same as an interval valued fuzzy set.

Mendel and others [17] defined IT2FSs using the footprint of uncertainty (FOU). An IT2FS $\tilde{A}$ for a primary variable $(x \in X)$ is characterized by its footprint of uncertainty, $FOU(\tilde{A})$, which in turn is completely described by its lower membership function, $LMF(\tilde{A})$, also denoted by $\underline{\mu}_{\tilde{A}}(x)$, and upper membership function $UMF(\tilde{A})$, also denoted by $\overline{\mu}_{\tilde{A}}(x)$, i.e., the lower and upper bounding functions of $FOU(\tilde{A})$ respectively.

Through the chapter we denote by $\mathcal{IT}2\mathcal{F}Ss(U)$ the set of all the interval type-2 fuzzy sets defined on U, and $\mathcal{F}Ss(U)$ all the fuzzy sets on U.

2 Construction Methods of IT2FSs

When we will develop an application using IT2FSs, the first step is to define the membership functions that will represent these sets. For example, if we use an IT2FS system then we must define the rules and the lower and upper membership functions of the linguistic labels. It is known that a key problem of the fuzzy systems is the definition of the membership functions, as we have previously stated.

Usually, IT2FS are defined manually or from data extracted [17]. Other typical method to obtain a good definition of the membership functions is to optimize their shape using genetic algorithms [14].

When we are working with IT2FSs, we must take into account that the FOU of the IT2FSs represents the uncertainty in the membership degree. Therefore the FOUs must represent the uncertainty that exists in the model.

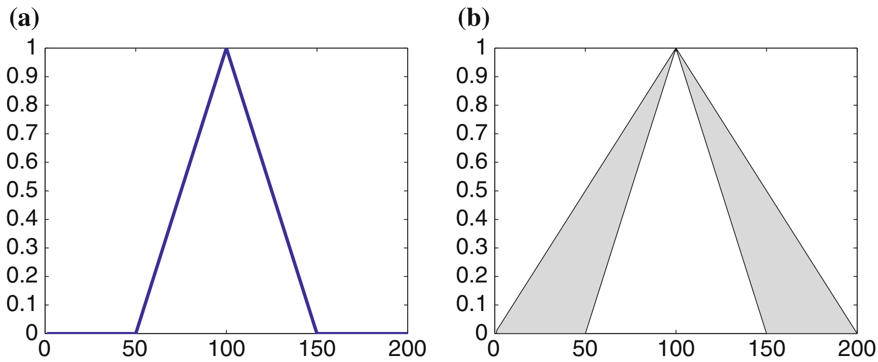


Fig. 1 **a** Triangular fuzzy membership function. **b** Triangular interval type-2 fuzzy membership function

In this work, we present three different methods to construct IT2FS from fuzzy sets that try to generate FOUs adapted to the model's uncertainty. We have studied three different cases:

- Using several fuzzy membership functions.
- Using two fuzzy membership functions that represent opposite objects or concepts.
- Using only one fuzzy membership function.

2.1 Construction of an IT2FS from Several Fuzzy Membership Functions

When we define a fuzzy system one key problem is the definition of the membership functions. In the context of fuzzy rule-based systems, sometimes the expert can choose between different functions (triangular, gaussian, etc.) and different parameters. Therefore, the expert is not sure about which is the best membership function, he can choose several adequate membership functions. If we want to construct an IT2FS from different membership functions, the IT2FS should be such that the lengths of the intervals represent the uncertainty that the expert has in the selection of these fuzzy sets. That is, if the expert is absolutely sure of the membership degree of an element, then the length of the interval associated to such element is zero (a fuzzy set). On the other hand, if the expert does not know the membership degree of an element at all, then the length of the interval associated to this element should be the maximum possible.

$$\begin{aligned}
&\Phi : \mathcal{FS}s(U)\overbrace{\times\cdots\times}^{k\text{ times}}\mathcal{FS}s(U) \longrightarrow \mathcal{IT2FS}s(U) \text{ given by}\\
&\Phi(A^1,\cdots,A^k) = \{(u,\Phi(A^1,\cdots,A^k)(u))|u\in U\} \quad \text{such that}\\
&\Phi(A^1,\cdots,A^k)(u) = [T(\mu_{A^1}(u),\cdots,\mu_{A^k}(u)) \quad S(\mu_{A^1}(u),\cdots,\mu_{A^k}(u))],
\end{aligned} \tag{4}$$

where T and S are a t-norm and a t-conorm, respectively, in $[0,1]$.

Remark The associativity of triangular norms and triangular t-conorms allows us to extend these mappings to an arbitrary finite number of arguments in a unique way, by means of a recursive definition. For example, n-ary triangular norms are defined as follows. Let $(x_i,\cdots,x_n)$ be a finite family in $[0,1]^n$. Then $T(x_1,\cdots,x_n) = T(T(x_1,\cdots,x_{n-1}),x_n)$.

We denote by W_{TS} the length of an interval constructed by the above method, where T is a t-norm and S is a t-conorm.

As U is discrete, our method can be seen as a construction of the footprint of uncertainty from several fuzzy sets. Suppose that the expert gives two different opinions (or two different experts each giving a single opinion). We can use a t-norm and a t-conorm to construct an IT2FS. Figure 2 depicted different IT2FSs

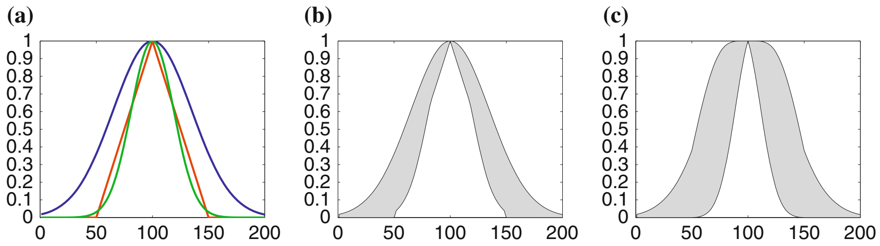


Fig. 2 (**a**) Three different fuzzy membership functions. (**b**) Interval type-2 fuzzy membership function generated using the t-norm minimum and the t-conorm maximum. (**c**) Interval type-2 fuzzy membership function generated using the t-norm product and the t-conorm probabilistic sum

constructed using different t-norms and t-conorms. We can also observe in Fig. 2 that the FOU generated with the t-norm product and the t-conorm probabilistic sum is wider than the one generated with the t-norm minimum and the t-conorm maximum.

Corollary 1 *Under the conditions of the construction method described in Eq.* (4), *the following statement is true:*

If T and S are any t-norm and t-conorm in [0,1], then

$$W_{TS}(\Phi(Q^1,\cdots,Q^k)(u)) \geq W_{\wedge\vee}(\Phi(A^1,\cdots,A^k)(u)) \forall u \in U.$$

Proof It is enough to take into account the fact that $\wedge$ is the largest t-norm and $\vee$ is the smallest t-conorm. □

This corollary proves that the FOU constructed with the t-norm minimum and t-conorm maximum is the smallest one. If other combination of t-norm and t-conorm is used the FOU will be greater.

2.2 Construction of an IT2FS from Two Fuzzy Membership Functions

Next we introduce the concept of *Ignorance function* and the way we use it to construct IT2FS from two related fuzzy sets. In this case, the fuzzy sets must be related with each other; they must represent opposite concepts. For example, one set represents the concept *near* and the other the concept *far*, or the concepts *small* and *big*. We have proposed this method in the context of image segmentation where we have two sets, one to represent the object and another to represent the background; but it can be used in any environment in which we have two different sets that represent opposite concepts.

The concept of *ignorance function* [6] tries to model the lack of knowledge that sometimes experts suffer when determining the membership degrees of some pixels of an image Q to the fuzzy set representing the background (B) of the image and to the fuzzy set representing the object (A) in the image.

For us, $\mu_B(x)$ $(\mu_A(x))$ is the quantification of the expert knowledge that the pixel with intensity x belongs to the background (object). In this sense, if $\mu_B(x) = 1(\mu_A(x) = 1)$, then the expert has total knowledge (total sureness) that the pixel belongs to the background (object). When $\mu_B(x) = 0.5$ $(\mu_A(x) = 0.5)$, we say that the expert is totally ignorant of whether the pixel belongs to the background (object) (total doubt). If the expert is totally sure that the pixel belongs to the background (object), then he should take $\mu_B(x) = 1$ and in this case the membership to the object (background) should be close to 0, ($\mu_A(x) \approx 0$). In spite of this, the simultaneous ignorance of a pixel's membership to the background and to the object will be given when the two membership functions are close to 0.5.

Evidently, there are pixels of the image for which the expert is absolutely sure that the chosen representation is the correct one. Nevertheless, there are also pixels for which the expert does not know if the representation taken is the best. We will represent the expert's ignorance in terms of μ_B and μ_A by means of what we denote as ignorance functions.

Under this interpretation, the following conditions must be fulfilled by these functions:

1. The ignorance function depends only on $\mu_B(x)$ and $\mu_A(x)$.
2. The ignorance does not depend on whether we first consider the membership to the background and then the membership to the object or we first consider the membership to the object and then the membership to the background.
3. (Representation of total knowledge) The ignorance of the expert in the choice of the membership of a pixel must be zero if and only if he is certain that the pixel belongs to the object or the background.
4. (Representation of total doubt) If $\mu_B(x) = 0.5$ and $\mu_A(x) = 0.5$; that is if the expert is not capable of distinguishing whether a pixel belongs to the background or to the object, then we will say that the expert's ignorance of the membership of this pixel to the background or to the object is one.
5. If the membership of the pixel to the background and its membership to the object are greater than 0.5, then the greater both memberships are, the smaller the ignorance should be.
6. If the membership of the pixel to the background and its membership to the object are smaller than 0.5, then the greater both memberships are, the greater the ignorance should be.

We just recall that these properties are equivalent for any problem in which there are two objects that represent opposite things, and therefore the mathematical definition is valid in those environments. The considerations above have led us to present the following definition.

Definition 1 A function $G_i : [0,1]^2 \rightarrow [0,1]$ is called an *ignorance function*, if it satisfies the following conditions:

(G_i1) $G_i(x,y) = G_i(y,x)$ for all $x, y \in [0,1]$;
(G_i2) $G_i(x,y) = 0$ if and only if $x = 1$ or $y = 1$;
(G_i3) If $x = 0.5$ and $y = 0.5$, then $G_i(x,y) = 1$;
(G_i4) G_i is decreasing in $[0.5,1]^2$;
(G_i5) G_i is increasing in $[0,0.5]^2$.

In some cases, it is advisable to require ignorance functionss to be continuous, since the ignorance must not present a chaotic reaction to small changes in the degree of knowledge that the experts possess regarding to the membership of the pixel in question to the background or to object. If this is the case, we will say that the ignorance functionss are continuous.

In the following theorem, we show a construction method of continuous ignorance functions from t-norms.

Theorem 1 [6] *Let T be a continuous t-norm such that*
$$T(x,y) = 0 \text{ if and only if } x \cdot y = 0.$$
Under these conditions, the function

$$G_i(x,y) = \begin{cases} \dfrac{T(1-x,1-y)}{T(0.5,0.5)} & \text{if } T(1-x,1-y) \le T(0.5,0.5) \\ \dfrac{T(0.5,0.5)}{T(1-x,1-y)} & \text{otherwise} \end{cases}$$

is a continuous ignorance function.

Example 1

(1) The t-norm minimum satisfies the conditions in Theorem 1, so

$$G_i(x,y) = \begin{cases} 2 \cdot \min(1-x,1-y) & \text{if } \min(1-x,1-y) \le 0.5 \\ \dfrac{1}{2 \cdot \min(1-x,1-y)} & \text{otherwise} \end{cases}$$

is a continuous ignorance function.

(2) The t-norm product satisfies the conditions in Theorem 1., so

$$G_i(x,y) = \begin{cases} 4 \cdot (1-x) \cdot (1-y) & \text{if } (1-x) \cdot (1-y) \le 0.25 \\ \dfrac{1}{4 \cdot (1-x) \cdot (1-y)} & \text{otherwise} \end{cases}$$

is a continuous ignorance function.

In [6], we developed a method to construct ignorance functions from functions different than the t-norms. Next, we show an example.

Example 2 If we take $\varphi(x) = \sqrt{x}$ for all $x \in [0, 1]$ we recover the following ignorance function:

$$G_i(x, y) = \begin{cases} 2\sqrt{(1-x)\cdot(1-y)} & \text{if } (1-x)\cdot(1-y) \leq 0.25 \\ \dfrac{1}{2\sqrt{(1-x)\cdot(1-y)}} & \text{otherwise} \end{cases}$$

Taking into account, the value of ignorance and the original fuzzy set we can construct the IT2FS. First, we assign the value of the ignorance function to the length W of the interval. Such a way the ignorance calculated represent the FOU:

$$W(x) = G_i(\mu_A(x), \mu_B(x)).$$

The main problem is that the lower membership function must always be greater than zero and the upper membership function must be lower than one. Therefore in [21], we propose the following method to construct IT2FS for two opposite fuzzy sets A and B:

$$\tilde{A}(u) = [S(0, \mu_A(u) + \lambda \times W(u)/2) \quad T(1, \mu_A(u) + \lambda \times W(u)/2)], \tag{5}$$

where T and S are a t-norm and a t-conorm, respectively, in $[0, 1]$ *and* $\lambda > 0$.

With this method the interval generated is always within $[0, 1]$. Also the parameter λ modifies the length of the intervals. If $\lambda = 1$ then the length of the interval is the same as the value of the ignorance function G_i calculated.

One of the advantages of this method is that the shape of the FOU is related with the shape of the membership functions, as we can see in Fig. 3.

2.3 Construction of an IT2FS from One Fuzzy Membership Function

If we have a membership function that represents the fuzzy set that modelizes certain concept, sometimes we know that there exist uncertainty in this membership. There exist several works that try to obtain from the proper membership function a value of the uncertainty and from this value to construct an IT2FS. Mainly two different approaches have been proposed. The first one intervals are generated using one or two parameters, we denote this method as interval generators. The second approach intervals are constructed by means of a function that only depends on the value of the membership function. This function gives an ignorance value, related with the membership degree, allowing us to construct an IT2FS.

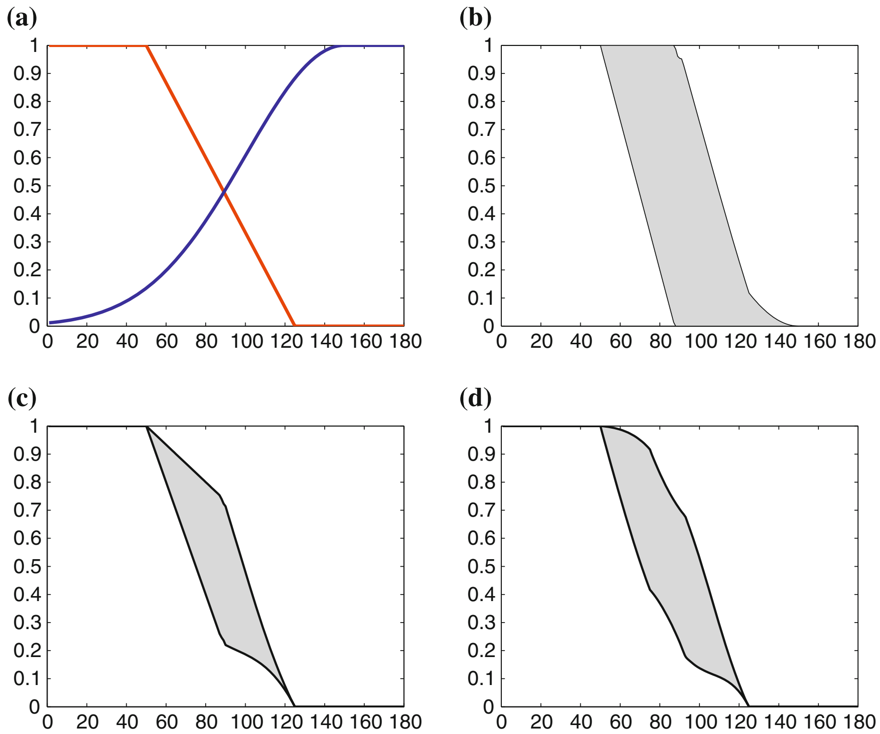


Fig. 3 **a** Two different fuzzy membership functions. **b** IT2FS generated from the ignorance function of example 1.1 and $\lambda = 1$. **c** IT2FS generated from the ignorance function of example 1.1 and $\lambda = 0.5$. **d** IT2FS generated from the ignorance function of example 1.2 and $\lambda = 0.5$

2.3.1 Interval Generators

If the uncertainty presented in the problem is due to a known cause, we can modelize it with some functions [3], called generators, and construct an IT2FS from the former fuzzy set.

In the following example we present an interval generator with two parameters.

Let $A \in \mathcal{FS}s(U)$ and let the functions:

$$\begin{cases} f : [0,1] \to [0,1] \text{ given by} \\ \qquad f(x) = x^{\alpha} \text{ with } \alpha \geq 1. \\ g : [0,1] \to [0,1] \text{ given by} \\ \qquad g(x) = x^{\frac{1}{\beta}} \text{ with } \beta \geq 1. \end{cases}$$

Under these conditions

$$\tilde{A}_{\alpha,\beta} = \{(u, [\mu_A^{\alpha}(u)\ \mu_A^{\frac{1}{\beta}}(u))]|u \in U\} \in \mathcal{IT2FS}s(U).$$

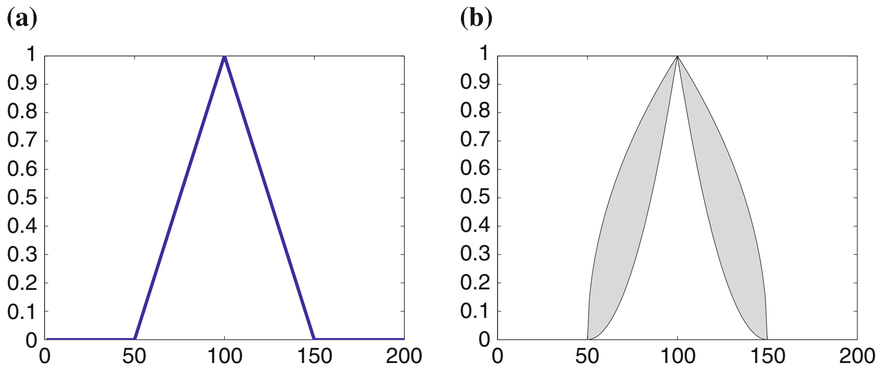


Fig. 4 **a** Original fuzzy set. **b** IT2FS generated from an interval generator with $\alpha = 2$ and $\beta = 2$.

The verification that $\tilde{A}_{\alpha,\beta} \in \mathcal{IT2FSs}(U)$ is evident: $0 \leq \mu_A^{\alpha}(u) \leq \mu_A^{\frac{1}{\beta}}(u) \leq 1$. The parameters α and β can be related with the ignorance of the expert in the membership function selection. An specific case with only one parameter α is:

$$\tilde{A}_{\alpha} = \{(u, [\mu_A^{\alpha}(u)\ \mu_A^{\frac{1}{\alpha}}(u))]|u \in U\} \in \mathcal{IT2FSs}(U). \tag{6}$$

Figure 4 depicted a fuzzy set and an IT2FS generated with values of $\alpha = 2$ and $\beta = 2$.

2.3.2 Weak Ignorance Function

The length of the IT2FSs can be seen as a representation of the ignorance when assigning punctual values as membership degrees. In order to measure the ignorance degree, we define the concept of weak ignorance functions [26], which are a particular case of ignorance functions depending on a single variable and demanding a less number of properties.

Definition 2 [26] A weak ignorance function is a mapping $g : [0, 1] \to [0, 1]$ that satisfies:

- $(g1)$ $g(x) = g(1 - x)$ for all $x \in [0, 1]$;
- $(g2)$ $g(x) = 0$ if and only if $x = 0$ or $x = 1$;
- $(g3)$ $g(0.5) = 1$.

Example 3 $g(x) = 2 \cdot \min(x, 1 - x)$ is a weak ignorance function.

We also present in [26] the following construction method of IT2FSs. First, we assign the length of the interval the value of ignorance of the membership degree of the fuzzy set A, i.e., $W(u) = g(\mu_A(u))$ and then we construct the IT2FSs $\tilde{A}$ in the following way:

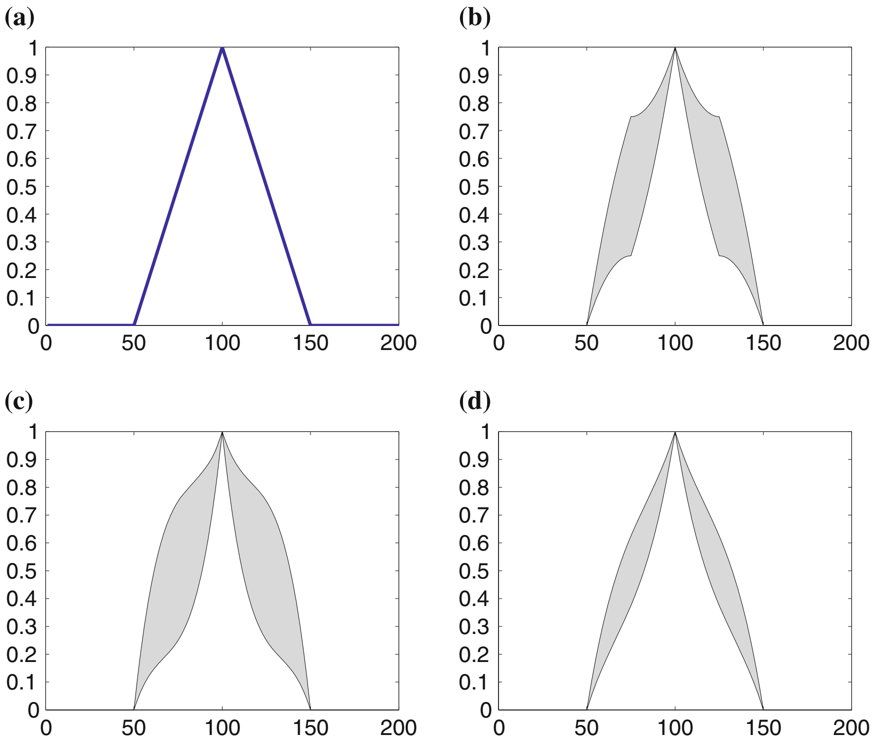


Fig. 5 **a** Original fuzzy set. **b** IT2FS generated with weak ignorance function of equation $g(x) = 2 \cdot \min(x, 1 - x)$ and $\lambda = 1$. **c** IT2FS generated with weak ignorance function of equation $g(x) = 4 \cdot (x \cdot (1 - x))$ and $\lambda = 1$. **d** IT2FS generated with weak ignorance function of equation $g(x) = 4 \cdot (x \cdot (1 - x))$ and $\lambda = 0.5$

$$\tilde{A} = \{(u, [\mu_A(u)(1 - \lambda \times W(u)) \quad \mu_A(u)(1 - \lambda \times W(u)) + \lambda \times W(u)] | u \in U\}. \tag{7}$$

Also the parameter λ modifies the length of the intervals. If $\lambda = 1$ then the length of the interval is the same as the value of the ignorance function g calculated.

Figure 5 depicted three different IT2FSs generated from different weak ignorance functions.

3 Applications

Next we present three different applications where we use IT2FSs constructed with the methods presented in the previous section.

3.1 Classification

Fuzzy rule-based classification systems (FRBCSs) are widely employed in classification tasks since they allow us to deal with noisy, imprecise or incomplete information which is often present in many real world problems. They provide a good trade-off between the empirical precision of traditional engineering techniques and the interpretability achieved through the use of linguistic labels whose semantic is close to the natural language.

However, FRBCSs can suffer a lack of system accuracy as a result of the uncertainty related to the definition of the membership functions.

In [25], we propose a methodology in which we use IT2FSs to model the linguistic labels of the classification system. To do so, we define a new parametrized IT2FSs construction method using triangular shaped membership IT2FSs. Specifically, the amplitude of the support of the upper bound of the IT2FSs is determined by the value of the parameter W, which establishes the relationship between the length of the lower and the upper bounds of each IT2FS. In this manner, we can build an IT2FSs model using the initial knowledge base generated by any fuzzy rule learning algorithm. Furthermore, the representation of the linguistic labels by means of IT2FSs leads to a natural extension of the classical fuzzy reasoning method (FRM) [8]. Specifically, we modified the two first steps out of the four, which compose the original FRM, in the following way:

- *Matching degree*: we apply a t-norm to the lower and upper bounds of the interval membership degrees of the elements to the IT2FSs composing the antecedent of the rules.
- *Association degree*: we take the mean between the product of the matching degree by the rule weight associated with the lower bound and the product of the matching degree by the rule weight associated with the upper bound.

In addition, we defined an evolutionary tuning in which we modified the value of the parameter W for each IT2FS used in the system. In this way, we tried to improve the system's performance by looking for the best amount of uncertainty that the FOU of each IT2FS represents.

In the experimental study, we used two well-recognized fuzzy rule learning methods, i.e., the algorithm proposed by Chi et al. [9] and the fuzzy hybrid genetics-based machine learning (FH-GBML) defined by Ishibuchi and Yamamoto [11]. In both cases, the application of our methodology (to the knowledge base generated by each algorithm) allowed to notably enhance the results provided by the initial nonIT2 fuzzy methods.

In [26], using the concept of weak ignorance function, we formalize the IT2FSs construction method introduced in [25] by establishing the relationship between the uncertainty represented by the FOUs of the IT2FSs and the ignorance degree. Specifically, we achieve that the length of the intervals, which are assigned as the membership degree of the elements to the set, are proportional to the weak ignorance degree computed by $g(x)$.

The experimental study supported the suitability of our method, since we outperformed the results of: (1) the original FH-GBML method; (2) the tuning approach based on the linguistic 3-tuples representation applied to the original fuzzy knowledge base, and (3) the lateral tuning applied to both the nonIT2 and the IT2 fuzzy versions of the knowledge base.

3.2 Image Segmentation

In 2005, Tizhoosh [28] presented an image thresholding approach using interval type 2 fuzzy sets (we must point out that he tries to use type 2 fuzzy sets, however he only uses interval type 2 fuzzy sets [7]). His study is based on the modification of the classical fuzzy algorithm of Huang and Wang [15], so that he applies an α factor as an interval generator to the membership function. Starting from a membership function, Tizhoosh obtains an interval type-2 fuzzy set that "contains" different membership functions and is useful for finding the threshold of an image. Tizhooh's algorithm is applied directly to color segmentation using RGB in [27] and it is also used to segment color image skin lesions [29]. Starting form the idea of obtaining the uncertainty from the information given by the user, we have proposed an approximation using interval type-2 fuzzy sets generated from interval generators [3] (where the key point is to choose the correct parameters). Also we have used interval type-2 membership functions within an algorithm of stereo matching [12] (in this case we use the terminology of interval-valued fuzzy sets). In said paper, we were interested in eliminating the sensitivity to the radiometric gain, bias, and noise using IT2FSs to represent the images. In this way, we managed the cited problems by splitting the image into two different areas (background and objects), where the membership degree of each pixel to an object or to the background is represented with an interval.We proposed a thresholding-based segmentation to build these interval type-2 fuzzy sets. These works led us to introduce the concept of ignorance function to try to model the lack of knowledge from which experts may suffer when determining the membership degrees of some pixels of a given image. This concept was presented in [5] and [6] where we modified the classical fuzzy thresholding algorithm such way the user should pick two functions, one to represent the background and another one to represent the object, instead of using one membership function to represent the whole image; that is, we proposed by means of *ignorance functions* to modelize the user's ignorance for choosing these two membership functions. From this value of ignorance we constructed the IT2FSs. The rest of the algorithm remained similar to the algorithm using IT2FS constructed from interval generators.

We evaluated the performance of the algorithm that uses ignorance functions in natural images and prostate ultrasound images. We must take into account that, since ultrasound images depend on the particular settings of the machine is very important that our algorithm gives good solutions even if some membership functions that do not represent accurately the background and the prostate are chosen. The IT2FS algorithm performance was compared with the classical fuzzy algorithm and we can conclude that *for the pairs of membership functions such that the fuzzy algorithm solution is good (small error), the IT2FS algorithm does not provide better results but if the error we get with the fuzzy algorithm begins to be high (i.e., if we have used bad-chosen membership functions), then the result of the IT2FS algorithm improves the other algorithm's result.*

3.3 Decision Making

Fuzzy preference relations have been widely used to model preferences for decision-making problems due to their high expressiveness and their effectiveness as a tool for modeling decision processes. In the fuzzy case, the experts express their opinions using a difference scale [0,1]. In [2] we presented a generalization of the nondominance criterion proposed by Orlovsky using interval preferences.

Our method starts from fuzzy preferences and by means of weak ignorance functions we construct an interval type-2 fuzzy preference matrix (in the paper we use the notation of interval valued fuzzy preference relation).

Let $R^* \in FR(X \times X)$ be a fuzzy preference relation over a set of alternatives $X = \{x_1, \ldots, x_n\}$; for each pair of alternatives x_i and x_j, $R^*_{ij} = R^*(x_i, x_j)$ represents a degree of (weak) preference of x_i over x_j, namely the degree to which x_i is considered as least as good as x_j.

Given $R^* \in FR(X \times X)$ we normalize it to $[0, 1]$ in such a way that for each element of the new relation, denoted by $R \in FR(X \times X)$, holds that $R_{ij} = 1 - R_{ji}$.

Next, from R we must extract a set of nondominated alternatives as the solution of the decision-making problem. Specifically, the maximal nondominated elements of R are calculated extending the nondominance criterion proposed by Orlovsky in [22] to intervals.

The *Non-dominance Interval Algorithm* that we proposed [2] is the following: Given a fuzzy preference relation R^* (without defined elements in the main diagonal) and a *weak fuzzy ignorance function* g,

1. Construct R normalizing R^*
2. Compute the fuzzy strict preference relation R^s in Orlovsky's sense
3. Build the interval type-2 fuzzy relation $\mathbf{r}$:

$$\mathbf{r_{ij}} = \begin{cases} [R^s_{ij} \cdot (1 - g(R_{ij})), R^s_{ij} \cdot (1 - g(R_{ij})) + g(R_{ij})] & \text{if } R_{ij} > R_{ji} \\ [0, g(R_{ij})] & \text{otherwise} \end{cases} \quad (8)$$

4. Build the interval type-2 fuzzy set:

$$ND_{IV} = \{(x_j, ND_{IV}(x_j))|x_j \in X\} \quad \text{where}$$
$$ND_{IV}(x_j) = \mathbf{S}(\mathbf{r}_{ij}) = \left[\bigvee_{i=1}^{n}(\underline{r}_{ij}), \bigvee_{i=1}^{n}(\overline{r}_{ij})\right] \tag{9}$$

5. Build the interval type-2 fuzzy set:

$$N_{IV}(ND_{IV}) = \{(x_j, N_{IV}(ND_{IV}(x_j)))|x_j \in X\} \quad \text{where} \tag{10}$$

$$N_{IV}(ND_{IV})(x_j) = \left[1 - \bigvee_{i=1}^{n}(\overline{r}_{ij}), 1 - \bigvee_{i=1}^{n}(\underline{r}_{ij})\right] \tag{11}$$

6. Order the elements of $N_{IV}(ND_{IV})$ in a decreasing way in terms of accuracy and score functions.
7. If there exist several alternatives occupying the firstplace, take as solution the alternative with the biggest upper bound of its interval associated.

We must remark that if for a majority of the elements $\mathbf{r_{ij}}$ we have that $g(R_{ij}) \to 0$, then the resulting intervals have a very small length and it is reasonable to assume that the result obtained with the algorithm is the same than the result obtained with the nondominance algorithm.

If for a majority of the elements $\mathbf{r_{ij}}$ we have that $g(R_{ij}) \to 1$, then the algorithm allows us to distinguish better than the nondominance algorithm the alternative or alternatives that we must take as solution.

4 Conclusions

A key problem of fuzzy systems and algorithms is the accurate election of the membership function. In this chapter, we have presented three different methods to generate interval type-2 fuzzy sets from fuzzy sets, such that they are very goodtools to represent the uncertainty existing in the problem or specifically in the election of the correct membership function. We have presented three different applications in which these methods have been applied successfully. In some cases, the IT2FS systems or algorithms achieved an improvement in the results of the original fuzzy cases.

As future research we plan to study different methods to construct IT2FS from data. Another interesting study is how to construct a general type-2 fuzzy set from a fuzzy set.

Acknowledgments This research was partially supported by grant TIN2010-15505 from the Government of Spain.

References

1. Aisbett, J., Rickard, J.T., Morgenthaler, D.G.: Type-2 fuzzy sets as functions on spaces. IEEE Trans. Fuzzy Syst. **18**(4), 841–844 (2010)
2. Barrenechea, E., Fernández, A., Herrera, F., Bustince, H.: Construction of Interval-valued fuzzy preference relations using ignorance functions. Interval-valued Non Dominance Criterion, Advances in Intelligent and Soft Computing 107, Eurofuse : Workshop on Fuzzy Models and. Knowledge-Based Systems, 243–257 (2011)
3. Bustince, H., Kacpryzk, J., Mohedano, V.: Intuitionistic fuzzy generators—application to intuitionistic fuzzy complementation. Fuzzy Sets Syst. **114**, 485–504 (2000)
4. Bustince, H., Barrenechea, E., Pagola, M.: Image thresholding using restricted equivalence functions and maximizing the measures of similarity. Fuzzy Sets Syst. **158**, 496–516 (2007)
5. Bustince, H., Pagola, M., Barrenechea, E., Orduna, R.: Representation of uncertainty associated with the fuzzification of an image by means of interval type 2 fuzzy sets. Application to threshold computing. In Proceedings of Eurofuse Workshop: New Trends in Preference Modelling, Eurofuse, (Spain) 73–78 (2007)
6. Bustince, H., Pagola, M., Barrenechea, E., Fernandez, J., Melo-Pinto, P., Couto, P., Tizhoosh, H.R., Montero, J.: Ignorance functions. An application to the calculation of the threshold in prostate ultrasound images. Fuzzy Sets Syst. **161**(1), 20–36 (2010)
7. Bustince, H., Barrenechea, E., Pagola, M., Fernandez, J., Sanz, J.: Comment on: image thresholding using type II fuzzy sets. Importance of this method. Pattern Recognit. **43**(9), 3188–3192 (2010)
8. Cordón, O., del Jesus, M.J., Herrera, F.: A proposal on reasoning methods in fuzzy rule-based classification systems. Int. J. Approximate. Reasoning. **20**(1), 21–45 (1999)
9. Chi, Z., Yan, H., Pham, T.: Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition. World Scientific, singapore (1996)
10. Deschrijver, G., Kerre, E.E.: On the relationship between some extensions of fuzzy set theory. Fuzzy Sets Syst. **133**(2), 227–235 (2003)
11. Isibuchi, H., Yamamoto, T., Nakashima, T.: Hybridization of fuzzy GBML approaches for pattern classification problems. IEEE Trans. Syst. Man Cybern. B **35**(2), 359–365 (2005)
12. Galar, M., Fernandez, J., Beliakov, G., Bustince, H.: Interval-Valued fuzzy sets applied to stereo matching of color Images. IEEE Trans. Image Process. **20**, 1949–1961 (2011)
13. Grattan-Guinness I.:Fuzzy membership mapped onto interval and many-valued quantities. Z. Math. Logik Grundlag. Mathe. **22**, 149–160 (1976)
14. Hidalgo, D., Melin, P., Castillo, O.: An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms. Expert Syst. Appl. **39**(4), 4590–4598 (2012)
15. Huang, L.K., Wang, M.J.: Image thresholding by minimimizing the measure of fuzziness. Pattern recognit. **28**(1), 41–51 (1995)
16. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, New York (1995)
17. Liu, F., Mendel, J.M.: Encoding words into interval type-2 fuzzy sets using an interval approach. IEEE Trans. Fuzzy Syst. **16**(6), 1503–1521 (2008)
18. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
19. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems. Prentice-Hall, Upper Saddle River (2001)
20. Mizumoto, M., Tanaka, K.: Some properties of fuzzy sets of type 2. Inform. Control **31**, 312–340 (1976)
21. Pagola, M.: Representation of uncertainty by interval-valued fuzzy sets. Application to image thresholding. Ph.D. dissertation, Departamento de Automática y Computacin, Universidad Pública de Navarra, Pamplona (2008)

22. Orlovsky, S.A.: Decision-making with a fuzzy preference relation. Fuzzy Sets Syst. **1**(3), 155–167 (1978)
23. Pal, S.K., King, R.A., Hashim, A.A.: Automatic grey level thresholding through index of fuzziness and entropy. Pattern Recognit. Lett. **1**(3), 141–146 (1983)
24. Sambuc, R.: Function *Φ*- Flous. Application a l'aide au Diagnostic en Pathologie Thyroidienne. These de Doctorat en Medicine, University of Marseille (1975)
25. Sanz, J., Fernandez, A., Bustince, H., Herrera, F.: Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. Inf. Sci. **180**, 3674–3685 (2010)
26. Sanz, J., Fernandez, A., Bustince, H., Herrera, F.: A genetic tuning to improve the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets: degree of ignorance and lateral position. Int. J. Approximate Reasoning **52**(6), 751–766 (2011)
27. Tehami, S., Bigand, A., Colot, O.: Color image segmentation based on type-2 fuzzy sets and region merging. Lect. Notes Comput. Sci. **4678**, 943–954 (2007)
28. Tizhoosh, H.R.: Image thresholding using type-2 fuzzy sets. Pattern Recognit. **38**, 2363–2372 (2005)
29. Yuksel, M.E., Borlu, M: Accurate segmentation of dermoscopic images by image thresholding based on type-2 fuzzy logi. IEEE Trans. Fuzzy Syst. 976–982 (2009)
30. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
31. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning-I. Inf. Sci. **8**, 199–249 (1975)

Interval Type-2 Fuzzy Membership Function Generation Methods for Representing Sample Data

Frank Chung-Hoon Rhee and Byung-In Choi

Abstract Type-2 fuzzy sets (T2 FSs) have been shown to manage uncertainty more effectively than type-1 fuzzy sets (T1 FSs) in several areas of engineering. However, computing with T2 FSs can require an undesirably large amount of computations since it involves numerous embedded T2 FSs. To reduce the complexity, interval type-2 fuzzy sets (IT2 FSs) can be used, since the secondary memberships are all equal to one. In this chapter, three novel interval type-2 fuzzy membership function (IT2 FMF) generation methods are proposed. The methods are based on heuristics, histograms, and interval type-2 fuzzy C-means (IT2 FCM). For each method, the footprint of uncertainty (FOU) is only required to be obtained, since the FOU can completely describe an IT2 FMF. The performance of the methods is evaluated by applying them to back-propagation neural networks (BPNNs). Experimental results for several data sets are given to show the effectiveness of the proposed membership assignments.

1 Introduction

For many pattern classification applications, type-1 fuzzy sets (T1 FSs) have been successfully used to model the various uncertainties associated with the sample data over conventional methods [1–10]. Although T1 FSs may properly model the uncertainties, there may still exist uncertainties in the fuzzy pattern classification algorithm. To manage the uncertainties more effectively, type-2 fuzzy sets (T2 FSs)

F. C.-H. Rhee (✉)
Computation Vision and Fuzzy Systems Laboratory, Department of Electronic and Communication Engineering, Hanyang University,
55 Hanyang Daehak-Ro, Sangnok-Gu, Ansan-Si, Gyeounggi-Do 426-791, Korea
e-mail: frhee@fuzzy.hanyang.ac.kr

B.-I. Choi
Strategic Sensor Group, Samsung Thales Co. Ltd, Yongin, Korea

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_11,

have been successfully applied to various engineering areas [11–15]. This is due to the extra degree of freedom that T2 FSs possess. However, operations of T2 FSs, such as union (join), intersection (meet), and type reduction, involve numerous combinations of embedded T2 FSs. To reduce the complexity, the use of interval type-2 fuzzy sets (IT2 FSs), where all secondary grades are uniformly weighted (i.e., all equal to one) have been introduced [16–24]. However, the type-reduction process in IT2 FSs can still require a large amount of computation. Methods for reducing the computational complexity have been proposed [15, 24].

The issue of automatic generation of T1 FMFs from pattern data is an important step in developing algorithms that can handle uncertainties. Several T1 FMF generation methods have been proposed [6–10]. Typical T1 FMFs were generated based on heuristics (e.g., concepts based on human perception and experts), histograms, probability, and entropy. Additionally, algorithms based on the fuzzy nearest neighbor, back-propagation neural network, fuzzy C-means (FCM), robust agglomerative Gaussian mixture decomposition (RAGMD), and self-organizing feature map (SOFM) were used to generate T1 FMFs as well.

Generation methods of T2 FMFs have been limitedly discussed although various algorithms based on T2 FMFs have been proposed. In this chapter, we focus on some simple and effective generation methods of interval type-2 fuzzy membership functions (IT2 FMFs). Three methods for generating IT2 FMFs automatically from sample data are presented [16]. The three methods are based on heuristics, histograms, and interval type-2 fuzzy C-means (IT2 FCM) clustering. The heuristic method generates IT2 FMFs by simply incorporating heuristic T1 FMFs. The histogram-based method generates IT2 FMF by performing parameterized function fitting to smoothed histograms of sample data. The IT2 FCM-based method uses the formulas of IT2 FCM to obtain IT2 FMFs. For each method, the footprint of uncertainty (FOU) is only required to be obtained, since the FOU can completely describe an IT2 FMF (i.e., all secondary grades are equal to one).

To validate the IT2 FMF design methods, we apply them to back-propagation neural networks (BPNNs). T1 FMF values for each pattern are computed from the interval centroids of the generated IT2 FMFs and are used as inputs to train the BPNN. The remainder of this chapter is organized as follows. In Sect. 2, we briefly explain IT2 FSs. In Sect. 3, we present the three IT2 FMF generation methods. In Sect. 4, we explain how the IT2 FMF generation methods can be applied in a BPNN. Section 5 gives several examples showing the validity of the generation methods. Finally, Sect. 6 gives the summary and conclusions.

2 Overview of Interval Type-2 Fuzzy Sets

The extension of T1 FSs to T2 FSs can be used to effectively describe uncertainties in situations where the available information is uncertain. T2 FSs include a secondary membership function to model the uncertainty of exact (crisp) T1 FSs. A T2 FS in the universal set $\mathbf{X}$, denoted as $\tilde{\mathbf{A}}$, can be characterized by a T2 FMF $\mu_{\tilde{\mathbf{A}}}(x, u)$ as

$$\tilde{\mathbf{A}} = \int_{x\in\mathbf{X}} \mu_{\tilde{\mathbf{A}}}(x)/x = \int_{x\in\mathbf{X}} \left[\int_{u\in J_x} f_x(u)/u\right]/x \quad J_x \subseteq [0,\ 1], \tag{1}$$

where $f_x(u)$ is the secondary membership function and J_x is the primary membership of x which is the domain of the secondary membership function [24]. The region bounded by an upper membership function (UMF) $\bar{\mu}_{\tilde{\mathbf{A}}}(x)$ and lower membership function (LMF) $\underline{\mu}_{\tilde{\mathbf{A}}}(x)$ is called FOU. The FOU of $\tilde{\mathbf{A}}$ can be expressed by the union of all the primary memberships as

$$\mathrm{FOU}(\tilde{\mathbf{A}}) = \bigcup_{\forall x\in\mathbf{X}} J_x = \{(x,u) : u \in J_x \subseteq [0,1]\}. \tag{2}$$

The secondary membership function is a vertical slice of $\mu_{\tilde{\mathbf{A}}}(x, u)$. Although T2 FSs may be useful in modeling uncertainty where T1 FSs cannot, the operations of T2 FSs involve numerous embedded T2 FSs which consider all possible combinations of secondary membership values. Therefore, undesirably large amounts of computations may be required. However, IT2 FSs can be used to reduce the computational complexity. IT2 FSs are specific T2 FSs whose secondary membership functions are interval sets expressed as

$$\tilde{\mathbf{A}} = \int_{x\in\mathbf{X}} \left[\int_{u\in J_x} 1/u\right] \Big/ x. \tag{3}$$

All the secondary memberships are uniformly weighted for each primary membership of x. Therefore, J_x can be expressed as

$$J_x = \left\{(x,u) : u \in \left[\underline{\mu}_{\tilde{\mathbf{A}}}(x), \bar{\mu}_{\tilde{\mathbf{A}}}(x)\right]\right\}. \tag{4}$$

Moreover, $\mathrm{FOU}(\tilde{\mathbf{A}})$ in (2) can also be expressed as

$$\mathrm{FOU}(\tilde{\mathbf{A}}) = \bigcup_{\forall x\in\mathbf{X}} \left[\underline{\mu}_{\tilde{\mathbf{A}}}(x), \bar{\mu}_{\tilde{\mathbf{A}}}(x)\right]. \tag{5}$$

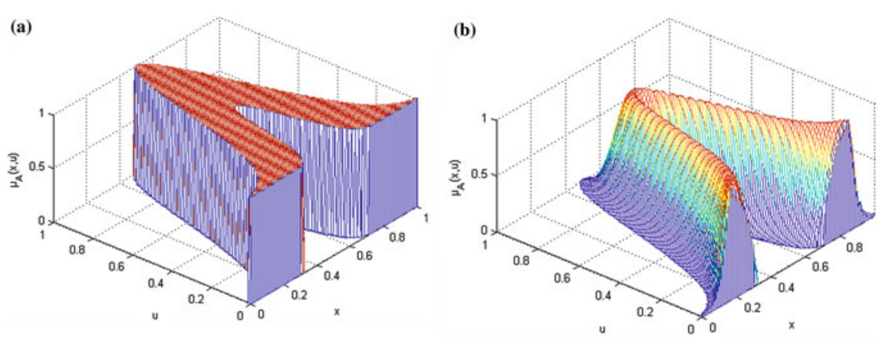


Fig. 1 Illustration of T2 FMFs: (**a**) IT2 Gaussian FMF and (**b**) general T2 Gaussian FMF

As a result, IT2 FSs requires only simple interval arithmetic for computing. Figure 1a and b shows the result of an IT2 Gaussian FMF and general T2 Gaussian FMF, respectively.

3 Interval Type-2 Fuzzy Membership Function Generation Methods

In this section, we present three methods for generating IT2 FMF automatically from pattern data. The methods are based on heuristics, histograms, and IT2 FCM. The heuristic-based design method simply generates the IT2 FMF using heuristic T1 FMFs and a scaling factor. The histogram-based method uses suitable parameterized functions chosen to model the histogram representing the sample data. The IT2 FCM-based method uses the derived formulas of the IT2 FMFs in the IT2 FCM algorithm [18]. A detailed description of each method is discussed as follows.

3.1 Heuristic Method

The heuristic method simply uses an appropriate predefined T1 FMF function, such as triangular, trapezoidal, Gaussian, *S*, or π function, to name a few, to initially represent the distribution of the pattern data. The following are some frequently used heuristic membership functions:

(1) Triangular function

$$\mu(x;a,b,c) = \begin{cases} 0 & \text{if} \quad x \le a \\ \frac{x-a}{b-a} & \text{if} \quad a \le x \le b \\ \frac{c-x}{c-b} & \text{if} \quad b \le x \le c \\ 0 & \text{if} \quad x \ge c. \end{cases} \tag{6}$$

(2) Trapezoidal function

$$\mu(x;a,b,c,d) = \begin{cases} 0 & \text{if} \quad x \le a \\ \frac{x-a}{b-a} & \text{if} \quad a \le x \le b \\ 1 & \text{if} \quad b \le x \le c \\ \frac{c-x}{c-b} & \text{if} \quad c \le x \le d \\ 0 & \text{if} \quad x \ge d. \end{cases} \tag{7}$$

(3) Gaussian function

$$\mu(x) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{8}$$

(4) S-function

$$S(x;a,b,c) = \begin{cases} 0 & \text{if} \quad x \le a \\ 2\cdot\left(\frac{x-a}{c-a}\right)^2 & \text{if} \quad a \le x \le b \\ 1-2\cdot\left(\frac{x-a}{c-a}\right)^2 & \text{if} \quad b \le x \le c \\ 1 & \text{if} \quad x \ge c \end{cases}, \tag{9}$$

where $b = \dfrac{a+c}{2}$.

(5) π-function

$$\Pi(x;a,b,c) = \begin{cases} S(x; c-b, c-\frac{b}{2}, c) & x \le c \\ 1 - S(x; c, c+\frac{b}{2}, c+b) & x \ge c. \end{cases} \tag{10}$$

Once a T1 FMF is selected such as the ones in (6–10), the heuristic T1 FMF $\mu_i(x)$ for feature i is designed by determining the parameters of the function which are usually provided by an expert. This becomes the UMF $\bar{u}_i(x)$ of the IT2 FMF. The LMF $\underline{u}_i(x)$ is obtained by scaling the UMF by a factor α between 0 and 1, which can also be provided by an expert. The FOU of the heuristic method for feature i can be expressed as

$$\bigcup_{\forall x \in \mathbf{X}} [\underline{u}_i(x), \bar{u}_i(x)] = \bigcup_{\forall x \in \mathbf{X}} [\mu_i(x),\ \alpha \cdot \mu_i(x)],\ 0 < \alpha < 1. \tag{11}$$

For input data of dimensions two or higher, we can obtain the overall FOU by taking intersections of all upper and lower memberships for all features. If we choose the min operation as intersection, the FOU can be expressed as

$$\bigcup_{\forall x \in \mathbf{X}} [\underline{u}(x), \bar{u}(x)] = \bigcup_{\forall x \in \mathbf{X}} \left[\min_i\{\underline{u}_i(x_i)\},\ \min_i\{\bar{u}_i(x_i)\}\right]. \tag{12}$$

Figure 2 shows a simple example of triangular IT2 FMFs that are generated by the heuristic method for the two feature sample data sets in Fig. 2a. For the UMF, we select the height in (6) (parameter b) as the center location of the patterns of each class. Next, we compute the distance between the center and most distant pattern. The spread (distance between a and c) is selected by subtracting and adding this distance from the center. The LMF is obtained by adequately scaling the height of the UMF. Figure 2b and c shows the resulting UMF and LMF for class "×" and class "o", respectively. The region between the UMF and corresponding LMF indicates the FOU for each class. Unfortunately, the shapes of heuristic FMFs are not flexible enough to model all types of data. The heuristic method is summarized as follows.

Heuristic-Based IT2 FMF Generation Method

(1) Select a heuristic T1 FMF that is suitable for a given data set.

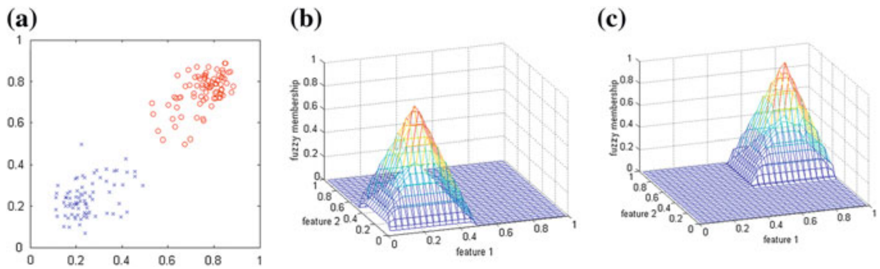


Fig. 2 Illustration of the heuristic-based IT2 FMF generation method using a triangular FMF for a given sample data set: (**a**) scatter plot, (**b**) UMF and LMF representing the FOU of class "×", and (**c**) UMF and LMF representing the FOU of class "o"

(2) Set the parameters for the membership function that is provided by an expert.
(3) Design the upper and lower membership functions using (11) and (12).

3.2 Histogram-Based Method

Histograms of features can be used to effectively describe the distribution of the feature values of the input data. Therefore, membership functions generated from histograms may be considered more suitable for arbitrary distributed data than from heuristics. We now present a method which designs IT2 FMFs by fitting parameterized functions to the histograms of the sample data. The method is described as follows.

3.2.1 Histogram Generation and Smoothing

First, the histogram of a given sample data for each labeled class is obtained. Next, the histograms are smoothed by sliding a symmetric window such as a hyper-cube or triangular window across the feature space and then normalized.

3.2.2 Polynomial Function Fitting

To extract the membership function, a suitable parameterized function is chosen to model the smoothed histograms. Approximate parameter values (e.g., the number of functions, height, and location of peaks) used to determine the optimal parameter values of the function can be obtained by polynomial function (PF) fitting. Using a least squares approximation, the PF of the lowest possible degree (i.e., to avoid over fitting) is chosen such that the fit to each smoothed histogram has a reasonably small error.

To illustrate the above procedure, we shall consider Gaussian functions as suitable parameterized functions to model the IT2 FMFs, and describe a method for estimating the parameters for a reasonable approximation [8].

3.2.3 Gaussian Function Modeling

Consider a parameterized function given by

$$G(\mathbf{x}) = a\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{13}$$

where a is the height, $\boldsymbol{\mu}$ is mean vector, and $\boldsymbol{\Sigma}$ is the covariance matrix. If the smoothed histogram has N significant peaks, we can model it as the sum of Gaussian functions by minimizing the objective function

$$J(\mathbf{p}) = \frac{1}{2}\left(\sum_{i=1}^{N} G_i(\mathbf{x}) - H(\mathbf{x})\right)^2, \tag{14}$$

where parameter vector $\mathbf{p}_i = (a_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is for the ith Gaussian function $G_i(\mathbf{x})$ and $H(\mathbf{x})$ is the smoothed histogram of the input data. The gradient descent method can be used to estimate the parameter vector $\mathbf{p}_i$ which minimizes (14) by the following update rule:

$$\mathbf{p}_i^{new} = \mathbf{p}_i^{old} - \rho\frac{\partial J}{\partial \mathbf{p}_i}, \tag{15}$$

where ρ is a positive learning constant.

For Gaussian functions, the partial derivatives of J with respect to each component of $\mathbf{p}_i$ are

$$\frac{\partial J}{\partial a_i} = \left(\sum_{j=1}^{N} G_j(\mathbf{x}) - H(\mathbf{x})\right) \cdot \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right), \tag{16}$$

$$\frac{\partial J}{\partial \boldsymbol{\mu}_i} = \frac{1}{2}\left(\sum_{j=1}^{N} G_j(\mathbf{x}) - H(\mathbf{x})\right) \cdot G_i(\mathbf{x}) \cdot (\mathbf{x}-\boldsymbol{\mu}_i)^T \cdot (\boldsymbol{\Sigma}_i^{-T} + \boldsymbol{\Sigma}_i^{-1}), \tag{17}$$

and

$$\frac{\partial J}{\partial \boldsymbol{\Sigma}_i} = \frac{1}{2}\left(\sum_{j=1}^{N} G_j(\mathbf{x}) - H(\mathbf{x})\right) \cdot G_i(\mathbf{x}) \cdot \left(\boldsymbol{\Sigma}_i^{-T}(\mathbf{x}-\boldsymbol{\mu}_i) \cdot (\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-T}\right). \tag{18}$$

For gradient descent methods, the choice of the initial values for the parameters is critical, due to the local minima problem. Therefore, for the case of Gaussian functions, we can use the following heuristic approach [8] that consists of 4 steps to obtain the initial parameters.

Step 1 Generate the smoothed histograms for the given sample data set as explained in Sect. 3.2.1.

Step 2 Using least squares approximation, fit a PF of the lowest possible degree (i.e., to avoid over fitting) such that the fit to each smoothed histogram has a reasonably small error.

Step 3 Calculate the extrema (maxima and minima) values for the PF in Step 2 and determine the number of Gaussians by the number of positive valued maxima, ignoring the ones that have small peaks.

Step 4 Initialize the heights of the Gaussians by the maxima values (peak values) and initialize the mean values of the Gaussians as the locations of these peaks. Initialize the standard deviation of each Gaussian as the shortest value among the distances between the mean of the Gaussian and the nearest minima or roots of the PF.

3.2.4 Upper and Lower Membership Function Modeling

To obtain the generated IT2 FMF, the FOU needs to be determined. The UMF and LMF are obtained by again fitting GFs to the smoothed histograms values that are above and below the fitted GF obtained previously.

As the final step, to obtain the FOU, the UMFs and LMFs are designed by using the upper and lower GFs as discussed above. The UMF can be obtained by normalizing the height to 1. Moreover, the height of the LMF is scaled in proportion to the UMF.

As the illustration in Fig. 3 shows, the smoothed histogram (using a 3-point triangular window) and resulting IT2 FMF is obtained by one-dimensional Gaussian function fitting for Feature 2 (y axis) of class "×" in Fig. 2a. The approximate parameters of the UMF and LMF are obtained after PF fitting of the smoothed histogram values that correspond to the upper and lower histogram

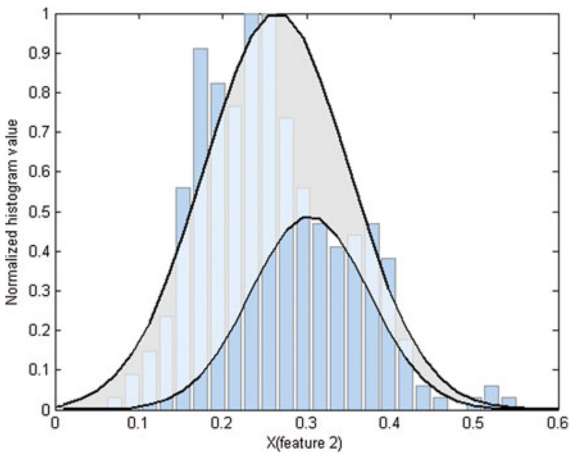


Fig. 3 Results of GF fitting for obtaining the UMF and LFM for feature 2 patterns labeled class "×" in Fig. 2a

values. The suitable degree of the PF is selected as the knee point of error. For this example, a third degree polynomial function is considered suitable. The resulting UMF and LMF are obtained by Gaussian function fitting the corresponding upper and lower histogram values. The shaded region between the UMF and LMF indicates the FOU. As shown in the figure, this method can effectively design IT2 FMFs based on the distribution of the input data.

Finally, Fig. 4 shows the results obtained by the histogram-based method for the sample data in Fig. 2a. Figure 4a and b shows the smoothed histograms for class "×" and class "o," respectively. As a result, Fig. 4c and d shows the FOU obtained for each class. The histogram-based method is summarized as follows:

Histogram-Based IT2 FMF Generation Method

(1) Construct and smooth the histogram of the sample data for each labeled class.
(2) Perform PF fitting to obtain the approximate parameter values (e.g., the number of functions, height, and location of peaks).
(3) Perform GF fitting using the values in step 2).

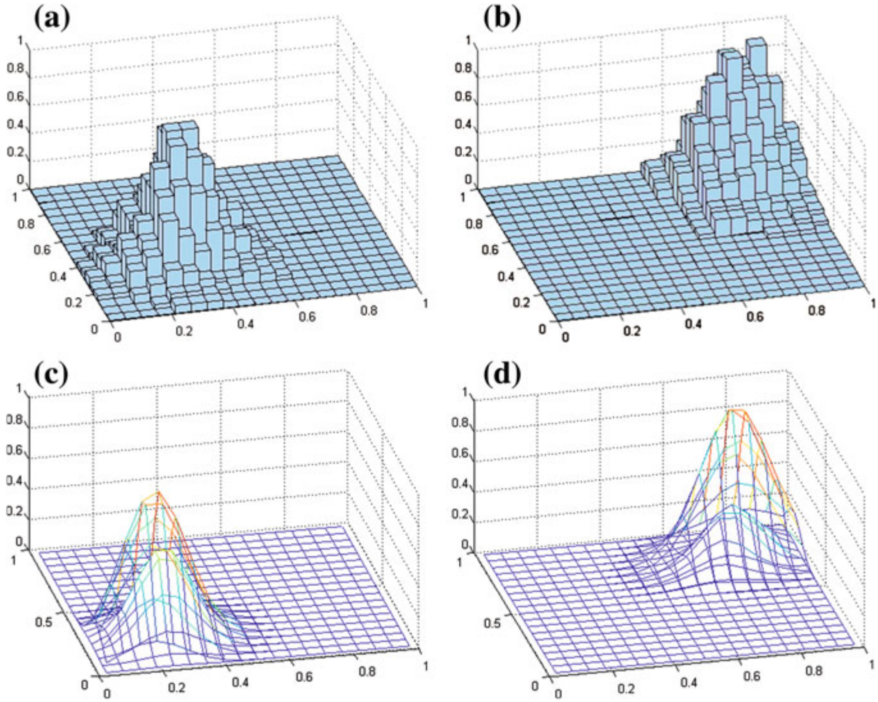


Fig. 4 Illustration of the histogram-based IT2 FMF generation method for the sample data in Fig. 2a: (**a**) smoothed histogram of class "×", (**b**) smoothed histogram of class "o", (**c**) UMF and LMF representing the FOU of class "×", and (**d**) UMF and LMF representing the FOU of class "o"

(4) Perform GF fitting for the upper and lower histogram values with respect to the GF in step 3).
(5) Determine UMF by normalizing the height of the upper GF and LMF by proportionally scaling the lower GF obtained in step 4).

For high-dimensional input data, the computational load can become undesirably high, due to the high-dimensional histogram smoothing process and fitting. As an alternative, first we find the one-dimensional UMF and LMF for each class label and feature by the histogram-based method. Next, we obtain the overall UMF and LMF by taking intersections of all UMF and LMF obtained for all features. If we use the min operation as our choice for intersection, the FOU can be expressed as

$$\left[f^L(\mathbf{x}), f^U(\mathbf{x})\right] = \left[\min_i\left\{f_i^L(\mathbf{x}_i)\right\}, \min_i\left\{f_i^U(\mathbf{x}_i)\right\}\right], \tag{19}$$

where f^{U} is the UMF, f^{L} is the LMF, and i is the feature number.

3.3 *Interval Type-2 Fuzzy C-Means Based Method*

The IT2 FCM algorithm was proposed to control the uncertainty of the fuzzifer m in FCM which affects the assignment of memberships for the patterns [18]. In this section, we present an IT2 FMF generation method based on the IT2 FCM algorithm. First, the IT2 FCM algorithm is briefly explained, and then the IT2 FCM-based method is described.

3.3.1 Interval Type-2 Fuzzy C-Means Algorithm

The FCM algorithm has been widely used for data partitioning [1]. FCM considers weight values (memberships) to control the contribution degree of a pattern in determining cluster prototypes. For example, a higher weighed pattern can play a more important role in determining the resulting cluster prototype. Thus, FCM can give more desirable cluster results than crisp C-means for pattern sets which contain overlapping clusters. The goal of FCM is to minimize the objective function

$$J(\mathbf{U}; \mathbf{X}, \mathbf{V}) = \sum_{j=1}^{C}\sum_{i=1}^{N} u_{ij}^m d_{ij}^2 \text{ subject to} \sum_{j=1}^{C} u_{ij} = 1, \tag{20}$$

where u_{ij} is the membership value of pattern $\mathbf{x}_i$ for cluster i, d_{ij}^2 is distance between $\mathbf{x}_i$ and the cluster prototype $\mathbf{v}_j$, and m represents the fuzzifier ($m > 1$). The partition matrix $\mathbf{U}$ represents the memberships for the patterns across each cluster having the elements u_{ij} and the matrix $\mathbf{V}$ is the collection of all cluster prototypes $\mathbf{v}_j$. The memberships and cluster prototypes that minimize the objective function in (20) can be obtained by

$$u_{ij} = \frac{\left(1/d_{ij}^2\right)^{1/(m-1)}}{\sum_{k=1}^{C}\left(1/d_{ik}^2\right)^{1/(m-1)}} \tag{21}$$

and

$$v_j = \frac{\sum_{k=1}^{N}\left(u_{kj}\right)^m x_k}{\sum_{k=1}^{N}\left(u_{kj}\right)^m}. \tag{22}$$

However, in FCM, fuzzifier m affects the membership assignment for the patterns. Suppose that there exist two cluster prototypes, namely centers $\mathbf{v}_1$ and $\mathbf{v}_2$, for a given pattern set. The membership plots corresponding to center $\mathbf{v}_1$ for patterns that lie between the two cluster centers for various values of fuzzifier m can be explained as follows. When $m \to 1$, the memberships are maximally crisp (hard), that is, patterns that are located just left (right) of the maximum fuzzy boundary are assigned full (0) membership for center $\mathbf{v}_1$ ($\mathbf{v}_2$). Conversely, when $m \to \infty$, the memberships are maximally fuzzy, which means that patterns that are only located at the centers are assigned full (0) membership, otherwise they are assigned memberships of 0.5. If the geometry of the clusters is of similar volume and density, change in fuzzifier m will not significantly affect the clustering result in FCM. This is illustrated in Fig. 5. As shown in the figure, the gray region indicates the maximum fuzzy region for various values of m. However, if there is a significant difference in density among clusters in a pattern set, the choice of m will give inconsistent clustering results in the FCM. The reason is that an impertinent establishment of maximum fuzzy boundary in FCM can provide undesirable updating of the cluster centers. This was previously emphasized in detail in [18]. A more desirable establishment of the maximum fuzzy region can allow for desirable clustering results in the FCM. Again, due to the constraint on the memberships we cannot design this region with any particular single value of fuzzifier m to be used in the FCM. However, if we can somehow simultaneously incorporate various values of fuzzifier m, we may perhaps be able to design a desirable maximum fuzzy region as reported in [18].

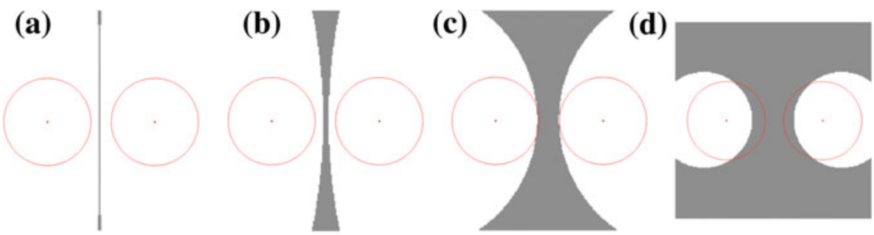


Fig. 5 Maximum fuzzy region (*gray area*) for various m: (**a**) 1.1, (**b**) 2.0, (**c**) 5.0, and (**d**) 10.0

To overcome this problem caused by the uncertainty of fuzzifier m, the IT2 FCM algorithm was proposed [18]. In IT2 FCM, the maximum fuzzy boundary can be controlled by incorporating two values of fuzzifier m. By doing so, the management of uncertainty can further improve clustering results obtained by FCM algorithm. The IT2 FMF $J_{x_i} = [\underline{u}(x_i),\ \bar{u}(x_i)]$ in IT2 FCM can be expressed as

$$\overline{u_j}(\mathbf{x}_i) = \begin{cases} \frac{1}{\sum_{k=1}^{C} \left(d_{ij}/d_{ik}\right)^{2/(m_1-1)}} & \text{if } \frac{1}{\sum_{k=1}^{C} \left(d_{ij}/d_{ik}\right)^{2/(m_1-1)}} > \frac{1}{\sum_{k=1}^{C} \left(d_{ij}/d_{ik}\right)^{2/(m_2-1)}} \\ \frac{1}{\sum_{k=1}^{C} \left(d_{ij}/d_{ik}\right)^{2/(m_2-1)}} & \text{otherwise} \end{cases} \tag{23}$$

and

$$\underline{u_j}(\mathbf{x}_i) = \begin{cases} \frac{1}{\sum_{k=1}^{C} \left(d_{ij}/d_{ik}\right)^{2/(m_1-1)}} & \text{if } \frac{1}{\sum_{k=1}^{C} \left(d_{ij}/d_{ik}\right)^{2/(m_1-1)}} \leq \frac{1}{\sum_{k=1}^{C} \left(d_{ij}/d_{ik}\right)^{2/(m_2-1)}} \\ \frac{1}{\sum_{k=1}^{C} \left(d_{ij}/d_{ik}\right)^{2/(m_2-1)}} & \text{otherwise} \end{cases}. \tag{24}$$

The procedure for updating cluster prototypes in IT2 FCM requires type-reduction and defuzzification using type-2 fuzzy operations. The generalized centroid (GC) type-reduction can be used as a type-reduction procedure. The centroid obtained by the type-reduction is shown as the following interval:

$$\mathbf{v}_{\tilde{\mathbf{x}}} = [\mathbf{v}_L, \mathbf{v}_R] = \sum_{u(x_1)\in I_{x_1}} \cdots \sum_{u(x_1)\in I_{x_1}} 1 \Bigg/ \frac{\sum_{i=1}^{N} \mathbf{x}_i u(\mathbf{x}_i)^m}{\sum_{i=1}^{N} u(\mathbf{x}_i)^m}. \tag{25}$$

The crisp center can be simply obtained by defuzzification as

$$\mathbf{v}_j = \frac{v_L + v_R}{2}. \tag{26}$$

3.3.2 Membership Generation Method Using Interval Type-2 Fuzzy C-Means Algorithm

For this method, we use (23) and (24) to generate IT2 FMF of patterns for each class. For pattern data that are labeled, it is not necessary to use unsupervised clustering as in FCM. Instead, we should perform "supervised" FCM clustering on each class separately. This is performed as follows:

Let the pth prototype obtained by FCM in class k be denoted by $\mathbf{v}_p^k$. Next, the minimum distance between the prototypes and input pattern $\mathbf{x}_j$ is

$$d_{kj}^* = \min_p \left\{ d\left(\mathbf{x}_j, \mathbf{v}_p^k\right) \right\},\ p \in \{n_k\}\ , \tag{27}$$

where n_k is the number of prototypes used for class k. The UMF and LMF for class k and input pattern $\mathbf{x}_j$ can be expressed by modifying (23) and (24) by replacing d_{ij} and d_{ik} with d_{ij}^* and d_{ik}^*, respectively.

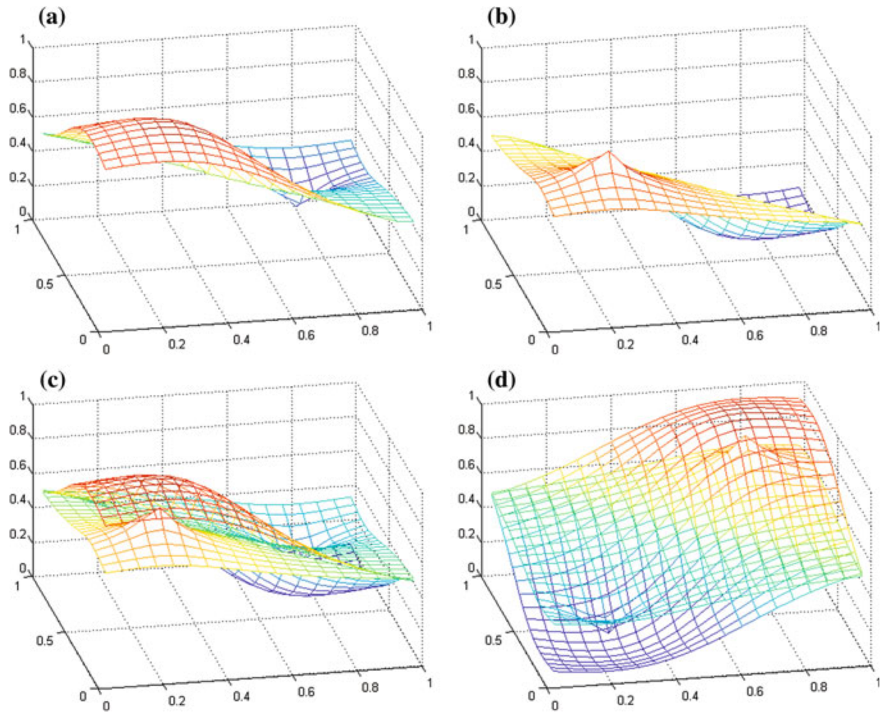


Fig. 6 Illustration of the IT2 FCM ($m_1 = 2$, $m_2 = 5$) based IT2 FMF generation method for the sample data in Fig. 2a: (**a**) UMF for class "×", (**b**) LMF for class "×", (**c**) FOU of class "×", and (**d**) FOU of class "o"

Figure 6 shows the IT2 FMF obtained by the IT2 FCM-based method for the same sample data in Fig. 2a. Fuzzifiers m_1 and m_2 are selected as 2 and 5 and the number of prototypes is chosen as one for each class. Figure 6a and b shows the generated UMF and LMF for class "×". As indicated, the membership values for the UMFs based on the lower fuzzifier value m_1 are selected when the memberships are above 0.5. On the contrary, the membership values based on the higher fuzzifier value m_2 are selected when the memberships are below 0.5. Figure 6c and d shows the FOU of the IT2 FMF for each labeled class. It is to be noted that the selection of fuzzifier values may affect the formation of the FOU.

The IT2 FCM-based membership generation method can desirably control the uncertainty of fuzzifier m for pattern distributions of different structure and density as in IT2 FCM. Additionally, the method is quite simple and can be used for all features simultaneously as in the case of high-dimensional input data. The method is summarized as follows:

IT2 FCM-Based IT2 FMF Generation Method

(1) Select fuzzifiers m_1 and m_2.
(2) Perform IT2 FCM on each class separately.
(3) Find the minimum distance between prototypes of each class and input patterns.
(4) Design the UMF and LMF using (23) and (24).

4 Application to Back-Propagation Neural Networks

To evaluate the performance of the IT2 FMF generation methods, we apply them to BPNNs [25]. In doing so, we extract T1 fuzzy membership values from the centroid of each IT2 FMF obtained by the methods. The centroid is obtained by performing type reduction. The type reduction procedure gives the GC for the IT2 FMF. The GC is an IT1 FS for the centroid of embedded T2 FSs and is expressed as

$$GC = [c_l, c_r] = \int_{x_1 \in X} \cdots \int_{x_N \in X} \cdots \int_{\mu_i \in M_1} \cdots \int_{\mu_N \in M_N} 1 \Big/ \frac{\sum_{i=1}^{N} x_i \mu_i}{\sum_{i=1}^{N} \mu_i}, \tag{28}$$

where x_i is input feature pattern and μ_i is the primary membership for x_i. This GC can be obtained using the Karnik-Mendel (KM) *iterative procedure* [15]. Next, the fuzzy membership value of input pattern $\mathbf{x}$ for class j is obtained according to the distance between the centroid and input patterns as

$$f_j(\mathbf{x}) = \max_{\forall k} \left\{ 1 - d\left(\mathbf{x}, C_j^k\right) \right\}, \tag{29}$$

where C_j^k denotes the kth centroid of class j and $d(\cdot)$ is distance between the pattern and the centroid. The distance measure can be described as

$$d\left(\mathbf{x}, C_j^k\right) = \begin{cases} 0 & \text{if } c_{jl}^k \le x_i \le c_{jr}^k \text{ for all } i \\ \sqrt{\sum_{i=1}^{M} \left(\min\left\{ \left| x_i - c_{jl}^k \right|, \left| x_i - c_{jr}^k \right| \right\} \right)^2} & \text{otherwise} \end{cases}, \tag{30}$$

where M is the number of features, c_{jl}^k (c_{jr}^k) is the left (right) value of the interval centroid C_j^k. The equation above is used to compute the distance between input pattern and interval centroid using the minimum values among the distances between input pattern and left and right values of the interval centroid for all features. The fuzzy memberships extracted from the IT2 FMF can be considered suitable in describing the class memberships of patterns, since the uncertainty of pattern data are desirably controlled by the IT2 FMF. The extracted fuzzy membership values of the patterns are used as input training data to the BPNN.

5 Experimental Results

To illustrate the performance of the three IT2 FMF generation methods, we apply them to BPNNs and the classification results are given for several data sets. We first obtain the centroid from the generated IT2 FMF by performing type-reduction using the KM iterative procedure as mentioned in the Sect. 4. Then, fuzzy membership values for patterns are extracted according to the distances between the centroids and patterns using (29) and (30). These fuzzy membership values are then used as inputs to the BPNN. The structure of the BPNN consists of a fully connected three-layered network. The classification results are reported by varying the number of neurons in the hidden layer fromone to ten for the three generation methods for several pattern sets. In addition, segmentation results for real images are also given. As a comparison, results are given for inputs to the BPNN that are from T1 FMF generation [7, 8, 21]. As in the generation of IT2 FMFs, fuzzy membership values for the patterns are extracted according to the distances between the centroids and patterns using (29) and (30), where in this case the centriod C_j^k (i.e., $c_{jl}^k = c_{jr}^k$) is simply considered as the peak (instead of interval) of the generated T1 FMF.

5.1 *"T-shape" Data Set*

The "T-shape" data set consists of 447 patterns (228 and 219 patterns for each class) of two features and two classes. We use one pattern for testing and the remaining patterns for training. This procedure was repeated for all patterns. Figure 7a shows the scatter plot of the data set. We assume that the patterns located outside of the shaded region do not have uncertainty. Therefore, we exclude those patterns (232 patterns) when comparing the classification results. The BPNNs are trained using all of the sample patterns, however, classification results are reported for only the patterns in the shaded region (215 patterns). Figure 7b–g shows the IT2 FMFs obtained by the three methods. As shown in Fig. 7b and c, the heuristic IT2 FMF was designed based on a triangular function. Figure 7d and e shows the membership designed by the histogram-based method. As shown in the figure, 2-D GF fitting was used. Figure 7f and g displays the memberships designed by the IT2 FC-based method. Experiments for several possible combinations of fuzzifiers m_1 and m_2 were performed and the combination $m_1 = 2$ and $m_2 = 5$ gave the best results. The IT2 FMF generation methods gave improved classification results (i.e., patterns in the shaded region) compared with the generated T1 FMF. Average improvement in classification resulted in about 0.6 % for the heuristic method, 2.1 % for the histogram-based method, and 1.7 % for the IT2 FCM-based method.

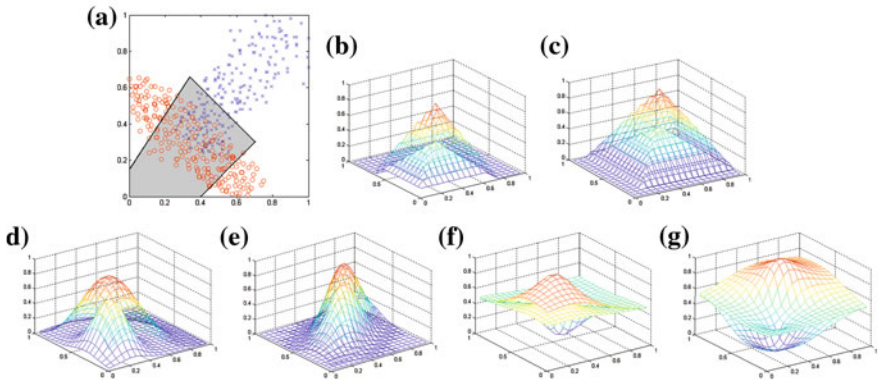


Fig. 7 IT2 FMFs generated for the "T-shaped" data set: (**a**) scatter plot, heuristic method for (**b**) class "o", and (**c**) class "×", histogram-based method for (**d**) class "o", and (**e**) class "×," and IT2 FCM-based method for (**f**) class "o" and (**g**) class "×" ($m_1 = 2$, $m_2 = 5$)

5.2 "Pima Indian Diabetes" Data Set

We now give classification results for a high-dimensional data set, namely "Pima Indian diabetes" data set. The data set consists of 768 patterns (500 and 268 patterns for each class) of eight features and two classes. The classification results are reported as follows. For the histogram-based method, due to the difficulty of processing the eight-dimensional histogram and Gaussian function fitting, we generated the IT2 FMF by (19) using a one-dimensional histogram and GF fitting for each feature. As in the above example, the IT2 FMF generation methods gave significantly improved classification results compared with the generated T1 FMF. Average improvement in classification resulted in about 0.92 % for the heuristic method, 2.16 % for the histogram-based method, and 1.5 % for the IT2 FCM-based method.

5.3 Real Image Segmentation

For the last example, we give segmentation results for 200 × 200 real scene images. The sample image consists of three regions namely, road, forest, and sky. The gray levels of median and excess green filtered images were used as feature values. We randomly selected 100 pixels in each region of the image, and obtained the sample patterns as the feature values of the selected pixels. Figure 8a shows the sample image. Next, BPNNs were trained the same as in the previous examples using the sample data. Then, all the pixels in the image were classified into three classes by the trained networks. For this example, we give segmentation results for

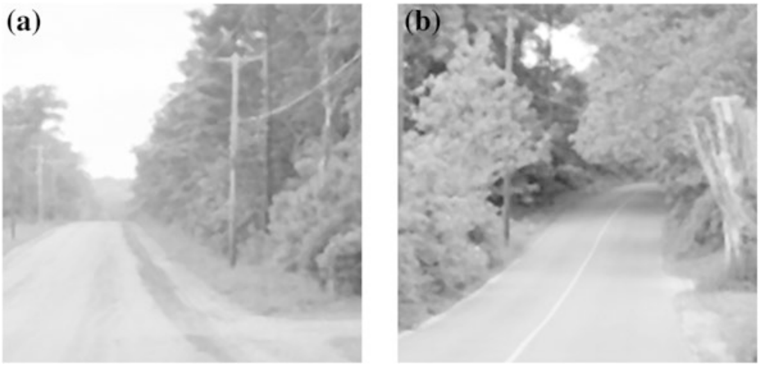


Fig. 8 Natural scenes used for training and testing: (**a**) scene 1 training image, (**b**) scene 2 testing image

the case of two hidden neurons. The results for networks with higher number of hidden neurons gave similar results.

5.3.1 Inference Results

We present inference (segmentation) results involving one training image (see Fig. 8a) and one test image (see Fig. 8b). Figure 9 shows the segmentation results for the test image. For all three IT2 FMF methods, one can visually see that the forest region was better classified than with the T1 FMF method.

6 Conclusions

In this chapter, we presented three IT2 FMF generation methods that were obtained from sample data. The heuristic method generated IT2 FMFs by simply incorporating heuristic type-1 FMFs provided by experts. The FOU was designed by assigning the UMF with the heuristic T1 FMF, and adequately scaling the UMF for the LMF. However, since the shapes of the heuristic FMF are not flexible, the obtained IT2 FMFs may not sufficiently represent the uncertainty for data that are complex.

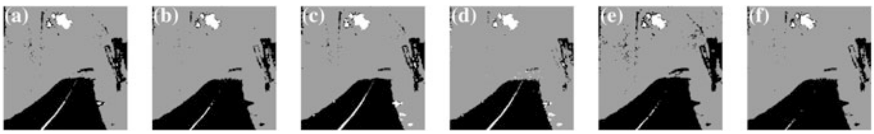


Fig. 9 Segmentation results for scene 2: heuristic-based method using (**a**) T1 FMF and (**b**) IT2 FMF, histogram-based method using (**c**) T1 FMF and (**d**) IT2 FMF, and IT2 FCM-based method using (**e**) T1 FMF and (**f**) IT2 FMF

In the histogram-based method, the FOU was designed by fitting Gaussian functions to the smoothed histograms of data. The UMF and LMF obtained by the histogram-based method have shown to effectively represent the distribution of patterns better than the heuristic method. In the IT2 FCM-based method, the FOU was designed using the UMF and LMF equations in the IT2 FCM. As with IT2 FCM, the membership generation method can control the uncertainty of the fuzzifer m in FCM which undesirably affects the fuzzy memberships when the clusters are significantly different in size and density. This suggests that the IT2 FCM based method can represent the uncertainty of fuzzifier m induced by arbitrary structures and densities of sample data.

Finally, we incorporated the IT2 FMF generation methods into the BPNN, where T1 fuzzy membership values were computed from the centroids of the IT2 FMFs. Then, the T1 fuzzy membership values representing the sample data were used as inputs to the BPNN. The membership assignment showed to improve the classification performance of the BPNN since the uncertainty of pattern data has been desirably controlled by the IT2 FMFs. Experimental results show that the IT2 FMF generation methods can effectively model the uncertainty of pattern data. Other methods of representing FMFs and the integration of IT2 FMFs into other types of neural networks may also be considered.

To comment on the possible future studies, we presented three IT2 FMF generation methods to represent the uncertainty of sample data. We believe that the methods may be considered to be early attempts for effectively generating IT2 FMFs automatically from pattern sets. However, there exist areas of improvement as described in the following.

In the heuristic method, we need to select the proper heuristic T1 FMF for the sample data and optimally find the parameters of the selected heuristic T1 FMF. The proper shape and parameters of the heuristic T1 FMF can be critical for properly designing the IT2 FMF. Moreover, the scaling factor α which determines the height of the LMF needs to be carefully selected, since the FOU is dependent on the scaling factor. If the value of α is poorly chosen, the IT2 FMF may not suitably represent the uncertainty of the sample data.

In the histogram-based method, we used symmetric Gaussian function fitting to the smoothed histograms of the sample data. However, non-symmetric Gaussian function fitting could be used to represent the distribution of sample data more effectively. Non-symmetric Gaussian functions can model the smoothed histograms with more flexibility since the symmetry constraint has been relaxed.

In the IT2 FCM-based method, the selection of the fuzzifier m_1 and m_2 is very important in designing the FOU for the sample data distribution. In general, if we select unsuitable fuzzifier m_1 and m_2, IT2 FCM can yield poor clustering results compared with FCM. For this reasoning, this method may not suitably model the uncertainty of the sample data as well. Therefore, the selection of m_1 and m_2 is an important research area to which various methods can be applied (e.g., neural networks and genetic algorithms).

As a final note, the methods presented can be used as major components of various applications in pattern recognition and computer vision. For example, they may be applied to the modeling of objects in images for object detection. Our studies show that applying generated IT2 FMFs into various fuzzy classification systems can improve their performance.

Acknowledgments This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (Grant number 2011-0025655).

References

1. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
2. Dave, R.: Fuzzy-shell clustering and applications to circle detection in digital images. Int. J. Gen Syst **16**, 343–355 (1990)
3. Keller, J., Gray, M., Givens, J.: A fuzzy K-nearest neighbor algorithm. IEEE Trans. Syst. Man Cybern. **30**, 258–263 (1985)
4. Keller, J., Hunt, D.: Incorporating fuzzy membership function into the perceptron algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **7**, 693–699 (1985)
5. Liang, Q., Mendel, J.: MPEG VBR video traffic modeling and classification using fuzzy techniques. IEEE Trans. Fuzzy Syst. **9**, 183–193 (2000)
6. Makrehchi, M., Basir, O., Karnel, M.: Generation of fuzzy membership function using information theory measures and genetic algorithm. Lect. Notes Comput. Sci. **275**, 603–610 (2003)
7. Medasani, S., Kim, J., Krishnapuram, R.: An overview of membership function generation techniques for pattern recognition. Int. J. Approximate Reasoning **19**, 391–417 (1998)
8. Rhee, F., Krishnapuram, R.: Fuzzy rule generation methods for high-level computer vision. Fuzzy Sets Syst. **60**, 245–258 (1993)
9. Wang, S.: Generating fuzzy membership functions: a monotonic neural network. Fuzzy Sets Syst. **61**, 71–81 (1994)
10. Yang, C., Bose, N.: Generating fuzzy membership function with self-organizing feature map. Pattern Recogn. Lett. **27**, 356–365 (2006)
11. Hagras, H.: A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Trans. Fuzzy Syst. **12**, 524–539 (2004)
12. John, R., Innocent, P., Barnes, M.: Neuro-fuzzy clustering of radiographic tiba image data using type-2 fuzzy sets. Inf. Sci. **125**, 65–82 (2000)
13. Karnik, N., Mendel, J.: Applications of type-2 fuzzy logic systems to forecasting of time series. Inf. Sci. **120**, 89–111 (1999)
14. Liang, Q., Karnik, N., Mendel, J.: Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **30**, 329–339 (2000)
15. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall, Upper Saddle River (2001)
16. Choi, B., Rhee, F.: Interval type-2 fuzzy membership function generation methods for pattern recognition. Inf. Sci. **179**, 2102–2122 (2009)
17. Hwang, C., Rhee, F.: An interval type-2 fuzzy C spherical shells algorithm. In: Proceedings of the IEEE International Conference on Fuzzy System, pp. 1117–1122 (2004)
18. Hwang, C., Rhee, F.: Uncertain fuzzy clustering: interval type-2 fuzzy approach to C-means. IEEE Trans. Fuzzy Syst. **15**, 107–120 (2007)
19. Liang, Q., Mendel, J.: Interval type-2 fuzzy logic systems: theory and design. IEEE Trans. Fuzzy Syst. **8**, 535–550 (2000)

20. Rhee, F.: Uncertain fuzzy clustering: insights and recommendations. IEEE Comput. Intell. Mag. **2**, 44–56 (2007)
21. Rhee, F., Choi, B.: Interval type-2 fuzzy membership function design and its application to radial basis function neural networks. In: Proceedings of the IEEE International Conference on Fuzzy System, pp. 2047–2052 (2007)
22. Rhee, F., Hwang, C.: An interval type-2 fuzzy perceptron. In: Proceedings of the IEEE International Conference on Fuzzy System, pp. 1331–1335 (2002)
23. Rhee, F., Hwang, C.: An interval type-2 fuzzy K-nearest neighbor. In: Proceedings of the IEEE International Conference on Fuzzy System, pp. 802–807 (2003)
24. Wu, H., Mendel, J.: Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **10**, 622–639 (2002)
25. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Englewood Cliffs (1999)

Part III
Applications

Type-2 Fuzzy Logic in Image Analysis and Pattern Recognition

Patricia Melin and Oscar Castillo

Abstract Interval type-2 fuzzy systems can be of great help in image analysis and pattern recognition applications. In particular, edge detection is a process usually applied to image sets before the training phase in recognition systems. This preprocessing step helps to extract the most important shapes in an image, ignoring the homogeneous regions and remarking the real objective to classify or recognize. Many traditional and fuzzy edge detectors can be used, but it is difficult to demonstrate which ones are better before the recognition results are obtained. In this work we show experimental results, where several edge detectors were used to preprocess the same image sets. Each resulting image set was used as training data for a neural network recognition system, and the recognition rates were compared. The goal of these experiments is to find the better edge detector that can be used to improve the training data of a neural network for an image recognition system.

1 Introduction

In previous work, we have proposed extensions to traditional edge detectors to improve their performance by using fuzzy systems [1–3]. The performed experiments have shown that the resulting images obtained with fuzzy edge detectors were visually better than the ones obtained with the traditional methods.

There is still work to be done on developing formal validation metrics for fuzzy edge detectors. In the literature we can find comparisons of edge detectors based

P. Melin (✉) · O. Castillo
Division of Graduate Studies and Research, Tijuana Institute of Technology, Tijuana, Mexico
e-mail: pmelin@tectijuana.mx

O. Castillo
e-mail: ocastillo@tectijuana.mx

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*, Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_12,

on human observations [4–8], and some others that found the optimal values for parametric edge detectors [9].

Edge detectors can be used in recognition systems for different purposes, but in this work we are particularly interested in knowing, which is the best edge detector for a neural recognition system. In this chapter we present some experiments that show that fuzzy edge detectors are a good method to improve the performance of neural recognition systems, and for this reason we propose that the recognition rate of the neural networks can be used as an edge detection performance index.

The rest of the chapter is organized as follows. Section 2 presents an overview of fuzzy edge detectors. Section 3 describes the experimental setup used to test the proposed fuzzy edge detectors in a neural recognition system. Section 4 presents the experimental results achieved with the proposed fuzzy edge detectors. Finally, Section 5 outlines the conclusions and future work.

2 Overview of Fuzzy Edge Detectors

In this section an overview of the previously proposed fuzzy edge detectors is presented. First, the Sobel edge detector improved with fuzzy logic is presented. Second, the morphological gradient edge detector enhanced with fuzzy logic is also presented.

2.1 Sobel Edge Detector Improved with Fuzzy Logic

In the Sobel fuzzy edge detector we used the individual operators $Sobel_x$ and $Sobel_y$ as in the traditional method, and then we substitute the Euclidean distance of Eq. (1) by a fuzzy system, as we show in Fig. 1 [3].

$$Sobel\_edges = \sqrt{Sobel_x^2 + Sobel_y^2} \quad (1)$$

The individual Sobel operators are the main inputs to the type-1 fuzzy inference system (FIS1) and type-2 fuzzy inference system (FIS2), and we have also considered adding two more inputs, which are filters that improve the final edge image. The fuzzy variables used in the Sobel+FIS1 and Sobel+FIS2 edge detectors are shown in Figs. 2 and 3 respectively.

The use of the FIS2 [10, 11] provided images with better defined edges than the FIS1, which is a very important result in providing better inputs to the neural networks that will perform the recognition task.

The fuzzy rules for both the FIS1 and FIS2 are the same and are shown below:

1. If (dh is LOW) and (dv is LOW) then (y1 is HIGH)
2. If (dh is MIDDLE) and (dv is MIDDLE) then (y1 is LOW)

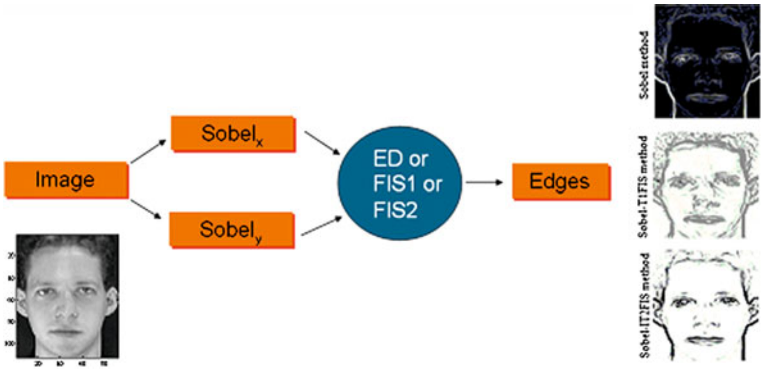


Fig. 1 Sobel edge detector improved with fuzzy logic

3. If (dh is HIGH) and (dv is HIGH) then (y1 is LOW)
4. If (dh is MIDDLE) and (hp is LOW) then (y1 is LOW)
5. If (dv is MIDDLE) and (hp is LOW) then (y1 is LOW)
6. If (m is LOW) and (dv is MIDDLE) then (y1 is HIGH)
7. If (m is LOW) and (dh is MIDDLE) then (y1 is HIGH).

The fuzzy rule base shown above infers the gray tone of each pixel for the edge image with the following reasoning: When the horizontal gradient d_h and vertical gradient d_v are LOW it means that there is not enough difference between the gray tones in its neighbors pixels, and hence the output pixel must belong to a homogeneous or not edges region, then the output pixel is HIGH or near WHITE. In the opposite case, when d_h and d_v are both HIGH this means that there is enough difference between the gray tones in its neighborhood, then the output pixel is an EDGE.

2.2 *Morphological Gradient Edge Detector Improved with Fuzzy Logic*

In the morphological gradient, we calculated the four gradients as in the traditional method [12, 13], and substitute the sum of gradients in Eq. (2) with a fuzzy inference system, as we show in Fig. 4.

$$MG\_edges = D_1 + D_2 + D_3 + D_4 \quad (2)$$

The linguistic variables used in the MG+FIS1 and MG+FIS2 edges detectors are shown in Figs. 5 and 6 respectively.

The rules for both the FIS1 and FIS2 are the same and are shown below:

1. If (D1 is HIGH) or (D2 is HIGH) or (D3 is HIGH) or (D4 is HIGH) then (E is BLACK)

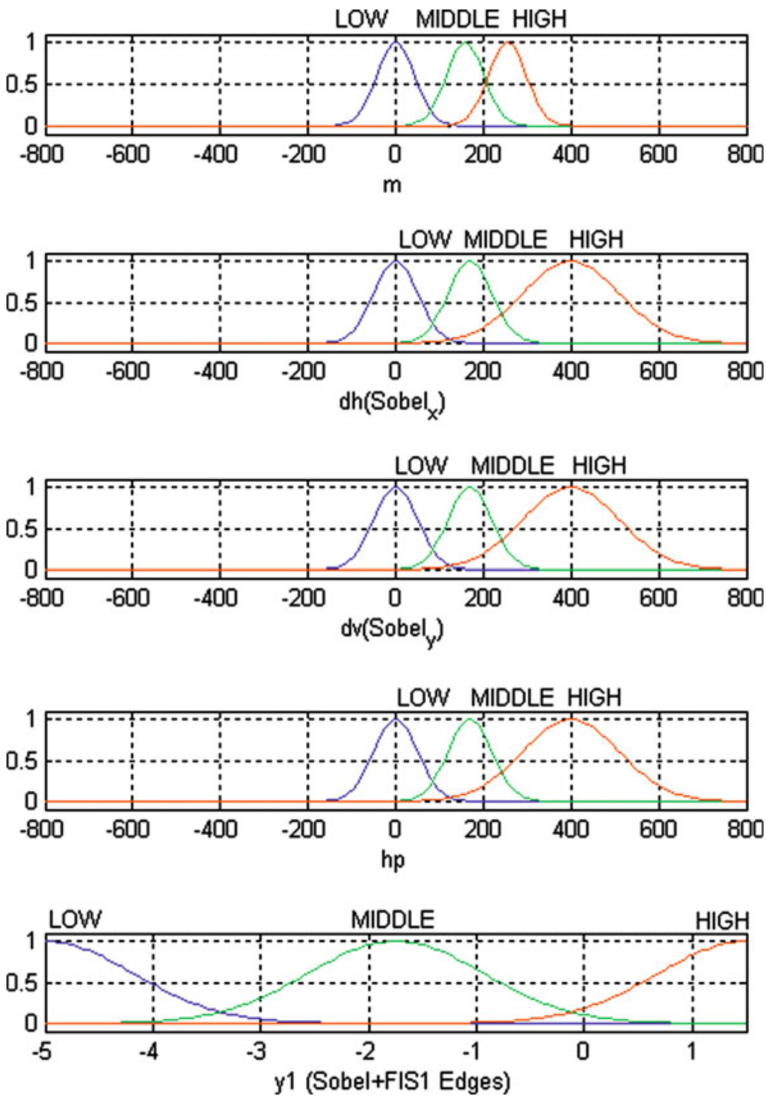


Fig. 2 Membership functions of the variables for the Sobel + FIS1 edge detector

2. If (D1 is MIDDLE) or (D2 is MIDDLE) or (D3 is MIDDLE) or (D4 is MIDDLE) then (E is GRAY)
3. If (D1 is LOW) and (D2 is LOW) and (D3 is LOW) and (D4 is LOW) then (E is WHITE)

After many experiments we found that an edge exists when any gradient D_i is HIGH, which means that a difference of gray tones in any direction of the image must produce a pixel with a BLACK value or EDGE. The same behavior occurs

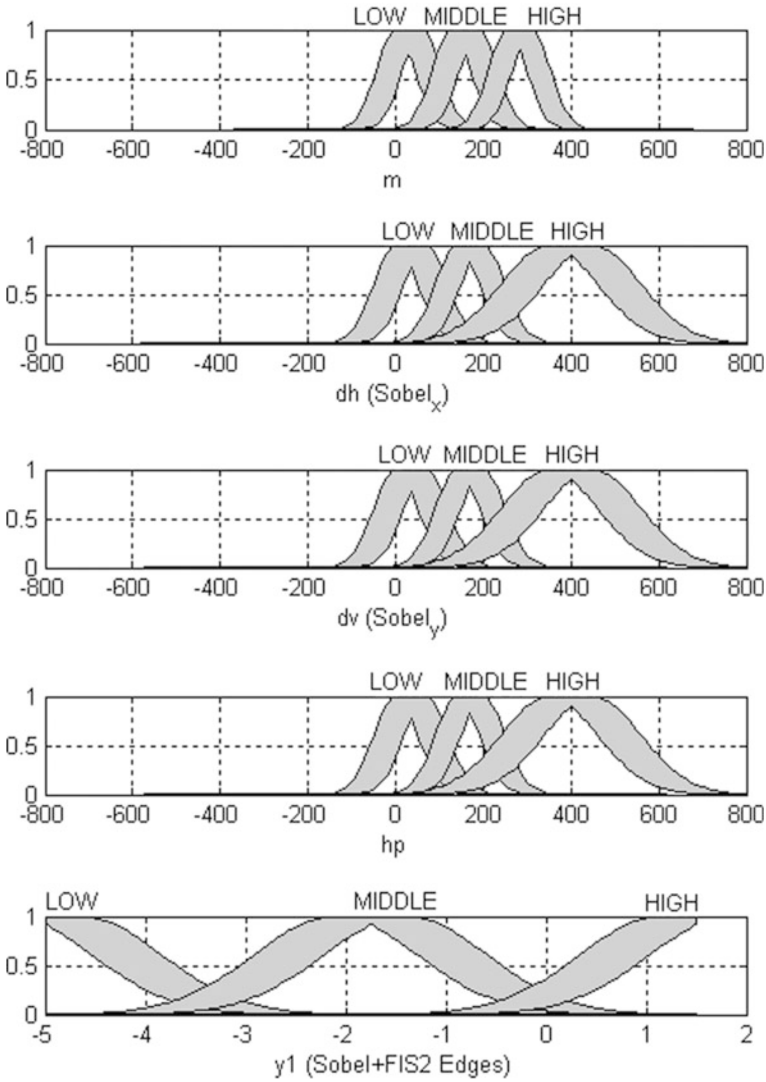


Fig. 3 Membership functions of the variables for the Sobel + FIS2 edge detector

when any gradient D_i is MIDDLE, which means that even when the differences in the gray tones are not maximal, the pixel is an EDGE, then the only rule that found a non-edge pixel is the number 3, where only when all the gradients are LOW, the output pixel is WHITE, which means a pixel belonging to a homogeneous region.

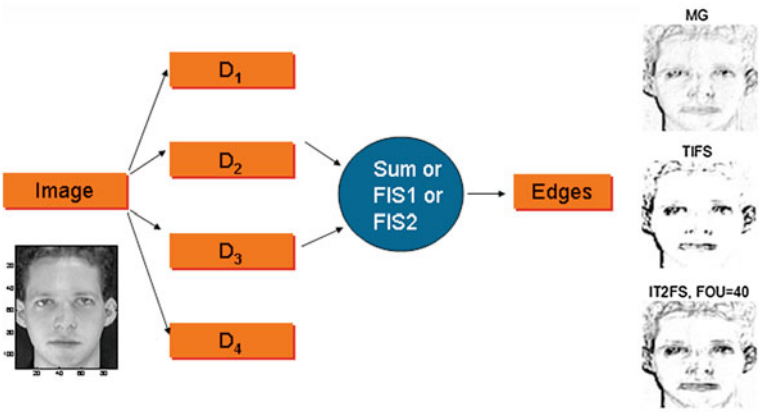


Fig. 4 Morphological gradient edge detector improved with fuzzy systems

3 Experimental Setup

The experiment consists on applying a neural recognition system using each of the previously presented edge detectors: Sobel, Sobel+FIS1, Sobel+FIS2, Morphological Gradient (MG), Morphological Gradient+FIS1, and Morphological Gradient+FIS2 and then comparing the results.

3.1 General Algorithm Used for the Experiments

1. Define the database folder.
2. Define the edge detector.
3. Detect the edges of each image as a vector and store it as a column in a matrix.
4. Calculate the recognition rate using the k-fold cross validation method.
    a. Calculate the indices for training and test k folds.
    b. Train the neural network $k - 1$ times, one for each training fold calculated previously.
    c. Test the neural network k times, one for each fold test set calculated previously.
5. Calculate the mean rate for all the k-folds.

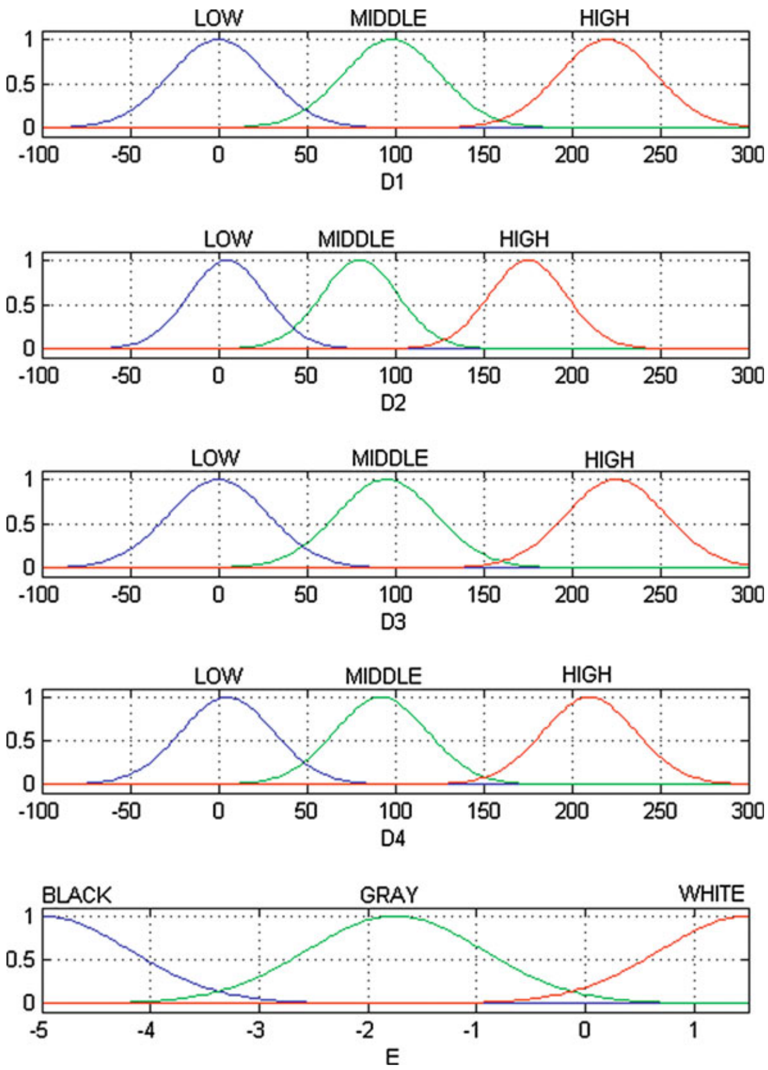


Fig. 5 Membership functions of the variables for the MG + FIS1 Edge Detector

3.2 Parameters for the Image Databases

The experiments can be performed with benchmark image databases used for identification purposes. This is the case of face recognition applications, then we use three of the most popular benchmark sets of images, the ORL face database [14], the Cropped Yale face database [15, 16], and the FERET face database [17].

For the three databases we defined the variable *p* as the person number and *s* as number of samples for each person. The tests were made with *k*-fold cross

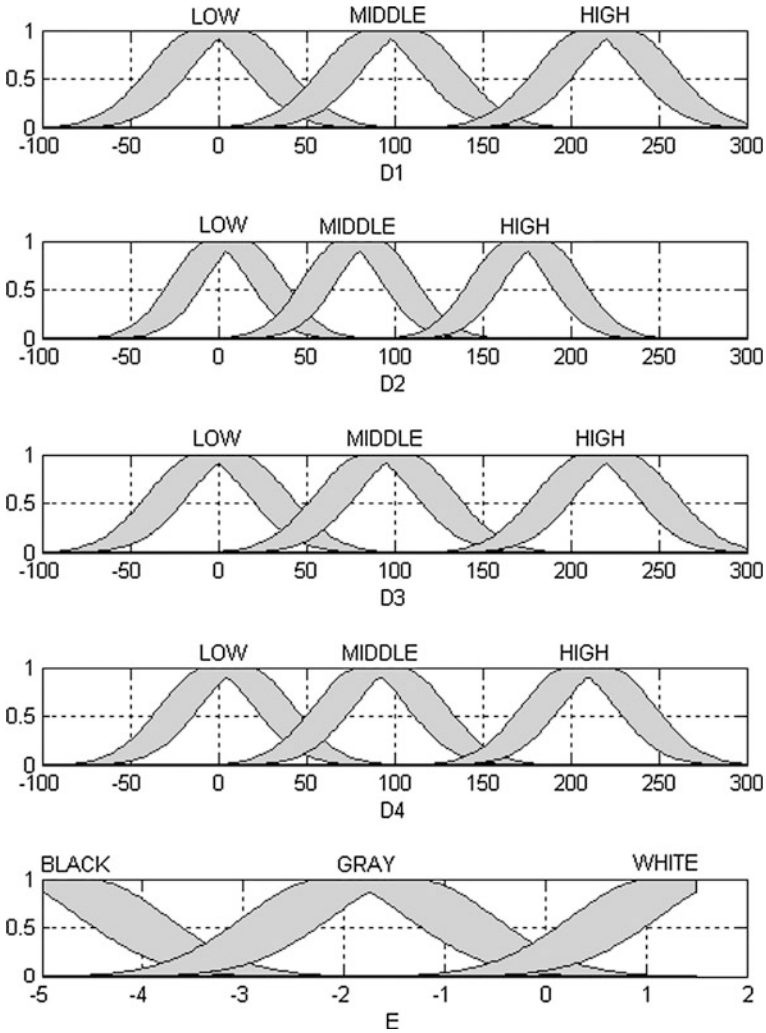


Fig. 6 Membership functions of the variables for the MG + FIS2 Edge Detector

validation method, with $k = 5$ for the three databases. We can generalize the calculation of fold size m or number of samples in each fold, dividing the total number of samples for each person s by the fold number, and then multiplying the result by the person number p (3), then the train data set size i (4) can be calculated as the number of samples in $k-1$ folds m, and test data set size t (5) are the number of samples in only one fold.

$$m = (s/k) * p \tag{3}$$

$$i = m(k - 1) \tag{4}$$

Table 1 Particular information for the tested benchmark face databases

Database	Person number (p)	Samples number (s)	Fold size (m)	Training set size (i)	Test set size (t)
ORL	40	10	80	320	80
Cropped Yale	38	10	76	304	76
FERET	74	4	74	222	74

$$t = m \tag{5}$$

The total number of samples used for each person were 10 for the ORL and YALE databases; then if the size m of each 5-fold is 2, the number of samples for training for each person is 8 and for testing is 2. For experiments with the FERET face database we use only the samples of 74 persons who have 4 frontal sample images. The particular information for each database is shown in Table 1.

3.3 The Monolithic Neural Network

In previous experiments with neural networks for image recognition, we have found a general structure with acceptable performance, even if it is not optimized. We used the same structure for multi-net modular neural networks, in order to establish a standard for comparison for all the experiments [18–23]. The general structure for the monolithic neural network is indicated below:

- Two hidden layers with 200 neurons.
- Learning Algorithm: Gradient descent with momentum and adaptive learning rate back-propagation.
- Error goal of 0.0001.

4 Experimental Results

In this section we show the numerical results of the experiments. Table 2 contains the results for the ORL face database, Table 3 contains the results for the Cropped Yale database and Table 4 contains the results for the FERET face database.

It can be noticed that mean times in Table 2 are the same and this is true because the ORL database is very uniform, in other words the figures (faces) do not have much diversity. The standard deviations appear to be relatively high, but considering that the neural networks are trained with different initial weights is not a concern. Table 3 shows larger standard deviations on the results due to the fact that the Yale database is more complicated for recognition on average and this causes some neural networks to produce sometimes bad results, although it is also possible to find very good solutions.

Table 2 Recognition rates for the ORL face database

Training set pre-processing method	Mean time (s)	Mean rate (%)	Standard deviation	Max rate (%)
MG+FIS1	1.2694	89.25	4.47	95.00
MG+FIS2	1.2694	90.25	5.48	97.50
Sobel+FIS1	1.2694	87.25	3.69	91.25
Sobel+FIS2	1.2694	90.75	4.29	95.00

Table 3 Recognition rates for the Cropped Yale face database

Training set pre-processing method	Mean time (*s*)	Mean rate (%)	Standard deviation	Max rate (%)
MG+FIS1	1.76	68.42	29.11	100
MG+FIS2	1.07	88.16	21.09	100
Sobel+FIS1	1.17	79.47	26.33	100
Sobel+FIS2	1.1321	90	22.36	100

Table 4 Recognition rates for the FERET face database

Training set pre-processing method	Mean time (*s*)	Mean rate (%)	Standard deviation	Max rate (%)
MG+FIS1	1.17	75.34	5.45	79.73
MG+FIS2	1.17	72.30	6.85	82.43
Sobel+FIS1	1.17	82.77	00.68	83.78
Sobel+FIS2	1.17	84.46	03.22	87.84

For a better appreciation of the results we made plots for the values presented in Tables 2, 3 and 4. Even if this work does not pretend to make a comparison based on the training times as performance index for the edge detectors, it is interesting to note that the necessary time to reach the error goal is established for each experiment.

As we can see in Fig. 7 the lowest training times are for the Morphological Gradient+FIS2 edge detector and Sobel+FIS2 edge detector. That is because both edge detectors were improved with interval type-2 fuzzy systems and produce images with more homogeneous areas; which means a high frequency of pixels near the WHITE linguistic values.

However, the main advantage of the interval type-2 edges detectors are the recognition rates plotted in Fig. 8, where we can notice that the best mean performance of the neural network was achieved when it was trained with the data sets obtained with the MG+FIS2 and Sobel+FIS2 edge detectors.

Figure 9 shows that the recognition rates are also better for the edge detectors improved with interval type-2 fuzzy systems. The maximum recognition rates could not be the better parameter to compare the performance of the neural networks depending on the training set; but it is interesting to note that the maximum recognition rate of 97.5 % was achieved when the neural network was trained with

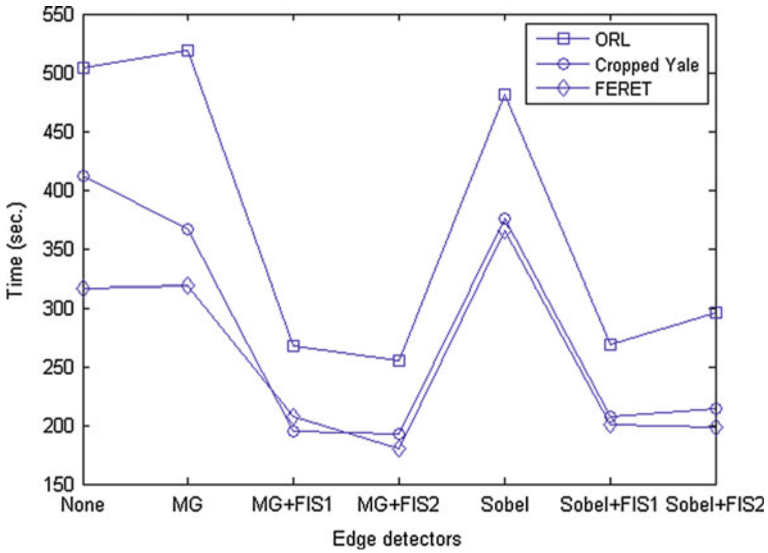


Fig. 7 Training time for the compared edge detectors tested with the ORL, Cropped Yale, and FERET face databases

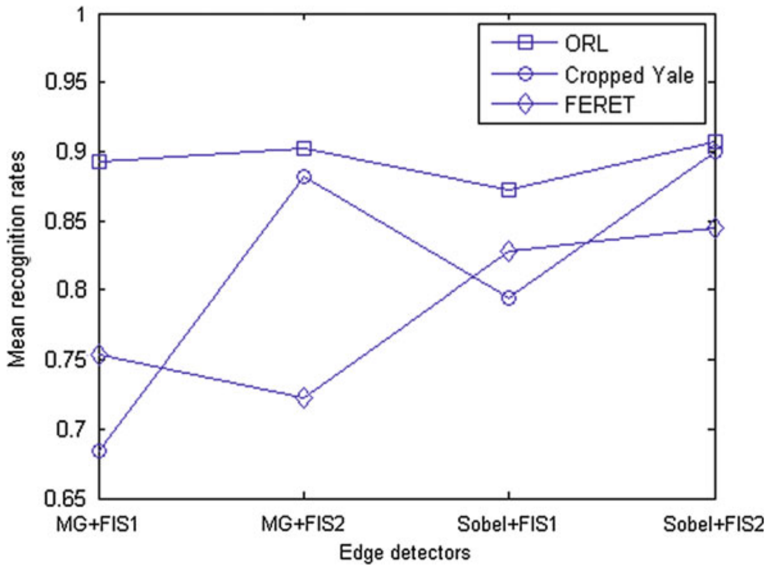


Fig. 8 Mean recognition rates for the compared edge detectors with ORL, Cropped Yale, and FERET face databases

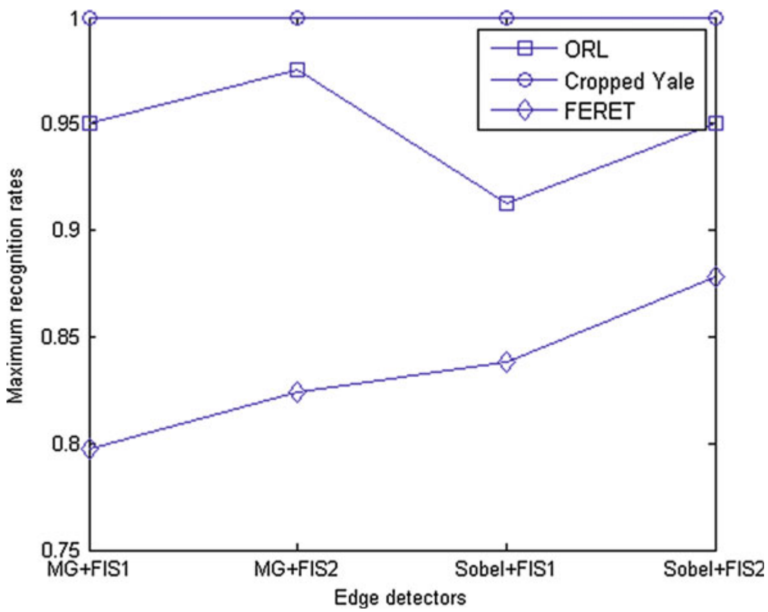


Fig. 9 Maximum recognition rates for the compared edge detectors with ORL, Cropped Yale, and FERET face database

the ORL data set preprocessed with the MG+FIS2. This is important because in a real-world system we can use this as the best configuration for images recognition, expecting to obtain good results.

The statistical analysis of the simulation results for the interval type-2 fuzzy method in comparison with the type-1 method is shown in Table 5, in which it is appreciated that in this case the difference between both methods is very significant ($t = -21.61$). This high t value can be interpreted as having sufficient statistical evidence (with more than 99 % degree of confidence) to say that that the interval type-2 method is better than the type-1 method in this case of face recognition.

Another way to illustrate the significant difference between the interval type-2 fuzzy logic method and the type-1 fuzzy method is with Fig. 10 in which the differences, between the mean recognition rates, as well as the individual values, are indicated. Here we have to say that the FERET face database is more complicated because the original photographs not only include the face, but in many cases other parts of the human body, and for this reason there is more uncertainty in the recognition, and as a consequence type-2 fuzzy logic is able to manage better this higher degree of uncertainty.

Figure 10 is a more descriptive representation of the difference between results with type-2 and type-1 fuzzy edge detector, in which is clearly evidenced the significant difference that can be achieved by using type-2 fuzzy logic. As a final note, we have to mention that the type-2 fuzzy logic toolbox developed by our

Table 5 Comparison of results for type-2 and type-1 for the FERET database

Method	Mean recognition rate	Standard deviation	n
Type-1	75.0500	0.224	20
Type-2	77.5320	0.327	10

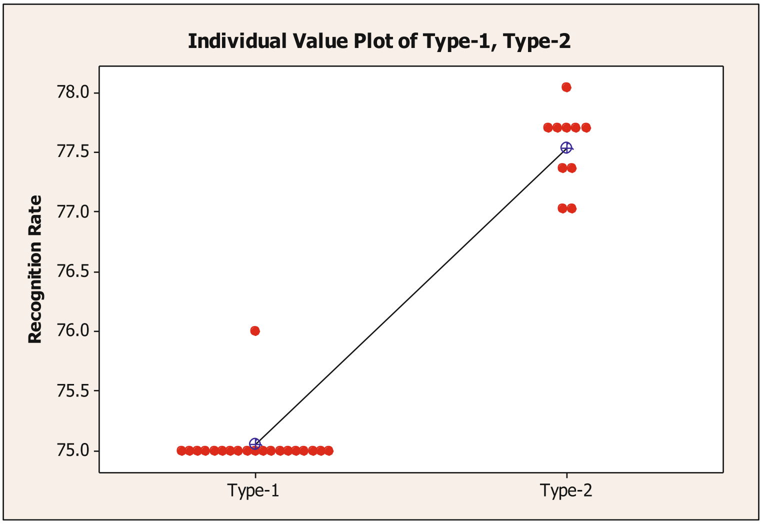


Fig. 10 Comparison of recognition results for type-2 and type-1 for the FERET database

research group was used in all the type-2 fuzzy calculations. This toolbox can be obtained following the instructions provided at the web page of our group: http://www.hafsamx.org/his/index_files/Page315.htm. The toolbox can be requested for academic and research use as indicated in the web page.

5 Conclusions

This chapter is the first effort to develop a comparison method for edge detectors as a function of their performance in different types of recognition systems. In this chapter we show that Sobel and Morphological Gradient edge detectors improved with type-2 fuzzy logic have better performance than the type-1 fuzzy edge detector and traditional methods in an image recognition system based on neural networks. Based on these results the type-2 fuzzy edge detectors can be recommended as good image processing techniques for complex recognition tasks, like face recognition. The future work includes the development of metrics for measuring the quality of edge detection and applying the type-2 fuzzy edge detector to other biometric datasets.

References

1. Mendoza, O., Melin, P., Licea, G.: A new method for edge detection in image processing using interval type-2 fuzzy logic, IEEE international conference on granular computing (GRC 2007), Silicon Valley. Ca, U.S.A (2007)
2. Mendoza, O., Melin, P., Licea, G.: Fuzzy inference systems type-1 and type-2 for digital images edges detection. J. Eng. Lett. (International Association of Engineers, Hong Kong, China) **15** (1). (Available on line: http://www.engineeringletters.com/issues_v15/issue_1/EL_15_1_7.pdf) (2007)
3. Mendoza, O., Melin, P., Licea, G.: Interval type-2 fuzzy logic for edges detection in digital images. Int. J. Intell. Syst. **24**(11), 1115–1134, Wiley, U.S.A (2009)
4. Bustince, H., Berrenechea, E., Pagola, M., Fernandez, J.: Interval-valued fuzzy sets constructed from matrices: Application to edge detection, Fuzzy Sets and Systems, Elsevier, available in: www.sciencedirect.com (2008)
5. Heath, M., Sarkar, S., Sanocki, T., Bowyer, K.W.: A robust visual method for assessing the relative performance of edge-detection algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **19**(12), 1338–1359 (1997)
6. Hua, L., Cheng, H.D., Zhang, M.: A High Performance Edge Detector Based On Fuzzy Inference Rules, Information Sciences: An International Journal. Vol. 177, Issue 21, pp. 4768–4784. Elsevier Science Inc, New York, NY, USA, (2007)
7. Revathy, K., Lekshmi, S., Prabhakaran-Nayar, S.R.: Fractal-based fuzzy technique for detection of active regions from solar. J. Sol. Phys. **228**, 43–53 (2005)
8. Suzuki, K., Horiba, I., Sugie, N., Nanki, M.: Contour extraction of left ventricular cavity from digital subtraction angiograms using a neural edge detector, Systems and Computers Japan, Wiley, Japan, 55–69 (2003)
9. Yitzhaky, Y., Peli, E.: A method for objective edge detection evaluation and detector parameter selection. IEEE Trans. Pattern Anal. Mach. Intell. **25**(8), 1027–1033 (2003)
10. Castro, J.R., Castillo, O., Melin, P., Rodriguez-Diaz, A.: Building fuzzy inference systems with a new interval type-2 fuzzy logic toolbox: In Transactions on Computational Science Book Chapter, Vol. 4750, pp. 104–114. Springer (2008)
11. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, U.S.A (2001)
12. Evans, A.N., Liu, X.U.: Morphological gradient approach for color edges detection. IEEE Trans. Image Process. **15**(6), 1454–1463 (2006)
13. Russo, F., Ramponi, G.: Edge extraction by FIRE operators Fuzzy Systems, IEEE World Congress on Computational Intelligence, pp. 249–253 (1994)
14. The ORL database of faces, AT&T Laboratories Cambridge, available at: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
15. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 643–660 (2001)
16. Lee, K.C., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 684–698 (2005)
17. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10), 1090–1104 (2000)
18. Mendoza, O., Melin, P.: The Fuzzy Sugeno Integral As A Decision Operator in The Recognition of Images with Modular Neural Networks, Hybrid Intelligent Systems. Springer, Germany, pp. 299–310 (2007)
19. Mendoza, O., Melin, P., Licea, G.: A Hybrid Approach for Image Recognition Combining Type-2 Fuzzy Logic, Modular Neural Networks and the Sugeno Integral, Information Sciences, Vol. 179(13), pp. 2078–2101. Elsevier, U.S.A. (2007)

20. Mendoza, O., Melin, P., Licea, G.: Interval Type-2 Fuzzy Logic for Module Relevance Estimation in Sugeno Integration of Modular Neural Networks, Soft Computing for Hybrid Intelligent Systems, pp. 115–127. Springer, Germany (2008)
21. Mendoza, O., Melin, P., Licea, G.: A Hybrid Approach For Image Recognition Combining Type-2 Fuzzy Logic, Modular Neural Networks and the Sugeno Integral, Information Sciences, Vol. 179(3), pp. 2078–2101. Elsevier (2008)
22. Mendoza, O., Melin, P., Licea, G.: Interval type-2 fuzzy logic and modular neural networks for face recognition applications, Appl. Soft Comput. J. **9**(4), 1377–1387 (2009)
23. Mendoza, O., Melin, P., Castillo, O., Licea, G.: Type-2 Fuzzy Logic for Improving Training Data and Response Integration in Modular Neural Networks for Image Recognition, Foundations of Fuzzy Logic and Soft Computing (LNCS), pp. 604–612. Springer, Germany (2007)

Reliable Tool Life Estimation with Multiple Acoustic Emission Signal Feature Selection and Integration Based on Type-2 Fuzzy Logic

Qun Ren, Luc Baron, Marek Balazinski and Krzysztof Jemielniak

Abstract Reliable tool life estimation of cutting tool in micromilling is essential for planning machining operations for maximum productivity and quality. This chapter presents type-2 fuzzy tool life estimation system. In this system, type-2 fuzzy analysis is used as not only a powerful tool to model acoustic emission signal features, but also a great estimator for the ambiguities and uncertainties associated with them. Depending on the estimation of root-mean-square-error and variations in modeling results of all signal features, reliable ones are selected and integrated to cutting tool life estimation.

1 Introduction

Tool wears quickly in micromilling with the microscaled cutting tool (diameter <1 mm) and the high speed (>10,000 rpm). High throughput and in-process measurement and monitoring become the central objective due to the high precision required.

Q. Ren (✉) · L. Baron · M. Balazinski
Mechanical Engineering Department, Polytechnique Montréal, University of Montreal, C.P. 6079, succ. Centre-Ville, Montréal, Québec H3C 3A7, Canada
e-mail: qun.ren@polymtl.ca

L. Baron
e-mail: luc.baron@polymtl.ca

M. Balazinski
e-mail: marek.balazinski@polymtl.ca

K. Jemielniak
Faculty of Production Engineering, Warsaw University of Technology, Narbutta 86 02-524 Warsaw, Poland
e-mail: k.jemielniak@wip.pw.edu.pl

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*, Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_13,

Currently, both experimental and commercially available tool condition monitoring (TCM) systems are based on the measurements of physical phenomena that are correlated with tool wear and can be exploited as tool wear symptoms. Cutting force and acoustic emission (AE) are most often used for TCM in micromilling process. Compared to conventional machining, the noise component in the signal for monitoring micro-machining is usually very high and difficult to separate [1]. AE is particularly well suited, because it is not only a capable detector of microscale deformation mechanisms but also a relatively uncontaminated signal within the noisy machining environment. Despite the small material removal rate in micromilling, AE signal is strong, easy to register, and shows a very short reaction time to the tool—workpiece contact, which makes it a very good means of detecting this contact and monitoring the integrity of the cutting process [2]. Very limited work has been conducted at the microscale TCM system [3–10].

The possibility of a reliable tool wear evaluation using conventional statistical methods based on one signal feature (SF) has been questioned, because the measured feature depends not only on tool wear but also on a variety of other process parameters and random disturbances. Attempts at rectifying these shortcomings have focused on pursuing a multi-sensor fusion strategy. Because of the dependence of the magnitude and frequency characteristics of AE signal on the nature of the transmission path of the signal and the sensor itself [11], the impossibility to detect damage of millimeter-sized end mill directly, and the difficulty in understanding the exact physics in micromilling processes, establishment of intelligent models for modeling and propagating uncertainties of multiple AE SFs to characterize the tool condition have been one of the key elements in making small parts accurately.

Recently, type-2 fuzzy sets and systems had become a very strategic and active research area around the world. When comparing it with traditional mathematical modeling methods and traditional fuzzy approach, type-2 fuzzy logic systems (FLSs) not only can obtain a modeling result directly from vague input–output information, but also can capture the uncertainties in the estimation results. This information is very helpful to a decision maker as he can better handle the decision. Type-2 FLSs moves the world of FLSs into a fundamentally new and important direction. Based on the literature review and our previous studies, type-2 fuzzy logic should be very suitable to identify the uncertainty in machining which has direct effect on products. And no research on this subject has been done.

In this chapter, type-2 fuzzy system based tool life estimation system is proposed. In the system, type-2 Takagi–Sugeno-Kang (TSK) FLSs are used to analyze AE SFs in micromilling process. Numerous SFs of AE signal obtained from micromilling are calculated, because it cannot be determined in advance which ones will appear to be useful in a particular application. To make the comparison and evaluation of the SFs easier and more transparent, Type-2 TSK fuzzy analysis is used as not only a powerful tool to model SFs, but also a great estimator for the ambiguities and uncertainties associated with them. Depending on the estimation of variations in modeling results of AE SFs. Reliable SFs are selected and integrated into tool life evaluation.

The rest of the chapter is organized as follows. Type-2 fuzzy logic and the strategy of type-2 fuzzy analysis method are introduced in Sect. 2. Proposed type-2 fuzzy tool life estimation system is presented in Sect. 3. A micromilling case study is presented in Sect. 4. The experimental results show the effectiveness of this method. Finally, conclusions are given in Sect. 5.

2 Type-2 Fuzzy Logic System

The universal approximation property of TSK fuzzy systems [12, 13] is well-known today. TSK FLS has a powerful capability of explaining complex relations among variables using rule consequents which are functions of the input variables.

A generalized type-1 TSK model can be described by IF–THEN rules which represent input–output relations of a system. For a multi-input–single-output (MISO) first-order type-1 TSK model, its *k*th rule can be expressed as:

$$\text{IF } x_1 \text{ is } Q_{1K} \text{ and } x_2 \text{ is } Q_{2K} \text{ and } \ldots \text{ and } x_n \text{ is } Q_{nK}$$
$$\text{THEN Z is } w^k = p_0^k + p_1^k x_1 + p_2^k x_2 + \ldots + p_n^k x_n$$

where $x_1, x_2, \ldots, x_n$ and Z are linguistic variables; $Q_{1k}, Q_{2k}, \ldots,$ and Q_{nk} are the fuzzy sets on universe of discourses U, V,…, and W, and p_0^k, p_1^k,…, p_n^k are regression parameters.

A Gaussian MF can be expressed by the following formula for the *v*th variable:

$$Q_{vk} = \exp\left[-\frac{1}{2}\left(\frac{x_v - x_v^{k*}}{\sigma}\right)^2\right] \tag{1}$$

where x_v^{k*} is the mean of the *v*th input feature in the *k*th rule for $v \in [0, n]$. σ is the standard deviation of Gaussian MF.

Based on Zadeh's conception of type-2 fuzzy sets and extension principle [14], practical algorithms for conjunction, disjunction, and complementation operations of type-2 fuzzy sets are obtained by extending previous studies [15]. Embedded interval valued type-2 fuzzy sets was introduced and a general formula was developed for the extended composition of type-2 relations which is considered as an extension of the type-1 composition [16, 17]. Based on this formula, a complete type-2 fuzzy logic theory with the handling of uncertainties was established [18]. The characterization in the definition of type-2 fuzzy sets uses the notion that type-1 fuzzy sets can be thought of as a first-order approximation to uncertainty and therefore type-2 fuzzy sets provide a second-order approximation. First-order type-2 TSK FLS and its structures were presented in 1999 [19]. High-order type-2 TSK FLS and generalized type-2 TSK FLS were presented in 2008 [20, 21].

A generalized *k*th rule in the first-order type-2 TSK fuzzy multiple-input–single-output (MISO) system can be expressed as

$$\text{IF } x_1 \text{ is } \tilde{Q}_{1K} \text{ and } x_2 \text{ is } \tilde{Q}_{2K} \text{ and } \ldots \text{ and } x_n \text{ is } \tilde{Q}_{nK},$$
$$\text{THEN Z is } \tilde{w}^k = \tilde{p}_0^k + \tilde{p}_1^k x_1 + \tilde{p}_2^k x_2 + \ldots + \tilde{p}_n^k x_n$$

where $\tilde{p}_0^k$, $\tilde{p}_1^k \ldots$, $\tilde{p}_n^k$ are consequent parameters, $\tilde{w}^k$ output from the kth IF–THEN rule in a total of M rules FLS, $Q_{\widetilde{1k}}$, $Q_{\widetilde{2k}},\ldots,Q_{\widetilde{nk}}$ are fuzzy sets on universe of discourses. For the most general model of type-2 TSK FLS, antecedents are type-2 fuzzy sets and consequents are type-1 fuzzy sets, then consequent parameter $\tilde{p}_0^k, \tilde{p}_1^k, \ldots, \tilde{p}_n^k$ are assumed as convex and normal type-1 fuzzy number subsets of the real numbers, so that they are fuzzy numbers. These rules let us simultaneously account for uncertainty about antecedent MFs and consequent parameter values. The output of type-2 fuzzy system is not only a crisp output but also an interval output. The interval output reveals the uncertainty due to antecedent or consequent parameter uncertainties.

To obtain a type-2 model directly form a type-1 model, a width a_j^k of cluster center x_{jk}^* is extended both directions as shown in Fig. 1. The cluster center x_{jk}^* becomes a constant width interval valued fuzzy set $\tilde{x}_{jk}^*$.

$$\tilde{x}_{jk}^* = [x_{jk}^*(1 - a_j^k), x_{jk}^*(1 + a_j^k)] \tag{2}$$

where a_j^k is spread percentage of cluster center x_{jk}^*.

Consequent parameter $\tilde{p}_j^k$ is obtained by extending the consequent parameter p_j^k from its type-1 counterpart using the following expression:

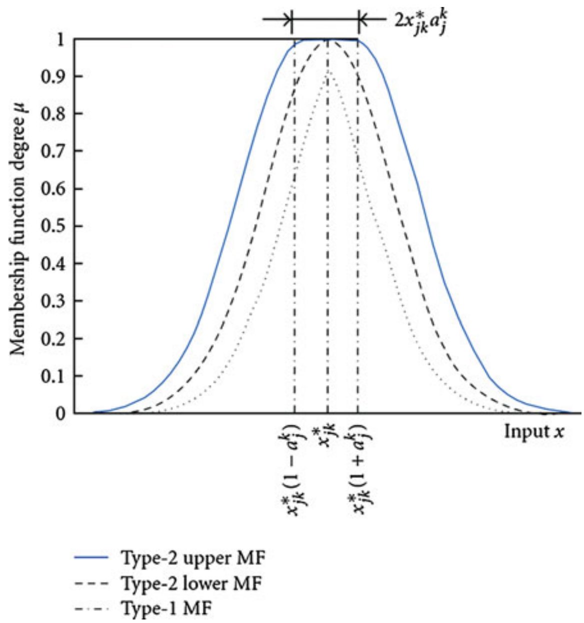


Fig. 1 Spread of cluster center

$$\tilde{p}_j^k = \left[p_j^k - s_j^k, p_j^k + s_j^k\right], \tag{3}$$

where $j \in [0, n]$, and s_j^k denotes the spread of fuzzy numbers $\tilde{p}_j^k$, where

$$s_j^k = p_j^k * b_j^k \tag{4}$$

with b_j^k denoting the spread percentage of consequent parameter p_j^k.

Hence, the premise MF is changed from type-1 fuzzy sets of (2) into type-2 fuzzy set, *i.e.*,

$$\underset{jk}{\tilde{Q}} = \exp\left[-\frac{1}{2}\left(\frac{x_j - x_{jk}^*\left(1 \pm a_j^k\right)}{\sigma_j^k}\right)^2\right] \tag{5}$$

where σ_j^k is the standard deviation of Gaussian MF.

Because of its larger number of design parameters for each rule, it was believed that type-2 FLS has the potential to be used in control [22] and other areas where a type-l model may be unable to perform well [23, 24]. Type-2 FLSs are very useful in circumstances in which it is difficult to determine an exact membership function for a fuzzy set. They can be used to handle rule uncertainties and even measurement uncertainties. Comparing with mathematical modeling methods and traditional fuzzy approach (type-1FLS), type-2 FLSs not only can obtain a modeling result directly from vague input–output information, but also they can capture the uncertainties in the estimation result. This information is very helpful to a decision maker. Type-2 FLSs move the world of FLSs into a fundamentally new and important direction. To date, type-2 FL moves in progressive ways where type-1 FL is eventually replaced or supplemented by type-2 FL [25, 26].

3 Type-2 Fuzzy Logic Based Tool Life Estimation

Key components of the proposed architecture of type-2 fuzzy analysis based tool life estimation system are shown in Fig. 2. Firstly, raw AE signal is obtained directly from the AE sensor, and the high and low frequency noise components in the AE signal are eliminated by using traditional high-pass and low-pass filter. Then the raw AE signal is demodulated to RMS value (AE_{RMS}) and different SFs are extracted. The next is type-2 fuzzy analyze each SF. Subtractive clustering based type-2 TSK fuzzy approach is adapted for modeling of the tool wear process in micro-milling. Different tool wear states based on the information from each AE_{RMS} SF are modeled as separate type-2 fuzzy models. The root-mean-square-error (RMSE) between original SF and modeled one is one of the two measures of SF usability for TCM. The other is the variation of SF. Furthermore, features with higher RMSE and bigger variation (difference between type-2 upper boundary and

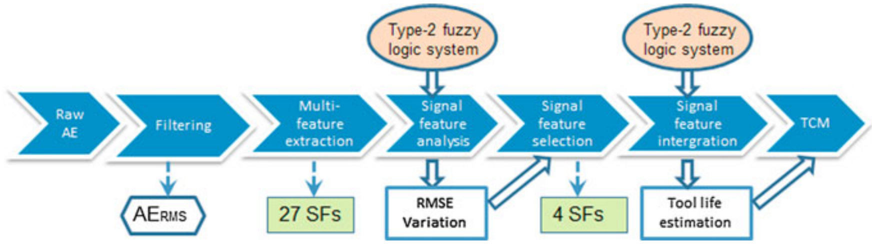


Fig. 2 Architecture of type-2 fuzzy analysis based tool condition monitoring system

lower boundary) are rejected from further evaluation. The determination of the number of SFs depends on the request of the accuracy of the TCM. In the final stage, type-2 TSK fuzzy approach is used again to estimate of the tool life estimation by integrating the selected SFs.

In this TCM, type-2 fuzzy analysis uses the type-2 TSK fuzzy modeling algorithm based on subtractive clustering method shown in Fig. 3. In this diagram, the subtractive clustering method is combined with least-square estimation algorithm to pre-identify a type-1 fuzzy model from input–output data [27, 28]. Then considering the type-1 membership functions as principal MFs of type-2 FLS, the antecedent MFs are extended as interval type-2 MFs by assigning uncertainty to cluster centers and the consequent parameters are extended as fuzzy numbers by assigning uncertainty to consequent parameter values. The best approach for AE SFs is obtained through enumerative search of optimum values for spreading percentage of cluster centers and consequent parameters.

4 Experimental Study

4.1 Experimental Setup and Measurement

The experiment of this chapter was taken on a high precision milling machine KERN Evo, equipped with a 50,000 rpm electrospindle and an HSK 25 tool holder [4, 5]. A laser system was used to measure the tool's length. The workpiece was a cold-work tool steel X155CrVMo12-1, 50HRC clamped on a three-axis Kistler 9256C1 mini-dynamometer side by side with Kistler 8152B221 AE sensor. Signals from those sensors were acquired at a sampling frequency of 50 kHz. Two-flute uncoated micro-grain WC ball end mills with 400 μm radii and 30^{o} helix angle were used for a side-milling operation performed on a 45^{o} tilted workpiece surface 20×20 mm^2 in subsequent cuts with cutting parameters: rotational speed $n = 36{,}210$ rpm, cutting speed $v_c = 68$ m/min, feed $f_z = 0.016$ mm/tooth, depth of cut $a_p = 0.05$ mm, width of cut $a_e = 0.05$ mm. Thus, one cut lased little more than one second, and the surface was machined in 400 cuts. The total wear in the

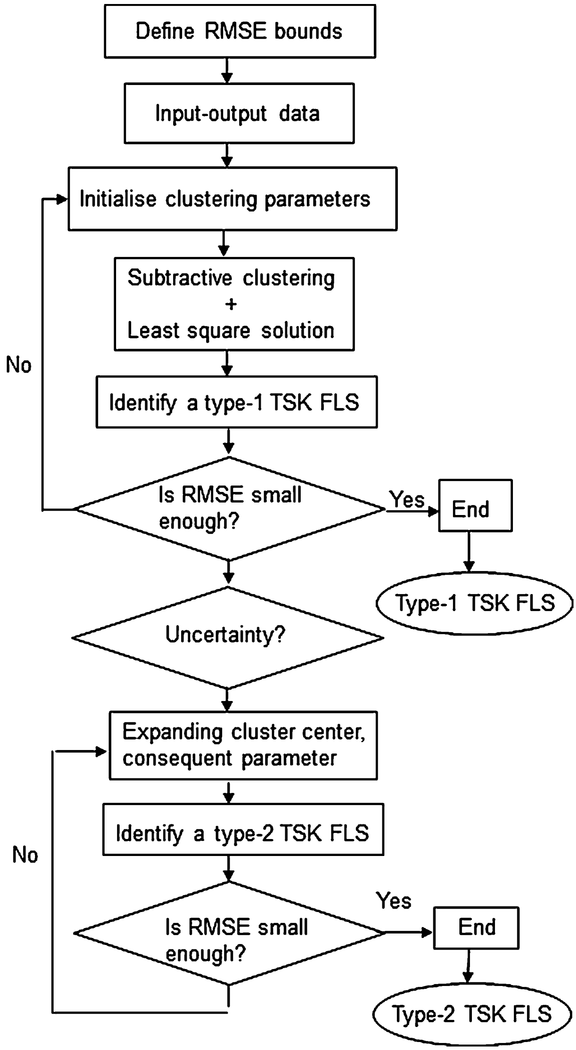


Fig. 3 Diagram of type-2 fuzzy analysis method

flank wear $VB_{Bmax} = 0.11$ mm, was used as the tool life criterion. The test was regularly interrupted to measure the wear in an optic stereo microscope.

In Fig. 4, examples of AE signal acquired in single tool pass (cut) are presented. A pass lasts only some 1.05 s and there were hundreds of them in every tool life. Since tool wear is a gradual process, only 10 passes every 50 s of cutting time were taken into further evaluation as separate operations.

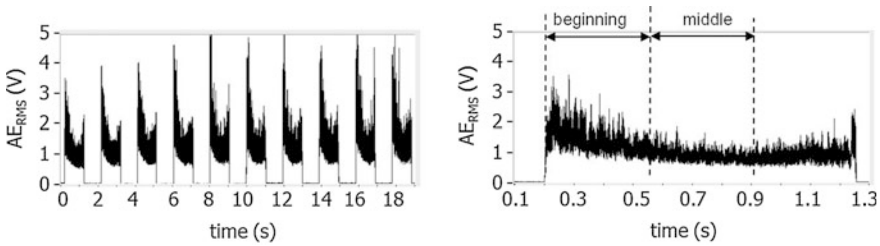


Fig. 4 Example of acoustic emission signals registered during tests

4.2 Signal Feature Calculation for Tool Wear Monitoring

Numerous SFs should be calculated by the TCM system, because it cannot be determined in advance which ones will appear to be useful in a particular application. The average value of the diagnostic signal is most often used in TCM. However, even under constant cutting conditions, the signals are not generally constant. Therefore, to minimize the diagnosis uncertainty, reduce the randomness in one SF and provide more reliable tool condition estimation, the number of SFs should be as big as possible. Therefore, from the available AE signal, nine SFs were calculated for each cut as listed in Table 1. As the signals are not generally constant, to separate initial signal disturbances from the rest of the signal, analogous features were also calculated for the beginning (first 20 %) of the cut and for the middle part (40–80 %) of the cut, and designated with the letters "BG" or "MD" in the indices of the SF's designation, respectively, $AE_{BG\text{-}MI}$ and AEMD-MI (see Fig. 4). In this experiment, Type-2 TSK fuzzy system is used to model and analyze all the 27 SFs of AE.

Table 1 Signal features from AE signal

Symbols	Definitions
AE_{AV}	The average value
AE_{RMS}	The RMS value
AE_{SD}	The standard deviation
AE_{MX}	The maximum value above which was 5 % of all values
AE_{MI}	The minimum value below which was 5 % of all values
AE_{RG}	The range (maximum–minimum)
$AE_{MX\text{-}AV}$	The maximum minus average
$AE_{MI\text{-}AV}$	The minimum minus average
AE_{Dy}	The absolute difference between subsequent signal values

4.3 Type-2 Fuzzy Acoustic Emission Signal Feature Selection

By following the step illustrated in the diagram of Fig. 3, each of the 27 SFs is analyzed. In this experiment, MFs are Gaussian MFs. There are 529 datasets from each AE SF. The clustering parameters are pre-initialized. The cluster radius is confined to the range [0.15; 1.0] with a step size of 0.15. The accept ratio and the reject ratio are both considered in the range [0; 1.0] with a step size of 0.1. The squash factor is considered in the range [0.05; 2] with a step size of 0.05. Combined with a least-square estimation algorithm, the fuzzy systems for each cutting length were identified. To pre-identify the type-1 fuzzy system, the clustering parameters used for subtractive clustering method in this experimental study are cluster radius = 0.15, accept ratio = 0.5, reject ratio = 0.15 and squash factor = 0.1.

In this experiment, RMSE and variations are the measures of SF usability for TCM. RMSEs for the type-1 and type-2 fuzzy estimation are calculated using the following expression:

$$RMSE = \sqrt{\sum \left(\tilde{V} - V\right)^2 / N} \tag{6}$$

where $\tilde{V}$ is the fuzzy estimated voltage of AE, either type-1 modeling or type-2 modeling. V is the voltage of AE recorded. N is number of the registered datasets.

The variation of $\tilde{V}$ is estimated with Eq. (6) which represents the difference between upper and lower values of the interval output of type-2 fuzzy approach, revealing the uncertainties of the AE in type-2 fuzzy estimation.

$$Variation = upper\_boundary - lower\_boundary \tag{7}$$

Moreover, SFs with RMSE higher than 60 mV and the maximum variation higher than 100 mV are rejected from further evaluation, these threshold values were selected the same as the ones used in [5]. The RMSE, spreading percentage for cluster centers and consequent parameters, and the maximum and minimum variation for the four selected SFs $AE_{MD\text{-}MI\text{-}AV}$, $AE_{MD\text{-}MI}$, $AE_{BG\text{-}MI\text{-}AV}$, and $AE_{MD\text{-}RG}$ are listed in Table 2. In this experiment, 6, 7, 6, 6-rules fuzzy systems are used for modeling the SF $AE_{MD\text{-}MI\text{-}AV}$, $AE_{MD\text{-}MI}$, $AE_{BG\text{-}MI\text{-}AV}$, and $AE_{MD\text{-}RG}$. Their type-2 fuzzy models are illustrated in Fig. 5. The RMSE, number of fuzzy rules, spreading percentage for cluster centers and consequent parameters, and the maximum and minimum variation for the four SFs are listed in Table 2 column 2–5. Further, they will be integrated to cutting tool life estimation.

Figure 6 depicts the type-1 fuzzy modeling results for the four selected SFs—$AE_{MD\text{-}MI\text{-}AV}$, $AE_{MD\text{-}MI}$, $AE_{BG\text{-}MI\text{-}AV}$, and $AE_{MD\text{-}RG}$. Comparing the curves for each SF to overall output curve from type-2 fuzzy modeling in Fig. 5, it is observed that they are similar and have slight differences in each cutting instant

Table 2 Type-2 fuzzy parameters for AE signal features

Symbols	$AE_{MD\text{-}MI\text{-}AV}$	$AE_{MD\text{-}MI}$	$AE_{BG\text{-}MI\text{-}AV}$	$AE_{MD\text{-}BG}$	AE_{AV}	$AE_{AV\text{-}BG\text{-}dY}$
RMSE	18.8897	20.8703	35.5409	46.5838	31.3703	203.9663
Number of rules	6	7	6	6	7	5
Spreading percentage of the cluster centers (%)		0.35 %			0.35 %	
	0.30 %	0.93 %	0.45 %	0.26 %	0.51 %	
	0.47 %	0.87 %	0.54 %	0.41 %	0.40 %	0.18 %
	0.23 %	0.55 %	0.29 %	0.59 %	0.07 %	90 %
	0.84 %	0.62 %	0.74 %	0.26 %	0.23 %	0.98 %
	0.19 %	0.58 %	0.18 %	0.60 %	0.12 %	0.44 %
	0.23 %	0.20 %	0.68 %	0.71 %	0.18 %	0.11 %
Spreading percentage of consequent parameters (%)	4 %	4 %	4 %	4 %	4 %	4 %
Max variation	37.5224	47.5750	66.0816	88.7114	101.7906	375.5222
Min variation	9.0635	14.8341	21.8028	20.8129	32.4195	118.4191

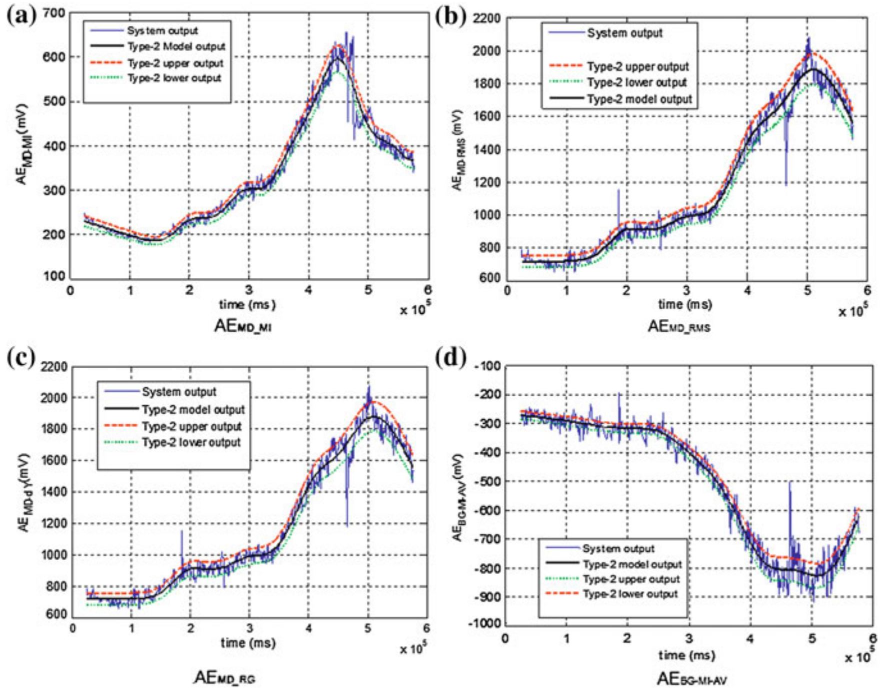


Fig. 5 Type-2 fuzzy model for the four selected acoustic emission signal features

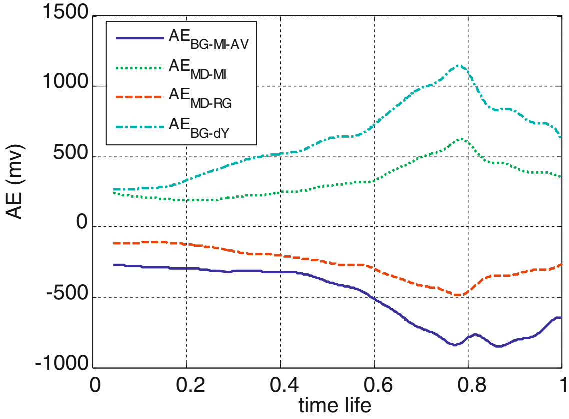


Fig. 6 Type-1 fuzzy modeling acoustic emission signal features

which are illustrated in Fig. 7. The type-2 fuzzy modeling has additional outputs for each SF—the upper boundary and lower boundary in Fig. 5. In this experimental study, the variation between the upper boundary and lower boundary calculated using Eq. (7) at each instant of the cutting process is of importance as it gives one of the information of uncertainty. To detect damage on the millimeter-sized end mill directly, knowing the difficulty in understanding the exact physics in micromilling processes, establishment of intelligent models for modeling, and propagating uncertainties of multiple AE SFs to characterize the tool condition is one of the key elements in being able to produce small parts with better accuracy.

The AE signal data used in this experiment is the same as that of the first tool in Jemielniak's paper [5] which use an approach based on hierarchical algorithms [29]. In Jemielniak's paper, RMSE is the only measure of SF usability for TCM. There were six SFs—AE_{AV}, $AE_{BG\text{-}MI\text{-}AV}$, $AE_{MD\text{-}MI}$, $AE_{MD\text{-}MI\text{-}AV}$, $AE_{MD\text{-}RG}$ and AEAV-BG-dY, and AE_{AV} were the best. In the type-2 fuzzy analysis, not only RMSE

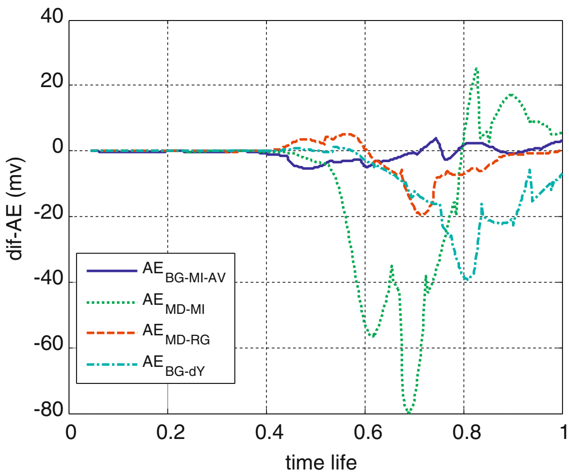


Fig. 7 Differences between type-2 fuzzy model and its type-2 counterpart

but also variations are used as the measure of SF usability. For example, AE_{AV} has less RMSE but bigger variation which has the information of associated ambiguities and uncertainties. AE_{AV} and $AE_{AV\text{-}BG\text{-}dY}$ are rejected because of their large RMSE and variation (listed in Table 2 column 6–7). The best SF is $AE_{MD\text{-}MI\text{-}AV}$. Obviously, the results using type-2 FLS are more reasonable than the one using hierarchical algorithms. $AE_{MD\text{-}MI\text{-}AV}$ is AE signal from the middle part of the cut, which has less noise and is more stable than that AE signal from the whole cutting process (AE_{AV}). It is more suitable for TCM.

4.4 Tool Life Estimation with Signal Feature Integration

The four selected SFs—$AE_{MD\text{-}MI\text{-}AV}$, $AE_{MD\text{-}MI}$, $AE_{BG\text{-}MI\text{-}AV}$, and $AE_{MD\text{-}RG}$ are integrated to evaluate the micromilling cutting tool life by using type-2 TSK fuzzy logic system. The tool life estimation is illustrated as the sold curve in Fig. 8. The information on uncertainty in the tool life fuzzy estimation is shown between the type-2 fuzzy output upper boundary (shown as dashed curve) and the lower boundary (shown as dash-dotted curve). The maximum difference between the upper boundary and the lower boundary is 0.5003 % and the minimum is 0.00032 %. The maximum difference between the overall tool life estimation and the model ideal (shown as dotted curve) is 0.0492 %.

Comparing the results from type-2 fuzzy estimation with that in Jemielniak's paper [5], Type-2 tool life fuzzy estimation not only estimate the tool life by integrating selected multiple AE SFs, but also predict the uncertainty in tool life in each cutting instant. During initial cutting period (before 50 % of cutting tool life), the estimated cutting tool life is almost the same as the situation ideal, which corresponds to the initial stages of wear occurring. After cutting tool reach 50 % of its life, uncertainty in the tool life estimation increases, even though the overall

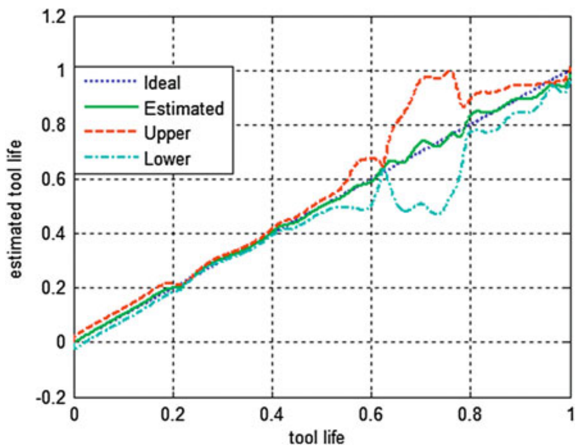


Fig. 8 Type-2 fuzzy tool life estimation

estimation of cutting tool life still has the same trend of the model ideal. During the period that cutting tool life reaches its 60–80 %, the interval of uncertainty becomes very large, while AE SFs vary significantly. This corresponds to the period of relatively rapid wear or failure of cutting tool. The information about uncertainty prediction has great meaning for decision making or tool condition investigation.

5 Conclusion

This chapter presented type-2 fuzzy tool life estimation system. In this system, type-2 fuzzy logic is used to analyze the AE SFs in TCM in micromilling process and integrate the selected SFs for tool life estimation. The interval output of type-2 approach provides an interval of uncertainty associated with SFs of AE signal and tool life prediction.

Nowadays, Type-2 fuzzy logic is the only intelligent method which not only can model the AE signal, but also estimate the uncertainties from the vague information obtained during high precision machining. The estimation of uncertainties can be used for proving the conformance with specifications for products or auto controlling of machine system. The application of type-2 fuzzy logic on uncertainty estimation in high precision machining has great meaning for continuously improvement in product quality, reliability, and manufacturing efficiency in machining industry.

One limitation of the results obtained in this chapter can be that the tool life determination is carried out solely using the AE SF information during the micromilling process. For high precision machining, cutting force could be combined with AE to effectively determinate cutting tool life at a higher precision scale.

Acknowledgments The data for the experimental study described in this chapter were collected by Micromachining Laboratory at Mondragón University in Spain, in collaboration with Prof. Krzysztof Jemielniak.

The authors wish to acknowledge the anonymous reviewers for their detailed and helpful comments to the manuscript.

References

1. Tansel, I., Rodriguez, O., Trujillo, M., Paz, E., Li, W.: Micro-end-milling—I, wear and breakage. Int. J. Mach. Tools Manuf. **38**, 1419–1436 (1998)
2. Wang, L.H., Gao Robert, X.: Condition monitoring and control for intelligent manufacturing. Springer Series in Advanced Manufacturing, ISBN: 978-1-84628-268-3 (2006)
3. Chae, J., Park, S.S., Freiheit, T.: Investigation of micro-cutting operations. J. Mach. Tools Manuf. **46**, 313–332 (2006)
4. Jemielniak, K., Arrazola, P.J.: Application of AE and cutting force signals in tool condition monitoring in micro-milling. CIRP J. Manuf. Sci. Technol. **1**, 97–102 (2008)

5. Jemielniak, K., Bombinski, S., Aristimuno, P.X.: Tool condition monitoring in micromilling based on hierarchical integration of signal measures. CIRP Ann. Manuf. Technol. **57**, 121–124 (2008)
6. Ren, Q., Baron, L., Balazinski, M.: Type-2 fuzzy modeling for acoustic emission signal in precision manufacturing, Model. Simul. Eng. **2011**(696947) (2011). doi: 10.1155/2011/696947
7. Ren, Q., Baron, L., Balazinski, M.: Fuzzy identification of cutting acoustic emission with extended subtractive cluster analysis. Nonlinear Dyn. **67**(4), 2599–2608 (2012)
8. Ren, Q., Baron, L., Jemielniak, K., Balazinski, M.: Acoustic emission signal feature analysis using type-2 fuzzy logic system. In: Proceedings of the 29th North American Fuzzy Information Processing Society Annual Conference NAFIPS 2010, pp. 1–6. Toronto, Canada (2010)
9. Tansel, I.N., Arkan, T.T., Bao, W.Y., et al.: Tool wear estimation in micro-machining. Int. J. Mach. Tools Manuf. **40**, 599–608 (2000)
10. Teti, R., Jemielniak, K., O'Donnell, G., Dornfeld, D.: Advanced monitoring of machining operations. CIRP Ann. Manuf. Technol. **59**, 717–739 (2010)
11. Moriwaki, T.: Application of acoustic measurements to sensing of wear and breakage of cutting tool, bull. Jpn. Soc. Precis. Eng. **17**(3) (1983)
12. Sugeno, M., Kang, G.: Structure identification of fuzzy model. Fuzzy Sets Syst. **28**(1), 15–33 (1988)
13. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man. Cybern. **15**(1), 116–132 (1985)
14. Zadeh, L.A.: The conception of a linguistic variable and its application in approximate reasoning—I. Inf. Sci. **8**, 199–249 (1975)
15. Mizumoto, M., Tanaka, K.: Fuzzy sets of type 2 under algebraic product and algebraic sum. Fuzzy Sets Syst. **5**(3), 277–290 (1981)
16. John, R.I.: Type-2 fuzzy sets: an appraisal of theory and applications. Int. J. Uncertain. Fuzziness Knowl. Based Syst. **6**(6), 563–576 (1998)
17. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Inf. Sci. **132**(1–4), 195–220 (2001)
18. Mendel, J.M.: Uncertain Rule-Based Fuzzy logic Systems—Introduction on New Directions. Prentice hall PTR, Upper Saddle River (2001)
19. Liang, Q., Mendel, J.M.: An introduction to type-2 TSK fuzzy logic systems, In: Proceedings of IEEE FUZZ Conference, Seoul, Korea (1999)
20. Ren, Q., Baron, L., Balazinski, M.: (2008) High order type-2 TSK fuzzy logic system. NAFIPS 2008. New York, USA, 1–6
21. Ren, Q., Balazinski, M., Baron, L.: High order interval type-2 Takagi-Sugeno-Kang fuzzy logic system and its application in acoustic emission signal modeling in turning process. Int. J. Adv. Manuf. Technol. (2012). doi:10.1007/s00170-012-3956-z
22. Hagras, H.: Type-2 FLCs: a new generation of fuzzy controllers. IEEE Comput. Intell. Mag. **2**(1), 30–43 (2007)
23. Castillo, O., Melin, P.: Intelligent systems with interval type-2 fuzzy logic. Int. J. Innov. Comput. Inf. Control **4**(4), 771–783 (2008)
24. Wu, D., Tan, W.W.: A simplified type-2 fuzzy logic controller for real-time control. ISA Trans. **45**(4), 503–516 (2006)
25. John, R., Coupland, S.: Type-2 fuzzy logic: a historical view. IEEE Comput. Intell. Mag. **2**(1), 57–62 (2007)
26. Mendel, J.M.: Advances in type-2 fuzzy sets and systems. Inf. Sci. **177**(1), 84–110 (2007)
27. Ren, Q., Baron, L., Balazinski, M., Jemielniak K.: tool condition monitoring using the tsk fuzzy approach based on subtractive clustering method. News Frontiers in Applied Artificial Intelligence, pp. 52–56, Springer-Verlag, Berlin (2008)

28. Ren, Q., Baron, L., Balazinski, M., Jemielniak, K.: TSK fuzzy modeling for tool wear condition in turning processes: an experimental study. Eng. Appl. Artif. Intell. **24**(2), 260–265 (2011)
29. Jemielniak, K., Bombinski, S.: Hierarchical strategies in tool wear monitoring. J. Eng. Manuf. **223**(B3), 375–382 (2006)

A Review of Cluster Validation with an Example of Type-2 Fuzzy Application in R

Ibrahim Ozkan and I. Burhan Türkşen

Abstract Interval valued type-2 fuzziness can be represented by means of membership functions obtained with upper and lower values of the level of fuzziness. These upper and lower values for the level of fuzziness in FCM algorithm were obtained in our previous studies. A particular application of Interval valued type-2 fuzziness is shown for cluster validity analysis in this chapter. For this purpose, we introduce a brief taxonomy for cluster validity indices to clarify the contribution of our novel approach. To provide reproducibility of our technique, the source code is written in freely available language 'R' and can be found on our web site.

1 Introduction

Cluster validity is an important task in unsupervised learning of data structure with an application of clustering algorithms based on predetermined similarity measure. These algorithms are aimed at extracting group structure in a data set. Such extractions are often validated by means of cluster validity indices. There is a vast literature on this subject in both crisp and fuzzy clustering domains. Researchers are still improving the reported validity measures. The aim of this chapter is twofold. One is to introduce a brief taxonomy of cluster validity indices.

I. Ozkan (✉)
Department of Economics, Hacettepe University, Beytepe 06800 Ankara, Turkey
e-mail: ozkan@mie.utoronto.ca

I. Burhan Türkşen
TOBB ETU, Ankara, Turkey
e-mail: turksen@mie.utoronto.ca;bturksen@etu.edu.tr

I. Burhan Türkşen
Department of Mechanical and Industrial Engineering, University of Toronto, Ontario M5S3G8, Canada

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*, 219
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_14,

The second aim is to provide an example of an application of interval valued type-2 fuzziness for identification of the number of clusters in Fuzzy C-Mean (FCM). To this end, upper and lower values of the level of fuzziness for FCM clustering found in our previous studies [32, 33] are used to assess the stability of clustering. In order to make this chapter fully reproducible, at least, as much as possible, the source code written in freely available language 'R'[1] for computing is available in our website: http://individual.utoronto.ca/ozkan/bk_chapters.htm.

1.1 A Brief Survey of Cluster Validity Indices

Since there is a vast literature on cluster validity indices (CVI), a list that includes all previous works may not be explanatory. Hence, a brief survey of CVI and their classification will be introduced for both fuzzy and crisp clustering algorithms.

In general, one can find three types of cluster validity indices in the literature [12, 26, 38]. These are:

(i) Internal criteria that look for both small dispersion in clusters and high dissimilarity between clusters;
(ii) External criteria in which pre-specified structures are used to validate the clustering results;
(iii) Relative criteria that need a decision objective to be chosen before clustering. Then the clustering algorithm is applied with different input parameter sets. The results are assessed and the optimal clustering structure is selected based on a previously chosen criterion [7, 8, 21, 38].

The objective ways of assessing clusters are: (i) to use the class membership information if available, (ii) to use internal criteria or mix with relative criteria if class membership information is not available.

Apart from this general classification, *Stability* and *Biological* types of validation can be seen in the literature. In case of biological criteria, the result is assessed by means of intuition to determine if the clustering produced meaningful results. Stability type validation is a special case of relative criteria approach in which the variables are removed one by one and the results are compared for the selection of the optimum number of clusters. (See for example, [1, 2, 11, 16, 29, 31]) Other stability approaches use resampling to assess the cluster stability (See for example, [17, 30, 31, 36]).

Since the clustering algorithms can be assessed under three basic criteria, namely *Compactness*, *Connectedness*, and *Separation* [25, 36], the validity indices naturally are formed based on the basis of such categories. These indices are also classified as internal type validation measures. Compactness increases with the number of clusters while separation decreases. Hence, compactness and separation

---

[1] 'R' SW can be downloaded from http://cran.r-project.org/ web site.

measurements are most common in classification of such indices within cluster variability and gap statistics type validation measures. (One may want to refer to the following brief list of papers for a discussion of the cluster validity: (i) Compactness and Separation [15, 18, 22–24, 27]; (ii) within cluster validity and gap statistics [10, 37, 40–42], and (iii) others, [43, 46, 49]).

There are several works on the validation of the number of clusters for FCM. Among them the following articles are worth mentioning: [4, 6, 9, 15, 19, 20, 27, 28, 35, 37, 44, 45, 47, 49, 50].

In this chapter, we demonstrate our recently proposed validation criterion called "MiniMax ε-stable cluster validity index", [34], using the free software 'R'. This validity index can be classified as a stability type validation where, the stability of the clusters is assessed by means of the uncertainty associated with the level of fuzziness in FCM clustering algorithms. In addition to this cluster validity index, some other variations will also be demonstrated. Since one of the aims of this chapter is to provide a good platform for researchers/practitioners, all scripts related with CVI analysis are prepared in pure 'R' language and one may get these functions from "http://individual.utoronto.ca/ozkan/bk_chapters.htm" web site.

In order to keep the chapter self-explanatory, we state a brief introduction of FCM algorithm with an overview of selected and widely used cluster validity indices together with packages that are necessary to perform fuzzy clustering which is given in the following section. Then, *MiniMax ε-stable index* is given together with the implementation of the selected indices. The performances of these indices and the conclusions are stated in the results section.

2 FCM, Upper and Lower Level of Fuzziness and Number of Clusters

2.1 *Fuzzy C-Means (FCM)*

FCM algorithm partitions data into clusters in which each observation is assigned a membership value between zero and one to each cluster. Bezdek [3] proposed the minimization of the following objective function:

$$J_m(U, V : X) = \sum_{k=1}^{nd} \sum_{c=1}^{nc} \mu_{c,k}^m \|x_k - v_c\|_A^2 \qquad (1)$$

where, $\mu_{c,k}$: membership value of kth vector in cth cluster such that $\mu_{c,k} \in [0, 1]$, nd is the number of vectors used in the analysis, nc is the number of clusters, $\|.\|_A$ is norm and m is the level of fuzziness, the membership values are calculated as:

$$\mu_{i,k} = \left[\sum_{c=1}^{nc} \left(\frac{\|x_k - v_i\|_A}{\|x_k - v_c\|_A} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{2}$$

where, $\sum_{c=1}^{nc} \mu_{k,c} = 1$ for some given $m > 1$, and finally the cluster centers are computed as:

$$v_c = \frac{\sum_{k=1}^{nd} \mu_{ck}^m x_k}{\sum_{k=1}^{nd} \mu_{ck}^m} \tag{3}$$

FCM algorithm is implemented in several packages in R. For example, "*fanny*" function in "*cluster*" package[2] performs fuzzy clustering with limited similarity/dissimilarity measures. Another package that provides FCM is "*e1071*" through "*cmean*" function. This function is used for the experiments given in this chapter.[3] This function accepts only two dissimilarity methods that are *Euclidian* and *Manhattan*.[4]

FCM algorithm requires that the number of clusters and the level of fuzziness need to be identified first. There are limited studies for the level of fuzziness (0), even though this parameter makes this algorithm a fuzzy algorithm. The most widely used value for the level of fuzziness is two. And this value is usually accepted as the rule of thumb. Pal and Bezdek [35] investigate that the value of the level of fuzziness should be between 1.5 and 2.5 based on their analysis on the performance of cluster validity indices. Yu et al. [48] suggest that the proper value of the level of fuzziness depends on the data itself. Ozkan and Turksen [33] show that the proper values for upper and lower bounds of level of fuzziness are 1.4 and 2.6, respectively. The upper and lower values of the level of fuzziness *do not depend on the number of clusters*. Thus one can use uncertainty related with fuzziness to find the optimum number of clusters among the possible stable clusters. Hence this becomes a Type-2 fuzzy application, because it incorporates the uncertainty of fuzziness.

2.2 Selected Validation Indices

In the current literature, several cluster validity indices have been introduced to identify the number of clusters in fuzzy data with FCM [6, 9, 13–15, 20, 28, 35, 37, 44]. In this section, we briefly review those cluster validity indices that are most frequently investigated.

2 See http://cran.r-project.org/web/packages/cluster/index.html.

3 http://www.stat.ucl.ac.be/ISdidactique/Rhelp/library/e1071/html/cmeans.html.

4 Euclidian distance is defined as square root of $\sum_i (x_i - y_i)^2$ and Manhattan is $\sum_i abs(x_i - y_i)$.

Bezdek proposed partition coefficient (PC) and partition entropy (PE) as cluster validity criteria [4, 5]. Both indices use only membership values of fuzzy clusters. Both seek higher membership values to any one of the clusters. These indices implicitly seek stable cluster centers. Minimizing PE or maximizing PC with respect to the number of clusters, nc, where $nc \in [c_{\min}, c_{\max}]$, are used to determine the optimum number of clusters:

$$\mathrm{PC}\,(U) = \frac{1}{nd}\left(\sum_{k=1}^{nd}\sum_{c=1}^{nc}\mu_{c,k}^{2}\right) \tag{4}$$

Partition coefficient is increasing with the membership values which also increase with the level of fuzziness approaches to its lower bound. Hence partition coefficient is not an appropriate measure if clustering is performed with the level of fuzziness close to 1.4.

$$\mathrm{PE}\,(U) = -\frac{1}{nd}\left(\sum_{k=1}^{nd}\sum_{c=1}^{nc}\mu_{c,k}\log_{a}(\mu_{c,k})\right) \tag{5}$$

Partition entropy is expected to increase with the level of fuzziness. For the sake of explanation of its behavior, this index is calculated for the values of the level of fuzziness 1.4, 2 and 2.6.

Fukuyama and Sugeno's [20] selection index uses both membership values and data as:

$$\mathrm{FS}\,(U,\,V:X) = \sum_{c=1}^{nc}\sum_{k=1}^{nd}\mu_{c,k}^{m}\left(\|x_k - v_c\|_d^2 - \|v_c - \overline{v}\|_d^2\right) \tag{6}$$

where, $\overline{v} = \frac{1}{nd}\sum_{k=1}^{nd} x_k$, is the mean of the whole data set.

Xie-Beni's index seeks compactness and separation through their ratios [47] as:

$$\mathrm{FS}\,(U,\,V:X) = \frac{\sum_{c=1}^{nc}\sum_{k=1}^{nd}\mu_{c,k}^{m}\|v_c - x_k\|^2}{nd\cdot\min_{i,j}\left(\|v_i - v_j\|\right)} \tag{7}$$

The source code to calculate these indices is given as "*cvindxs_cmeans*" ("*cvindxscmeans.r*" file) together with "*sugenocmeans*" ("*sugenocmeans.r*" file) function. *MiniMax ε-stable index*, [34] and its variations worth explained explicitly, since this is a unique example of an application of interval valued type-2 fuzziness to cluster validity.

2.3 MiniMax ε-stable Index, l_∞ Norm

Ozkan and Turksen [34] proposed an application of interval valued type-2 fuzziness to find the optimum number of clusters which is called *MiniMax ε-stable index*. This approach is based on the uncertainty associated with the level of fuzziness. Pal and Bezdek [35] report that the level of fuzziness affects the success of cluster validation and they suggest that the value of the level of fuzziness should be between 1.5 and 2.5. If the value of the level of fuzziness is important in FCM, *the uncertainty associated with the value of the level of fuzziness is also important*. *MiniMax ε-stable index* incorporates this uncertainty in validation. This method seeks cluster center value stability with respect to the level of fuzziness where its upper and lower bounds [33] are used as a guide to find the number of clusters in FCM.[5] This also means that the change in entropy is minimized with respect to the change in the level of fuzziness if the stability of cluster centers is maximized.

For the well-separated clusters where membership values are closer to either 0 or 1, cluster center values become more stable with respect to the level of fuzziness. Thus, the effect of the changes of the level of fuzziness to the changes in the cluster centers should be minimized for the optimum number of clusters. Hence the objective function is given as

$$\min\left(\Delta v_c\big|_{m_l,m_u} = \Delta\left[\frac{\sum_{k=1}^{nd}\mu_{ck}^m x_k}{\sum_{k=1}^{nd}\mu_{ck}^m}\right]_{m_l,m_u}\right) \tag{8}$$

where m_l and m_u are the lower and upper values of the level of fuzziness to be used to assess the stability of cluster centers.

Authors [34] suggest using at least one half of the range of boundary values to get a proper validity measure based on the behavior of the membership function. Assessing the stability of cluster center values obtained with $m = 2.6$ and $m = 1.4$ should create enough information to decide the optimum number of clusters. In order to measure the stability they suggest the l_∞ norm.[6] This distance measure seeks the maximum changes in cluster center values in its dimensions. Euclidian, l_2 norm, or Manhattan, l_1 norm can also be considered as stability measures although they are computationally expensive. Calculation of these measures can be performed by the following steps:

1. I: Run FCM algorithm with $m = m_l$ and $m = m_u$
   II: Measure center dissimilarities by means of l_∞, l_2 and l_1 norms.

---

5 Kim et al. [27], used similar intuition and suggested that the optimal number of clusters can be found by minimizing the change in cluster center with respect to the number of clusters.

6 l_∞ norm is defined as $\lim_{p\to\infty} l_p$ where l_p is p-norm. Since p-norm is given as $\left\|v_c^{m_u} - v_c^{m_l}\right\|_p = \sum_{i=1}^{nv}((v_{c,i}^{m_u})^p - (v_{c,i}^{m_l})^p)^{\frac{1}{p}}$, hence, $\lim_{p\to\infty}\left\|v_c^{m_u} - v_c^{m_l}\right\|_\infty = \max_{i=1}^{nv}\left|v_{c,i}^{m_u} - v_{c,i}^{m_l}\right|$.

2. I: Run FCM algorithm with $m = m_u$. Then change $m = m_l$ and calculate new cluster centers (in one step only)
   II: Measure center dissimilarities by means of l_∞, l_2 and l_1 norms.

Authors present four indices in order to find the optimum number of clusters in fuzzy data sets. They also explicitly state that there can be more approaches similar to theirs. The example of indices are listed below (all with respect to the different number of clusters):

- "Average change in cluster centers measured by the values of a distance measure, (*Average change in distance*, ACI)
- The change in specific entropy measure Ozkan and Turksen [32] used in their analysis, (*Change in Entropy*, CIE, based on the adapted version of Shannon's Entropy [39])
- The minimum of the maximum changes in cluster centers measured by the values of a distance measure, (*Minimum of the Maximum change in distance*, MMI)
- The minimum of the maximum changes in cluster centers in any dimension, *MiniMax ε-stable index*. *MiniMax ε-stable index* tries to identify stability in every dimension and finds out the minimum of all maximum distances in all dimensions. For example two sets of cluster centers are obtained with the perturbation of the level of fuzziness, *m*. Thus, the first set of cluster centers is calculated with an application of FCM by setting the level of fuzziness 2.6. Then the second set of centers is calculated with the perturbation of the level of fuzziness from the value of 2.6 to the value of 1.4, in equation…" [34].

Since the first three indices are computationally expensive, one can apply these measures to cluster centers and membership values calculated as given in the fourth index. The experimentation is left to the reader. The code is given as "*cvindxs_cmean*" function in "*cvindxscmeans.r*" file.

In this chapter, several data sets are used for CVI experiments. In order to make this chapter fully reproducible some data sets are obtained by means of random number generations.Two other are selected among well-known real-world data sets, "*iris*" and "*wine*" data sets, used in the literature.[7]

3 Experiment

In the first part of this section, several artificial data sets are used. The second part contains real-world data sets, which are *iris* and *wine* data sets.

[7] Iris data is available in R SW. Wine data can be downloaded from http://archive.ics.uci.edu/ml/machine-learning-databases/wine/ manually or by using R SW as it is shown in "iris and wine data ex.r" script file.

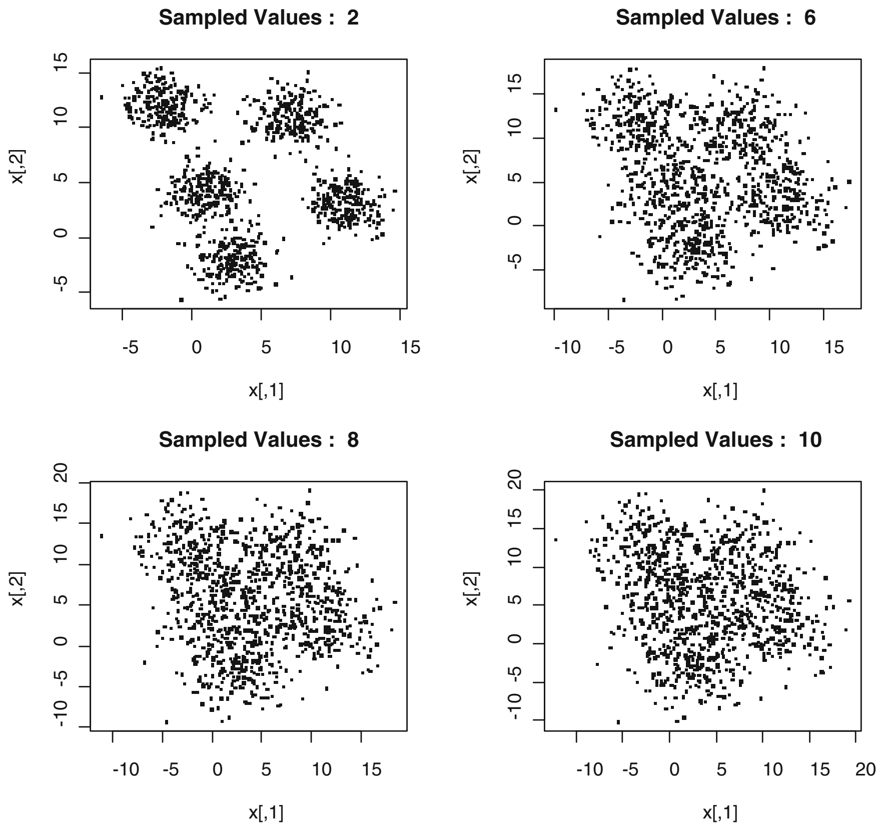


Fig. 1 *Scatter graph* of artificial data sets

3.1 Artificial Data

There are nine artificial data sets created by application of normal random values with different variances (using "*rmvnorm*" function of "*mvtnorm*" package in R). There are 1000 pairs divided equally into 5 clusters centered as randomly picked {(1,4), (7,11), (11,3), (3,−2), (−2,12)} values. For the sake of simplicity, the number of variables is set as $nv = 2$. The random numbers are drawn with equal variances of each dimension and zero covariance.

In Fig. 1, randomly generated data sets are shown. Each data set is generated with a diagonal covariance matrix where the values of the diagonals are set to sampled values. The R code to create this figure is given as "*fig1.r*" file.

The boundaries of clusters are clear for the first few data sets but the boundaries of clusters cannot be recognized in the last few of them. Up to sampled values with covariance matrix whose diagonal values are six, the boundaries of the clusters can

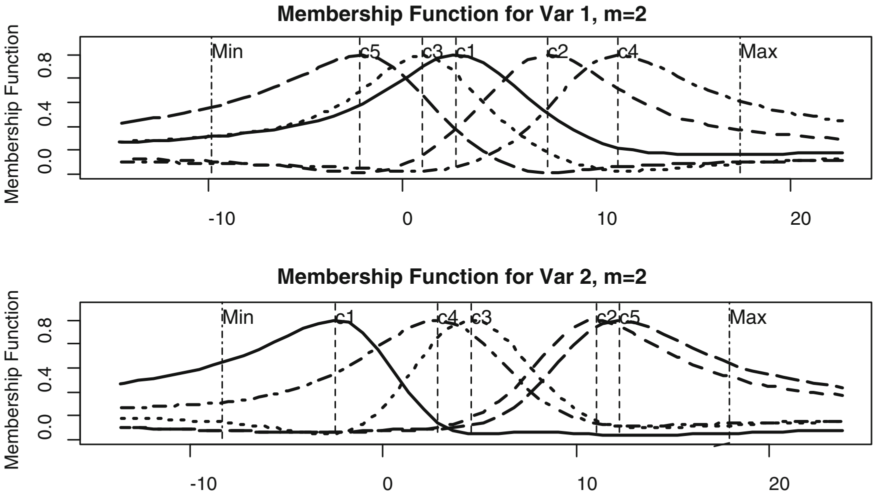


Fig. 2 Membership values for sampled values 6 with the level of fuzziness, m = 2

easily be identified. However the recognition becomes increasingly difficult for data sets with higher values.

Figure 2 shows the membership functions for the sixth sample using the value of the level of fuzziness 2. The implementation of this figure is given as fig2.r file ("*plotMemcmeans"* function is given as "*plotmemcmeans.r" file*).

3.2 Real-World Data

Aside from the experiments that are designed to be simple yet informative, some real-world data sets are used for presentation. To this extend, selected cluster validity indices are applied to "*wine*" and "*iris*" data sets. Both data sets can be downloaded from the web site of, Machine Learning Repository of University of California, Irvine.[8] "*iris*" data set is readily available within R software. "*wine*" data set can be downloaded as shown in "*iris and wine data ex.r*" file.

Table 1 shows the summary statistics of *wine* data set. Since variables statistics are significantly different to each other, variables are scaled to have zero mean and unity variance (and they are not decorrelated) before clustering algorithm is applied.

This data set contains 178 observations each 13 variables and three classes (cultivars) named 1, 2, and 3 with 59, 71, and 48 observations each.

[8] http://www.ics.uci.edu/~mlearn/databases.html

Table 1 Summary statistics for wine data

	Minimum	First Qu.	Median	Mean	Third Qu.	Maximum
Alcohol	11.03	12.36	13.05	13	13.68	14.83
Malic acid	0.74	1.603	1.865	2.336	3.083	5.8
Ash	1.36	2.21	2.36	2.367	2.558	3.23
Alkalinity of ash	10.6	17.2	19.5	19.49	21.5	30
Magnesium	70	88	98	99.74	107	162
Total phenols	0.98	1.742	2.355	2.295	2.8	3.88
Flavanoids	0.34	1.205	2.135	2.029	2.875	5.08
Nonflavanoid phenols	0.13	0.27	0.34	0.3619	0.4375	0.66
Proanthocyanins	0.41	1.25	1.555	1.591	1.95	3.58
Color intensity	1.28	3.22	4.69	5.058	6.2	13
Hue	0.48	0.7825	0.965	0.9574	1.12	1.71
OD280/OD315 of diluted wines	1.27	1.938	2.78	2.612	3.17	4
Proline	278	500.5	673.5	746.9	985	1680

Table 2 Summary statistics for iris data

	Minimum	First Qu.	Median	Mean	Third Qu.	Maximum
Sepal length	4.3	5.1	5.8	5.843	6.4	7.9
Sepal width	2	2.8	3	3.057	3.3	4.4
Petal length	1	1.6	4.35	3.758	5.1	6.9
Petal width	0.1	0.3	1.3	1.199	1.8	2.5

Summary statistics of iris data is shown in Table 2. Statistics suggest performing scaling before clustering. There are 150 observations with four variables and three types of species.

4 Results

4.1 Artificial Data Sets

Table 3 shows the numbers of clusters found by different cluster validity indices for each of the nine artificially generated data sets. PE, PC, Xie-Beni and Fukuyama and Sugeno's cluster validation indices are used for comparison. Although our intention is not to compare the result, for the sake of presentation it is given. All these indices are calculated with the values of the level of fuzziness, 1.4, 2, and 2.6. The value 2 is rule of thumb and others are lower and upper bound values suggested by Ozkan and Turksen [33]. The list of observations is:

- l_∞ indices perform well. It seems the lower half of the range, using the value of the level of fuzziness 2 then changing to 1.4 works slightly better. Instead

Table 3 Validation indices performances

Artificially generated data set	l_∞ : 1[a]	l_∞ : 2[a]	F–S[a,b]	XB[c]	PE[b]	PC[b]
$1 - \sigma^2 = 2$	5	5	5	5	5	5
$2 - \sigma^2 = 3$	5	5	5	5	5	5
$3 - \sigma^2 = 4$	5	5	5	5	3	5
$4 - \sigma^2 = 5$	5	5	5	5	3	3
$5 - \sigma^2 = 6$	5	5	5	5	3	3
$6 - \sigma^2 = 7$	5	5	5	5	3	3
$7 - \sigma^2 = 8$	5	5	8	5	3	3
$8 - \sigma^2 = 9$	5	5	–	5	3	3
$9 - \sigma^2 = 10$	5	5	6	5	3	3

[a] Results for 1:$m_l = 1.4$, $m_u = 2.6$ 2: $m_l = 1.4$, $m_u = 2$. Manhattan distance works well too. In general setting first to upper bound, then changes to the lower bound values of the level of fuzziness performs well. F–S: Fukuyama-Sugeno index

[b] Partition Coefficient identifies correctly up to number $\sigma^2 = 4$. However when the level of fuzziness is set to 1.4, it identifies the number of clusters up to $\sigma^2 = 8$. Partition Entropy identifies correctly up to number $\sigma^2 = 3$. However when the level of fuzziness is set to 1.4, it identifies the number of clusters up to $\sigma^2 = 7$

[c] Xie-Beni's index performs well only when the value of the level of fuzziness is set to 2. The first local min is taken as the optimum number of clusters.

of global minimum, using the first local minimum is the best decision criterion.

- Xie-Beni's index performs quite well when the value of the level of fuzziness is set to two. But it gets better when the value of the level of fuzziness is set to upper bound if the first local minimum is taken as the number of clusters. The result is poorer when the value of the level of fuzziness is 1.4.
- Fukuyama and Sugeno's index also performs well when the first local minimum is used as the decision criterion. Similar to Xie-Beni's index, the performance is increasing with the value of the level of fuzziness up to the upper bound.
- Partition Coefficient and Entropy indices perform better when the value of the level of fuzziness is set to the lower bound.
- The performances of all indices are good up to $\sigma^2 = 7$ when their best performances are used.

4.2 Real-World Data Sets

4.2.1 Iris Data Set

Figure 3 shows the results of *MiniMax ε-stable index* for *iris* data set. This validity index identifies three clusters since the first (local) minimum is at three

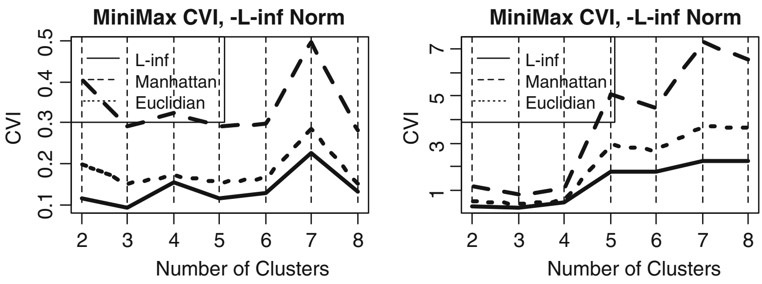


Fig. 3 MiniMax CVI for iris data l_∞ norms. *Left* $m_1 = 2$, $m_2 = 1.4$, *Right* $m_1 = 2.6$, $m_2 = 1.4$

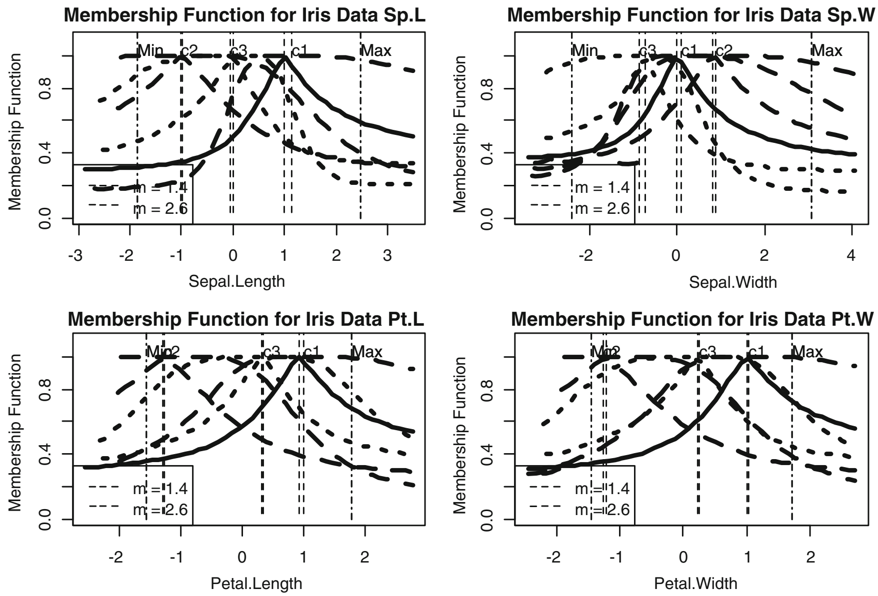


Fig. 4 Membership function for iris data set

and also there is another local minimum at six which is two times of the first minimums place as suggested by Ozkan and Turksen [34]. Hence, three is an optimum number of clusters based on authors' recommendation. There are also two other variants given in this figure. These are the Manhattan distance (l_1 norm) and the Euclidian distance (l_2 norm). Both identify the correct number of clusters as l_∞ norm for *iris* data set. The averaging (or taking all the changes in all dimensions into account may result in misleading information. Stability measure in this example of usage should not be the average). Setting the upper

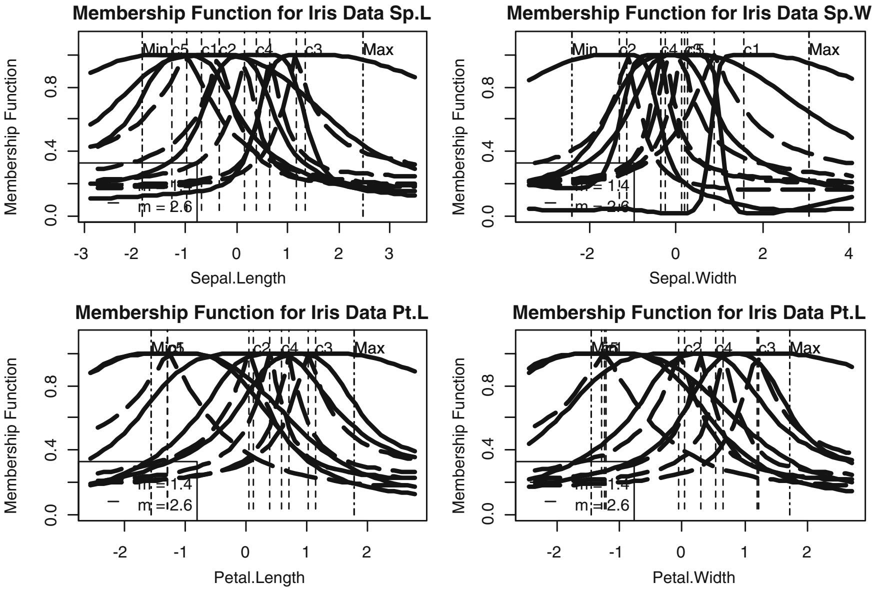


Fig. 5 Membership function of iris data with 5 clusters

and lower values of the level of fuzziness as $m_l = 1.4$, $m_u = 2.6$ or $m_l = 1.4$, $m_u = 2$ perform well.

Since *iris* data set is a pretty small data sample and the number of variables is four it may be informative to see the behavior of the membership function. Figure 4 shows the membership functions obtained for *iris* data set. These figures in fact demonstrate the existence of interval valued type-2 membership function obtained with an application of upper and lower fuzziness. It appears that, the change in the value of the '*sepal. length*' in cluster labeled as *c1* and the change in the value of the '*sepal. width*' in cluster labeled as *c3* are significantly larger as shown by vertical dotted lines. This change in cluster centers is the main idea for this cluster validity index. The intuition is that the cluster center values become more unstable if the number of clusters is set less than or greater than the optimal number of clusters.

Membership function of iris data set with 5 clusters is given in Fig. 5. One can observe that the significantly large changes in cluster center values in all dimensions as shown by vertical dotted lines. Hence the stability reduces if this data set is over-clustered.

Figure 6 shows the result of other selected cluster validity indices. Xie-Beni's indices for iris data set reach minimum when the number of clusters is set to three. Also, Fukuyama and Sugeno's indices for three different values of the level of fuzziness identify the optimal number of clusters as three. The behavior of all

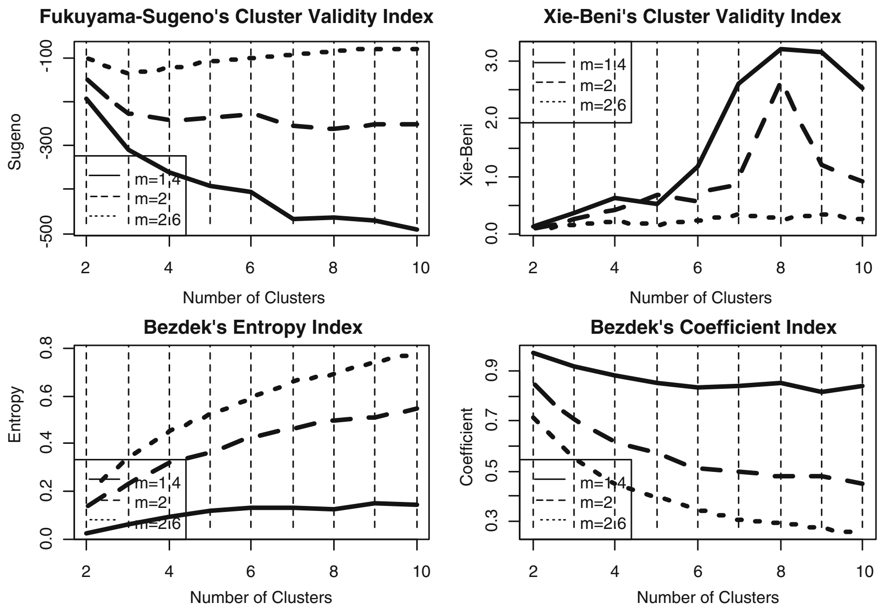


Fig. 6 Selected indices for *iris* data set

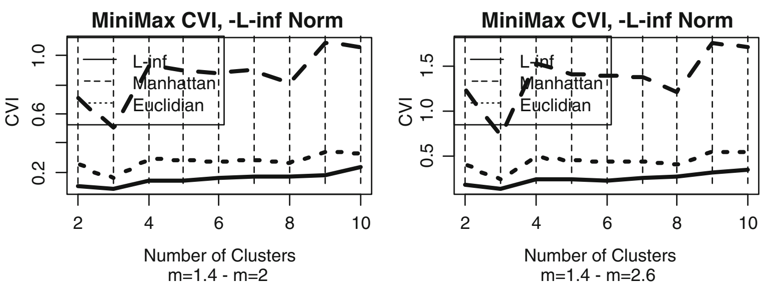


Fig. 7 MiniMax CVI for wine data l_∞ norms. *Left* $m_1 = 1.4$, $m_2 = 2$, *Right* $m_1 = 1.4$, $m_2 = 2.6$

indices clearly changes with the level of fuzziness. The list of observations given above seems valid for iris data set experiment.

4.2.2 Wine Data Set

Figures 7 and 8 show the result of *MiniMax ε-stable index* and other selected indices respectively for *wine* data set. In order to avoid repetition, the discussion given above is valid for this *wine* data experiment also. Since there are 13 variables, creating the membership functions and discussion about them are left to the reader.

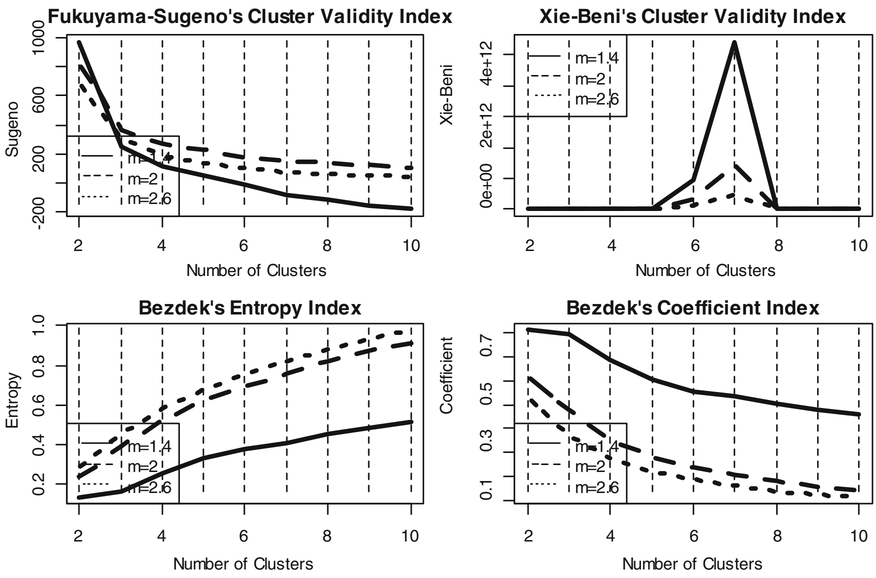


Fig. 8 Selected indices for wine data set

5 Conclusion

One of the aims of this chapter is to introduce a possible use of the interval valued type-2 assessment for cluster validation. To this end, a brief survey of cluster validation approaches including their basic taxonomy is given first. Then Ozkan and Turksen's *MiniMax ε-stable index* is introduced as a recent example of how type-2 fuzziness can be used for cluster validation. This approach is a novel one in the sense that it is the first example of the use of upper and lower bounds of the level of fuzziness used. This method seeks stability of the values of cluster centers with respect to the level of fuzziness. One can create many possible measures for cluster validation with an application of interval valued type-2 fuzziness. It appears that the extra information that can be extracted from interval valued type-2 fuzziness opens a new path for the extensions of the research already conducted for type-1 fuzzy systems.

The second aim of this chapter is to provide a platform for researchers and practitioners to let them fully reproduce all examples given in this chapter. All the source codes of the functions used in this chapter are written in freely available R[9] software. The data sets used in this chapter are (i) artificial data sets created by

[9] This software can be downloaded from http://cran.r-project.org/. There are also several good documents related to this statistical computing environment and there are more than 3500 packages prepared already.

multivariate sampling from normal distribution, (ii) *iris* ,and (iii) *wine* data sets. Iris data set is available in R. Wine data set is available on the Internet and can be downloaded within R. We believe researchers and practitioners can modify/extend the given source code for further experimentation and research.

It is reasonable to suggest that the level of the fuzziness is a very powerful parameter. It certainly helps us to understand both the relation between the data vectors and the overall structure within a data set.

Acknowledgments This work was partially supported by the Natural Science and Engineering Research Council (NSERC) Grant (RPGIN 7698-05) to University of Toronto. Also, partial support is provided by Hacettepe University and TOBB Economics and Technology University. Their support is greatly appreciated.

References

1. Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. In: Pacific Symposium on Biocomputing, vol. 7, pp. 6–17. World Scientific Publishing Co., New Jersey (2002)
2. Ben-Hur, A., Guyon, I.: Detecting stable clusters using principal component analysis. In: Brownstein, M.J., Kohodursky, A. (eds.) Methods in Molecular Biology, pp. 159–182 Humana Press, Clifton, (2003)
3. Bezdek, JC.: Fuzzy mathematics in pattern classification. Dissertation, Applied Mathematics Center, Cornell University, Ithaca, (1973)
4. Bezdek, J.C.: Cluster validity with fuzzy sets. J. Cybernet. **3**, 58–72 (1974)
5. Bezdek, JC.: Mathematical models for systematics and taxonomy. In: Estabrook, G. (ed.) Proceeding of 8th International Conference on Numerical Taxonomy, pp. 143–166. Freeman, San Francisco (1975)
6. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
7. Bolshakova, N., Azuaje, F.: Cluster validation techniques for genome expression data. Sig. Process **83**, 825–833 (2003)
8. Bolshakova, N., Azuaje, F., Cunningham, P.: A knowledge-driven approach to cluster validity assessment. Bioinformatics **21**, 2546–2547 (2005)
9. Boudraa, A.O.: Dynamic estimation of number of clusters in data set. Electron. Lett. **35**, 1606–1608 (1999)
10. Bouguessa, M., Wang, S., Sun, H.: An objective approach to cluster validation. Pattern Recogn. Lett. **27**, 1419–1430 (2006)
11. Breckenridge, J.: Replicating cluster analysis: method, consistency and validity. Multivar. Behav. Res. **24**, 147–161 (1989)
12. Brock, G.N., Pihur, V., Datta, S., Datta, S.: clValid: an R package for cluster validation. J. Stat. Softw. **251**, 22 (2008). http://www.jstatsoft.org/v25/i04
13. Celikyilmaz, A., Türksen, I.B.: Enhanced fuzzy system models with improved fuzzy clustering algorithm. IEEE Trans. Fuzzy Syst. **16**, 779–794 (2008)
14. Celikyilmaz, A., Türksen, I.B.: Validation criteria for enhanced fuzzy clustering. Pattern Recogn. Lett. **29**, 97–108 (2008)
15. Chen, M.Y., Linkens, A.: Rule-base self-generation and simplification for data-driven fuzzy models. J. Fuzzy Sets Syst. **142**, 243–265 (2004)
16. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **1**, 224–227 (1979)

17. Dudoit, S., Fridlyand, J.: A prediction based re-sampling method for estimating the number of clusters in a data set. Genome Biol. **3**, 1–21 (2002)
18. Dunn, J.C.: Well separated clusters and fuzzy partitions. J. Cybern. **4**(1974), 95–104 (1974)
19. Falasconi, M., Gutierrez, A., Pardo, M., Sberveglieri, G., Marco, S.: A stability based validity method for fuzzy clustering. Pattern Recogn. **43**, 1292–1305 (2010)
20. Fukuyama, Y., Sugeno, M.: A new method of choosing the number of clusters for the fuzzy c-means method. In: Proceedings of Fifth Fuzzy Systems Symposium, pp. 247–250 (in Japanese) (1989)
21. Gan, G., Chaoqun, M., Jianhong, W.: Data Clustering: Theory, Algorithms, and Applications. ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia, ASA, Alexandria (2007)
22. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering validity methods part I. ACM SIGMOD Rec. **31**, 40–45 (2002)
23. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering validity methods part II. ACM SIGMOD Rec. **31**, 19–27 (2002)
24. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Intell. Inf. Syst. J. **17**, 107–145 (2001). Kluwer Pulishers
25. Handl, J., Knowles, J., Kell, D.B.: Computational cluster validation in post-genomic data analysis. Bioinformatics **21**, 3201–3212 (2005)
26. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
27. Kim, D.W., Lee, K.H., Lee, D.: On cluster validity index for estimation of the optimal number of fuzzy clusters. Pattern Recogn. **37**, 2009–2025 (2004)
28. Kwon, S.K.: Cluster validity index for fuzzy clustering. Electron. Lett. **34**, 2176–2177 (1998)
29. Lange, T., Roth, V., Braun, M.L., Buhmann, J.M.: Stability based validation of clustering solutions. Neural Comput. **16**, 1299–1323 (2004)
30. Levine, E., Domany, E.: Resampling method for unsupervised estimation of cluster validity. Neural Comput. **13**, 2573–2593 (2001)
31. Mufti, G.B., Bertrand, P., El Moubarki, L.: Determining the number of groups from measures of cluster validity. In: Proceedings of ASMDA2005, pp. 404–414 (2005)
32. Ozkan, I., Türksen, I.B.: Entropy assessment of type-2 fuzziness. IEEE Int. Conf. Fuzzy Syst. **2**, 1111–1115 (2004)
33. Ozkan, I., Turksen, I.B.: Upper and lower values for the level of fuzziness in FCM. Inf. Sci. **177**, 5143–5152 (2007)
34. Ozkan, I., Turksen, I.B.: MiniMax ε-stable cluster validity index for type-2 fuzziness. Inf. Sci. **184**, 64–74 (2007)
35. Pal, N.R., Bezdek, J.C.: On cluster validity for the fuzzy c-means model. IEEE Trans. Fuzzy Syst. **3**, 370–379 (1995)
36. Pascual, D., Pla, F., Sánchez, J.S.: Cluster validation using information stability measures. Pattern Recogn. Lett. **31**, 454–461 (2010)
37. Rezaee, M.R., Lelieveldt, B.P.F., Reiber, J.H.C.: A new cluster validity index for the fuzzy c-means. Pattern Recogn. Lett. **19**, 237–246 (1998)
38. Salem, S.A., Nandi, A.K.: Development of assessment criteria for clustering algorithms. Pattern Anal. Appl. **12**, 79–98 (2009)
39. Shannon, C.E.: The mathematical theory of communication. Bell Syst. Tech. J. **27**,379–423 and **27**, 623–656 (in two parts) (1948)
40. Sugar, C., James, G.: Finding the number of clusters in a data set: an information theoretic approach. J. Am. Stat. Assoc. **98**, 750–763 (2003)
41. Sugar, C., Lenert, L., Olshen, R.: An application of cluster analysis to health services research: empirically defined health states for depression from the sf-12. Technical Report, Stanford University, Stanford (1999)
42. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. J. R. Stat. Soc. B Part 2, **63**, 411–423 (2001)

43. Tibshirani, R., Walther, G.: Cluster validation by prediction strength. J. Comput. Graph. Stat. **14**, 511–528 (2005)
44. Volkovich, Z., Barzily, Z., Morozensky, L.: A statistical model of cluster stability. Pattern Recogn. **41**, 2174–2188 (2008)
45. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. Fuzzy Sets Syst. **158**, 2095–2117 (2007)
46. Wu, K.L., Yang, M.S., Hsieh, J.N.: Robust cluster validity indexes. Pattern Recogn. **42**, 2541–2550 (2009)
47. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. **13**, 841–847 (1991)
48. Yu, J., Cheng, Q., Huang, H.: Analysis of the weighting exponent in the FCM. IEEE Trans. Syst. Man Cybern. B **34**, 634–639 (2004)
49. Zhang, Y., Wang, W., Zhang, X., Li, Y.: A cluster validity index for fuzzy clustering. Inf. Sci. **178**, 1205–1218 (2008)
50. Yue, S., Wang, J.S., Wu, T., Wang, H.: A new separation measure for improving the effectiveness of validity indices. Inf. Sci. **180**, 748–764 (2010)0

Type-2 Fuzzy Set and Fuzzy Ontology for Diet Application

Chang-Shing Lee, Mei-Hui Wang, Chin-Yuan Hsu and Zhi-Wei Chen

Abstract Nowadays, most people can get enough energy to maintain one-day activity, while few people know whether they eat healthily or not. It is quite important to analyze nutritional facts of foods eaten for those who are losing weight or suffering chronic diseases such as diabetes. However, diet is a problem with a high uncertainty, and it is widely pointed out that classical ontology is not sufficient to deal with imprecise and vague knowledge for some real-world applications like diet. On the other hand, a fuzzy ontology can effectively help handle and process uncertain data and knowledge. This chapter proposes a type-2 fuzzy set and fuzzy ontology for diet application and uses the type-2 fuzzy markup language (T2FML) to describe the knowledge base and rule base of the diet, including ingredients and the contained servings of six food categories of some common foods in Taiwan. The experimental results show that type-2 fuzzy logic system (FLS) performs better than type-1 FLS, proving that type-2 FLS can provide a powerful paradigm to handle the high level of uncertainties present in diet.

C.-S. Lee (✉) · M.-H. Wang · Z.-W. Chen
Department of Computer Science and Information Engineering,
National University of Tainan, Tainan, Taiwan
e-mail: leecs@mail.nutn.edu.tw

M.-H. Wang
e-mail: mh.alice.wang@gmail.com

Z.-W. Chen
e-mail: a301203034@gmail.com

C.-Y. Hsu
Chunghwa Telecom, Taipei, Taiwan
e-mail: hsucy0320@cht.com.tw

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6_15,

1 Introduction

According to the Food and Drug Administration, Department of Health, Executive Yuan, Taiwan [5], a balanced diet is defined as follows: All of the nutrients a human needs daily come from various kinds of foods. However, various types of foods do not provide the same nutrients, so each food group cannot replace each other. Unprocessed food has a higher priority than processed food because the former is much healthier than the latter. Additionally, according to the daily dietary guidelines suggested by [5] and according to each person's age and daily physical activity, a dietician is able to find suitable daily caloric requirement and then plan the unique dietary goal for a person. Finally, a person can acquire balanced nutrition by following a balanced intake of six food groups and various choices from each food group. However, each dietician might have different views on the daily caloric requirement and dietary goal even for the same person. What is more, dietary habits might have something to do with religion, culture, or personal preferences. Consequently, diet is a problem with high uncertainty and vagueness, and also a highly personalized problem.

Because of the highly personalized feature of diet, ontology is a good method to construct unique dietary information for each unique human. Ontology is an explicit specification of a conceptualization and it is also a knowledge representation and communication model for intelligent agents. However, it is widely pointed out that classical ontology is not sufficient to deal with imprecise and vague knowledge for some real-world applications. A fuzzy ontology is an extension of the classical ontology that is more suitable to describe the domain knowledge for solving the uncertainty in reasoning problems [9, 10]. As a result, a fuzzy ontology can effectively help handle and process uncertain data and knowledge. There has been considerable research on the fuzzy ontology: Trappey et al. [18] presented a novel hierarchical clustering approach for knowledge document self-organization, particularly for patent analysis. Bobillo and Straccia [4] proposed a concrete methodology to represent fuzzy ontologies using web ontology language (OWL) 2 to deal with vagueness or imprecision in the knowledge of Semantic web. Afacan and Demirk [3] proposed an ontology-based universal design knowledge system to support the cognitive activities of universal design process. Gaeta et al. [6] proposed an integrated approach to manage the life cycle of ontologies to model educational domains and to build, organize, and update specific learning resources. Lee et al. [9, 10] proposed a type-2 fuzzy ontology to apply to a personal diabetic–diet recommendation and a fuzzy ontology for news summarization.

Type-2 Fuzzy Logic Systems (FLSs) could be used to handle uncertainties in the group decision-making process as they can model the uncertainties between expert preferences using type-2 fuzzy sets. A type-2 fuzzy set is characterized by a fuzzy Membership Function (MF), i.e., the membership value (or membership grade) for each element of this set is a fuzzy set in [0,1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0,1] [9, 15, 16, 17]. Fuzzy Markup Language (FML) permits to model only type-1 fuzzy logic controllers with great success, but real-world environments and applications are usually with high level of uncertainties [7]. Hence, type-2 FLSs have the potential to overcome the limitations of type-1 FLSs and to produce a new generation of fuzzy systems with improved performance for many applications required to handle high levels of uncertainty [7, 15]. For this reason, Mendel et al. [15, 16] proposed type-2 fuzzy systems to overcome the limitations of type-1 Fuzzy Logic Controllers (FLCs). Therefore, in order to allow FML to model also type-2 FLCs in a transparent way, an extension of FML, named type-2 FML (T2FML), dealing with type-2 fuzzy sets, is defined [11]. Since the mathematics needed for interval type-2 fuzzy sets is simpler than the ones needed for general type-2 fuzzy sets, the interval type-2 fuzzy sets are more popular. Consequently, T2FML describes type-2 fuzzy systems based on interval type-2 fuzzy sets.

A balanced diet must contain correctly proportioned carbohydrates, proteins, fats, vitamins, mineral salts, and fiber because each nutrient plays an important role in supporting humans' everyday physical activities [13, 14]. However, what is a balanced diet? According to [5, 8], eating a balanced diet means choosing a wide variety of foods and drinks from all the food groups. It also means eating certain things in moderation. The goal of a balanced diet is to take in nutrients you need for health at the recommended levels. However, culture, religion, and lifestyle deeply affect people's diet and each person has a specific eating habit such that the diet behavior is highly personalized [13, 14]. Consequently, this chapter proposes a T2FML to describe the knowledge base and rule base of T2FML-based diet healthy assessing system. Additionally, this chapter also tries to compare the performance between type-1 FLSs and type-2 FLSs to show that integrating type-2 fuzzy sets with fuzzy ontology is suitable for dietary applications. The remainder of this chapter is as follows. Section 2 describes the fuzzy ontology and type-2 fuzzy markup language, and Sect. 3 introduces the T2FML-based fuzzy inference for dietary assessment to obtain the dietary healthy level of the food eaten. Section 4 shows some experimental results and conclusions are given in Sect. 5.

2 Fuzzy Ontology and Type-2 Fuzzy Markup Language

This section introduces an extension of the previous works on domain ontology [12] and fuzzy ontology [10] to describe the fuzzy concepts and relations in the dietary domain. Section 2.1 introduces the fuzzy food ontology and fuzzy personal food ontology,, and then the type-2 fuzzy set and type-2 fuzzy markup language are described in Sects. 2.2 and 2.3, respectively.

2.1 Fuzzy Food Ontology and Fuzzy Personal Food Ontology

A fuzzy ontology is a knowledge representation model for describing the uncertain domain knowledge, such as diet. Different people eat different types of food. Food items are divided into six major groups: grains and starches, vegetables, fruits, milk, meats and proteins, and fats. Each portion of food contains some information such as servings of six food groups, nutrition facts, and contained calories. If there is such an ontology existing to represent the above-mentioned information, then people can easily know the calories of each food item that are high, medium, or low for an adult or a child. Based on such ideas, we provide illustrative examples of the fuzzy food ontology and fuzzy personal food ontology in Figs. 1 and 2, respectively.

- Fuzzy Food Ontology

The daily dietary guideline from [5] suggests that a person should eat 1.5–4 bowls of grains and starches (1 bowl = 200 ml), 3–5 plates of vegetables (1 plate = 1 serving), 2–4 servings of fruits, 1.5–2 glasses of milk (1 glass = 240 ml = 1 serving), 3–8 servings of meats and proteins, and 3–7 teaspoons of fats. Besides, according to [5], each gram of carbohydrate, protein, and fact contains 4 kcal, 4 kcal, and 9 kcal, respectively, and the standard of one serving of each food group is listed in Table 1. There are various kinds of food items to choose from for one's daily meal. For example, rice is a very common food item in Taiwan, and one bowl of rice

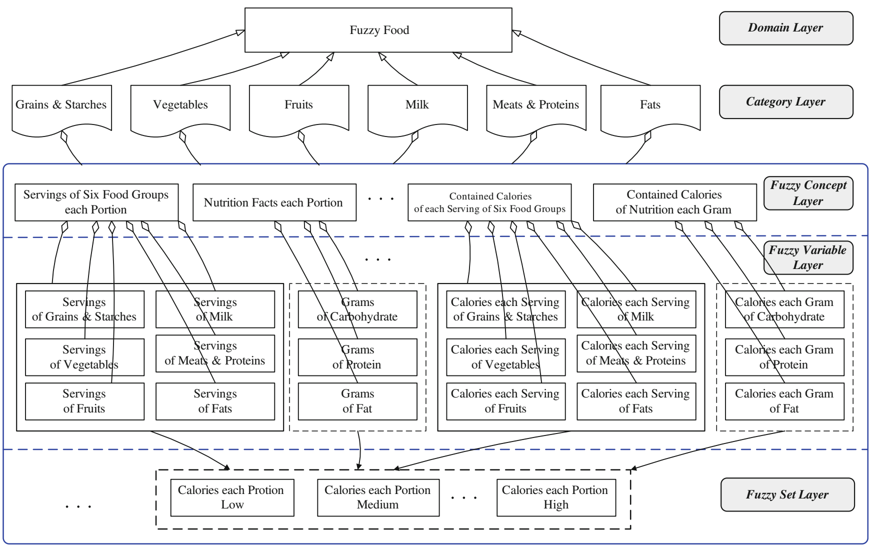


Fig. 1 Structure of the fuzzy food ontology [14].

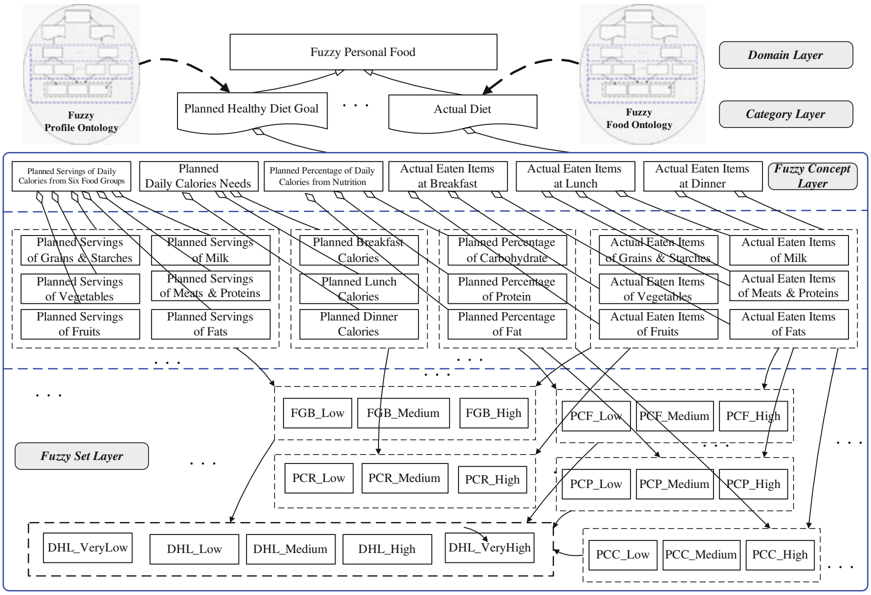


Fig. 2 Structure of the fuzzy personal ontology [14].

(=one portion of rice) contains 4 servings of the grains the starches group. That is, consuming one bowl of rice can produce about 280 kcal.

According to the knowledge mentioned in Table 1, the fuzzy food ontology is constructed in Fig. 1. The domain layer represents the domain name of the fuzzy food ontology. The categories in the category layer include Grains and Starches, Vegetables, Fruits, Milk, Meats and Proteins, and Fats. The fuzzy food ontology contains four concepts, Servings of Six Food Groups each Portion (SSFGP), Nutrition Facts each Portion (NFP), Contained Calories of each Serving of Six Food Groups (CCSSFG), and Contained Calories of Nutrition each Gram (CCNG). The fuzzy variables, Servings of Grains and Starches (SGS), Servings of Vegetables (SV), Servings of Fruits (SF), Servings of Milk (SM), Servings of Meats and

Table 1 Standard of one serving of each food group [5]

One serving of each food group	Nutrients (g)			Calories(kcal)
	Carbohydrate	Protein	Fat	
Grains and Starches	15	2	+	70
Vegetables	5	1	0	25
Fruits	15	+	0	60
Milk	12	8	4	120
Meats and Proteins	+	7	5	75
Fats	0	0	5	45

Note + means minute quantity

Proteins (SMP), and Servings of Fats (SF), are defined in the fuzzy concept Servings of Six Food Groups each Portion (SSFGP). In the fuzzy set layer, there are three fuzzy sets, including Calories each Portion Low (CPL), Calories each Portion Medium (CPM), and Calories each Portion High (CPH), defined to describe the linguistic meaning of calories of the food item.

- Fuzzy Personal Food Ontology

According to [5], the daily calories that a person needs to digest are mainly decided according to his/her age, weight, height, and physical activity. For example, if a man aged 19–30 likes outdoor sport, such as swimming, hiking, playing tennis, and so on, then his daily physical activity is high and caloric requirement should be about 2700 kcal. With this information, the dietician plans that this man should digest 4 servings of grains and starches group, 5 plates of vegetables group, 4 servings of fruits group, 2 glasses of milk group, 8 servings of meats and proteins group, and 8 servings of fats group per day. However, sometimes the consumed calories are not exactly equal to the planned ones so that some people need to gain weight and others lose weight, instead. In this chapter, we divide one-day meals into breakfast, lunch, and dinner in order to simplify the problem. With planned servings of daily calories from the six food groups, planned daily calories needs, planned percentage of daily calories from nutrition, and three-meal actual eaten items, the possibility of daily healthy level diet can be known.

With the above-mentioned knowledge, the fuzzy personal food ontology, shown in Fig. 2, is constructed to describe personal planned calories and one-day actual eaten items. The domain name is fuzzy personal food. There are two categories, Planned Healthy Diet Goal and Actual Diet, in the category layer defined in this ontology. In addition, the fuzzy concept layer contains six fuzzy concepts, including Planned Servings of Daily Calories from Six Food Groups, Planned Daily Calories Needs, Planned Percentage of Daily Calories from Nutrition, Actual Eaten Items at Breakfast, Actual Eaten Items at Lunch, and Actual Eaten items at Dinner. The fuzzy variables, Planned Percentage of Carbohydrate, Planned Percentage of Protein, and Planned Percentage of Fat, are defined for the concept Planned Percentage of Daily Calories from Nutrition. The fuzzy sets, "PCC_Low, PCC_Medium, PCC_High", "PCP_Low, PCP_Medium, PCP_High", "PCF_Low, PCF_Medium, PCF_High", "PCR_Low, PCR_Medium, PCR_High", and "FGB_Low, FGB_Medium, FGB_High," are defined to describe the linguistic meaning of Percentage of Calories from Carbohydrate (PCC), Percentage of Calories from Protein (PCP), Percentage of Calories from Fat (PCF), Percentage of Caloric Ratio (PCR), and Food Group Balance (FGB), respectively. Based on the information from PCC, PCP, PCF, PCR, and FGB, the Dietary Healthy Level (DHL) can be obtained.

2.2 Type-2 Fuzzy Set

Because the words mean different things to different people, T2FLS has the potential to provide better performance than a type-1 FLS (T1FLS) when such linguistic uncertainties are presented [15]. A T2FS $\tilde{A}$ is characterized by a type-2 membership function $\mu_{\tilde{A}}(x, u)$ where $x \in X$, $u \in J_x \subseteq [0, 1]$, and $\tilde{A}$ is denoted by Eq. 1 [9]. $\tilde{A}$ can also be expressed by Eq. 2 for discrete universes of discourse. When all $\mu_{\tilde{A}}(x, u)$=1, then $\tilde{A}$ is an interval T2FS (IT2FS) [9, 15, 17]. Figure 3 shows an example of type-2 fuzzy set $\tilde{A}$ which is with the upper bound of $FOU(\tilde{A})$, called A_U, and the lower bound of $FOU(\tilde{A})$, called A_L. $\tilde{A}$ is represented by the following parameters on the x-axis $\{\tilde{l}, \tilde{m}_l, \tilde{m}_r, \tilde{r}\}$ = $\{[l_L, m_{l_L}, m_{r_L}, r_L], [l_U, m_{l_U}, m_{r_U}, r_U]\}$. A_U and A_L are defined in Eqs. 3 and 4, respectively [9]. Unlike a type-1 fuzzy set whose membership degree is a crisp value in [0, 1], a type-2 membership grade can be described as any fuzzy subset in [0, 1]. This fuzzy subset is called the primary membership. And, for each primary membership, there is a secondary grade that defines the possibility for the primary membership [9, 15, 17].

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

$$\tilde{A} = \sum_{x \in X} \sum_{\mu \in J_x} \mu_{\tilde{A}}(x, \mu)/(x, \mu) \tag{2}$$

$$A_U(x) = [l_U, m_{l_U}, m_{r_U}, r_U] = \begin{cases} 0, \ x < l_U \\ (x - l_U)/(m_{l_U} - l_U), l_U \le x \le m_{l_U} \\ 1, \ m_{l_U} \le x < m_{r_U} \\ (r_U - x)/(r_U - m_{r_U}), m_{r_U} \le x \le r_U \\ 0, \ x > r_U \end{cases} \tag{3}$$

$$A_L(x) = [l_L, m_{l_L}, m_{r_L}, r_L] = \begin{cases} 0, \ x < l_L \\ \beta(x - l_L)/(m_{l_L} - l_L), l_L \le x \le m_{l_L} \\ \beta, \ m_{l_L} \le x < m_{r_L} \\ \beta(r_L - x)/(r_L - m_{r_L}), m_{r_L} \le x \le r_L \\ 0, \ x > r_L \end{cases} \tag{4}$$

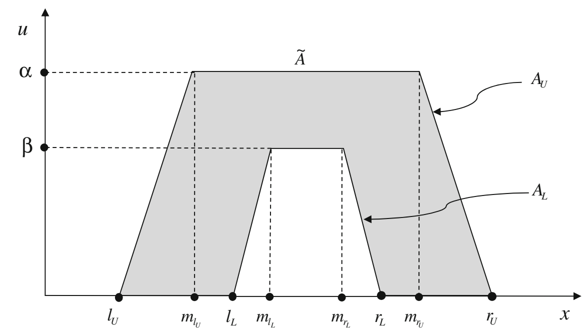


Fig. 3 Example of type-2 fuzz set [9]

2.3 Type-2 Fuzzy Markup Language

Fuzzy Markup Language (FML) is a fuzzy-based markup language that can manage fuzzy concepts, fuzzy rules, and a fuzzy inference engine [1, 2]. The fuzzy knowledge base contains the domain knowledge used by human experts. The fuzzy rule base represents the set of relations among fuzzy variables and fuzzy sets defined in the controller system. The inference engine is the fuzzy controller component able to extract new domain knowledge from fuzzy knowledge base and fuzzy rule base [1]. Additionally, FML is composed of three layers—an Extensible Markup Language (XML), a document-type definition, and extensible stylesheet language transformations. It is also one of the most important results because it allows fuzzy scientists to express their ideas in an abstract and interoperable way by improving their productivity and, at the same time, increasing the average quality of their works [2]. There has been considerable research on FML applications, such as Lee et al. [11, 19] applied FML to dietary domain, and Wang et al. [20] applied to Electrocardiogram (ECG) domain to express the ECG's knowledge base and rule base between before and after examinations. In order to allow FML to model type-2 FLCs in a transparent way, an extension of FML, named type-2 FML (T2FML), dealing with type-2 fuzzy sets, is defined [11]. In particular, since FML is a markup language, its extension is realized by adding some tags and attributes useful to model type-2 fuzzy sets and the inference features of a type-2 FLC. The definitions of the tags are described as follows: [11]:

- **<FuzzyController>**: The root of fuzzy controller taxonomy, the Controller node, is represented through the FML tag <FuzzyController> . Such tag represents the root tag of T2FML programs, that is, the opening tag of each T2FML program.
- **<KnowledgeBase>**: The fuzzy knowledge base is defined by means of fuzzy concepts used to model the fuzzy rule base. In order to define the type-2 fuzzy concept related controlled system, <KnowledgeBase> tag uses a set of nested tags:
  - <**Type2FuzzyVariable**>: This tag is used to define the type-2 fuzzy concept.
  - <**Type2FuzzyTerm**>: This tag is used to define a linguistic term describing the type-2 fuzzy concept, nested in <**Type2FuzzyVariable**> tag.
  - <**Type2TriangularShape**>, <**Type2PIShape**>, <**Type2GaussianShape**>, <**Type2LinearShape**>, <**Type2TrapezoidShape**>, <**Type2SShape**>, <**Type2ZShape**>: These tags are used to define the shape of a type-2 fuzzy set, including the shape of triangular, *PI*, Gaussian, linear, trapezoid, *S*, or *Z*.
  - <**UMF**> and <**LMF**>: Every shaping tag uses these two nested tags to define the upper membership function (UMF) and the lower membership function (LMF) of a term of type-2 fuzzy concept, represented by two type-1 fuzzy sets. The number of these attributes depends on the shape of the chosen fuzzy set.

- <**RuleBase**>: The root of fuzzy rule base component is modeled by the <RuleBase> tag.
  - <**Rule**>: Define the single rule.
  - <**Antecedent**> and <**Consequent**>: These two tags are used to define the antecedent and consequent rule parts, respectively.
  - <**Clause**>: This tag is used to model the fuzzy clauses in antecedent and consequent parts.
  - <**Variable**>, <**Term**>, and <**TSKParam**>: The pair "<Variable> , <Term>" is used to define fuzzy clauses in antecedent and consequent parts of Mamdani controllers rules as well as in antecedent part of TSK controllers rules. While, the pair "<Variable>, <TSKParam>" is used to model the consequent part of TSK controllers rules.

3 T2FML-Based Fuzzy Inference Mechanism for Dietary Assessment

This section utilizes the predefined fuzzy food ontology and fuzzy personal food ontology to perform the T2FML-based fuzzy inference mechanism for dietary assessment. The system structure is first introduced in Sect. 3.1, then some considered factors for dietary healthy level, T2FML view of the dietary assessment, and type-2 fuzzy inference mechanism are described in Sects. 3.2, 3.3, and 3.4, respectively.

3.1 System Structure

Figure 4 shows the structure of the T2FML-based fuzzy inference mechanism for dietary assessment and it operates as follows [21]:

Step 1. The ingredients extraction mechanism retrieves the nutrition facts of each eaten item from the food ontology.

Step 2. The nutrients analysis mechanism diagnoses the percentage of calories from carbohydrates, proteins, and fats for one meal.

Step 3. The calorie computation mechanism calculates the caloric difference between actual calories people consume and planned caloric intake set by dieticians.

Step 4. The balance evaluation mechanism is to consider the balance of six food groups. That is, the more variety the eaten food contains, the more balanced the eaten food is.

Step 5. The type-2 fuzzy inference mechanism infers the dietary healthy level of the eaten food based on the predefined ontology and the outputs from steps 1 to 4.

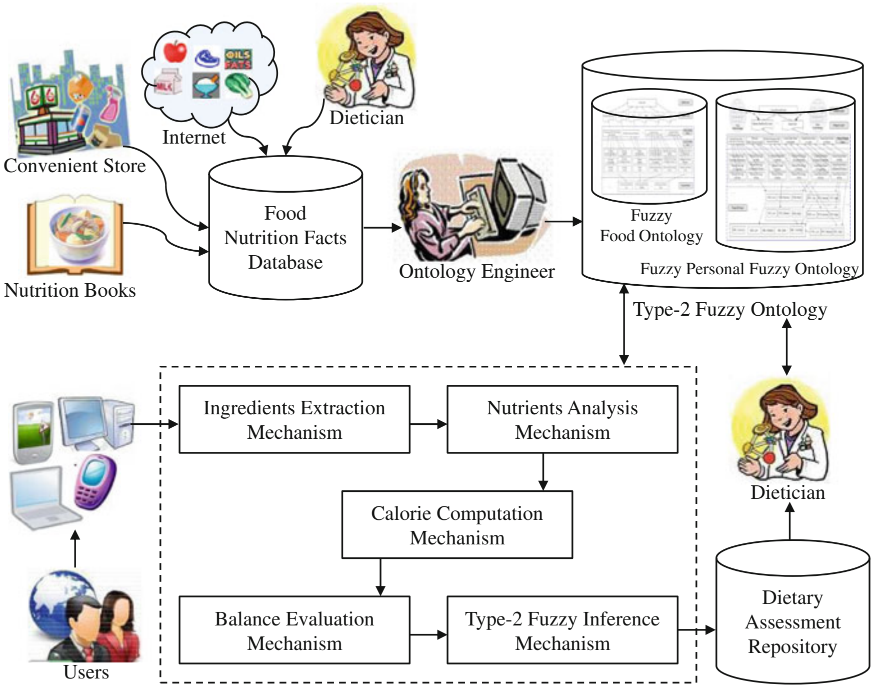


Fig. 4 System structure [21]

Step 6. Finally, the dietary healthy level is stored into the dietary assessment repository and validated by the dieticians to evaluate the performance of the proposed approach.

3.2 Considered Factors for Dietary Healthy Level

PCC, PCP, PCF, PCR, and FGB of everyday meals are acquired from the collected meal records and the pre-defined type-2 fuzzy food ontology. The detailed information about these five fuzzy variables is described as follows:

- PCC, PCP, and PCF

In order to support one-day energy, it is necessary to consume the required calories via the everyday diet. However, the main source of the calories is decided by the fact of how many grams of the carbohydrate, protein, and fat the food contains. As a result, to eat healthily, it is necessary for people to consider the balance of these three main nutrients, namely the carbohydrate, protein, and fat, when consuming your one-day meals. For this reason, this chapter sets PCC, PCP, and PCF to represent the percentage of calories from carbohydrate, protein, and fat, respectively. According to [5, 8], it indicates that (1) the suggested percentages of the daily intake for carbohydrate, protein, and fat are 55∼65 %, 10∼20 %, and 25∼35 %, respectively, and (2) the contained calories for each gram of the carbohydrate, protein, and fat are 4 kcal, 4 kcal, and 9 kcal, respectively. Hence, the values of PCC, PCP, and PCF are defined in Eqs. 5, 6, and 7, respectively [13, 14].

$$PCC = \frac{\text{Contained Grams of Carbohydrate } \times 4}{\text{Total Consumed Carloies}} \times 100\,\% \tag{5}$$

$$PCP = \frac{\text{Contained Grams of Protein } \times 4}{\text{Total Consumed Carloies}} \times 100\,\% \tag{6}$$

$$PCF = \frac{\text{Contained Grams of Fat } \times 9}{\text{Total Consumed Carloies}} \times 100\,\% \tag{7}$$

- PCR

In addition to considering the balance of the three main nutrients, the ratio of actual calories people consume to the planned caloric intake set by nutritionists is also an important factor to affect the healthy diet, especially for the fat and the slim. According to the involved domain experts' opinions, the one-day actually consuming calories for common people should be between 90 and 110 % of the planned calories set by the dieticians. The domain experts set a planned goal for each person according to individual's height, weight, and daily physical activities. Based on the planned goal and the actual acquiring energies, the value of PCR is obtained by Eq. 8 [13, 14].

$$PCR = \frac{\text{Actual Consuming Calories}}{\text{Planned Consuming Calories}} \times 100\,\% \tag{8}$$

- FGB

The balance of six food groups is the most important factor to decide how much at healthy level the eaten meals are. This is because the lack of some beneficial nutrients for our bodies such as fiber cannot be known if we only consider the balanced intake of carbohydrate, protein, and fat. So, in addition to planning the daily calories, dieticians also plan one-day unique consuming servings of each food group for each person. Moreover, the contained quantity of sugar also affects the balance of intake-in food. Owing to the present-day living style, most people purchase processed foods as their usual meals from shopping malls. However, unfortunately, processed foods frequently contain much more sugar than natural foods, and this is harmful to our bodies. The suggested percentage of the daily sugar intake is less than 10 % of the planned caloric intake each day. The value of FGB is calculated by Eq. 9 [13, 14].

$$FGB = BL_{\text{Grains\&Starches}} + BL_{\text{Vegetables}} + BL_{\text{Fruits}} + BL_{\text{Milk}} + BL_{\text{Meats\&Proteins}} + BL_{\text{Fats}} - CL_{\text{Sugar}} \quad (9)$$

where (1) $BL_{\text{Grains\&Starches}}$, $BL_{\text{Vegetables}}$, BL_{Fruits}, BL_{Milk}, $BL_{\text{Meats\&Proteins}}$, and BL_{Fats} represent the balanced level of the food groups grains and starches, vegetables, fruits, milk, meats and proteins, and fats, respectively. The higher the value, the healthier the meals are; (2) CL_{Sugar} denotes the contained level of sugar for the meals. The lower the value, the healthier the meals are; and (3) FGB value lies in the interval [0, 6] [13, 14].

3.3 T2FML View of Dietary Assessment

Based on the T2FML, a T2FML editor, developed by the Ontology Application and Software Engineer (OASE) Laboratory, National University of Tainan (NUTN), Taiwan, is used to construct the knowledge base and rule base of the T2FML-based fuzzy inference mechanism for dietary assessment. The knowledge base describes fuzzy concepts, including fuzzy variables, fuzzy terms, and membership functions of fuzzy sets. And, the rule base describes the fuzzy rule sets, including the antecedent and consequent rules. Figure 5 shows the controller tree of T2FML-based fuzzy inference mechanism for dietary assessment. Table 2 lists the partial knowledge base and rule base of the proposed system. It has five input fuzzy variables—PCC, PCP, PCF, PCR, FGB, one output fuzzy variable DHL, and 243 fuzzy rules. Each fuzzy variable has several fuzzy terms.

3.4 Type-2 Fuzzy Inference Mechanism

How to embed domain experts' knowledge into the design of the fuzzy rules is an important issue. The involved dieticians consider that the FGB plays the most important role on the dietary healthy level. But, FCR is as important as FGB when people who are losing weight or gaining weight are neglected. As for PCC, PCP, and PCF, generally speaking, these three factors are equally important, but the dietary healthy level becomes lower when PCF is not within the balanced range. Eventually, the involved domain experts determine that the fuzzy relation ratio of PCC:PCP:PCF:PCR:FGB is 1:1:2:3:3. Table 3 shows fuzzy relation of each input fuzzy variable. Take one fuzzy rule as example. Assume there is one fuzzy rule, and the if-part is "IF PCC is Medium AND PCP is Medium AND PCF is Medium AND PCR is Medium AND FGB is Medium." Computed by Eq. 10, this fuzzy rule's score (S) can be obtained according to Table 3. After the normalization, the normalized rule score (S_N) is acquired by Eq. 11, where $S_{\min}$ and $S_{\max}$ are the

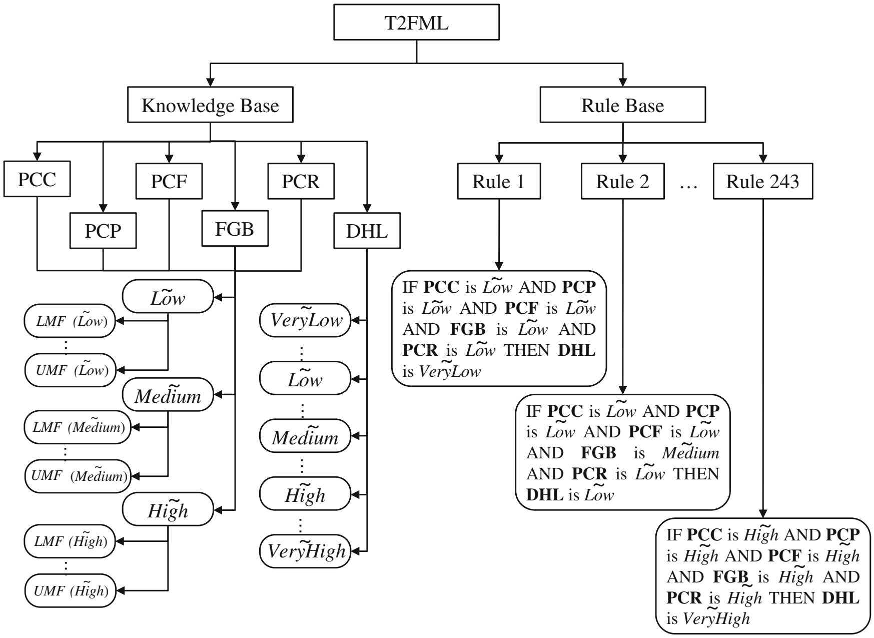


Fig. 5 Controller tree

minimum and maximum rule scores, respectively. Based on the normalized rule score, the then-part of the fuzzy rule could be determined by Table 4. Finally, the completed fuzzy rule is represented as "IF PCC is Medium AND PCP is Medium AND PCF is Medium AND PCR is Medium AND FGB is Medium THEN DHL is VeryHigh" [13, 14].

$$S = \frac{1 \times 2 + 1 \times 2 + 2 \times 2 + 3 \times 2 + 3 \times 1}{1 + 1 + 2 + 3 + 3} = 1.7 \tag{10}$$

$$S_N = \frac{S - S_{\min}}{S_{\max} - S_{\min}} = 0.83 \tag{11}$$

DHL is then inferred by carrying out the proposed method to indicate how much healthy, namely very high, high, medium, low, or very low, the eaten meals are. Type-2 fuzzy inference mechanism for dietary assessment is composed of five components, including type-2 fuzzy ontology, fuzzifier, inference, type reducer, and defuzzifier [15]. Herein, the type-2 fuzzy ontology stores the established knowledge base and rule base that are provided by domain experts. Additionally, the Karnik–Mendel (KM) algorithms [15] are used to compute the centroids of type-2 fuzzy sets. Figures 6a–f show the type-2 fuzzy sets for fuzzy variables PCC, PCP, PCF, PCR, FGB, and DHL, respectively. Table 5 shows parameters of the T2FSs for fuzzy variables PCC, PCP, PCF, PCR, FGB, and DHL [21].

Table 2 Partial knowledge base and rule base of the proposed system

```
<?xml version = "1.0"?>
<FuzzyController ip ="localhost" name = "">
  <KnowledgeBase>
    <Type2FuzzyVariable domainleft = "0" domainright = "100" name = "PCC"
    scale = "percentage" type = "input">
     <Type2FuzzyTerm name = "Low" hedge = "Normal">
      <Type2TrapezoidShape>
       <UMF Param1 = "0" Param2 = "0" Param3 = "35" Param4 = "50"/>
       <LMF Param1 = "0" Param2 = "0" Param3 = "35" Param4 = "40"/>
      </Type2TrapezoidShape>
     </Type2FuzzyTerm>
  ...
   </KnowledgeBase>
  <RuleBase activationMethod = "MIN" andMethod = "MIN" orMethod = "MAX"
    name = "RuleBase1" type = "mamdani">
    <Rule name = "Rule1" connector = "and" weight = "1" operator = "MIN">
     <Antecedent>
      <Clause>
       <Variable>PCC </Variable>
       <Term>Low </Term>
      </Clause>
  ...
      <Clause>
       <Variable>DHL </Variable>
       <Term>VeryLow </Term>
      </Clause>
     </Consequent>
    </Rule>
  ...
  </RuleBase>
</FuzzyController>
```

Table 3 Fuzzy relation of each input fuzzy variable [13, 14]

No.	Fuzzy variable	Fuzzy relation weight	Fuzzy term (Fuzzy term weight)		
1	PCC	1	Low (1)	Medium (2)	High (1)
2	PCP	1	Low (1)	Medium (2)	High (1)
3	PCF	2	Low (1)	Medium (2)	High (0)
4	PCR	3	Low (1)	Medium (2)	High (1)
5	FGB	3	Low (0)	Medium (1)	High (2)

Table 4 S_N range of the output fuzzy variable [13, 14]

Fuzzy term	Very low	Low	Medium	High	Very high
S_N Range	[0, 0.2]	(0.2, 0.4]	(0.4, 0.6]	(0.6, 0.8]	(0.8, 1.0]

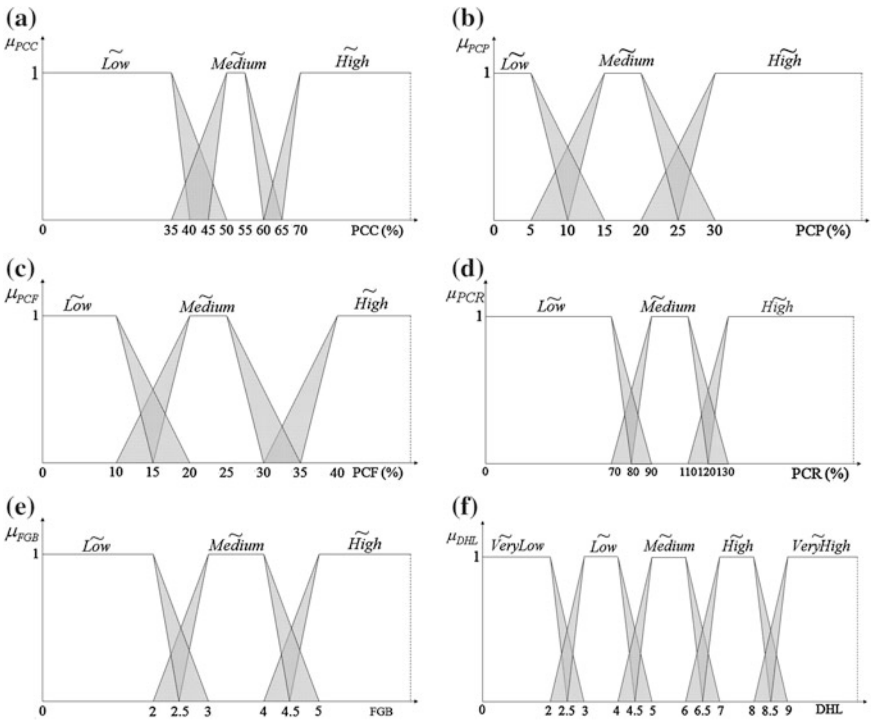


Fig. 6 T2FSs for fuzzy variables (**a**) PCC, (**b**) PCP, (**c**) PCF, (**d**) PCR, (**e**) FGB, and (**f**) DHL

4 Simulation Results

The system was implemented with the Microsoft Visual Studio C# programming language. We have conducted several experiments focusing on people who are aged from 20- to 30-years old and with the average levels of physical activity. Seven students of OASE Lab. of NUTN in Taiwan were involved in this experiment. They recorded their three meals from Monday to Friday for 20 days. The research performance presented in this chapter is a research project involving with NUTN and National Cheng Kung University Hospital in Taiwan.

Three factors, including PCR, FGB, and FR (Fuzzy Rule), are considered in the experiment to evaluate the performance of the proposed approach. Each factor has two different conditions, so totally eight main experiments are shown in Table 6. In addition, three domain experts, including DE_1, DE_2, and DE_3, are invited to involve this research project. Because each domain expert is requested to give his/her own opinions on FGB, Table 6 shows that Exps. 3, 4, 7, and 8 are with four sub-experiments, where A and B denote that input of the factors are gained by using the proposed approach in [21] and in this chapter [13], respectively. DE_1, DE_2, and DE_3 are responsible for providing his/her own opinions on FGB in (1) Exps. 3-1, 4-1, 7-1, & 8-1, (2) Exps. 3-2, 4-2, 7-2, & 8-2, and (3) Exps. 3-3, 4-3,

Table 5 Parameters of the T2FSs

Fuzzy variable	Linguistic term	Parameters of the T2FSs of the x-axis
		$\{[l_L, m_{l_L}, m_{r_L}, r_L], [l_U, m_{l_U}, m_{r_U}, r_U]\}$
PCC	$\tilde{Low}$	{[0, 0, 35, 40], [0, 0, 35, 50]}
	$\tilde{Medium}$	{[45, 50, 55, 60], [35 50 55 65]}
	$\tilde{High}$	{[65, 70, 100, 100], [60, 70, 100, 100]}
PCP	$\tilde{Low}$	{[0, 0, 5, 10], [0, 0, 5, 15]}
	$\tilde{Medium}$	{[10, 15, 20, 25], [5, 15, 20, 30]}
	$\tilde{High}$	{[25, 30, 100, 100], [20, 30, 100, 100]}
PCF	$\tilde{Low}$	{[0, 0, 10, 15], [0, 0, 10, 20]}
	$\tilde{Medium}$	{[15, 20, 25, 30], [10, 20, 25, 35]}
	$\tilde{High}$	{[35, 40, 100, 100], [30, 40, 100, 100]}
PCR	$\tilde{Low}$	{[0, 0, 70, 80], [0, 0, 70, 90]}
	$\tilde{Medium}$	{[80, 90, 110, 120], [70, 90, 110, 130]}
	$\tilde{High}$	{[120, 130, 200, 200], [110, 130, 200, 200]}
FGB	$\tilde{Low}$	{[0, 0, 2, 2.5], [0, 0, 2, 3]}
	$\tilde{Medium}$	{[2.5, 3, 4, 4.5], [2, 3, 4, 5]}
	$\tilde{High}$	{[4.5, 5, 6, 6], [4, 5, 6, 6]}
DHL	$\tilde{VeryLow}$	{[0, 0, 2, 2.5], [0, 0, 2, 3]}
	$\tilde{Low}$	{[2.5, 3, 4, 4.5], [2, 3, 4, 5]}
	$\tilde{Medium}$	{[4.5, 5, 6, 6.5], [4, 5, 6, 7]}
	$\tilde{High}$	{[6.5, 7, 8, 8.5], [6, 7, 8, 9]}
	$\tilde{VeryHigh}$	{[8.5, 9, 10, 10], [8, 9, 10, 10]}

Table 6 Experiments under different conditions

Factor	Exp. No							
	1	2	3-1 ~ 3-4	4-1 ~ 4-4	5	6	7-1 ~ 7-4	8-1 ~ 8-4
PCR	A	A	A	A	B	B	B	B
FGB	A	A	B	B	A	A	B	B
FR	A	B	A	B	A	B	A	B

Note *FR* means the abbreviation of the fuzzy rule
A means to use the method proposed in [21] to obtain *PCR*, *FGB*, and *FR*
B means to use the method proposed in this chapter to obtain *PCR*, *FGB*, and *FR*, that is, the method in [13]
The difference between (1) Exps. 1 & 5, (2) Exps. 2 & 6, (3) Exps. 3 & 7, and (4) Exps. 4 & 8 is *PCR*
The difference between (1) Exps. 1 & 3, (2) Exps. 2 & 4, (3) Exps. 5 & 7, and (4) Exps. 6 & 8 is *FGB*
The difference between (1) Exps. 1 & 2, (2) Exps. 3 & 4, (3) Exps. 5 & 6, and (4) Exps. 7 & 8 is *FR*

7-3, & 8-3, respectively. FGB comments in Exps. 3-4, 4-4, 7-4, and 8-4 are acquired by averaging three involved domain experts' opinions and DE_{avg} is used to represent this virtual domain expert. In addition, we use mean square error (MSE), calculated by Eq. 12, to evaluate the performance of the proposed method.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (12)$$

where, Y_i represents the inferred results via the type-2 fuzzy inference mechanism, $\widehat{Y}_i$ denotes the desired output recommended from the dieticians, and n is the number of the meal records.

Table 6 indicates that the difference between Exps. 1 & 5 and Exps. 2 & 6 is PCR, and the difference between Exps. 1 & 2 and Exps. 5 & 6 is FR. Therefore, Fig. 7a shows that the considered factor for the dotted lines and the solid lines is PCR.

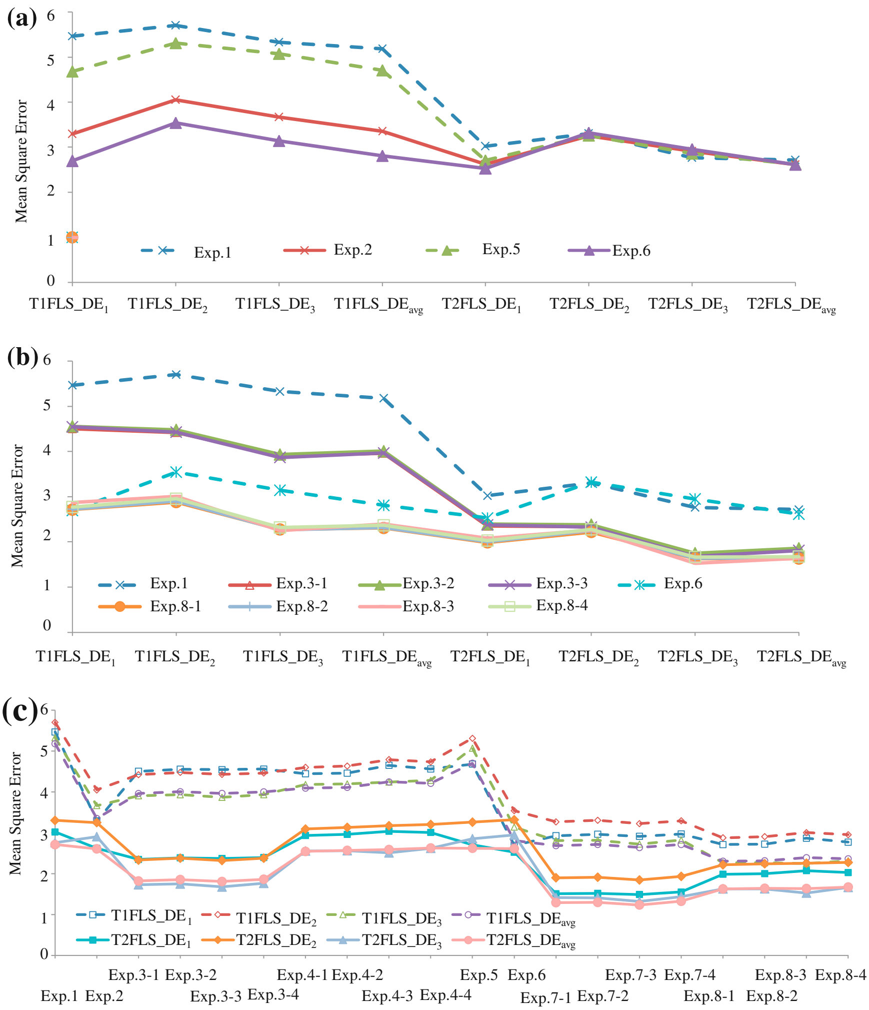


Fig. 7 Curves for considering the variance (**a**) in *PCR* and *FR*, (**b**) in *FGB*, and (**c**) between T1FS and T2FS for all of the experiments

Figure 7a also shows that the considered factor for the marked-cross lines and the marked-triangular lines is FR. From the curves in Fig. 7a, they indicate the *MSE* of the solid lines is smaller than the one of the dotted lines, showing that using the proposed method in this chapter to consider the calories people consume too much more or fewer than the planned caloric intake set by nutritionists can get much smaller *MSE*. Figure 7a also shows that the *MSE* of the marked-crossed lines is smaller than the one of the marked-triangular lines, which demonstrating that using the method proposed in this chapter to construct the fuzzy rules can get the much better performance, especially for T1FLS. Table 6 indicates that the difference between Exps. 1 & 3 and Exps. 6 & 8 is FGB. Therefore, Fig. 7b shows that the *MSE* of Exp. 3 is much smaller than the one of Exp. 1, and the *MSE* of Exp. 8 is also much smaller than the one of Exp. 6, demonstrating that the performance of considering the FGB method proposed in this chapter is much better. For most points in Fig. 7c, the *MSE* of the solid lines is smaller than the one of the dotted lines under the same experiment's condition, which means that there is a tendency for the *MSE* to get down when T2FLS is used to implement all of the experiments.

5 Conclusions

In this chapter, a type-2 FML-based dietary assessment system is proposed. Based on the viewpoint from dieticians, the percentage of calories from carbohydrate, the percentage of calories from protein, the percentage of calories from fat, and six food groups balance, are considered as one of the features to evaluate if the eaten food is healthy or not. From the experimental results, it shows that type-2 FLS performs better than type-1 FLS, proving that type-2 FLS can provide a powerful paradigm to handle the high level of uncertainties present in diet. Additionally, owning to the change in the lifestyle, more and more processed food is generated to meet modern-people's requirements. However, natural food like fresh fruit juice is much healthier than processed food like packaged fruit juice even though they are with the same provided calories and nutrients. This is because valuable nutrients contained in raw food are definitely partially washed away after the food is processed. Consequently, the food-processed level and the genetic learning machine will be considered to improve the experimental results in the future.

Acknowledgments This work is supported by the National Science Council (NSC) of Taiwan under the grant NSC98-2221-E-024-009-MY3 and 99-2622-E-024-003-CC3. The authors wish to thank Su-E Kuo, Hui-Ching Kuo, and Hui-Hua Chen, for their support with the experimental results.

References

1. Acampora, G., Loia, V.: Using FML and fuzzy technology in adaptive ambient intelligence environments. International Journal of Computational Intelligence Research **1**(2), 171–182 (2005)
2. Acampora, G., Loia, V.: Fuzzy control interoperability and scalability for adaptive domotic framework. IEEE Trans. Industr. Inf. **1**(2), 97–111 (2005)
3. Afacan, Y., Demirkan, H.: An ontology-based universal design knowledge support system. Knowl.-Based Syst. **24**(4), 530–541 (2011)
4. Bobillo, F., Straccia, U.: Fuzzy ontology representation using OWL 2. Int. J. Approximate Reasoning **52**(7), 1073–1094 (2011)
5. Food and Drug Administration, Department of Health, Executive Yuan, Taiwan, Daily dietary guideline. http://consumer.fda.gov.tw/Pages/List.aspx?nodeID=3. Accessed 12 July 2012
6. Gaeta, M., Orciuoli, F., Ritrovato, P.: Advanced ontology management system for personalized e-Learning. Knowledge-Based System **22**(4), 292–301 (2009)
7. Hagras, H.: Type-2 FLCs: A new Generation of Fuzzy Controllers. IEEE Comput. Intell. Mag. **2**(1), 30–43 (2007)
8. Katz, D.L., Gonalez, H.: The way to eat: a six-step path to lifelong weight control. Sourcebooks Inc, Napervilee (2004)
9. Lee, C.S., Wang, M.H., Hagras, H.: A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation. IEEE Trans. Fuzzy Syst. **18**(2), 374–395 (2010)
10. Lee, C.S., Jian, Z.W., Huang, L.K.: A fuzzy ontology and its application to news summarization. IEEE Trans. Syst. Man Cybern. B Cybern. **35**(5), 859–880 (2005)
11. Lee, C.S., Wang, M.H., Acampora, G., Hsu, C.Y., Hagras, H.: Diet assessment based on type-2 fuzzy ontology and fuzzy markup language. International Journal of Intelligent System **25**(12), 1187–1216 (2010)
12. Lee, C.S., Kao, Y.F., Kuo, Y.H., Wang, M.H.: Automated ontology construction for unstructured text documents. Data Knowl. Eng. **60**(3), 547–566 (2007)
13. Lee, C.S., Wang, M.H., Chen, Z.W., Hsu, C.Y., Kuo, S.E., Kuo, H.C., Cheng, H.H., Naito, A.: Genetic fuzzy markup language for diet application. Proceeding of 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Taipei, Taiwan, Jun. 27–30, 2011, pp. 1791–1798 (2011)
14. Lee, C.S., Wang, M.H., Hagras, H., Chen, Z.W., Lan, S.T., Kuo, S.E., Kuo, H.C., Cheng, H.H.: A novel genetic fuzzy markup language and its application to healthy diet assessment. International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems. **20**(2), 247–278 (2012)
15. Mendel, J.M.: Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice Hall, Upper Saddle River, NJ (2001)
16. Mendel, J.M., John, R.I.B.: Type-2 fuzzy sets made simple. IEEE Transactions on Fuzzy Sets **10**(2), 117–127 (2002)
17. Mendel, J.M., John, R.I.B., Liu, F.: Interval type-2 fuzzy logic systems made simple. IEEE Trans. Fuzzy Syst. **14**(6), 808–821 (2006)
18. Trappey, A.J.C., Trappey, C.V., Hsu, F.C., Hsiao, D.W.: A fuzzy ontological knowledge document clustering methodology. IEEE Transactions on Systems, Man. And Cybernetics Part B. Cybernetics **39**(3), 806–814 (2009)
19. Wang, M.H., Lee, C.S., Hsieh, K.L., Hsu, C.Y., Acampora, G., Chang, C.C.: Ontology-based multi-agents for intelligent healthcare applications. Journal of Ambient Intelligence and Humanized Computing **1**(2), 111–131 (2010)

20. Wang, M.H., Lee, C.S., Acampora, G., Loia, V.: Electrocardiogram application based on heart rate variability ontology and fuzzy markup language. In: Gacek, A., Pedrycz, W. (eds.) ECG Signal Processing, Classification and Interpretation: A Comprehensive Framework of Computational Intelligence, pp. 155–178. Springer-Verlag, Germany (2011)
21. Wang, M.H., Lee, C.S., Chen, Z.W., Lo, C.F., Kuo, S.E., Kuo, H.C., Cheng, H.H.: (2010) Property and application of fuzzy ontology for dietary assessment: IEEE World Congress on Computational Intelligence (IEEE WCCI 2010), pp. 18–23. Barcelona, Spain (2010)

About the Editors

Alireza Sadeghian received the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada. Since 1999, he is the Professor of Computer Science at Ryerson University in Toronto where he also holds the position of Chair and the founding Director of the Computational Intelligence Initiative (CI2) Lab. He has published over 90 technical papers in the areas of type-2 fuzzy sets, knowledge-based expert systems, artificial neural networks, adaptive neuro-fuzzy networks, and nonlinear modeling.

Jerry M. Mendel received the Ph.D. degree in electrical engineering from the Polytechnic Institute of Brooklyn, Brooklyn, NY. Currently he is the Professor of Electrical Engineering and Systems Architecting Engineering at the University of Southern California in Los Angeles, where he has been since 1974. He has published over 500 technical papers and is author and/or editor of nine books, including *Uncertain Rule-based Fuzzy Logic Systems Introduction and New Directions* (Prentice-Hall, 2001) and *Perceptual Computing: Aiding People in Making Subjective Judgments* (Wiley & IEEE Press, 2010). His present research interests include: type-2 fuzzy logic systems and their applications to a wide range of problems, including smart oil field technology, computing with words, and fuzzy set qualitative comparative analysis. He is a Life Fellow of the IEEE, a Distinguished Member of the IEEE Control Systems Society, and a Fellow of the International Fuzzy Systems Association. Among his awards are the 1983 Best Transactions Paper Award of the IEEE Geoscience and Remote Sensing Society, the 1992 Signal Processing Society Paper Award, the 2002 *Transactions on Fuzzy Systems* Outstanding Paper Award, a 1984 IEEE Centennial Medal, an IEEE Third Millenium Medal, and a Fuzzy Systems Pioneer Award (2008) from the IEEE Computational Intelligence Society.

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6,

Hooman Tahayori received his Ph.D. degree in Informatics from Università Degli Studi di Milano, Italy in 2009. He completed his Honors master degree in Robotics and Artificial Intelligence and his Bachelor's degree with Honors in Computer Science-Hardware design respectively in 2000 and 1997 at Shiraz University. Currently, he is a post-doctoral research fellow in the Computational Intelligence Initiative (CI2) Lab in the Department of Computer Science at Ryerson University. His main areas of research interests are fuzzy sets of higher order, granular computing, and their applications in various disciplines. He has published over 40 technical papers in the areas of type-2 fuzzy sets, shadowed sets, and granular computing.

Index

A. Sadeghian et al. (eds.), *Advances in Type-2 Fuzzy Sets and Systems*,
Studies in Fuzziness and Soft Computing 301, DOI: 10.1007/978-1-4614-6666-6,