Alireza Sadeghian
Hooman Tahayori *Editors*

Frontiers of Higher Order Fuzzy Sets

Springer

Frontiers of Higher Order Fuzzy Sets

Alireza Sadeghian • Hooman Tahayori
Editors

Frontiers of Higher Order Fuzzy Sets

Springer

Editors
Alireza Sadeghian
Ryerson University
Toronto
Ontario
Canada

Hooman Tahayori
Ryerson University
Toronto
Ontario
Canada

ISBN 978-1-4614-3441-2 978-1-4614-3442-9 (eBook)
DOI 10.1007/978-1-4614-3442-9
Springer NewYork Heidelberg Dordrecht London

Library of Congress Control Number: 2014957730

Preface

Frontiers of Higher Order Fuzzy Sets

Uncertainty is the result of imperfection of knowledge about a state or a process. Inevitable errors that occur in the process of measurement on one hand, together with the limited accuracy and resolution level of measuring instruments on the other hand, constitute empirical uncertainty level. Natural language as the main carrier of human knowledge, with its intrinsic ambiguity and vagueness, results in cognitive uncertainty level. Sometimes, knowledge is intentionally made uncertain for specific strategic usages, which constitutes the social level of uncertainty.

Real world problems require exploitation of frameworks that enable handling different types and levels of uncertainty. Type-2 fuzzy sets enable handling intra- and inter-uncertainties, i.e., uncertainty of a subject and uncertainties among different subjects.

In the implementation of fuzzy systems, in addition to the explicit reasons of uncertainty associated with membership grades related to the empirical and cognitive levels of uncertainty, implicit sources of uncertainty are to be recognized that rooted at the methods that may be used to tune the membership values. Uncertain data or uncertain resources that may be used for tuning the membership grades themselves will introduce new sources of uncertainties. Type-2 fuzzy sets enable *capturing the uncertainty* on membership functions of fuzzy sets through blurring the membership function of type-1 fuzzy sets. In general, as the order of fuzzy sets increases, their degrees of freedom increase, and hence, provide more potential for handling uncertainties.

The book intends to be a valuable source of recent knowledge about higher types and orders of fuzzy sets. New capable fuzzy frameworks are discussed and their applicability is shown. There are elaborations on providing a basis for selecting fuzzy sets of higher order which are suitable for addressing various types of uncertainty issues in problems. New areas in which fuzzy sets would be applicable are also introduced.

This book outlines notable achievements in the realm of higher order and higher type of fuzzy set to date. The editors hope the materials covered in this book, provided

by the leading scholars in the field, motivate and accelerate future progress and introduce new branches off the fuzzy set theory. Of course, there are still many related theoretical and applied issues that need to be addressed . This book is organized in three parts.

Part 1 is dedicated to the theoretical foundations of type-2 fuzzy sets. In Chap. 1, Tahayori and Sadeghian have introduced a disjointing difference operator on fuzzy sets. Based on the properties of their proposed operator, a novel and easy algorithm for performing the union and intersection operations on type-2 fuzzy sets with respect to min t-norm and max t-conorm is proposed. Through defining robustness in terms of the maximum output tolerance of the system to a given output deviation, Biglarbegian in Chap. 2, has presented a rigorous mathematical methodology for the robustness analysis of Interval Type-2 Takagi-Sugeno-Kang fuzzy logic systems.

Chapters in Part 2 discuss different methodologies of fuzzy modeling. Pedrycz in Chap. 3 has investigated potential and algorithmic implications of fuzzy sets of higher order and higher type, specially fuzzy sets of type-2 and order-2, in the realm of fuzzy modeling. In Chap. 4, Türkşen has discussed a framework for modeling the human decision-making process with type-1 and full type-2 fuzzy logic methodology. He has also proposed a new algorithm for generating type-2 membership value distributions for the development of second order fuzzy system models. Chapter 5, by Liu and Gomide, introduces participatory evolutionary learning as a framework for data driven fuzzy modeling. Despite the focus on participatory learning and the selective transfer to build first order fuzzy rule-based models, the use of the genetic fuzzy systems to develop higher order fuzzy rule-based models is also discussed. Chapter 6, by Frantz et al., provides an instance that shows the strong potential of IT2FS to establish a comparatively simple aggregation of opinions into a fuzzy set. They have proposed a mechanism to integrate numerous opinions to model the establishment of economic institutional rules.

Finally, chapters in Part 3 introduce novel application of higher order fuzzy sets. Livi and Rizzi, in Chap. 7, have defined a framework for designing and evaluating uncertainty preserving transformation procedures for generating type-2 fuzzy sets from raw input data. They have applied their method on the set of labeled graphs as input data. Yazdanbakhsh and Dick, in Chap. 8, have studied the performance of adaptive neuro-complex-fuzzy inference system (ANCFIS) which is a neuro-fuzzy system that employs complex fuzzy sets for time-series forecasting. Chapter 9 by Niewiadomski and Superson has enhanced basics of type-2 linguistic summarization of data. They have introduced new forms of linguistic summaries that use type-2 fuzzy sets as representations of linguistic information. Castillo, in Chap. 10, has presented a general framework for designing interval type-2 fuzzy controllers based on bio-inspired optimization techniques. He has shown how bio-inspired optimization techniques can be used to obtain results that outperform traditional approaches in the design of optimal type-2 fuzzy controllers. In Chap. 11, Melin has reported some experiments that clearly show that fuzzy edge detectors are a good choice to improve the performance of neural or other types of recognition systems. Hence, she has argued that the recognition rate of the neural networks can be used as an edge detection performance index. Jamshidi et al., in Chap. 12, attempts to construct

a bridge between system-of-systems and data analytic to develop reliable models for operating, nonhomogeneous systems. They have applied big data analytic approaches to predict or forecast the behavior of stock market and renewable energy availability.

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, Rebecca Hytowitz, and Charles Glaser.

Contents

Contributors

Mohammad Biglarbegian School of Engineering University of Guelph Guelph, Ontario, Canada

Oscar Castillo Tijuana Institute of Technology, Tijuana, CP, Mexico

Scott Dick Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada

Christopher Frantz Department of Information Science, University of Otago, Dunedin, New Zealand

Fernando Gomide School of Electrical and Computer Engineering, University of Campinas, Sao Paulo, Brazil

Mo Jamshidi ACE Laboratory, The University of Texas, San Antonio, TX, USA

Halid Kaplan ACE Laboratory, The University of Texas, San Antonio, TX, USA

Nathan D. Lewis Department of Information Science, University of Otago, Dunedin, New Zealand

Yi Ling Liu School of Electrical and Computer Engineering, University of Campinas, Sao Paulo, Brazil

Lorenzo Livi Department of Information Engineering, Electronics, and Telecommunications, SAPIENZA University of Rome, Rome, Italy

Patricia Melin Division of Graduate Studies and Research, Tijuana Institute of Technology, Tijuana, Mexico

Adam Niewiadomski Institute of Information Technology, Łódź University of Technology, Łódź, Poland

Mariusz Nowostawski Department of Information Science, University of Otago, Dunedin, New Zealand

Witold Pedrycz Department of Electrical Computer & Engineering, University of Alberta, Edmonton, Canada

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Department of Electrical and Computer Engineering Faculty of Engineering, King Abdulaziz University Jeddah, Saudi Arabia

Maryam A. Purvis Department of Information Science, University of Otago, Dunedin, New Zealand

Martin K. Purvis Department of Information Science, University of Otago, Dunedin, New Zealand

Antonello Rizzi Department of Information Engineering, Electronics, and Telecommunications, SAPIENZA University of Rome, Rome, Italy

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

Izabela Superson Institute of Information Technology, Łódź University of Technology, Łódź, Poland

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

Barney Tannahill Southwest Research Institute, San Antonio, TX, USA

I. Burhan Türkşen Department of Industrial Engineering, TOBB-ETU, Ankara, Turkey

University of Toronto, Toronto, ON, Canada

Omolbanin Yazdanbakhsh Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada

Yunus Yetis ACE Laboratory, The University of Texas, San Antonio, TX, USA

About the Book

- The book presents new variations of fuzzy set frameworks and new areas of applicability of fuzzy theory
- It discusses different methodologies for fuzzy modeling
- It studies Perceptual Computing with higher order fuzzy sets

This book is a valuable source of recent knowledge about higher types and orders of fuzzy sets. New capable fuzzy frameworks are discussed and their applicability is shown. Moreover, there are elaborations on providing a basis for selecting fuzzy sets of higher order which are suitable for addressing various types of uncertainty issues in problems. New areas in which fuzzy sets would be applicable are also introduced.

In the book, efficient algorithms for performing operations on general type-2 fuzzy sets (T2FSs) are proposed. Also, rigorous mathematical methodology for the robustness analysis of interval type-2 fuzzy logic systems is presented. Participatory evolutionary learning as a framework for data driven fuzzy modeling and implications of fuzzy sets of higher order and higher type, in the realm of fuzzy modeling is studied. Moreover, a general framework for designing interval type-2 fuzzy controllers based on bio-inspired optimization techniques is discussed in the book. Also, the potential of IT2FS to establish a comparatively simple aggregation of opinions into a fuzzy set is studied. Uncertainty-preserving transformation procedures for generating T2FSs from raw input data are introduced in the book. Moreover, new forms of linguistic summaries that use T2FSs as representations of linguistic information are discussed.

Chapter 1
A New Fuzzy Disjointing Difference Operator to Calculate Union and Intersection of Type-2 Fuzzy Sets

Hooman Tahayori and Alireza Sadeghian

Abstract This chapter introduces a fuzzy disjointing difference operator. Based on the ordering of the disjoint fuzzy sets of the real line, a novel algorithm for calculation of the union and intersection of type-2 fuzzy sets with convex fuzzy grades using min t-norm and max t-conorm is proposed. The algorithm can be easily extended to the problems of ordering fuzzy numbers and calculation of the extended *max* and *min* of fuzzy sets.

1.1 Introduction

In type-1 fuzzy sets (T1FSs), membership values are real numbers that are linearly ordered. Although the membership values may be chosen from a partially ordered set [1], in practice the values are from the unit interval [0, 1]. Hence, the operations *max* and *min* can be easily exploited to determine the ordering of any pair of the membership values. Consequently, calculating the union and intersection of T1FSs using any t-norm and t-conorm operation is straightforward.

It has been argued in the literature that the membership functions of T1FSs are precise and therefore they lack the ability to handle uncertainties [2–5]. Hence, type-2 fuzzy sets (T2FSs) have been proposed as an extension to T1FSs to handle uncertainties [4, 5]. It is also argued that the more imprecise or vague the data are the better applicability of T2FSs can be assumed [5]. However, T2FSs advantage comes at a cost which may hinder their applicability. In particular, T2FSs are more complex than T1FSs, and importantly operations on T2FSs are more computationally expensive.

In T2FSs, membership values are of T1FSs and extending t-norm and t-conorm operations on them are not as straightforward as performing the same operations on the membership grades in T1FSs. In particular, calculating their maximum and

H. Tahayori (✉) · A. Sadeghian
Department of Computer Science, Ryerson University,
Toronto, ON M5B 2K3, Canada
e-mail: htahayor@scs.ryerson.ca

A. Sadeghian
e-mail: asadeghi@scs.ryerson.ca

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_1

minimum cannot be achieved by direct exploitation of *max* and *min* operations. Fulfilling such operations naturally demands for Zadeh's extension principle [6] which is known to be neither simple nor efficient [7]. There have been extensive efforts to overcome the aforementioned difficulties associated with T2FSs. For example, [8, 9] provide the underlying principle of T2FSs, and [7, 9, 21] elaborate on algorithms to perform operations on T2FSs. The existing algorithms for operations on T2FSs are not as efficient as the operations on T1FSs [14]. Moreover, most of the existing algorithms are limited to T2FSs with normal convex fuzzy membership grades. That is why despite their rather Boolean behavior in modeling uncertainties, a special case of T2FSs, i.e., interval type-2 fuzzy sets (IT2FSs) [22], is more preferred by researchers. Recently, shadowed fuzzy sets (SFSs) as another variation of T2FSs are proposed [23, 24]. SFSs through redistribution of fuzziness associated with fuzzy grades in shadowed sets [25] provide more freedom degrees for handling uncertainties in comparison with IT2FSs but with lower computational complexity than general T2FSs.

In this chapter, we introduce a novel algorithm for calculating the union and intersection operations on T2FSs using min t-norm and max t-conorm. The underlying idea is inspired by the concept of ordering the disjoint fuzzy sets of the real line. To this end, we will define the *disjointing difference operation* on fuzzy sets based on which we determine the minimum and maximum of the fuzzy membership grades. The proposed algorithm can be easily extended to the problem of ordering fuzzy numbers and calculation of the extended *max* and *min* of fuzzy sets.

The rest of this chapter is organized as follows. In Sect. 1.2, we provide a brief review of the basics of T1FSs and T2FSs. We introduce the disjointing difference operation on fuzzy sets and will investigate some of its properties in Sect. 1.3. The proposed algorithm for performing union and intersection operations on T2FSs is discussed in Sect. 1.4, and the conclusions are given in Sect. 1.5.

1.2 Preliminaries and Notations

A fuzzy set F in the universe of discourse X is characterized by its membership function $f : X \rightarrow [0,1]$ that assigns a crisp number from unit interval [0, 1] to each element of X. The *support* of fuzzy set F is defined to be $S_F = \{x \in X | f(x) > 0\}$ while its *height* is $H_F = \underset{x \in X}{Sup}\,(f(x))$. A fuzzy set is *normal* if its height is 1 otherwise it is considered as *subnormal*. The set $C_F = \{x \in X | f(x) = H_F\}$ constitutes the *core* of the fuzzy set F. Fuzzy set F is said to be *empty* if $\underset{x \in X}{Sup}\,(f(x)) = 0$, i.e., $H_F = 0$. Fuzzy set F is convex if for all $x_1, x_2 \in X$ and $\lambda \in [0,1]$, $f(\lambda x_1 + (1-\lambda)x_2) \geq Min(f(x_1), f(x_2))$. Given $f : X \rightarrow [0,1]$ and $g : X \rightarrow [0,1]$ be membership functions of the fuzzy sets F and G that are defined on the same domain. F is a subset of G, i.e., F is included in G, if for all $x \in X$, $f(x) \leq g(x)$.

T2FS, F, a fuzzy set with fuzzy membership grades over the universe of discourse X, is denoted as

$$\begin{aligned}\mathcal{F} &= \{(x, F_x), x \in X, F_x = \{(u, f_x(u)), u \in [0,1], f_x(u) \in [0,1]\}\} \\ &= \left\{(x, (u, f_x(u))), x \in X, u \in J_x^{\mathcal{F}} \subseteq [0,1], f_x(u) \in [0,1]\right\}, \end{aligned} \tag{1.1}$$

where x denotes *primary variable* and $J_x^{\mathcal{F}}$ represents the *primary membership values* of x in F. We refer to

$$F_x = \left\{(u, f_x(u)), u \in J_x^{\mathcal{F}} \subseteq [0,1], f_x(u) \in [0,1]\right\}, \tag{1.2}$$

which is a T1FS, as the *fuzzy grade* of x in F, and it is also known as *secondary membership function* or *secondary set* [8]. The domain of f_x constitutes the *primary membership values* of $x \in X$ and if all uncertainty on membership values disappear, the domain can be reduced to a point. The union of all primary membership values constitutes a region known as footprint of uncertainty (FOU), that is, $FOU_{\mathcal{F}} = \bigcup_{x \in X} \left\{\left(x, J_x^{\mathcal{F}}\right)\right\}$. The amplitudes of the primary membership values, i.e., $f_x(u)$ are referred to as the *secondary grades*. In (1.2), if all $f_x(u) = 1$ then F is considered as a special case of T2FS that is known as IT2FS.

1.3 Fuzzy Disjointing Difference Operation

The difference of two sets, A and B, in the classical set theory is defined as $A - B = \{x | x \in A \text{ and } x \notin B\}$. It is evident that based on the given difference definition, $A - B$ and $B - A$, are disjoint sets. The notion of the set difference can be extended to the fuzzy sets using several proposed operators, e.g., see [26, 27]. However, the common characteristic of these operators is the fact that they do not necessarily result in disjointing fuzzy sets. Fuzzy sets F and G are disjoint if their intersection is empty. However, since this condition is very restrictive; instead, the *highest degree of separation* of two convex fuzzy sets F and G is defined as $1 - \underset{x \in X}{Sup}\,(\mu_{F \cap G}(x))$ (see [1]). In this section, we define a difference operation on fuzzy sets that leads to disjoint fuzzy sets. Throughout the chapter, we assume all T1FSs are defined on real line unless explicitly expressed otherwise. Initially we define packed fuzzy sets.

Definition 1: Fuzzy set $F = \{(x, f(x)), x \in R\}$ defined on real line is *packed* if and only if S_F is a convex set.

The disjointing difference operation (DDO) on fuzzy sets of real line is then defined as follows.

Definition 2: Let $F = \{(x, f(x)), x \in R\}$ and $G = \{(x, f(x)), x \in R\}$ be two fuzzy sets of real line. The disjointing difference of fuzzy sets F and G, is the fuzzy set $F' = F - G = \left\{\left(x, f'(x)\right), x \in R\right\}$, where

$$f'(x) = \begin{cases} f(x) & f(x) > g(x) \\ 0 & \text{otherwise} \end{cases}.$$

Definition 3: Symmetrical difference of fuzzy sets F and G is defined as, $D = F\Delta G = (F - G) \cup (G - F) = \bigcup_k D_k$, where $D_m \cap D_n \neq \emptyset, m \neq n$.

In the case of F and G, which are two regular sets represented by their characteristic functions, Definitions 2 and 3 are valid to denote their difference and symmetrical difference. Theorem 1 in the following proves the main property of DDO, i.e., $F' = F - G$ and $G' = G - F$ are disjoint.

Theorem 1: Let $F' = F - G$ and $G' = G - F$ then $F' \cap G' = \emptyset$.

Proof: Assume $F' \cap G' \neq \emptyset$, so $\exists x \in X$ such that $f'(x) \wedge g'(x) > 0$, i.e., $f'(x) > 0$ and $g'(x) > 0$. Based on Definition 2, $f'(x) > 0$ connotes $f(x) > g(x)$and similarly $g'(x) > 0$ means $g(x) > f(x)$ that is a contradiction. ■

Based on Definition 2, generally F' cannot be a packed fuzzy set. Figure 1.1a shows an example where F' has ended in a single-packed fuzzy set; however Fig. 1.1b demonstrates that F' is not necessarily a packed fuzzy set. In Theorem 2, we prove that the disjointing difference of two fuzzy sets will in general result in a number of packed-disjoint fuzzy sets.

Theorem 2: Let F and G be arbitrary fuzzy sets and $F' = F - G \neq \emptyset$. F' is composed of a number of packed disjoint fuzzy sets, i.e., $F' = F - G = \bigcup_i F_i'$, where $\forall i, F_i'$ is packed fuzzy set and $F_i{}' \cap F_j{}' = \emptyset, i \neq j$.

Proof: First, we prove that F' is composed of a set of packed fuzzy sets. Assume $\exists i$ such that F_i' is not packed; hence, $S_{F_i'}$ is not convex, and is composed of a number of intervals, i.e., $S_{F_i'} = \bigcup_{k>1} S^k_{F_i'}$. $S^k_{F_i'}$ is the kth interval that constitutes $S_{F_i'}$ and is convex, so its underlying fuzzy set is a packed fuzzy set. This signifies that even if F_i' is not a packed fuzzy set, it is composed of a set of packed fuzzy sets.

To prove that the constituting packed fuzzy sets of F' are disjoint, we assume $\exists i \neq j, F_i' \cap F_j' \neq \emptyset$ which means $\exists x, f_i'(x) > 0$ and $f_j'(x) > 0$. This indicates that $\forall x \in S_{F_i'} \cup S_{F_j'}$, $f(x) > g(x)$ and since $S_{F_i'} \cap S_{F_j'} \neq \emptyset$, based on Definition 1, $S_{F_i'} = S_{F_j'}$. ■

Throughout the chapter, we refer to the fuzzy sets that are composed of a number of packed fuzzy sets as compound fuzzy sets.

Theorem 3 proves that DDO on respectively a convex and arbitrary fuzzy set results in a compound fuzzy set that each of its constituting packed fuzzy set is convex. The obvious consequence of this theorem is that DDO of convex fuzzy sets results in a compound fuzzy set with all of its constituting packed fuzzy sets convex.

Theorem 3: Let F be a convex fuzzy set and G an arbitrary fuzzy set, and let $F' = F - G = \bigcup_i F_i'$. Then for all i, F_i' is a convex fuzzy set.

Proof: The proof is by contradiction. Imagine $\exists i$ such that F_i' is not convex. Given $x_1, x_2 \in S_{F'_i}$, where $x_1 \leq x_2$. There should exist a $x_3 \in S_{F'_i}, x_1 \leq x_3 \leq x_2$ such that $f_i'(x_3) < \min(F_i'(x_1), F_i'(x_2))$. Based on Definition 2, $x_1, x_2, x_3 \in S_{F_i'}$ implies that $f_i'(x_1) = f(x_1) > g(x_1)$, $f_i'(x_2) = f(x_2) > g(x_2)$, and $f_i'(x_3) = f(x_3) > g(x_3)$. F is convex; hence, $f(x_3) \geq \min(f(x_1), f(x_2))$ and consequently $f_i'(x_3) \geq \min(F_i'(x_1), F_i'(x_2))$. ■

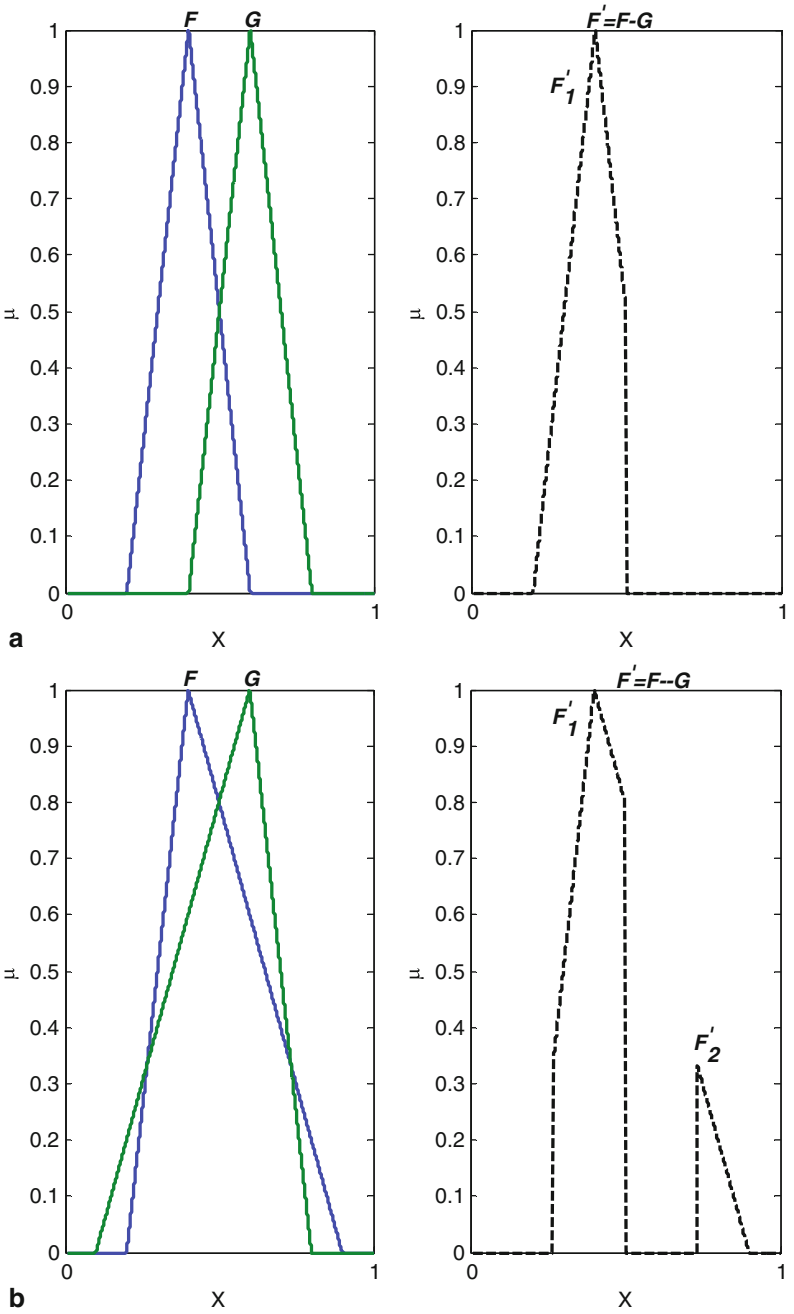


Fig. 1.1 Disjoint difference of two convex fuzzy sets. (**a**) The difference has resulted in one packed fuzzy set. (**b**) the difference has resulted in two disjoint packed fuzzy sets

1.4 Fuzzy Disjointing Difference Operation to Calculate Join and Meet of Fuzzy Grades

Let $\mathcal{F}= \left\{(x,(u,f_x(u))), x \in X, u \in J_x^{\mathcal{F}} \subseteq [0,1], f_x(u) \in [0,1]\right\}$ and $\mathcal{G}= \left\{(x,(u,g_x(u))), x \in X, u \in J_x^{\mathcal{G}} \subseteq [0,1], g_x(u) \in [0,1]\right\}$ be the two T2FSs defined in the universal set X. The membership value of x in F and G are, respectively $F_x = \left\{(u, f_x(u)), u \in J_x^{\mathcal{F}} \subseteq [0,1], f_x(u) \in [0,1]\right\}$ and $G_x = \left\{(u, g_x(u)), u \in J_x^{\mathcal{G}} \subseteq [0,1], g_x(u) \in [0,1]\right\}$. The membership degree of x in the union and intersection of F and G, using min t-norm and max t-conorm respectively represented by $\wedge$ and $\vee$, is calculated through exploitation of Zadeh's extension principle as [28]

$$\mu_{\mathrm{F}\cup\mathrm{G}}(x) = F_x \sqcap G_x = \left\{\left(\theta, \underset{u\wedge w=\theta}{Sup}\ (f(u)\wedge g(w))\right), u,w \in U\right\} \tag{1.3}$$

$$\mu_{\mathrm{F}\cup\mathrm{G}}(x) = F_x \sqcup G_x = \left\{\left(\theta, \underset{u\vee w=\theta}{Sup}\ (f(u)\wedge g(w))\right), u,w \in U\right\}, \tag{1.4}$$

where $\sqcup$ and $\sqcap$ denote the so-called *join* and *meet* operations [14, 28]. As evident from (1.3) and (1.4), the join and meet operations through the direct application of Zadeh's extension principle are computationally and conceptually complex. The main goal of the chapter is to propose an algorithm for performing the join and meet operations based on the properties of disjoint fuzzy grades. To this end, in the following, we study some properties of join and meet of two fuzzy grades.

Theorem 4: Let F and G be packed fuzzy grades of x respectively in T2FSs, F and G, where $S_{F\cap G} = \left[\underline{si}, \overline{si}\right] \neq \emptyset$, then

(i) $\forall\theta > \overline{\mathrm{si}}$ then $\mu_{F\sqcap G}(\theta) = 0$,

(ii) $\forall\theta < \underline{\mathrm{si}}$ then $\mu_{F\sqcup G}(\theta) = 0$,

Proof: Using (1.3) and calculating the meet operation we have

$$\mu_{F\sqcap\mathrm{G}}(\theta) = \underset{u\wedge w=\theta}{Sup}\ (f(u)\wedge g(w)) = \underset{\substack{u=\theta\\ w\geq\theta}}{Sup}\ (f(\theta)\wedge g(w)) \vee \underset{\substack{w=\theta\\ u\geq\theta}}{Sup}\ (f(u)\wedge g(\theta))$$

$$= \left(f(\theta)\wedge \underset{w\geq\theta}{Sup}\ (g(w))\right) \vee \left(g(\theta)\wedge \underset{u\geq\theta}{Sup}\ (f(u))\right). \tag{1.5}$$

Given $\theta > \overline{si}$, we know that $f(\theta)\wedge g(\theta) = 0$. Without loss of generality, we assume $f(\theta) = 0$ and since F is packed fuzzy set; hence, $\forall\theta > \overline{si}$, $f(\theta) = 0$ and consequently $\forall\theta > \overline{si}$, $\underset{u\geq\theta}{Sup}\ (f(u)) = 0$; hence,

$$\mu_{F\sqcap G}(\theta) = \left(0\wedge \underset{w\geq\theta}{Sup}\ (g(w))\right) \vee (g(\theta)\wedge 0) = 0. \tag{1.6}$$

Proof of (ii) for the join operation is similar. ■

Corollary 1: Let F and G be packed fuzzy grades of x respectively in T2FSs, F and G, then

$$\text{(i)} \quad F \cap G \subseteq F \sqcap G \subseteq F \cup G, \tag{1.7}$$

$$\text{(ii)} \quad F \cap G \subseteq F \sqcup G \subseteq F \cup G. \tag{1.8}$$

Proof: We will first prove (i), and then (ii) can also be proved similarly. First, we will prove that $F \cap G \subseteq F \sqcap G$. Based on (1.3) we have

$$\begin{aligned}
\mu_{F\sqcap G}(\theta) &= \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right) \\
&= \left((f(\theta) \wedge g(\theta)) \vee \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right)\right) \\
&\quad \vee \left((f(\theta) \wedge g(\theta)) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right)\right). \\
&= (f(\theta) \wedge g(\theta)) \vee \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right)
\end{aligned} \tag{1.9}$$

It is evident that $\mu_{F\sqcap G}(\theta) \geq (f(\theta) \wedge g(\theta))$.

To prove that $F \sqcap G \subseteq F \cup G$, we have

$$\begin{aligned}
\mu_{F\sqcap G}(\theta) &= \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right) \\
&= (f(\theta) \vee g(\theta)) \wedge \left(f(\theta) \vee \underset{u\geq\theta}{Sup}\,(f(u))\right) \wedge \left(g(\theta) \vee \underset{w\geq\theta}{Sup}\,(g(w))\right) \\
&\quad \wedge \left(\underset{u\geq\theta}{Sup}\,(f(u)) \vee \underset{w\geq\theta}{Sup}\,(g(w))\right) \\
&= (f(\theta) \vee g(\theta)) \wedge \left(\underset{u\geq\theta}{Sup}\,(f(u))\right) \wedge \left(\underset{w\geq\theta}{Sup}\,(g(w))\right) \\
&\quad \wedge \left(\underset{u\geq\theta}{Sup}\,(f(u)) \vee \underset{w\geq\theta}{Sup}\,(g(w))\right) \\
&= (f(\theta) \vee g(\theta)) \wedge \left(\underset{u\geq\theta}{Sup}\,(f(u))\right) \wedge \left(\underset{w\geq\theta}{Sup}\,(g(w))\right) \leq (f(\theta) \vee g(\theta)). \blacksquare
\end{aligned} \tag{1.10}$$

Corollary 2: Let F and G be fuzzy grades of x respectively in T2FSs, F and G, where $S_F = S_G$, then $S_{F\sqcap G} = S_{F\sqcap G} = S_F$.

Proof: Let $\theta \in S_F$, then $f(\theta) \neq 0$, $g(\theta) \neq 0$; hence, $\underset{u\geq\theta}{Sup}\,(f(u)) > 0$, $\underset{w\geq\theta}{Sup}(g(w)) > 0$ and consequently,

$\mu_{F\sqcap G}(\theta) = \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right) > 0$. However, for all $\theta \notin S_F$, both $f(\theta) = 0$ and $g(\theta) = 0$; hence,

$\mu_{F\sqcap G}(\theta) = \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right) = 0.$ ■

Corollary 3: Let F and G be packed fuzzy grades of x respectively in T2FSs, F and G, then $H_{F\sqcap G} \leq H_F \wedge H_G$ and $H_{F\sqcup G} \leq H_F \wedge H_G$.

Proof: With respect to the second part of the proof provided for Corollary 1, it is straightforward.

1.4.1 Calculation of the Join and Meet of Disjoint Fuzzy Grades

Real numbers are linearly ordered; hence, there is no doubt of finding minimum and maximum of a set of real numbers. Definition 4, however, describes how ordering can be extended on the intervals of real line.

Definition 4 [29]: Let $A = \left[\underline{a}, \overline{a}\right] = \left\{x \in R | \underline{a} \leq x \leq \overline{a}\right\}$ and $B = \left[\underline{b}, \overline{b}\right] = \left\{x \in R | \underline{b} \leq x \leq \overline{b}\right\}$ be two intervals defined on the real line. Then $A < B$ if and only if $\overline{a} < \underline{b}$.

Considering the interval ordering (see Definition 4) in Theorem 5, we prove that given two disjoint fuzzy grades, finding their meet (join) can be reduced to the identification of the fuzzy grade that is located to the left/right of the other. It should be stressed that since the fuzzy grades are disjoint, there is no ambiguity regarding their ordering.

Theorem 6 hence provides a generalization of Theorem 5 over two disjoint compound fuzzy sets. Although ordering compound fuzzy grades is not straightforward, in Theorem 6 it is shown that calculation of the join (meet) of two disjoint compound fuzzy grades can be reduced to the calculation of the join (meet) of all possible pairs of their constituting packed fuzzy sets.

Theorem 5: Let F and G be disjoint packed fuzzy grades of x respectively in T2FSs, F and G with the heights of H_F and H_G and $S_F < S_G$, then $F \sqcap G = F \wedge H_G$ and $F \sqcup G = G \wedge H_F$.

Proof: Using (1.3) and calculating the meet of the fuzzy grades F and G, the membership value of θ in the result is

$$\mu_{F\sqcap G}(\theta) = \underset{u\wedge w=\theta}{Sup}\,(f(u) \wedge g(w)) = \underset{\substack{u=\theta \\ w\geq\theta}}{Sup}\,(f(\theta) \wedge g(w)) \vee \underset{\substack{w=\theta \\ u\geq\theta}}{Sup}\,(f(u) \wedge g(\theta))$$

$$= \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right). \tag{1.11}$$

We evaluate (1.11) in three cases:

Case I: $\theta < S_F$. In this region, $f(\theta) = 0$ and $g(\theta) = 0$; hence, $\mu_{F \sqcap G}(\theta) = 0$.
Case II: $\theta \in S_F$. Since $F \cap G = \emptyset$ and $S_F < S_G$, we have $g(\theta) = 0$ and $\underset{w \geq \theta}{Sup}\,(g(w)) = H_G$. So

$$\mu_{\mathrm{F} \sqcap \mathrm{G}}(\theta) = \left(f(\theta) \wedge \underset{w \geq \theta}{Sup}\,(g(w)) \right) \vee \left(g(\theta) \wedge \underset{u \geq \theta}{Sup}\,(f(u)) \right)$$

$$= (f(\theta) \wedge H_G) \vee \left(0 \wedge \underset{u \geq \theta}{Sup}\,(f(u)) \right) = f(\theta) \wedge H_G. \tag{1.12}$$

Case III: $\theta > S_F$. We have $f(\theta) = 0$ and $Sup_{u \geq \theta}\,(f(u)) = 0$. So

$$\mu_{F \sqcap G}(\theta) = \left(f(\theta) \wedge \underset{w \geq \theta}{Sup}\,(g(w)) \right) \vee \left(g(\theta) \wedge \underset{u \geq \theta}{Sup}\,(f(u)) \right)$$

$$= \left(0 \wedge \underset{w \geq \theta}{Sup}\,(g(w)) \right) \vee (g(\theta) \wedge 0) = 0. \tag{1.13}$$

The proof for the joint operation is similar. ■

Theorem 6: Let $F = \bigcup_{i=1}^{M} F_i$ and $G = \bigcup_{j=1}^{N} G_j$ be disjoint compound fuzzy grades of x respectively in T2FSs, F and G, where $M, N \geq 1$ and $S_{F_i} < S_{F_{i+1}}$ and $S_{G_j} < S_{G_{j+1}}$, then

$$\text{(i)} \quad F \sqcap G = \left(\bigcup_{i=1}^{M} F_i \right) \sqcap \left(\bigcup_{j=1}^{N} G_j \right) = \bigcup_{i,j} \left(F_i \sqcap G_j \right)$$

$$= \left(\bigcup_{i} \left(F_i \wedge \left(\underset{S_{G_j} > S_{F_i}}{\vee} H_{G_j} \right) \right) \right) \cup \left(\bigcup_{j} \left(G_j \wedge \left(\underset{S_{F_i} > S_{G_j}}{\vee} H_{F_i} \right) \right) \right) \tag{1.14}$$

$$\text{(ii)} \quad F \sqcup G = \left(\bigcup_{i=1}^{M} F_i \right) \sqcup \left(\bigcup_{j=1}^{N} G_j \right) = \bigcup_{i,j} \left(F_i \sqcup G_j \right)$$

$$= \left(\bigcup_{i} \left(F_i \wedge \left(\underset{S_{G_j} < S_{F_i}}{\vee} H_{G_j} \right) \right) \right) \cup \left(\bigcup_{j} \left(G_j \wedge \left(\underset{S_{F_i} < S_{G_j}}{\vee} H_{F_i} \right) \right) \right). \tag{1.15}$$

Proof: We will first prove (i), and then (ii) can also be proved similarly. $F_l \cap F_k = \emptyset, l \neq k$, $G_l \cap G_k = \emptyset, l \neq k$. Since F and G are disjoint, for any θ either $\theta \in S_F$or $\theta \in S_G$. Given $\theta \in S_{F_i}$, hence $\forall j, g_j(\theta) = 0$,
Case I: Given $\theta \in S_{F_i}$, $1 \leq i \leq M$. Since F is a compound fuzzy grade, $F_l \cap F_k = \emptyset, l \neq k$. Moreover, F and G are disjoint; hence, $g_j(\theta) = 0$, $1 \leq j \leq N$, so

$$\begin{aligned}\mu_{F\sqcap G}(\theta) &= \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right) \\ &= \left(f_i(\theta) \wedge \left(\underset{w\geq\theta}{Sup}\,\big(g_j(w)\big) \vee \ldots \vee \underset{w\geq\theta}{Sup}\,(g_N(w))\right)\right), S_{G_j} > S_{F_i} \\ &= \big(f_i(\theta) \wedge \big(H_{G_j} \vee \ldots \vee H_{G_N}\big)\big), S_{G_j} > S_{F_i} \\ &= \big(f_i(\theta) \wedge H_{G_j}\big) \vee \ldots \vee \big(f_i(\theta) \wedge H_{G_N}\big), S_{G_j} > S_{F_i}. \end{aligned} \tag{1.16}$$

Considering Theorem 5, then for all G_j, $S_{G_j} > S_{F_i}$

$$(F_i \sqcap G_j) \cup \ldots \cup (F_i \sqcap G_N) = (F_i \wedge H_{G_j}) \cup \ldots \cup (F_i \wedge H_{G_N}). \tag{1.17}$$

Hence, for all $\theta \in S_{F_i}$,

$$\mu_{(F_i\sqcap G_j)\cup\ldots\cup(F_i\sqcap G_N)}(\theta) = (f_i(\theta) \wedge H_{G_j}) \cup \ldots \cup (f_i(\theta) \wedge H_{G_N}). \tag{1.18}$$

Case II: Given $\theta \in S_{G_j}$, $1 \leq j \leq N$. Since G is a compound fuzzy grade, $G_l \cap G_k = \emptyset, l \neq k$. Moreover, F and G are disjoint; hence, $f_i(\theta) = 0$, $1 \leq i \leq M$, so

$$\begin{aligned}\mu_{F\sqcap G}(\theta) &= \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right) \\ &= \left(g_j(\theta) \wedge \left(\underset{w\geq\theta}{Sup}\,(f_i(w)) \vee \ldots \vee \underset{w\geq\theta}{Sup}\,(f_M(w))\right)\right), S_{F_i} > S_{G_j} \\ &= \big(g_j(\theta) \wedge \big(H_{F_i} \vee \ldots \vee H_{F_M}\big)\big), S_{F_i} > S_{G_j} \\ &= \big(g_j(\theta) \wedge H_{F_i}\big) \vee \ldots \vee \big(g_j(\theta) \wedge H_{F_M}\big), S_{F_i} > S_{G_j}, \end{aligned} \tag{1.19}$$

Based on Theorem 5, then for all F_i, $S_{F_i} > S_{G_j}$

$$(G_j \sqcap F_i) \cup \ldots \cup (G_j \sqcap F_N) = (G_j \wedge H_{F_i}) \cup \ldots \cup (G_j \wedge H_{F_N}). \tag{1.20}$$

Hence, for all $\theta \in S_{G_j}$,

$$\mu_{(G_j\sqcap F_i)\cup\ldots\cup(G_j\sqcap F_N)}(\theta) = (g_j(\theta) \wedge H_{F_i}) \vee \ldots \vee (g_j(\theta) \wedge H_{F_N}). \tag{1.21}$$

Case III: Given $\nexists i, j$ such that $\theta \in S_{F_i}$ or $\theta \in S_{G_j}$, hence $\forall i, j$, $f_i(\theta) = g_j(\theta) = 0$ and consequently $\mu_{F_i\sqcap G_j}(\theta) = 0$. ■

Corollary 4: Let F and G be disjoint fuzzy grades as defined in Theorem 6, then

(i) $\forall\theta \in S_{F_i}, i \in \{1,2,\ldots,M\}$, where $S_{F_i} > S_{G_N}$ then $\mu_{F\sqcap G}(\theta) = 0$

(ii) $\forall\theta \in S_{F_i}, i \in \{1,2,\ldots,M\}$, where $S_{F_i} < S_{G_N}$ then $\mu_{F\sqcap G}(\theta) = 0$

Proof: Based on Theorem 6, the proof is straightforward.

1.4.2 Join and Meet Operations on Non-Disjoint Convex Fuzzy Grades

Theorem 6 proved in Sect. 1.4.1 is restricted to disjoint fuzzy grades. However, in this section we elaborate on convex fuzzy grades that are not essentially disjoint. Using the fuzzy disjointing difference operator and the properties proved in Theorems 1–3 and corollaries 1–3, we will extend Theorem 6 to non-disjoint convex fuzzy grades. We will show that calculating join (meet) of two fuzzy grades can be fulfilled based on the join (meet) of the disjoint fuzzy grades corresponded to the fuzzy grades that are calculated using the introduced DDO.

Theorem 7: Let F and G be normal convex fuzzy grades of x respectively in T2FSs, F and G, where $C_F \not\subseteq C_G, C_G \not\subseteq C_F$, $F' = F - G = \bigcup_{i=1}^{M} {F_i}'$ and $G' = G - F = \bigcup_{j=1}^{N} {G'}_j$,

$$\text{(i)} \quad F \sqcap G = \left(F' \sqcap G'\right) \cup (F \cap G) \tag{1.22}$$

$$\text{(ii)} \quad F \sqcup G = \left(F' \sqcup G'\right) \cup (F \cap G). \tag{1.23}$$

Proof: We will prove (i), and (ii) can be proved in the same manner. We investigate $\mu_{F \sqcap G}(\theta)$ in three cases. In each case we compare membership grades in $F \sqcap G$ with $\left(F' \sqcap G'\right) \cup (F \cap G)$.
(a) Given $f(\theta) > g(\theta)$, then the membership grade of θ in $F \sqcap G$ is

$$\begin{aligned}
\mu_{F \sqcap G}(\theta) &= \left(f(\theta) \wedge \underset{w \geq \theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u \geq \theta}{Sup}\,(f(u))\right) \\
&= \left(f(\theta) \wedge \underset{w \geq \theta}{Sup}\,(g(w))\right) \vee (g(\theta)) \\
&= (f(\theta) \vee g(\theta)) \wedge \left(g(\theta) \vee \underset{w \geq \theta}{Sup}\,(g(w))\right).
\end{aligned} \tag{1.24}$$

Knowing $f(\theta) > g(\theta)$, indicates $f'(\theta) = f(\theta)$ and $g'(\theta) = 0$; hence, the membership grade of θ in $(F' \sqcap G') \cup (F \cap G)$ is

$$\begin{aligned}
\mu_{(F' \sqcap G') \cup (F \cap G)}(\theta) &= \left(f'(\theta) \wedge \underset{w \geq \theta}{Sup}\left(g'(w)\right)\right) \vee \left(g'(\theta) \wedge \underset{u \geq \theta}{Sup}\left(f'(u)\right)\right) \vee (f(\theta) \wedge g(\theta)) \\
&= \left(f'(\theta) \wedge \underset{w \geq \theta}{Sup}\left(g'(w)\right)\right) \vee (f(\theta) \wedge g(\theta)) \\
&= \left(f'(\theta) \wedge \underset{w \geq \theta}{Sup}\left(g'(w)\right)\right) \vee g(\theta) \\
&= \left(f'(\theta) \vee g(\theta)\right) \wedge \left(\underset{w \geq \theta}{Sup}\left(g'(w)\right) \vee g(\theta)\right).
\end{aligned} \tag{1.25}$$

Since $C_G \not\subseteq C_F$, hence $\exists \theta' s.t. g'(\theta') = 1$. Consequently, if $\theta \leq \theta'$, then (1.20) and (1.21) are

$$\mu_{F \sqcap G}(\theta) = (f(\theta) \vee g(\theta)) \wedge \left(g(\theta) \vee \underset{w \geq \theta}{Sup} (g(w)) \right) = (f(\theta) \vee g(\theta)) \tag{1.26}$$

$$\begin{aligned} \mu_{(F' \sqcap G') \cup (F \cap G)}(\theta) &= \left(f'(\theta) \vee g(\theta) \right) \wedge \left(\underset{w \geq \theta}{Sup} \left(g'(w) \right) \vee g(\theta) \right) \\ &= f'(\theta) \vee g(\theta) = f(\theta) \vee g(\theta). \end{aligned} \tag{1.27}$$

However, if $\theta \geq \theta'$, due to the convexity of G, $Sup_{w \geq \theta} (g(w)) = g(\theta)$; hence, (1.20) and (1.21) are

$$\begin{aligned} \mu_{F \sqcap G}(\theta) &= (f(\theta) \vee g(\theta)) \wedge \left(g(\theta) \vee \underset{w \geq \theta}{Sup} (g(w)) \right) \\ &= (f(\theta) \vee g(\theta)) \wedge g(\theta) = g(\theta) \end{aligned} \tag{1.28}$$

$$\begin{aligned} \mu_{(F' \sqcap G') \cup (F \cap G)}(\theta) &= \left(f'(\theta) \vee g(\theta) \right) \wedge \left(\underset{w \geq \theta}{Sup} \left(g'(w) \right) \vee g(\theta) \right) \\ &= \left(f'(\theta) \vee g(\theta) \right) \wedge g(\theta) = g(\theta). \end{aligned} \tag{1.29}$$

(b) Given $g(\theta) > f(\theta)$, then the membership grade of θ in $F \sqcap G$ is

$$\begin{aligned} \mu_{F \sqcap G}(\theta) &= \left(f(\theta) \wedge \underset{w \geq \theta}{Sup} (g(w)) \right) \vee \left(g(\theta) \wedge \underset{u \geq \theta}{Sup} (f(u)) \right) \\ &= f(\theta) \vee \left(g(\theta) \wedge \underset{u \geq \theta}{Sup} (f(u)) \right) \\ &= (f(\theta) \vee g(\theta)) \wedge \left(f(\theta) \vee \underset{u \geq \theta}{Sup} (f(u)) \right). \end{aligned} \tag{1.30}$$

Knowing $g(\theta) > f(\theta)$, indicates $g'(\theta) = g(\theta)$ and $f'(\theta) = 0$; hence, the membership grade of θ in $\left(F' \sqcap G' \right) \cup (F \cap G)$ is

$$\begin{aligned} \mu_{(F' \sqcap G') \cup (F \cap G)}(\theta) &= \left(f'(\theta) \wedge \underset{w \geq \theta}{Sup} \left(g'(w) \right) \right) \vee \left(g'(\theta) \wedge \underset{u \geq \theta}{Sup} \left(f'(u) \right) \right) \vee (f(\theta) \wedge g(\theta)) \\ &= \left(g'(\theta) \wedge \underset{u \geq \theta}{Sup} \left(f'(u) \right) \right) \vee f(\theta) \\ &= \left(g'(\theta) \vee f(\theta) \right) \wedge \left(\underset{u \geq \theta}{Sup} \left(f'(u) \right) \vee f(\theta) \right). \end{aligned} \tag{1.31}$$

Since $C_F \not\subseteq C_G$, hence $\exists \theta' \ s.t. \ f'(\theta') = 1$. Consequently, if $\theta \leq \theta'$, then (1.26) and (1.27) are

$$\mu_{F \sqcap G}(\theta) = (f(\theta) \vee g(\theta)) \wedge \left(f(\theta) \vee \underset{u \geq \theta}{Sup} (f(u)) \right) = (f(\theta) \vee g(\theta)) = g(\theta) \tag{1.32}$$

$$\begin{aligned}\mu_{(F'\sqcap G')\cup(F\cap G)}(\theta) &= \big(g'(\theta)\vee f(\theta)\big)\wedge\left(\underset{u\geq\theta}{Sup}\big(f'(u)\big)\vee f(\theta)\right)\\ &= \big(g'(\theta)\vee f(\theta)\big) = g(\theta).\end{aligned} \tag{1.33}$$

However, if $\theta \geq \theta'$ and due to convexity of F, $Sup_{u\geq\theta}(f(u)) = f(u)$; hence, (1.26) and (1.27) are

$$\begin{aligned}\mu_{F\sqcap G}(\theta) &= (f(\theta)\vee g(\theta))\wedge\left(f(\theta)\vee\underset{u\geq\theta}{Sup}(f(u))\right)\\ &= (f(\theta)\vee g(\theta))\wedge f(\theta) = f(\theta)\end{aligned} \tag{1.34}$$

$$\begin{aligned}\mu_{(F'\sqcap G')\cup(F\cap G)}(\theta) &= \big(g'(\theta)\vee f(\theta)\big)\wedge\left(\underset{u\geq\theta}{Sup}\big(f'(u)\big)\vee f(\theta)\right)\\ &= \big(g'(\theta)\vee f(\theta)\big)\wedge f(\theta) = f(\theta).\end{aligned} \tag{1.35}$$

(c) $f(\theta) = g(\theta)$, then $f'(\theta) = g'(\theta) = 0$, so

$$\begin{aligned}\mu_{F\sqcap G}(\theta) &= \left(f(\theta)\wedge\underset{w\geq\theta}{Sup}(g(w))\right)\vee\left(g(\theta)\wedge\underset{u\geq\theta}{Sup}(f(u))\right)\\ &= f(\theta)\vee g(\theta) = f(\theta) = g(\theta)\end{aligned} \tag{1.36}$$

$$\begin{aligned}\mu_{(F'\sqcap G')\cup(F\cap G)}(\theta) &= \left(f'(\theta)\wedge\underset{w\geq\theta}{Sup}\big(g'(w)\big)\right)\vee\left(g'(\theta)\wedge\underset{u\geq\theta}{Sup}\big(f'(u)\big)\right)\vee(f(\theta)\wedge g(\theta))\\ &= f(\theta)\wedge g(\theta) = f(\theta) = g(\theta).\end{aligned} \tag{1.37}$$

■

Theorem 8: Let F and G be fuzzy grades as defined in Theorem 6, where $C_F \subseteq C_G$, then

$$\text{(i)}\quad F\sqcap G = \left(\bigcup_{S_{F'_i}<C_F} F'_i\right)\cup\left(\bigcup_{S_{G'_j}<C_F} G'_j\right)\cup(F\cap G) \tag{1.38}$$

$$\text{(ii)}\quad F\sqcup G = \left(\bigcup_{S_{F'_i}>C_F} F'_i\right)\cup\left(\bigcup_{S_{G'_j}>C_F} G'_j\right)\cup(F\cap G). \tag{1.39}$$

Proof: We will prove (i), and (ii) can then be proved similarly. We investigate $\mu_{F\sqcap G}(\theta)$ in two cases. In each case we compare corresponding membership grades in $\varphi_1 = F\sqcap G$ with $\varphi_2 = \Big(\underset{S_{F'_i}<C_F}{\cup} F'_i\Big)\cup\Big(\underset{S_{G'_j}<C_F}{\cup} F'_i\Big)\cup\big(F\cap G\big)$.

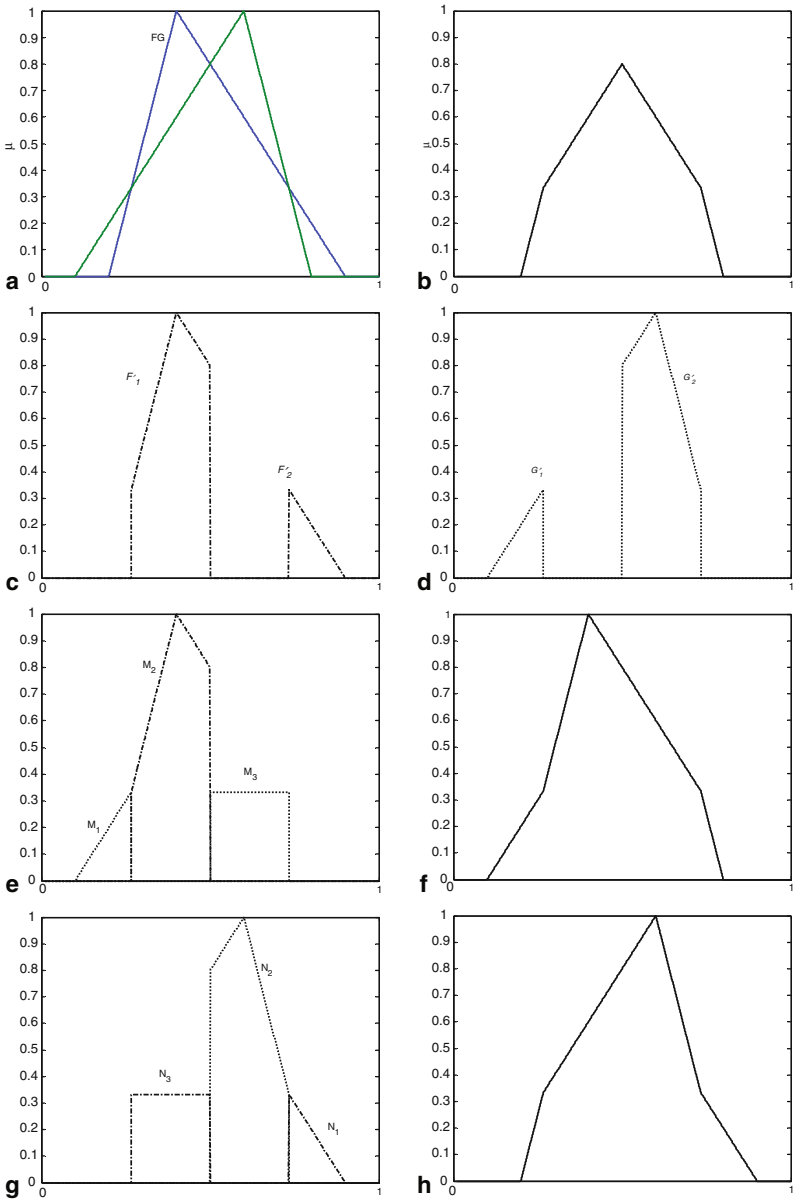


Fig. 1.2 Calculating meet and join of two convex normal fuzzy grades based on Theorems 6 and 7. (**a**) convex fuzzy grades F and G, (**b**) intersection of the fuzzy grades, i.e., $(F \cap G)$ (**c**) disjoin difference F–G, i.e., $F' = F - G = \bigcup_{i=1}^{2} {F'}_i$, (**d**) disjoin difference G–F, i.e., $G' = G - F = \bigcup_{j=1}^{2} {G'}_j$, (**e**) meet of the disjoint fuzzy sets F' and G', i.e., $\left(F' \sqcap G'\right)$ (**f**) meet of the fuzzy grades F and G, $F \sqcap G = \left(F' \sqcap G'\right) \cup (F \cap G)$ (**g**) join of the disjoint fuzzy sets F' and G', i.e., $\left(F' \sqcup G'\right)$ (**h**) join of the fuzzy grades F and G, $F \sqcup G = \left(F' \sqcup G'\right) \cup (F \cap G)$

(a) Given, $\theta > C_F$, then

$$\begin{aligned}\mu_{\varphi_1}(\theta) &= \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right) \\ &= \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee (g(\theta) \wedge f(\theta)) \\ &= g(\theta) \wedge f(\theta). \end{aligned} \tag{1.40}$$

However, for $\theta > C_F$, we have, $\left(\underset{S_{F'_i}<C_F}{\cup} F'_i\right)(\theta) = 0$ and $\left(\underset{S_{G'_j}<C_F}{\cup} F'_i\right)(\theta) = 0$, it follows then

$$\mu_{\varphi_2}(\theta) = 0 \vee 0 \vee (g(\theta) \wedge f(\theta)) = g(\theta) \wedge f(\theta). \tag{1.41}$$

(b) Given $\theta \in C_F$ or $\theta < C_F$, since $C_F \subseteq C_G$ so $\exists x \geq \theta \; s.t. \; f(x) = g(x) = 1$; hence,

$$\mu_{\varphi_1}(\theta) = \left(f(\theta) \wedge \underset{w\geq\theta}{Sup}\,(g(w))\right) \vee \left(g(\theta) \wedge \underset{u\geq\theta}{Sup}\,(f(u))\right) = f(\theta) \vee g(\theta) \tag{1.42}$$

(b.1) if $f(\theta) > g(\theta)$, then $\mu_{\varphi_1}(\theta) = f(\theta)$ and

$$\mu_{\varphi_2}(\theta) = f'(\theta) \vee g'(\theta) \vee (f(\theta) \wedge g(\theta)) = f(\theta) \vee (f(\theta) \wedge g(\theta)) = f(\theta) \tag{1.43}$$

(b.2) if $g(\theta) > f(\theta)$, then $\mu_{\varphi_1}(\theta) = g(\theta)$ and

$$\mu_{\varphi_2}(\theta) = f'(\theta) \vee g'(\theta) \vee (f(\theta) \wedge g(\theta)) = g(\theta) \vee (f(\theta) \wedge g(\theta)) = g(\theta) \tag{1.44}$$

(b.3) if $f(\theta) = g(\theta)$, then $\mu_{\varphi_1}(\theta) = f(\theta) = g(\theta)$ and

$$\mu_{\varphi_2}(\theta) = f'(\theta) \vee g'(\theta) \vee (f(\theta) \wedge g(\theta)) = f(\theta) \wedge g(\theta) = f(\theta) = g(\theta). \tag{1.45}$$

■

Corollary 5: Let F and G be fuzzy grades with the heights of H_F and H_G, then

$$\text{(i)} \quad F \sqcap G = (F \wedge H_G) \sqcap (G \wedge H_F) \tag{1.46}$$

$$\text{(ii)} \quad F \sqcup G = (F \wedge H_G) \sqcup (G \wedge H_F) \tag{1.47}$$

Proof: Straightforward.

Based on Corollary 6, we observe that Theorems 6 and 7 can be easily extended to the convex fuzzy grades with different heights. Figure 1.2 depicts an example of calculating join and meet of two convex fuzzy grades. Fuzzy grades are plotted in Fig. 1.2a and their intersection is shown in Fig. 1.2b. Figures 1.2c and 1.2d,

respectively represent $F' = F - G = F_1{}' \cup F_2{}'$ and $G' = G - F = G'_1 \cup G'_2$, where $F' \cap G' = \emptyset$, $F_1{}' \cap F_2{}' = \emptyset$ and $G'_1 \cap G'_2 = \emptyset$. Since $F' \cap G' = \emptyset$, hence $F' \sqcap G'$ (and similarly $F' \sqcup G'$) is calculated using Theorem 5:

$$\begin{aligned} F' \sqcap G' &= \left(F'_1 \sqcap G'_1\right) \cup \left(F'_1 \sqcap G'_2\right) \cup \left(F'_2 \sqcap G'_1\right) \cup \left(F'_2 \sqcap G'_2\right) \\ &= \left(G'_1 \wedge H_{F'_1}\right) \cup \left(F'_1 \wedge H_{G'_2}\right) \cup \left(G'_1 \wedge H_{F'_2}\right) \cup \left(G'_2 \wedge H_{F'_2}\right) \\ &= \underbrace{G'_1}_{M_1} \cup \underbrace{F'_1}_{M_2} \cup \underbrace{\left(G'_2 \wedge H_{F'_2}\right)}_{M_3} \end{aligned} \tag{1.48}$$

1.5 Conclusion

In this chapter, we have introduced a disjointing difference operator on fuzzy sets and investigated its related properties. Based on the proposed disjointing difference operator, we have discussed a novel and easy-to-follow algorithm for performing the union and intersection operations on T2FSs with respect to min t-norm and max t-conorm. The idea is based on the fact that if we remove the commonalities between two convex fuzzy grades that are defined on the real line, then calculating the join and meet operations using min t-norm and max t-conorm on the resulting disjoint fuzzy subsets is straightforward.

References

1. L.A. Zadeh, Fuzzy sets. Inform. Control **8**, 338–353 (1965)
2. G.J. Klir, T.A. Folger, *Fuzzy Sets, Uncertainty, and Information* (Prentice Hall, Englewood Cliffs, 1988)
3. W. Pedrycz, Shadowed sets: Representing and processing fuzzy sets. IEEE Trans. Syst. Man Cybern. B **28**, 103–109 (1998)
4. J.M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions* (Prentice-Hall, Upper Saddle River, 2001)
5. R.I. John, S. Coupland, in *Computational Intelligence: Principles and Practice.* Extensions to Type-1 Fuzzy: Type-2 Fuzzy Logic and Uncertainty (IEEE Computational Intelligence Society, 2006), pp. 89–102
6. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-1. Inform. Sci. **8**, 199–249 (1975)
7. H. Tahayori, A.G.B. Tettamanzi, G. Degli Antoni, A. Visconti, in *Fuzzy Sets and Systems*. On the Calculation of Extended Max and Min Operations between Convex Fuzzy Sets of the Real Line vol. 160(21), pp. 3103–3114, 2009.
8. J.M. Mendel, R.I. John, Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
9. H. Tahayori, A.G.B. Tettamanzi, G. Degli Antoni, in *IEEE World Congress on Computational Intelligence*. Approximated type-2 fuzzy set operations, Vancouver, 2006, pp. 1910–1917
10. H. Tahayori, G. Degli Antoni, in *Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS).* A simple method for performing type-2 fuzzy set operations based on highest degree of intersection hyperplane, San Diego, 2007, pp. 404–409

11. S. Coupland, R. John, New geometric inference techniques for type-2 fuzzy sets. J Fuzzy Syst. **10**(4), 276–286 (2008)
12. H. Tahayori, A.G.B. Tettamanzi, G. Degli Antoni, A. Visconti, M. Moharrer, Concave type-2 fuzzy sets: Properties and operations. Soft Comput. J. **14**(7), 749–756 (2010)
13. H. Tahayori, A. Sadeghian, A. Visconti, in *Proc. of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*. Operations on Type-2 fuzzy sets based on the set of pseudo-highest intersection points of convex fuzzy sets, Toronto, 2010, pp. 1–6
14. N.N Karnik, J.M. Mendel, Operations on type-2 fuzzy sets. Fuzzy Set. Syst. **122**, 327–348 (2001)
15. C.L. Walker, E.A. Walker, The algebra of fuzzy truth values. Fuzzy Set. Syst. **149**, 309–347 (2005)
16. Z. Gera, J. Dombi, Exact calculations of extended logical operations on fuzzy truth values. Fuzzy Set. Syst. **159**, 1309–1326 (2008)
17. S. Coupland, R.I. John, Geometric type-1 and type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **15**(1), 3–15 (2007)
18. J.T. Starczewski, Efficient triangular type-2 fuzzy logic systems. Int. J. Approx. Reason **50**(5), 799–811 (2009).
19. S. Coupland, R. John, New geometric inference techniques for type-2 fuzzy sets. Int. J. Approx. Reason **49**(1), 198–211 (2008)
20. J.M. Mendel, F. Liu, D. Zhai, Alpha-plane representation for type-2 fuzzy sets: Theory and applications. IEEE Trans. Fuzzy Syst. **17**(5), 1189–1207 (2009)
21. C. Wagner, H. Hagras, in *IEEE International Conference on Fuzzy Systems*. zSlices—Towards bridging the gap between interval and general type-2 fuzzy logic, Hong Kong, 2008, pp. 489–497
22. J.M. Mendel, R.I. John, F. Liu, Interval type-2 fuzzy logic systems made simple. IEEE Trans. Fuzzy Syst. **14**(6), 808–821 (2006)
23. H. Tahayori, A. Sadeghian, in *World Automation Congress.* Handling uncertainties of membership functions with shadowed fuzzy sets (2012), pp. 1–5
24. H. Tahayori, A. Sadeghian, in *New Concepts and Applications in Soft Computing, SCI 417*, eds. by V.E. Balas et al. 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 (Springer-Verlag, Berlin Heidelberg, 2013), pp. 97–117
25. H. Tahayori, A. Sadeghian, W. Pedrycz, Induction of shadowed sets based on the gradual grade of fuzziness. IEEE Trans. Fuzzy Syst. **21**(5), 937–949 (2013)
26. D.W. Roberts, An anticommutative difference operator for fuzzy sets and relations. Fuzzy Set. Syst. **21**(1), 35–42 (1987)
27. L.A. Fono, H. Gwet, B. Bouchon-Meunier, Fuzzy implication operators for difference operations for fuzzy sets and cardinality-based measures of comparison. Eur. J. Oper. Res. **183**(1), 314–326 (2007)
28. M. Mizumoto, K. Tanaka, Some properties of fuzzy sets of type-2. Inf. Control **31**, 312–340 (1976)
29. E.R. Moore, *Interval Analysis* (Prentice-Hall, Englewood Cliffs, 1966)

Chapter 2
Robustness of Higher-Order Fuzzy Sets

Mohammad Biglarbegian

Abstract Robustness is an important metric in analysis and design of interval type-2 fuzzy logic systems (IT2 FLSs). This chapter presents a mathematical approach to determine the robustness of IT2 FLSs that have Takag-Sugeno-Kang (TSK) structure. We present numerical examples to demonstrate how the developed methodologies can be applied. The presented approach herein provides a systematic method for robust analysis of IT2 FLSs to further enhance their applications.

2.1 Introduction

This chapter presents a systematic approach for the analysis and design of robust interval type-2 fuzzy logic systems (IT2 FLSs) with Takag-Sugeno-Kang (TSK) structure. The reason that we adopt the TSK structure is because this model structure allows for rigorous mathematical analyses on IT2 TSK FLSs.

IT2 FLSs have proven to handle uncertainties better than their type-1 (T1) counterparts. Robustness of any nonlinear system is inherently very essential when it is used for modeling or control. Robustness of T1 FLSs has been investigated in the literature [4]. To further expand their range of applications of IT2 FLSs, it is, hence, needed to investigate their important properties, such as robustness. Therefore, we focus on this important topic and present methodologies to determine the robustness of IT2 TSK FLSs and demonstrate how the developed methods can be used to design IT2 TSK FLSs.

This chapter is organized as follows: Sect. 2.2 presents background on IT2 FLSs, Sect. 2.3 presents an inference engine for analyses, Sect. 2.4 presents the main portion

The majority of this chapter has been adopted by the authors previous work [1–5]. (Biglarbegian et al. 2011).

M. Biglarbegian (✉)
School of Engineering University of Guelph Guelph,
Ontario, Canada
e-mail: mbiglarb@uoguelph.ca

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_2

of the chapter by providing the robustness analyses, Sect. 2.5 provides examples and conclusions are given in Sect. 2.6.

2.2 Background

This section provides preliminaries on IT2 TSK FLSs that are necessary to establish the main robustness development. We first introduce the rule structure, followed by some well-known inference mechanisms used in the literature.

2.2.1 IT2 FLSs

The ith rule for a continuous system is written as

$$R^i : \text{If } x_1 \text{ is } \widetilde{F}_1^i \text{ and } x_2 \text{ is } \widetilde{F}_2^i \text{ and} \cdots \text{and } x_p \text{ is } \widetilde{F}_p^s, \text{ then}$$
$$y_i = a_0^i + a_1^i x_1 + \cdots + a_p^i x_p \tag{2.1}$$

The lower and upper firing strengths of the ith rule are given by

$$\underline{f}^i(\boldsymbol{x}) = \underline{\mu}_{\widetilde{F}_1^i}(x_1) * \cdots * \underline{\mu}_{\widetilde{F}_p^i}(x_p) \tag{2.2}$$

$$\overline{f}^i(\boldsymbol{x}) = \overline{\mu}_{\widetilde{F}_1^i}(x_1) * \cdots * \overline{\mu}_{\widetilde{F}_p^i}(x_p) \tag{2.3}$$

where $\boldsymbol{x} = [x_1, x_2, \cdots, x_p]^T$ is the state vector containing all the inputs, and $\underline{\mu}_{\widetilde{F}}$ and $\overline{\mu}_{\widetilde{F}}$ are lower and upper membership functions, respectively; the t-norm operator is shown by $*$.

2.2.2 IT2 TSK FLS

Using the Karnik–Mendel algorithms [5], the final output of an IT2 FLS is given as follows:

$$Y_{\text{IT2}}(\boldsymbol{x}) = [y_l(\boldsymbol{x}), y_r(\boldsymbol{x})] = \int_{f^1 \in \left[\underline{f}^1, \overline{f}^1\right]} \cdots \int_{f^M \in \left[\underline{f}^M, \overline{f}^M\right]} 1 \Big/ \frac{\sum_{i=1}^M f^i(\boldsymbol{x}) y_i}{\sum_{i=1}^M f^i(\boldsymbol{x})} \tag{2.4}$$

where y_i is given by 2.1, and Eqs. 2.2 and 2.3 are used to determine the firing strengths.

Since Y_{IT2} is an interval type-1 set and only a function of its left and right end points, i.e., y_l and y_r. These end points can be computed using the KM algorithms. The final output is thus given by

$$Y_{\text{output}}(\boldsymbol{x}) = \frac{y_l(\boldsymbol{x}) + y_r(\boldsymbol{x})}{2}. \tag{2.5}$$

Needless to say, 2.5 does not represent a closed-form expression because of 2.4.

Having a closed-form expression for rigorous mathematical analysis on the robustness is preferred. Since the KM algorithms cannot provide a closed-form expression, we introduce a simple and an innovative inference mechanism that will be used throughout the rest of the chapter for our analysis.

2.3 An Innovative Inference Mechanism for Analysis of IT2 TSK FLSs

As mentioned earlier, we need to use an inference engine that offers a closed-form. We adopt a novel inference engine that was proposed in [1, 2] and has been used successfully in several IT2 FLSs design, analysis, and control. This inference engine, called $m-n$, is used in the rest of the paper. The $m-n$ inference engine is [2]

$$Y_{m-n}(\boldsymbol{x}) = m\frac{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})\,y_i}{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})} + n\frac{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})\,y_i}{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})} \tag{2.6}$$

where y_i is the output of each rule, and m and n are design parameters and will be chosen according to the design/control criteria.

2.4 Robustness of IT2 TSK FLSs

We formally define robustness in this section and present methodologies for the analysis of robust IT2 TSK FLSs.

We define robustness as follows: The maximum deviations of inputs of a system resulting in a maximum allowable output deviation. In other words, if we show the desired (allowable) output deviation with ΔY_{des}, the robustness problem is to find $|\Delta x_i|$'s such that the system output, Y_{out}, satisfies the following: $|\Delta Y_{out}| \leq |\Delta Y_{desired}|$.

We formulate this definition as a classical optimization problem:

$$\begin{cases} \text{Maximize :} & |\Delta x_1|, \cdots, |\Delta x_p| \\ & |\Delta Y_{out}| \leq |\Delta Y_{des}| \\ & \Delta x_{1\min} \leq \Delta x_1 \leq \Delta x_{1\max} \\ & \Delta x_{2\min} \leq \Delta x_2 \leq \Delta x_{2\max} \\ & \vdots \\ \text{Subject to :} & \Delta x_{p\min} \leq \Delta x_p \leq \Delta x_{p\max} \end{cases}$$

Next, we derive an expression for ΔY_{out}. The deviation in the output as a result of the change in the inputs, $\boldsymbol{x} + \Delta \boldsymbol{x}$, is calculated as follows:

$$\begin{aligned}\Delta Y &= Y(\boldsymbol{x} + \Delta \boldsymbol{x}) - Y(\boldsymbol{x}) \\ &= m \sum_{i=1}^{M} \underline{h}^i (\boldsymbol{x} + \Delta \boldsymbol{x})\, y_i(\boldsymbol{x} + \Delta \boldsymbol{x}) + n \sum_{i=1}^{M} \overline{h}^i (\boldsymbol{x} + \Delta \boldsymbol{x})\, y_i(\boldsymbol{x} + \Delta \boldsymbol{x}) \\ &\quad - m \sum_{i=1}^{M} \underline{h}^i (\boldsymbol{x})\, y_i(\boldsymbol{x}) - n \sum_{i=1}^{M} \overline{h}^i (\boldsymbol{x})\, y_i(\boldsymbol{x}) \end{aligned} \tag{2.7}$$

where $\underline{h}$ and $\overline{h}$ are given by

$$\underline{h}_i (\boldsymbol{x}) = \frac{\underline{f}_i (\boldsymbol{x})}{\sum_{i=1}^{M} \underline{f}_i (\boldsymbol{x})} \tag{2.8}$$

$$\overline{h}_i (\boldsymbol{x}) = \frac{\overline{f}_i (\boldsymbol{x})}{\sum_{i=1}^{M} \overline{f}_i (\boldsymbol{x})}. \tag{2.9}$$

Note that for small input deviations, we can use the following approximations:

$$\underline{h}^i (\boldsymbol{x} + \Delta \boldsymbol{x}) \simeq \underline{h}^i (\boldsymbol{x}) + \sum_{j=1}^{p} \frac{\partial \underline{h}^i (\boldsymbol{x})}{\partial x_j} \Delta x_j \tag{2.10}$$

$$y^i (\boldsymbol{x} + \Delta \boldsymbol{x}) \simeq y^i (\boldsymbol{x}) + \sum_{j=1}^{p} \frac{\partial y^i (\boldsymbol{x})}{\partial x_j} \Delta x_j. \tag{2.11}$$

We can similarly use

$$\overline{h}^i (\boldsymbol{x} + \Delta \boldsymbol{x}) \simeq \overline{h}^i (\boldsymbol{x}) + \sum_{j=1}^{p} \frac{\partial \overline{h}^i (\boldsymbol{x})}{\partial x_j} \Delta x_j. \tag{2.12}$$

For simplicity in notations, we drop $\boldsymbol{x}$ in the rest of the following derivations. Considering the first and fourth terms of 2.7:

$$\begin{aligned} & m \sum_{i=1}^{M} \left[\left(\underline{h}^i + \sum_{j=1}^{p} \frac{\partial \underline{h}^i}{\partial x_j} \Delta x_j \right) \left(y^i + \sum_{j=1}^{p} \frac{\partial y^i}{\partial x_j} \Delta x_j \right) \right] - m \sum_{i=1}^{M} \underline{h}^i y_i \\ &= m \sum_{i=1}^{M} \left[\underline{h}^i y^i + \underline{h}^i \sum_{j=1}^{p} \frac{\partial y^i}{\partial x_j} \Delta x_j + y^i \sum_{j=1}^{p} \frac{\partial \underline{h}^i}{\partial x_j} \Delta x_j \right. \\ &\quad \left. + \left(\sum_{j=1}^{p} \frac{\partial \underline{h}^i}{\partial x_j} \Delta x_j \right) \left(\sum_{j=1}^{p} \frac{\partial y^i}{\partial x_j} \Delta x_j \right) \right] - m \sum_{i=1}^{M} \underline{h}^i y_i. \end{aligned} \tag{2.13}$$

Since input deviations are small, the quadratic terms of input deviations are negligible. Thus, we can simply express 2.13 as

$$m\sum_{i=1}^{M}\left[\underline{h}^i y^i + \underline{h}^i \sum_{j=1}^{p}\frac{\partial y^i}{\partial x_j}\Delta x_j + y^i \sum_{j=1}^{p}\frac{\partial \underline{h}^i}{\partial x_j}\Delta x_j\right] - m\sum_{i=1}^{M}\underline{h}^i y_i$$
$$= m\sum_{i=1}^{M}\left[\underline{h}^i \sum_{j=1}^{p}\frac{\partial y^i}{\partial x_j}\Delta x_j + y^i \sum_{j=1}^{p}\frac{\partial \underline{h}^i}{\partial x_j}\Delta x_j\right]. \tag{2.14}$$

Similarly, the second and third terms of 2.7 are expressed as

$$n\sum_{i=1}^{M}\left[\overline{h}^i \sum_{j=1}^{p}\frac{\partial y^i}{\partial x_j}\Delta x_j + y^i \sum_{j=1}^{p}\frac{\partial \overline{h}^i}{\partial x_j}\Delta x_j\right]. \tag{2.15}$$

Using 2.14 and 2.15, the output of the IT2 TSK FLS is simply expressed as

$$\Delta Y = m\sum_{i=1}^{M}\left[\underline{h}^i \sum_{j=1}^{p}\frac{\partial y^i}{\partial x_j}\Delta x_j + y^i \sum_{j=1}^{p}\frac{\partial \underline{h}^i}{\partial x_j}\Delta x_j\right]$$
$$+n\sum_{i=1}^{M}\left[\overline{h}^i \sum_{j=1}^{p}\frac{\partial y^i}{\partial x_j}\Delta x_j + y^i \sum_{j=1}^{p}\frac{\partial \overline{h}^i}{\partial x_j}\Delta x_j\right] \tag{2.16}$$

which can be expressed as

$$\Delta Y_{out} = \sum_{j=1}^{p}\sum_{i=1}^{M}\left[m\underline{h}^i \frac{\partial y^i}{\partial x_j} + my^i \frac{\partial \underline{h}^i}{\partial x_j}\right].\Delta x_j + \sum_{j=1}^{p}\sum_{i=1}^{M}\left[n\overline{h}^i \frac{\partial y^i}{\partial x_j} + ny^i \frac{\partial \overline{h}^i}{\partial x_j}\right].\Delta x_j. \tag{2.17}$$

We can express 2.17 in the following compact form:

$$\Delta Y_{out} = \sum_{j=1}^{p}\alpha_j \Delta x_j \tag{2.18}$$

where

$$\alpha_j = \sum_{i=1}^{M}\left[m\underline{h}^i \frac{\partial y^i}{\partial x_j} + my^i \frac{\partial \underline{h}^i}{\partial x_j}\right] + \sum_{i=1}^{M}\left[n\overline{h}^i \frac{\partial y^i}{\partial x_j} + ny^i \frac{\partial \overline{h}^i}{\partial x_j}\right]. \tag{2.19}$$

Therefore, the expression given by 2.18 can be used to solve the optimization problem formulated above.

2.5 Examples

This section present three examples to demonstrate the proposed methodologies.

2.5.1 Example 1

This example is adopted from [3] where T1 and IT2 TSK FLSs were developed to approximate a nonlinear function with two inputs given by $f(x_1, x_2) = \frac{\sin(x1)}{x1} . \frac{\sin(x2)}{x2}$. Gaussian membership functions were used to design the FLSs; the membership parameters are summarized in Table 2.1. Note that the mean of the Gaussian functions for T1 and IT2 are the same. The two free parameters of the IT2 TSK FLS (tuning parameters) are chosen as $m = 0.3$ and $n = 0.1$.

Performance of the two system in terms of modeling and robustness to different input perturbations are shown in Table 2.2; e indicates the error in function approximation, EPI stands for error performance improvement, $R_i = \frac{\max|\Delta x_i|}{\Delta Y_{des}}$, and RPI is an indication for robust performance improvement. It is evident that the IT2 reveals a better approximator in modeling the target nonlinear function. In terms of robustness, the two systems exhibit similar performances, although T1 is slightly more robust.

2.5.2 Example 2

In this example, we study the robustness of another nonlinear system with two inputs. The function considered is $f(x_1, x_2) = \frac{cos(x_1)+sin(x_2)}{x_1+x_2}$. The parameters of the designed T1 and T2 FLSs are summarized in Table 2.3; the means of the Gaussian memberships for T1 and IT2 are chosen to be the same, and the tuning parameters of the IT2 FLS are: $m = 1.1, n = 0.1$.

The error generated in function approximation by T1 and IT2 are $T1_{error} = 46.8473$ and $T2_{error} = 33.6858$. For a given output deviation of $\Delta Y_{des} = 0.34$, the maximum output deviations for T1 are: $\delta x_1 = 3.6843e-4$ and $\delta x_2 = 0.1998$, and for IT2 $\delta x_1 = 8.8789e-4$ and $\delta x_2 = 0.1142$. Therefore, IT2 is significantly better function approximator. With regards to robustness, the first input IT2 can reveal more robustness, however, for the second input T1 can have a better robustness.

2.5.3 Example 3

In this example, we examine the robustness of another nonlinear system which has three inputs. This expression for this function is given by $f(x_1, x_2, x_3) = x_1^{0.12}/x_2 + x_2.x_3$. The membership function parameters are given in Table 2.4; the values for the IT2 tuning parameters are $m = 1.1$ and $n = -0.41$.

Table 2.1 Membership function parameters of T1 and IT2 TSK FLSs: first example

	First Input				Second Input			
	MF1	*MF2*	*MF3*	*MF4*	*MF1*	*MF2*	*MF3*	*MF4*
T1	$\sigma = 2.9290$ $\mu = -9.7910$	$\sigma = 1.7180$ $\mu = -3.9740$	$\sigma = 2.2510$ $\mu = 1.1110$	$\sigma = 3.7520$ $\mu = 8.0220$	$\sigma = 3.0390$ $\mu = -9.3970$	$\sigma = 2.7360$ $\mu = -3.2000$	$\sigma = 2.8220$ $\mu = 2.8460$	$\sigma = 3.4400$ $\mu = 9.2040$
IT2	$\sigma_u = 2.9290$ $\sigma_l = 2.6361$	$\sigma_u = 1.7180$ $\sigma_l = 1.5462$	$\sigma_u = 2.2510$ $\sigma_l = 2.0259$	$\sigma_u = 3.7520$ $\sigma_l = 3.3768$	$\sigma_u = 3.0390$ $\sigma_l = 2.7351$	$\sigma_u = 2.7360$ $\sigma_l = 2.4624$	$\sigma_u = 2.8220$ $\sigma_l = 2.5398$	$\sigma_u = 3.4400$ $\sigma_l = 3.0960$

Table 2.2 T1 and IT2 output performances

	e	$EPI\%$	R_1	R_2	$RPI_1\%$	$RPI_2\%$
$\Delta Y_{des} = 0.01$						
T1	0.1491	–	0.0114	0.0100	–	–
IT2	0.1476	1.0231 %	0.0166	0.0100	45.61	0
$\Delta Y_{des} = 0.03$						
T1	0.1491	–	0.0034	0.0033	–	–
IT2	0.1449	2.81 %	0.0034	0.0034	0	3.03
$\Delta Y_{des} = 0.05$						
T1	0.1499	–	0.0066	0.9400	–	–
IT2	0.1450	3.26 %	0.0081	0.5120	44.26	-45.53
$\Delta Y_{des} = 0.07$						
T1	0.1615	–	0.0021	2.8471	–	–
IT2	0.1534	5.04 %	0.0067	2.6029	> 100	-9.38
$\Delta Y_{des} = 0.1$						
T1	0.1763	–	1	3	–	–
IT2	0.1688	4.28 %	1	3	0	0

The error generated in approximating the function by T1 and IT2 are $6.0181e5$ and $1.8663e3$, respectively. This means that IT2 FLS obviously maps the function much better than its T1 counterpart.

For a given output deviation of $\Delta Y_{des} = 0.18$, the maximum output deviations for T1 are: $\delta x_1 = 2.2273e-4$ and $\delta x_2 = 2.8427e-4$, $\delta x_3 = 6.5783$, and for IT2 $\delta x_1 = 0.0227$, $\delta x_2 = 0.0527$, and $\delta x_3 = 6.1664$. Therefore, IT2 significantly approximates the function better. In terms of robustness, for the first and second inputs, the IT2 reveals enhanced improvement. For the third input, T1 is slightly more robust.

From the examples, it is concluded that decision on the robustness of IT2 versus T1 depends on the nonlinear function. However, IT2 has the potential to exhibit more robustness in some cases, which is of interest for the design of robust FLSs.

2.6 Conclusion

This chapter presented a rigorous mathematical methodology for the robustness analysis of IT2 TSK FLSs. We adopted the TSK model structure because it offers a closed form and, hence, enabling a mathematical formulation. The robustness is defined in terms of the maximum output tolerance of the system to a given output deviation. We formulate the robustness in terms of an optimization problem and present numerical examples to show how the developed contribution can be used in robustness analyses.

Table 2.3 Membership function parameters of T1 and IT2 TSK FLSs: second example

	First Input				Second Input			
	MF1	*MF2*	*MF3*	*MF4*	*MF1*	*MF2*	*MF3*	*MF4*
T1	$\sigma = 3.123$ $\mu = -9.551$	$\sigma = 2.796$ $\mu = -3.044$	$\sigma = 1.941$ $\mu = 2.934$	$\sigma = 2.455$ $\mu = 9.814$	$\sigma = 2.649$ $\mu = -10.01$	$\sigma = 1.677$ $\mu = -4.137$	$\sigma = 2.682$ $\mu = 3.008$	$\sigma = 2.786$ $\mu = 9.675$
IT2	$\sigma_u = 2.8107$ $\sigma_l = 3.123$	$\sigma_u = 2.5164$ $\sigma_l = 2.796$	$\sigma_u = 1.7469$ $\sigma = 1.941$	$\sigma_u = 2.2095$ $\sigma_l = 2.455$	$\sigma_u = 2.3841$ $\sigma_l = 2.649$	$\sigma_u = 1.5093$ $\sigma_l = 1.677$	$\sigma_u = 2.4138$ $\sigma_l = 2.682$	$\sigma_u = 2.5074$ $\sigma_l = 2.786$

Table 2.4 Membership function parameters of T1 and IT2 TSK FLSs: third example

	First Input			Second Input			Third Input		
	MF1	*MF2*	*MF3*	*MF1*	*MF2*	*MF3*	*MF1*	*MF2*	*MF3*
T1	$\sigma = 0.3910$	$\sigma = 0.2099$	$\sigma = 0.1402$	$\sigma = 0.2824$	$\sigma = 0.2246$	$\sigma = 0.2642$	$\sigma = 1.1980$	$\sigma = 1.1370$	$\sigma = 0.6363$
	$\mu = 1.0160$	$\mu = 1.9050$	$\mu = 2.9380$	$\mu = -0.9792$	$\mu = 0.1206$	$\mu = 1.0340$	$\mu = -4.7510$	$\mu = -1.8130$	$\mu = 1.2680$
IT2	$\sigma_u = 0.3910$	$\sigma_u = 0.2099$	$\sigma_u = 0.1402$	$\sigma_u = 0.2824$	$\sigma_u = 0.2246$	$\sigma_u = 0.2642$	$\sigma_u = 1.1980$	$\sigma_u = 1.1370$	$\sigma_u = 0.6363$
	$\sigma_l = 0.2737$	$\sigma_l = 0.1469$	$\sigma_l = 0.0981$	$\sigma_l = 0.1977$	$\sigma_l = 0.1572$	$\sigma_l = 0.1849$	$\sigma_l = 0.8386$	$\sigma_l = 0.7959$	$\sigma_l = 0.4454$

As higher-order fuzzy sets are finding more applications, it is important to investigate some of their main properties such as robustness especially for modeling and control. Therefore, the methodologies in this chapter will help designers to have a tool in determining the robustness of their system, as well as having a tool for the design on robust system that can tolerate certain input perturbations.

References

1. M. Biglarbegian, W.W. Melek, J.M. Mendel. Stability analysis of type-2 fuzzy systems. In *Proceedings of IEEE World Congress on Computational Intelligence (WCCI)*, pages 947–953, Hong Kong, June 2008
2. M. Biglarbegian, W.W. Melek, J.M. Mendel, On the stability of interval type-2 TSK fuzzy logic control systems. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics **40**(3), 798–818 (2010)
3. M. Biglarbegian, W.W. Melek, J.M. Mendel, On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling. Elsevier: Information Sciences **181**(7), 1325–1347 (2011)
4. W.W. Melek, A. Goldenberg, The development of a robust fuzzy inference mechanism. Elsevier: International Journal of Approximate Reasoning **39**(1), 29–47 (2005)
5. M. J. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice-Hall, Upper Saddle River, NJ, 2001

Chapter 3
Fuzzy Sets of Higher Type and Higher Order in Fuzzy Modeling

Witold Pedrycz

Abstract Fuzzy sets of higher order and higher type form one of the interesting conceptual and methodological pursuits in the development of the fundamentals of fuzzy sets. The objective of this study is to investigate a role of these constructs in the realm of fuzzy modeling. Rather than venturing into detailed algorithmic developments, we highlight key motivating factors behind the use of type-2 and order-2 in fuzzy models, especially fuzzy rule-based models. Linkages between type-n fuzzy sets and hierarchical fuzzy models are discussed. An overall setting of the study concerns granular computing (GC) along with its two fundamental ideas of the principle of justifiable granularity and an optimal allocation of information granularity.

3.1 Introductory Notes

Fuzzy models and fuzzy modeling have been around since the inception of fuzzy sets. There has been a plethora of different approaches supporting sound design practices, detailed algorithms, and applications. There has been a growing interest in the development of fuzzy models through a multiobjective optimization process with accuracy and interpretability being regarded as the two essential objectives. With the ever-growing challenges in system modeling, there is a visible need to exploit more advanced constructs of fuzzy sets and engage more advanced methodologies encountered in the realm of fuzzy sets.

Along with the generic constructs of fuzzy sets treated as mappings from a certain universe of discourse to the unit interval, we have witnessed a great deal of studies aimed at the generalizations of fuzzy sets. These developments were stimulated

W. Pedrycz (✉)
Department of Electrical Computer & Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada
e-mail: wpedrycz@ualberta.ca

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Department of Electrical and Computer Engineering Faculty of Engineering, King Abdulaziz University Jeddah, 21589, Saudi Arabia

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_3

by the common interest to develop more abstract and generalized notions of fuzzy sets, investigate their properties, and reveal relationships among them. Some of the generalizations are directly linked with applications; however, most of them exhibit some theoretical underpinnings not necessarily being fully justified.

In general, fuzzy sets are generalized in two main directions: fuzzy sets of higher order and fuzzy sets of higher type. Fuzzy sets of higher order generalize the original constructs of fuzzy sets by generalizing their universes of discourse. While in the generic constructs of fuzzy sets, the universe of discourse is composed of single elements, in order-2 fuzzy sets, their universes of discourse are composed of information granules, say sets, fuzzy sets, rough sets, etc. In fuzzy sets of type-2 or type-n, in general, instead of numeric membership grades, we admit granular membership grades such as intervals (interval-valued fuzzy sets), fuzzy sets defined in [0, 1] (type-2 fuzzy sets [8, 9]), probability density functions, say probabilistic fuzzy sets [5, 6], and probability-fuzzy set constructs [17]. Similarly, instead of the plain concept of quantification realized in the [0, 1] interval, more abstract situations are considered such as L-fuzzy sets. In all these cases, it becomes apparent that membership grades articulated in terms of information granules tend to capture and quantify a concept of membership, which spreads far beyond a simple numeric quantification.

In a nutshell, the origin of order-2 fuzzy sets is motivated by the complexity of the notion itself to be captured by a fuzzy set. Examples such as *comfortable* climate, *high*-quality car, and *strong* economy are convincing examples in this regard. *High*-quality car is a multifaceted concept: We consider *good* fuel economy, *high* reliability, *low* maintenance costs, etc. On the other hand, type-2 fuzzy sets address the point of difficulty of quantifying membership grades by single numeric entities. We can envision constructs where type-2 and order-2 aspects are brought together.

Both categories of these constructs presented above can be generalized further by forming fuzzy sets in a recursive way so we can talk about type-n and order-n fuzzy sets. While these generalizations could be appealing when looked at these more abstract fuzzy sets, one has to be cognizant that they come with (a) visible cost of processing (where in some cases the computational overhead could be quite substantial) and (b) estimation costs; the determination of membership functions of these fuzzy sets (again, one cannot ignore the algorithmic and experimental costs present here). Whether they are legitimate, this depends upon a problem at hand.

The ultimate objective of this study is to discuss a role of higher-order and higher-type fuzzy sets in fuzzy modeling. Having this objective in mind, we are concerned about ensuing algorithmic pursuits, which are supported by the use of these higher-type or -order fuzzy sets. In particular, we assess the current situation as to the position of these constructs in system modeling and identify to which extent they are considered in fuzzy modeling. We highlight a number of essential developments in system modeling which call for the use of type-2 and order-2 fuzzy sets. This research area is of growing interest considering a number of studies focused on the use of type-2 (mostly interval valued) fuzzy sets in control and system modeling [3, 4, 7, 9], decision making [16], and data analysis [15].

Quite often one encounters modeling scenarios where a number of *local* sources of knowledge (local models) become available and need to be used *en block* in further

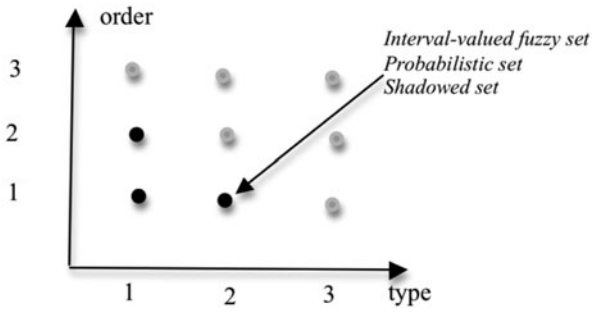


Fig. 3.1 A landscape of fuzzy sets of higher type and higher order; identified are main alternatives

processing to arrive at a holistic view of the system under discussion. This diversity of these sources has to be taken into account when constructing a piece of knowledge of *global* nature. For instance, considering that the local sources are descriptors of some decision-making processes realized by humans (and in this way exhibiting a quite local character confined to a single individual), we are interested in retaining and quantifying the diversity of the local sources of knowledge when arriving at the model formed at the higher level of abstraction. These considerations call for the use of fuzzy sets of higher type. In this suite of scenarios, fuzzy sets of higher type serve as a vehicle to quantify the diversity of the models present at the lower levels of hierarchy.

The final outcome should be reflective of the existing diversity offering an important overview of the classification pursuits completed so far and, if necessary, produce some guidelines for the enhancements of the local sources of knowledge (classifiers).

The landscape of fuzzy constructs of higher order and higher type is displayed in Fig. 3.1. Note that we may have constructs that generalize along the two directions, viz. we encounter fuzzy sets type-2 and order-2. Likewise, what falls under the rubric of type-2 fuzzy sets may exhibit a visible diversity bringing ideas of interval-valued fuzzy sets, shadowed sets, and probabilistic fuzzy sets.

The main objectives of the study can be succinctly outlined as follows. We investigate potential and algorithmic implications of fuzzy sets of higher order and higher type in fuzzy modeling. While the underlying concepts were intensively investigated in the realm of fuzzy sets and their formal structures are well investigated, it is not apparent how much and in which way they can impact current methodologies and practices of fuzzy modeling. Various topologies of fuzzy models along with related algorithms are critically assessed and contrasted.

The backbone of this study links with the concepts, methodologies, and algorithms of granular computing (GC); see the recent comprehensive treatise of the subject [14]. The reader may also refer to [1, 18–21]. In this regard, we elaborate on two principles; principle of justifiable granularity (which supports a construction of information granules, and fuzzy sets, in particular), and an allocation of information granularity where granularity is sought as an important modeling asset making the fuzzy model being more in rapport with reality.

The overall structure of the chapter is outlined as follows. We start with a brief review of fuzzy models cast in the context of fuzzy clustering by stressing a role of information granules highlighting the nature of modeling as predominantly based on and operating at the level of fuzzy sets rather than numeric entities. This feature is profoundly visible in fuzzy rule-based models (Sect. 3.2). In Sect. 3.3, we present a way of constructing information granules on a basis of existing experimental evidence by introducing a principle of justifiable granularity. The generality of the idea is that both the experimental evidence and the resulting information granules can be expressed in different ways not being necessarily confined to fuzzy sets. In the sequel, in Sect. 3.4, we discuss type-2 and order-2 fuzzy sets in system modeling by articulating a number of compelling reasons behind involving fuzzy sets of type-2. This is directly related to the treatment of information granularity as an important design asset in system modeling (Sects. 3.5 and 3.6). The role of type-2 and order-2 fuzzy sets in the model design is demonstrated in Sects. 3.7 and 3.8. Hierarchical modeling, consensus formation, and a role of type-n fuzzy sets is outlined in Sect. 3.8.

When it comes to fuzzy models, we concentrate on the use of fuzzy clustering—fuzzy c-means (FCM) [2, 12] and refer to its algorithmic settings to proceed into more detailed explanatory discussion.

3.2 Design of Fuzzy Models Through Fuzzy Clustering: A View at Type-2 Constructs

If we wish to point at the essence of fuzzy models, it is very likely we will be stressing the following characterization: fuzzy models are modeling constructs, which are built at the level of information granules—fuzzy sets—and operate by processing information granules realized at this level. Fuzzy rule-based models are convincing examples of fuzzy models in which these two features are highly visible. The dominant two categories of rule-based models (a) Mamdani and (b) Takagi-Sugeno are illustrated in a general way in Fig. 3.2.

Note that in the case of rules of Mamdani, the information granules (fuzzy sets) are present both in the input and in the output space. For the Takagi-Sugeno, we have fuzzy sets forming the condition parts of the rules and conclusions realized as local functions. Fuzzy sets A_i and B_j form a backbone (blueprint) of the fuzzy model.

Fuzzy clustering is used as a common vehicle to construct fuzzy sets out of a collection of experimental data. When FCM is considered, the membership functions of A_i (input space) read as follows:

$$A_i(\mathrm{x}) = \frac{1}{\sum_{\mathrm{j}=1}^{\mathrm{c}} \left(\frac{||\mathbf{x}-\mathbf{v}_\mathrm{i}||}{||\mathbf{x}-\mathbf{v}_\mathrm{j}||} \right)^{2/(\mathrm{m}-1)}}, \tag{3.1}$$

where v_i are the prototypes (centers) of the corresponding fuzzy set, $\|.\|$ stands for a distance function while m stands for the fuzzification coefficient [2].

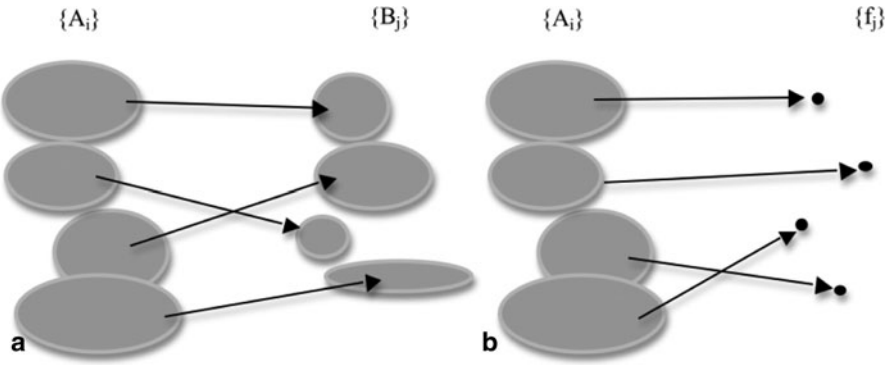
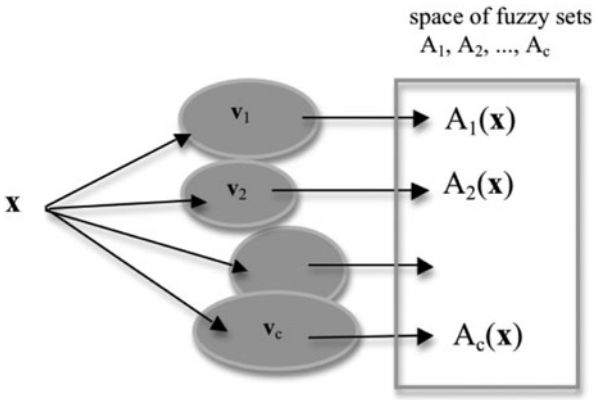


Fig. 3.2 A general view at fuzzy rule-based models. Mamdani (**a**) and Takagi-Sugeno (**b**) architectures and a web of link between functional elements in the input and output spaces

Fig. 3.3 Order-2 fuzzy set of activation of fuzzy sets in the input space



Once the fuzzy sets have been constructed, the next step of the design of the fuzzy model is to determine relationships among fuzzy sets A_i and B_j (Mamdani) or establish local models f_i.

In both categories of the models presented above, it is worth stressing that order-2 fuzzy sets are well visible. Any input **x** "activates" information granules and is translated into membership grades $A_1(\mathbf{x})\ A_2(\mathbf{x}) \ldots\ A_c(\mathbf{x})$. Alluding to the FCM (using which the input and output information granules are formed), the order-2 fuzzy set of activation levels is described in the vector form $[A_1(\mathrm{x})\ A_2(\mathrm{x}) \ldots\ A_c(\mathrm{x})]$. We underline this fact by using the following notation $[m_1/A_1\ m_2/A_2 \ldots\ m_c/A_c]$. It now becomes apparent that the universe of discourse of this fuzzy set is a family of information granules $\{A_1,\ A_2,\ \ldots,\ A_c\}$ and the corresponding degree of membership is $A_i(\mathbf{x})$ (Fig. 3.3).

In what follows, we look at the design of information granules by engaging the principle of justifiable granularity and link it with the construction of fuzzy sets of higher type.

3.3 Formation of Information Granules: The Principle of Justifiable Granularity

In what follows, we briefly highlight the concept of justifiable information granularity concentrating on the computational aspects in case of interval and fuzzy set-based formalisms of information granules.

The requirement of experimental evidence is quantified by counting the number of data falling within the bounds of information granule Ω. More generally, we may consider an increasing function of this cardinality, say $f_1(\text{card}\{x_k|x_k \in \Omega\})$, where f_1 is an increasing function of its argument. The simplest example is a function of the form $f_1(u) = u$. The specificity of the information granule Ω associated with its well-defined semantics (meaning) can be articulated in terms of the length of the interval. In case of $\Omega = [a, b]$, any continuous nonincreasing function f_2 of the length of this interval, say $f_2(m(\Omega))$ where $m(\Omega) = |b-a|$, can serve as a sound indicator of the specificity of the information granule. The shorter the interval (the higher the value of $f_2(m(\Omega))$), the better the satisfaction of the specificity requirement. It is evident that two requirements identified above are in conflict. The increase in the values of the criterion of experimental evidence (justifiable) comes at an expense of a deterioration of the specificity of the information granule (specific). As usual, we are interested in forming a sound compromise between these requirements.

The construction of the interval information granule comprises two steps. We start with a numeric representative of the set of data **D** around which the information granule Ω is created. A sound numeric representative of the data is its median, med(**D**). Recall that the median exhibits an appealing behavior by being a robust estimator of the sample and typically comes as one of the elements of **D**. Once the median has been determined, Ω (the interval $[a, b]$) is formed by specifying its lower and upper bounds, denoted here by a and b. As the determination of these bounds is realized independently, we discuss the optimization of the upper bound (b). The optimization of the lower bound (a) is carried out in an analogous fashion.

In the calculations of the cardinality of the information granule, we take into consideration the elements of **D** positioned to the right from the median, that is card $\{x_k \in \mathbf{D} | \text{med}(\mathbf{D}) \leq x_k \leq b\}$. As the requirements of experimental evidence (*justifiable granularity*) and specificity (*semantics*) are in conflict, we resort ourselves to a maximization of the composite index in which we form a product of the two expressions governing the requirements. This is done independently for the lower (a) and upper (b) bound of the interval.

In light of the conflicting requirements elaborated above, we form a multiplicative form of the optimization criterion:

$$V(b) = f_1(\text{card}\{x_k \mathbf{D} | \text{med}(\mathbf{D}) \leq x_k \leq b\}) * f_2(|\text{med}(\mathbf{D}) - b|) . \tag{3.2}$$

We obtain the optimal upper bound b_{opt}, by maximizing the value of $V(b)$, namely

$$V(b_{\text{opt}}) = \max_{b > \text{med}(\mathbf{D})} V(b). \tag{3.3}$$

Among numerous possible design alternatives regarding functions f_1 and f_2, we consider the following quite appealing alternatives $f_1(u) = u$ and $f_2(u) = \exp(-au)$ where α is a positive parameter delivering some flexibility when optimizing the information granule Ω. Under these assumptions, the optimization problem takes on the following form:

$$V(b) = \text{card}\{x_k \in \mathbf{D} | \text{med}(\mathbf{D}) \leq x_k \leq b\} * \exp(-\alpha |\text{med}(\mathbf{D}) - b|) . \tag{3.4}$$

The essential role of the parameter α is to calibrate an impact of the specificity criterion on the constructed information granule. Note that if $\alpha = 0$, then the value of the exponential function is 1; hence, the criterion of specificity of information granule is completely ruled out (ignored). In this case, $b = x_{\max}$ with $x_{\max}$ being the largest element in $\mathbf{D}$. Higher values of α stress the increasing importance of the specificity criterion.

The maximal value of α, say $\alpha_{\max}$, is determined by requesting that the optimal interval is the one for which $b_{\text{opt}} = x_1$, where x_1 is the data point closest to the median and larger than it. More specifically, we determine $\alpha_{\max}$ so that it is the smallest positive value of α for which the satisfaction of the following collection of inequalities holds,

$$\begin{aligned} &1 * \exp(-\alpha |\text{med}(\mathbf{D}) - x_1|) > 2 * \exp(-\alpha |\text{med}(\mathbf{D}) - x_2|) \\ &1 * \exp(-\alpha |\text{med}(\mathbf{D}) - x_1|) > 3 * \exp(-\alpha |\text{med}(\mathbf{D}) - x_3|) \\ &\qquad\qquad \ldots \\ &1 * \exp(-\alpha |\text{med}(\mathbf{D}) - x_1|) > p * \exp(-\alpha |\text{med}(\mathbf{D}) - x_p|), \end{aligned} \tag{3.5}$$

where the data $x_1, x_2, \ldots, x_p$ form a subset of $\mathbf{D}$ and are arranged as follows: $\text{med} < x_1 < x_2 < \ldots < x_p$.

Once the largest value of $\alpha_{\max}$ has been determined, the range of these values [0, $\alpha_{\max}$] can be normalized to [0, 1], and then the corresponding intervals $[a, b]$ indexed by α can be sought as a union of α-cuts of a certain fuzzy set of information granule A. In this way, the principle of justifiable granularity gives rise to a fuzzy set.

The concept of justifiable information granule has been presented in its simplest, illustrative version. In case of multivariable data, each variable is treated separately giving rise to the corresponding information granules and afterwards a Cartesian product of them is formed. The principle of justifiable granularity can be applied to experimental data being themselves information granules rather than numeric data. In these situations, some modifications of the coverage criterion are required.

An important design aspect of the discussed concept is concerned with the formation of the information granule in the presence of weighted experimental evidence. In other words, we have the data x_k associated with some weight coefficient f_k assuming values in the unit interval. The higher the value of the weight, the more substantial is a contribution of the data to the resulting information granule.

The underlying optimization process is arranged as follows. We again start with a numeric representative. The weighted median, med, is considered. It is constructed

by determining a value for which the following sum attains its minimum:

$$\text{Min}_{\xi} \sum_{k=1}^{M} f_k |z_k - \xi| = \sum_{k=1}^{M} f_k |z_k - \text{med}|. \tag{3.6}$$

Subsequently, the detailed calculations of (3.6) are slightly modified by incorporating the values of the weights f_k associated with the corresponding data.

Consider the objective function (3.4) where $\alpha = 0.0$ and rewrite the remaining part in a different way:

$$V(b) = \text{card}\{x_k \in \mathbf{D} | \text{med}(\mathbf{D}) \leq x_k \leq b\}/M, \tag{3.7}$$

where M is the number of data located on the right-hand side of the median.

The request that $V(b)$ exceeds a certain threshold brings a criterion of probabilistic character. Simply by requesting that $V(b)$ is not lower than 0.25, 0.50, 0.75, we are forming interval information granules implied by the corresponding quartiles of the experimental evidence.

The generality of the principle of justifiable granularity stems from the fact that the principle is associated with the idea, which inherently links with the fundamental notion of information granularity but does not specify directly which formalism is to be used. We showed how the construct works in case of intervals (sets) and fuzzy sets, but one can easily contemplate the use of other formal setup of GC. Likewise, we do not restrict ourselves to a specific construct. For instance, if $\{x_1, x_2, \ldots, x_M\}$ are membership values reported for a given element of the universe of discourse, the resulting interval of membership grades produces an interval-valued fuzzy set. If for the same case, we consider a fuzzy set being constructed, the result is a type-2 fuzzy set. In a similar way, we can anticipate a realization of fuzzy sets of higher type, say type-3 fuzzy sets, etc.

3.4 Type-2 and Order-2 Fuzzy Sets in Fuzzy Models

Order-2 fuzzy sets are elevated to order 2 and type-2 fuzzy sets. There are several compelling reasons behind the emergence and usage of type-2 fuzzy sets in fuzzy models:

Construction of Models with Granular Parameters The original numeric parameters of the fuzzy model are augmented and made granular to make a model being in rapport with the real world. Information granularity is regarded as an important design asset whose prudent usage improves the quality of the model. The construction of the model is supported by the principle of optimal allocation of information granularity being one of the underlying ideas of GC.

Use of Fuzzy Models in Presence of Input Granular Information In contrast to the commonly considered scenario where the input of the model is numeric, one

considers inputs that might be information granules. These granules could be a result of dealing with available data, which in virtue of their nature or the way in which the problem is formulated are nonnumeric.

Use of Granular Parameters and the Granular Inputs This is a combination of the two scenarios outlined above. One can strike a certain balance as to the allocation of information granularity in the sense that both parameters and inputs are made granular.

3.5 Allocation of Information Granularity in the Emergence of Granular Fuzzy Models

The problem of allocation of granularity across the parameters of the function f is regarded as a way of assigning a given level of information granularity $\varepsilon \in [0, 1]$ being viewed as a design asset. It transforms the vector of numeric parameters $\boldsymbol{a}$ into a vector whose coordinates are information granules $\mathrm{A} = [A_1\ A_2\ \ldots\ A_p]$ such that the level of admissible granularity ε is allocated to A_is in such a way that a balance of levels of information granularity, with $\varepsilon_1\ \varepsilon_2\ \ldots\ \varepsilon_p$ being the levels of information granularity, is satisfied, that is $\sum_{i=1}^{p} \varepsilon_i = p\varepsilon$, i.e., $\varepsilon = \sum_{i=1}^{p} \varepsilon_i / p$. Concisely, we can articulate this process of granularity allocation as follows:

$$\begin{array}{lll} f(\mathbf{x}, \mathbf{a}) \ \rightarrow \text{granularity allocation } (\varepsilon) \rightarrow & f(\mathbf{x}, \mathbf{A}) = f(\mathbf{x}, G(\mathbf{a})), & \\ \text{numeric mapping} & \text{granular mapping} & (3.8) \end{array}$$

that is, $A_i = \boldsymbol{G}(a_i)$ with $\boldsymbol{G}(.)$ denoting a transformation of the numeric parameter a_i to a certain granular counterpart A_i. Note that this expression is general and we are not confined to any particular formalism of information granules used here [13].

The mapping itself can be formed in various ways depending upon its original realization and a way in which information granules are represented, we come up with a plethora of modeling constructs with some representative examples listed in Table 3.1.

3.6 Information Granules: Formal Models and Characterization of Granularity

The information granules of the parameters of the mapping can be realized as intervals, fuzzy sets, or probability density functions (PDFs), to recall some commonly encountered alternatives. All of them are well documented in the literature. Hybrid constructs such as fuzzy probabilities, rough–fuzzy, or fuzzy–rough constructs are also quite visible. An information granule can be characterized by its specificity. In a descriptive way, one can think of specificity as a measure quantifying how detailed (specific) a piece of knowledge—information granule—is. If one regards information

Table 3.1 A collection of selected examples of granular mappings developed on a basis of well-known numeric modeling constructs

Model	Granular model	Examples of granular models
Linear regression	Granular linear regression	Fuzzy linear regression Rough linear regression Interval-valued linear regression Probabilistic linear regression
Rule-based model	Granular rule-based model	Fuzzy rule-based model Rough rule-based model Interval-valued rule-based model Probabilistic rule-based model
Fuzzy model	Granular fuzzy model	Fuzzy fuzzy model = fuzzy^2 model Rough fuzzy model Interval-valued fuzzy model Probabilistic fuzzy model
Neural network	Granular neural network	Fuzzy neural network Rough neural network Interval-valued neural network Probabilistic neural network
Polynomial	Granular polynomial	Fuzzy polynomial Rough polynomial Interval-valued polynomial Probabilistic polynomial

granule as a certain constraint expressed over a certain variable, the more specific this constraint is, the more useful the piece of knowledge (information granule) becomes. Granularity of information granule relates with the number of elements associated with the granule. The highest granularity characterizes an information granule composed of a single element, $\{x\}$. When the number of such elements (or some related characterization of the entities associated with the information granule) increases, the granularity decreases. In other words, the granularity is a nonincreasing or decreasing continuous function of this number of elements. Formally speaking, consider an information granule A and denote by Φ the function operating on A, $\Phi(A)$ and returning a nonnegative value characterizing the number of elements, dispersion or related measure of dispersion of A over the universe of discourse $\mathbf{X}$, $\Phi: A \rightarrow \mathbf{R}_+ \cup \{0\}$. The granularity of A, $g(A)$, is any continuous nonincreasing (decreasing) mapping defined over $\Phi(A)$, $g(\Phi(A))$. Let us consider some illustrative examples. For sets defined over a certain discrete space $\mathbf{X}$, a cardinality of A, card (A), can be viewed as the number of elements in $\mathbf{X}$ belonging to A. When $\mathbf{X} = \mathbf{R}$, where $A = [a, b]$, $\Phi(A)$ can be expressed as the length of this interval, namely $\Phi(A) = |b-a|$. In the sequel, the granularity can be taken as, e.g., $g(A) = \exp(-\Phi(A)) = \exp(-|b-a|)$. If A collapses to a single point, $\{a\}$, then $g(A) = 1$.

For fuzzy sets, as we are concerned with elements associated with information granule at some levels of belongingness (membership), the notion of cardinality is generalized in the form of so-called σ-count where one computes the overall sum of membership degrees. For the discrete space, we compute $\Phi(A) = \sum_{x \in \mathbf{X}} A(x)$ with $A(x)$ being a degree of membership of x in A. For $\mathbf{X} = \mathbf{R}$, one has $\Phi(A) = \int_{\mathbf{X}} A(x)\mathrm{d}x$ (assuming that the integral does exist).

For probabilistic information, granules described by probability density function, a measure of dispersion, say a standard deviation or variance, var(A), of the density function could be sought as a suitable representative of Φ, $\Phi(A) = \mathrm{var}(A)$. As before, $g(A)$ is a nonincreasing function of Φ(A).

Considering possible ways of allocating granularity and in order to arrive at its optimization throughout the mapping, we have to translate the allocation problem to a certain optimization task with a well-defined performance index and the ensuing optimization framework. In the evaluation, we use a collection of input–output data $\{(\mathbf{x}_1, \mathrm{target}_1), (\mathbf{x}_2, \mathrm{target}_2) \ldots (\mathbf{x}_N, \mathrm{target}_N)\}$. For $\mathbf{x}_k$, the granular mapping return Y_k, $Y_k = f(\mathbf{x}_k, \mathbf{A})$. There are two criteria of interest which are afterwards used to guide the optimization of the allocation of information granularity:

a. *Coverage criterion*. We count the number of cases when Y_k "covers" target_k. In other words, one can engage a certain inclusion measure, say, incl (target_k, Y_k) quantifying an extent to which target_k is included in Y_k. The computing details depend upon the nature of the information granule Y_k. If Y_k is an interval, then the measure returns 1 if target_k Y_k. In case Y_k is a fuzzy set, the inclusion measure returns $Y_k(\mathrm{target}_k)$, which is a membership degree of of target_k in Y_k. The overall coverage criterion is taken as a sum of degrees of inclusions for all data relative to all data, namely

$$Q = \frac{1}{N} \sum_{k=1}^{N} \mathrm{incl}(\mathrm{target}_k, Y_k). \tag{3.9}$$

b. *Specificity criterion.* Here our interest is in quantifying the specificity of the information granules $Y_1, Y_2, .., Y_N$. A simple alternative using the f-measure could be an average length of the intervals $V = 1/N \sum_{k=1}^{N} \left| y_k^+ y_k^- \right|$ in case of interval-valued formalism of information granules, $Y_k = [y_k^-, y_k^+]$ or a weighted length of fuzzy sets when this formalism is used.

Two optimization problems are formulated:

Maximization of the coverage criterion, maximize Q realized with respect to allocation of information granularity ε, that is

$$\mathrm{Max}_{\varepsilon_1, \varepsilon_2 \ldots \varepsilon_p} Q$$

subject to constraints

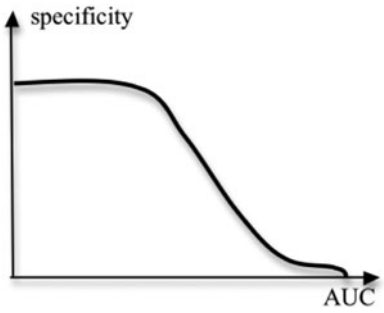


Fig. 3.4 Plot of a Pareto front displayed in AUC-length of interval coordinates

$\varepsilon_i > 0$ and the overall level of information granularity requirement

$$\sum_{i=1}^{p} \varepsilon_i = p\varepsilon. \tag{3.10}$$

Minimize average length of intervals V,

$$\text{Min}_{\varepsilon_1,\varepsilon_2\ldots\varepsilon_p} V$$

subject to constraints

$\varepsilon_i > 0$ and the overall level of information granularity requirement

$$\sum_{i=1}^{p} \varepsilon_i = p\varepsilon. \tag{3.11}$$

This optimization is about the maximization of specificity of the granular mapping (quantified by the specificity of the output of the mapping). Note that both Q and V depend upon the predetermined value of ε. Evidently, Q is a nondecreasing function of ε. If the maximization of Q is sought, the problem can be solved for each prespecified value of ε and an overall performance of the granular mapping can be quantified by aggregation over all levels of information granularity, namely

$$\text{AUC} = \int_0^1 Q(\varepsilon)\text{d}\varepsilon, \tag{3.12}$$

which represents an area under curve (AUC). The higher the AUC value, the higher the overall performance of the granular mapping.

The criteria of coverage and specificity of the granular outputs are in conflict. One can also consider a two-objective optimization problem and as a result develop a Pareto front of nondominated solutions (see Fig. 3.4).

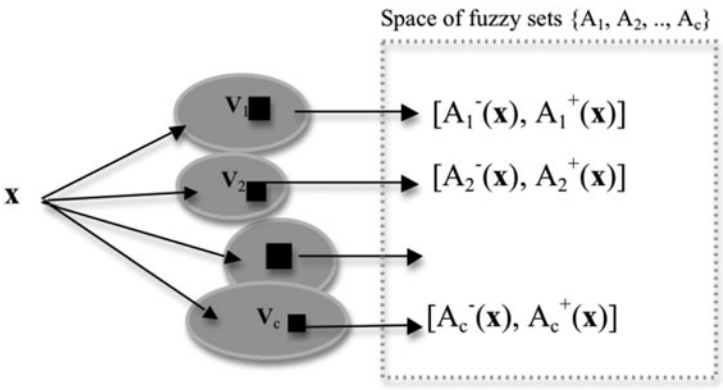


Fig. 3.5 From granular (interval-valued) prototypes to type-2 and order-2 fuzzy sets

3.7 Granular Fuzzy Rule-Based Models: Towards a Formation of Granular Prototypes in Fuzzy Clustering

Information granularity as a design asset gives rise to more realistic rule-based models, which could be referred to as granular fuzzy models. The granularity of information coming into the picture could be expressed in many different ways, however, the simplest version involve intervals. In a nutshell, we build granular prototypes, see Fig. 3.5, on a basis of the numeric prototypes (formed by the FCM), which in turn give rise to intervals of activation levels of each rule. This, in turn, produces interval (rather than numeric) outputs of the model "covering" the numeric data. The maximization of coverage is achieved by the allocation of information granularity as discussed in the previous section.

To come up with the details, let us look into calculations in the presence of granular prototypes. We compute the interval-valued membership grades in case the data x is not fully included in the granular prototype. Recall that for the j-th variable, the bounds of the granular prototype $\mathbf{V}_i$ form the interval $[v_{ij}^-, v_{ij}^+]$, refer to Fig. 3.6.

For the j-th coordinate of $\boldsymbol{x}$, x_j, we consider the following two situations:

1. $x_j \notin [v_{ij}^-, v_{ij}^+]$ The bounds of the distance are taken by considering the pessimistic and optimistic scenario, and computing the distances from the bounds of the interval, that is $\min((x_j^- v_{ij}^-)^2, (x_j^- v_{ij}^+)^2)$ and $\max((x_j^- v_{ij}^-)^2, (x_j^- v_{ij}^+)^2)$.
2. $x_j \in [v_{ij}^-, v_{ij}^+]$ It is intuitive to accept that the distance is equal to zero (as x_j is included in this interval).

The distance being computed on a basis of all variables, $\|\mathbf{x}-\mathbf{V}_i\|^2$, is determined coordinatewise by involving the two situations outlined above. The minimal distance obtained in this way is denoted by $d_{\min}(\mathbf{x}, \mathbf{V}_i)$ while the maximal one is denoted by

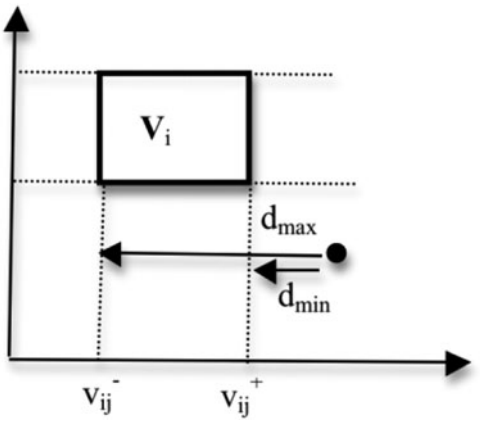


Fig. 3.6 Granular (interval) prototypes $\mathbf{V}_i$, numeric patterns, and computations of distances

$d_{\max}(\mathbf{x}, \mathbf{V}_i)$. More specifically, we have,

$$\begin{aligned} d_{\min}(\mathbf{x}, \mathbf{V}_i) &= \sum_{j \in K} \min\left(\left(x_j - v_{ij}^-\right)^2, \left(x_j - v_{ij}^+\right)^2\right) \\ d_{\max}(\mathbf{x}, \mathbf{V}_i) &= \sum_{k \in K} \max\left(\left(x_j - v_{ij}^-\right)^2, \left(x_j - v_{ij}^+\right)^2\right), \end{aligned} \tag{3.13}$$

where $K = \{j = 1, 2, \ldots, n | x_j \notin [v_{ij}^-, v_{ij}^+]\}$. Having the distances determined, we compute the following expressions:

$$\begin{aligned} w_1(\mathbf{x}) &= \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{\min}(\mathbf{x}, \mathbf{V}_i)}{d_{\min}(\mathbf{x}, \mathbf{V}_j)}\right)^{1/(m-1)}} \\ w_2(\mathbf{x}) &= \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{\max}(\mathbf{x}, \mathbf{V}_i)}{d_{\max}(\mathbf{x}, \mathbf{V}_j)}\right)^{1/(m-1)}}. \end{aligned} \tag{3.14}$$

Notice that these two formulas resemble the expression used to determine the membership grades in the FCM algorithm. In the same way as in the FCM, the weighted Euclidean distance is considered here, namely

$d_{\min}(\mathbf{x}, \mathbf{V}_i) = \sum_{j \in K} \min\left(\left(x_j - v_{ij}{}^-\right)^2/\sigma_j{}^2, \left(x_j - v_{ij}{}^+\right)^2/\sigma_J^2\right)$, and $d_{\max}(\mathbf{x}, \mathbf{V}_i) = \sum_{j \in K} \max\left(\left(x_j - v_{ij}{}^-\right)^2/s_j{}^2, \left(x_j - v_{ij}{}^+\right)^2/\sigma_j^2\right)$ with σ_j being the standard deviation of the j-th variable. These two are used to calculate the lower and upper bounds of the interval-valued membership functions (induced by the granular prototypes). Again, one has to proceed carefully with this construct. Let us start with a situation when $\mathbf{x}$ is not included in any of the granular prototypes. In this case, we compute

$$u_i^-(\mathbf{x}) = \min(w_1(\mathbf{x}), w_2(\mathbf{x}))$$

and

$$u_i^+(\mathbf{x}) = \max(w_1(\mathbf{x}), w_2(\mathbf{x})). \tag{3.15}$$

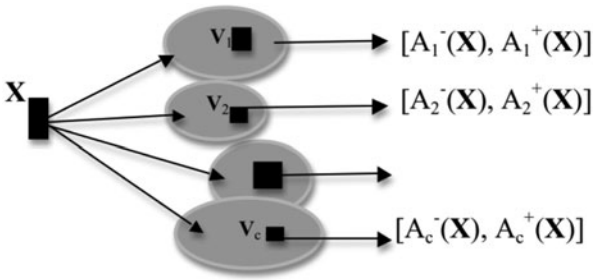


Fig. 3.7 Granular fuzzy model realized in presence of granular prototypes and granular (nonnumeric) input

So, in essence we arrive at the granular (interval-valued) membership function $U_i(\mathbf{x}) = [u_i^-(\mathbf{x}), u_i^+(\mathbf{x})]$. If $\mathbf{x}$ belongs to $\mathbf{V}_i$, then apparently $u_i^-(\mathbf{x}) = u_i^+(\mathbf{x}) = 1$ (and this comes as a convincing assignment as $\boldsymbol{x}$ is within the bounds of the granular prototype). Obviously, in this case, $u_j^-(\mathrm{x})$ as well as $u_j^+(\mathbf{x})$ for all indexes j different from i are equal to zero.

3.8 Distribution of Information Granularity Across the Model and Its Inputs: A Hybrid Design Scenario

An extension of the previous construct comes in the form illustrated in Fig. 3.7. Here the input $\mathbf{x}$ is also regarded to be nonnumeric, namely it is represented as a hyperbox (hypercube).

Here the main difference comes with the calculations of the distances. As both $\mathbf{X}$ and $\mathbf{V}_i$ are information granules (hypercubes)

$$\begin{aligned} d_{\min}(\mathbf{X}, \mathbf{V}_i) &= \sum_{j \in K} \min[(\mathrm{x}_j^- - \mathrm{v}_{ij}^-)^2, (\mathrm{x}_j^+ - \mathrm{v}_{ij}^-)^2, (\mathrm{x}_j^- - \mathrm{v}_{ij}^+)^2, (\mathrm{x}_j^+ - \mathrm{v}_{ij}^+)^2] \\ d_{\max}(\mathbf{X}, \mathbf{V}_i) &= \sum_{j \in K} \max[(\mathrm{x}_j^- - \mathrm{v}_{ij}^-)^2, (\mathrm{x}_j^+ - \mathrm{v}_{ij}^-)^2, (\mathrm{x}_j^- - \mathrm{v}_{ij}^+)^2, (\mathrm{x}_j^+ - \mathrm{v}_{ij}^+)^2], \end{aligned} \tag{3.16}$$

where $\mathbf{K} = \{j = 1, 2, \ldots, n | [\mathrm{x}_j^-, \mathrm{x}_j^+] \neq [\mathrm{v}_{ij}^-, \mathrm{v}_{ij}^+]\}$.

The motivation behind these formulas is made clear from Fig. 3.8.

3.9 Models of Consensus Formation with Higher-Type Information Granules: An Emergence of Type-N Fuzzy Sets

As discussed so far, the two-level hierarchy leads to granular (interval) fuzzy sets. This means that, as a result of reconciliation, information granules are made more abstract. More specifically, the type of the fuzzy sets has been elevated. Once working with type-1 fuzzy sets (with numeric membership grades), the result formed at the

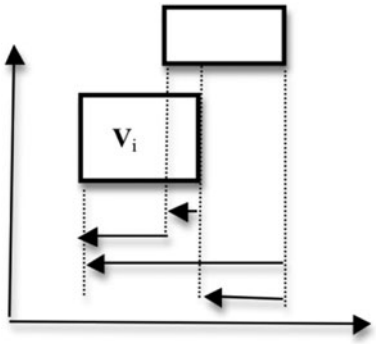


Fig. 3.8 Computing of distances in case of granular (interval) prototypes $\mathbf{V}_i$ and granular input $\mathbf{X}$

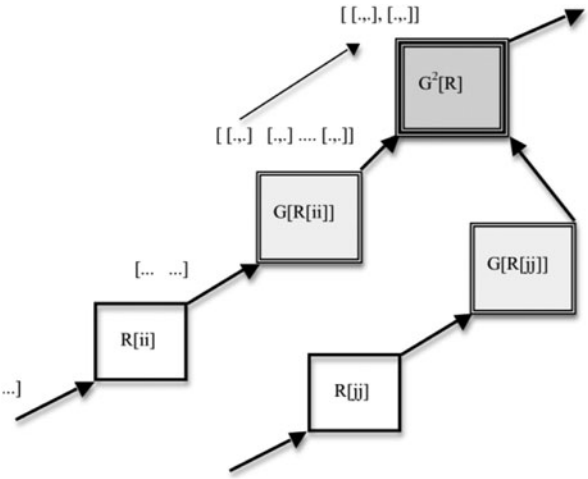


Fig. 3.9 Emergence of information granules (fuzzy sets) of higher type—symbolically shown are increased types of fuzzy sets when moving up along the hierarchy. Numeric values of the membership grades are denoted by a series of *dots*

higher level of hierarchy is expressed as a type-2 (interval) fuzzy set. The emergence of higher types of fuzzy sets is a result of processing information granules of a lower type. This effect arises in many different ways (this issue will be discussed in a while). Let us consider that at the lower level of hierarchy, we obtain interval-valued fuzzy sets (viz. type-2 fuzzy sets) and the results are subject to reconciliation. The result of this aggregation becomes a type-3 fuzzy set. For illustration, see Fig. 3.9.

Computationally, we note that the level of the fuzzy set increases; we show this as the following sequence: [. . . .] [. . . ..] [[.,.] [.,.]. . . . [.,.]] . . . . The aggregation of the bounds of the intervals of membership grades, say $a_i[1]$, $i = 1, 2, \ldots,$ c gives rise to a certain interval formed again in the unit interval (more precisely, the subinterval of the [0, 1] with the lower and upper bound expressed as $\min_{ii} a_i[ii]$ and $\max_{ii} a_i[ii]$). In a graphic form, the resulting construct is displayed in Fig. 3.10.

More descriptively, in contrast to type-2 (interval) fuzzy set where membership bounds are numbers, here the bounds are intervals (see shadowed regions shown in Fig. 3.10). This construct relates in a convincing way to shadowed sets [11] or rough

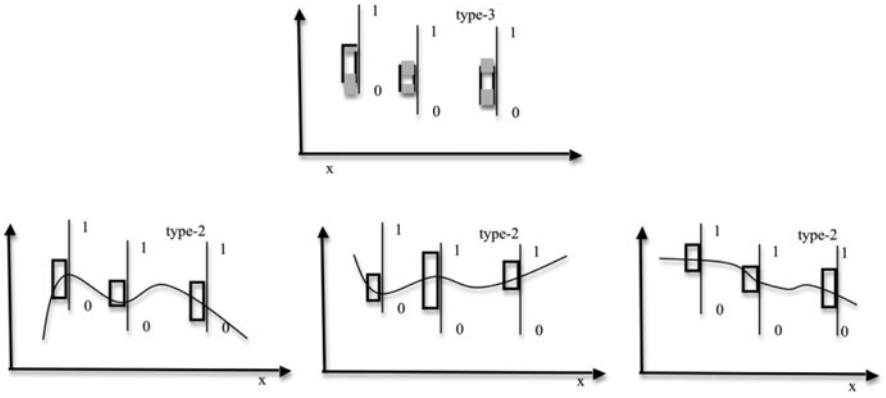


Fig. 3.10 Formation of type-3 fuzzy with the use of type-2 (interval) fuzzy sets

sets [10]—in both constructs, we deal with boundary regions of the information granules and their description.

When it comes to the emergence of type-2 fuzzy sets at the lower level of the hierarchy, there are at least two compelling scenarios supporting their emergence, see also Fig. 3.7:

a. Formation of local models of higher relevance captured by the granular (interval) fuzzy relations. Instead of numeric membership functions of R, following the scheme of allocation of information granularity [18], the entries or R are interval-valued. As a result of the composition of type-1 fuzzy set and type-2 fuzzy relation, the results are type-2 fuzzy sets.
b. Inputs are granular, viz. type-2 fuzzy sets. In this case, even though the fuzzy relation is of type-1, the result is type-2 fuzzy set. This is an inherent effect of the propagation of information granularity—the result inherits the highest level of the type of information granules present in the chain of processing. This scenarios emerge when dealing, e.g., in models of time series to be classified—the models come with granular features.
c. A combination of the situations outlined above—the models come with the type-2 fuzzy sets and the inputs are also type-2 information granules.

3.10 Conclusions

Fuzzy sets of higher order and higher type, especially type-2 and order-2, provide new opportunities to system modeling. In this study, we identified main directions and pointed out the main motivating factors behind the emergence of these modeling generalizations.

GC plays a pivotal role in this setting. As visualized in Fig. 3.11, the fundamental concepts such as the principle of justifiable granularity and the idea of allocation

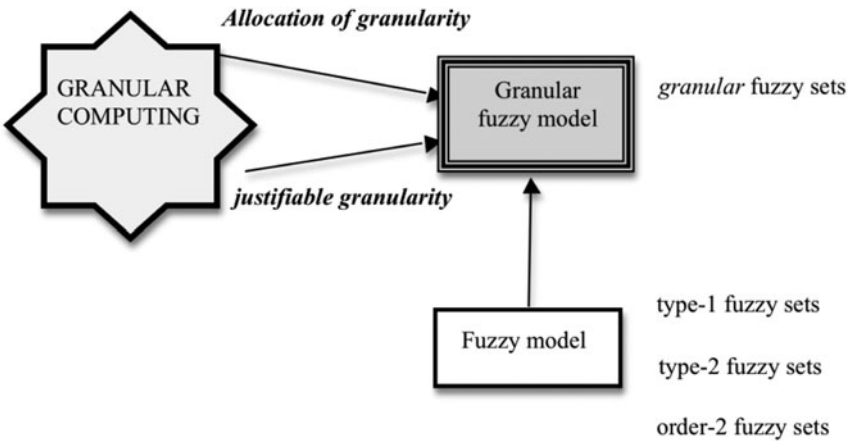


Fig. 3.11 Granular fuzzy models and fuzzy sets of higher order and higher type: a general perspective

of information granularity are essential in supporting the design of granular fuzzy models (in which type-2 and order-2 as well as higher order/type fuzzy sets provide interesting design features). Hierarchical fuzzy models naturally engage fuzzy sets of higher type; however, one has to be cognizant that high-order/high-type fuzzy sets invoke significant computing overhead and call for more elaborate acquisition (estimation) procedures required to construct information granules.

References

1. A. Bargiela, W. Pedrycz, *Granular Computing: An Introduction* (Kluwer Academic Publishers, London, 2003)
2. J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms* (Plenum Press, New York, 1981)
3. E. Czogala, W. Pedrycz, On the concept of fuzzy probabilistic controllers. Fuzzy Sets Syst. **10**, 109–121 (1983)
4. E. Czogala, S. Gottwald, W. Pedrycz, Logical connectives of probabilistic sets. Fuzzy Sets Syst. **10**, 299–308 (1983)
5. K. Hirota, Concepts of probabilistic sets. Fuzzy Sets Syst. **5**(1), 31–46 (1981)
6. K. Hirota, W. Pedrycz, Characterization of fuzzy clustering algorithms in terms of entropy of probabilistic sets. Pattern Recognit. Lett. **2**(4), 213–216 (1984)
7. E. Hisdal, The IF THEN ELSE statement and interval-valued fuzzy sets of higher type. Int. J. Man–Machine Stud. **15**, 385–455 (1981)
8. N.M. Karnik, J. M. Mendel, Q. Liang, Type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst. **7**, 643–658 (1999)
9. J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions* (Prentice Hall, Upper Saddle River, 2001)
10. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data* (Kluwer Academic Publishers, Dordrecht, 1991)

11. W. Pedrycz, Shadowed sets: Representing and processing fuzzy sets, IEEE Trans. Syst. Man, Cybern. Part B **28**, 103–109 (1998)
12. W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules* (John Wiley, Hoboken, 2005)
13. W. Pedrycz, Allocation of information granularity in optimization and decision-making models: towards building the foundations of granular computing. Eur. J. Oper. Res. (2012, to appear)
14. W. Pedrycz, *Granular Computing: Analysis and Design of Intelligent Systems* (CRC Press/Francis Taylor, Boca Raton, 2013)
15. W. Pedrycz, P. Rai, Collaborative clustering with the use of Fuzzy C-Means and its quantification. Fuzzy Sets Syst. **15**, 2399–2427 (2008)
16. W. Pedrycz, M.L. Song, Analytic Hierarchy Process (AHP) in group decision making and its optimization with an allocation of information granularity, IEEE Trans. Fuzzy Syst. **19**, 527–539 (2011)
17. L.A. Zadeh, Probability measures of fuzzy events. J. Math. Anal. Appl. **23**, 421–427 (1968)
18. L.A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. **90**, 111–117 (1997)
19. L.A. Zadeh, From computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions. IEEE Trans. Circuits Syst. **45**, 105–119 (1999)
20. L.A. Zadeh, Toward a perception-based theory of probabilistic reasoning with imprecise probabilities. J. Stat. Plan. Inference **105**, 233–264 (2002)
21. L.A. Zadeh, A note on Z-numbers. Inf. Sci. **181**, 2923–2932 (2011)

Chapter 4
Recent Advances in Fuzzy System Modeling

I. Burhan Türkşen

Abstract Decision making under uncertainty is an interdisciplinary research field. In this chapter, we attempt to create a framework for the human decision-making processes with Type 1 and Full Type 2 Fuzzy Logic methodology. For this purpose, we first present a brief review of the essentials of (1) Zadeh's rule base model, (2) Takagi and Sugeno's model which is partly a rule base and partly a regression function, and (3) Türkşen's model of fuzzy regression functions where a fuzzy regression function corresponds to each fuzzy rule in a fuzzy rule base model. Next, we review the well-known fuzzy C-means (FCM) algorithm which lets one to extract Type 1 membership values from a given data set for the development of Type 1 fuzzy system models as a foundation for the development of Full Type 2 fuzzy system models. For this purpose, we provide an algorithm which lets one to generate Full Type 2 membership value distributions for a development of second-order fuzzy system models with our proposed second-order data analysis. If required, one can generate Full Type 3, . . . , Full Type n fuzzy system models with an iterative execution of our proposed algorithm. We present our applied results graphically for TD_Stockprice data with respect to two validity indices, namely (1) Çelikyılmaz–Türkşen and (2) Bezdek indices.

4.1 Introduction

After Zadeh's introduction of fuzzy logic and fuzzy sets, a vast volume of literature appeared about fuzzy logic and fuzzy system modeling (FSM). There are at least two advantages of FSM that attracts researchers: (1) its power of linguistic explanation with resulting ease of understanding and (2) its tolerance to imprecise data which provides flexibility and stability for prediction.

Briefly, in the fuzzy theory, every element belongs to a concept class, say A, to a partial degree, i.e., $\mu_A : X \rightarrow [0, 1]$, $\mu_A(x) = a \in [0, 1]$, $x \in X$, where $\mu_A(x)$ is the

I. B. Türkşen (✉)
Department of Industrial Engineering, TOBB-ETU, Ankara, Turkey
e-mail: bturksen@etu.edu.tr; ibturksen@gmail.com

University of Toronto, Toronto, ON, Canada

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_4

membership assignment of an element 'x' to a concept class A in a proposition. All concepts in fuzzy theory are assumed to be definable to be true to a matter of degree.

Treating the membership assignment as perfectly known or calculated may be seen as contradiction. There are several types of fuzzy theory called as Type 1, Full Type 2, . . . , Full Type n fuzziness. Type 1 fuzzy sets are obtained in both subjective and objective manners and are well established in this area. Full Type 2 and higher types of fuzziness, on the other hand, are still a very active area of recent research. Here, we present an objective algorithmic Full Type 2 fuzzy systems investigations.

4.2 Fuzzy System Models

Let us first review historically significant fuzzy system model developments in order to identify their unique structures and to point out how they differ from each other. Then let us show the details of our Full Type 2 fuzzy system developments with an iterative algorithm that can generate Full Type 3, . . . , Full Type n fuzzy system models.

4.2.1 Type 1 Fuzzy Rule Base Models

The most commonly applied fuzzy system models are fuzzy rule bases. Here, we only deal with multi-input single output (MISO) systems. Generally, fuzzy system models represent relationships between the input and output variables which are expressed as represented with fuzzy sets. The general fuzzy rule base structure is a collection of IF–THEN rules that utilize linguistic labels, which is known as Zadeh Fuzzy Rule Base (Z-FRB). Z-FRB, can be written as follows:

$$R : \underset{i=1}{\overset{c*}{ALSO}} \text{ (IF } antecedent_i \text{ THEN } consequent_i),$$

where $c*$ is the number of rules in a rule base either given by experts or determined by a fuzzy clustering algorithm such as fuzzy C-means (FCM) [1] or improved fuzzy clustering (IFC) [2]. The fuzzy rule base structures determined by various alternatives mainly differ in the representation of the consequents. If the consequent is represented with fuzzy sets, then the fuzzy rule base is known as Z-FRB [3] a modified version of which is proposed by Sugeno and Yasukawa, SY-FRB [4], whereas if the consequents are represented with linear equations of input variables, then the rule base structure is the Takagi–Sugeno Fuzzy Rule Base (TS-FRB) [5] structure. These are the main models among others which we do not review in this chapter. In particular, Z-FRB can be formalized as follows, where the multidimensional antecedent fuzzy subset of the i^{th} rule is A_i:

$$R : \underset{i=1}{\overset{c*}{ALSO}} \text{ (IF } x \in X \text{ isr } A_i \text{ THEN } y \in Y \text{ isr } B_i).$$

In general, let *nv* be the number of selected input variables in the system. Then, the multidimensional antecedent, x, can be defined as $x = (x_1, x_2, \ldots, x_{nv})$, where x_j is the j^{th} input variable of the antecedent and the domain of x in X can be defined as $X = X_1 \times X_1 \times \ldots \times X_{nv}$, $X_j \subseteq \Re$. This multidimensional antecedent fuzzy subset determination eliminates the search for the appropriate t-norm for the combination of antecedent fuzzy subsets with "AND."

Thus, variations of Z-FRB are SY-FRB and TS-FRB structures:
(SY-FRB):

$$R : \underset{i=1}{\overset{c*}{ALSO}} \text{ (IF } x \in X \text{ isr } A_i \text{ THEN } y \in Y \text{ isr } B_i)$$

(TS-FRB):

$$R : \underset{i=1}{\overset{c*}{ALSO}} \text{ (IF } antecedent_i \text{ THEN } y_i = a_i x^T + b_i),$$

where $antecedent_i = x \in X \; isr \; A_i$, and $a_i = (a_i, 1, \ldots, a_i, NV)$ is the regression coefficient vector associated with the i^{th} rule together with b_i which is the scalar associated with the i^{th} rule. For these special cases of Z-FRB, again each degree of firing, d_i, associated with the i^{th} rule is determined directly from the corresponding i^{th} multidimensional antecedent fuzzy subset A_i and applied to the consequent fuzzy subset for the SY-FRB or to the classical ordinary regression for the case of TS-FRB.

4.2.2 Fuzzy Regression Functions

There are a number of variations of fuzzy regression functions proposed by Turksen [6]. We discuss only one alternative in this chapter, namely, fuzzy regression functions which we have proposed with least square estimation (LSE).

Fuzzy Regression Functions with LSE (FF-LSE) In ordinary LSE method, the dependent variable, y, is assumed to be a linear function of input variables, x, plus an error component:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_{nv} x_{nv} + \varepsilon,$$

where y is the dependent output, x_j's are the explanatory input variables, for $j = 1, \ldots, nv$, nv is the number of selected inputs, and ε is the independent error term which is typically assumed to be normally distributed. The goal of the least-squares method is to obtain estimates of the unknown parameters, β_j's, $j = 0, 1, \ldots, nv$, which indicate how a change in one of the independent variables affects the dependent variable and are calculated as

$$\beta = (X^T X)^{-1} X^T y.$$

The proposed generalization of LSE as FF-LSE (fuzzy functions with LSE, more appropriately known as fuzzy regression functions with LSE) requires that a fuzzy

clustering algorithm, such as FCM, or IFC be available to determine the interactive (joint) membership values of input–output variables in each of the fuzzy clusters that can be identified for a given training data set. Let (X_k, Y_k), $k = 1, \ldots, nd$,be the set of observations in a training data set, such that $X_k = (x_{jk} | j = 1, \ldots, nv)$. First, one determines the optimal (m^*, c^*) pair for a particular performance measure, i.e., cluster validity indices such as Bezdek [1] and Celikyılmaz and Türkşen [2] with an iterative search and an application of FCM or IFC algorithm, where m is the level of fuzziness (in our experiments, we usually take $m = 1.4, \ldots, 2.5$; Ozkan and Türkşen [8]) and c is the number of clusters (in our experiments, we usually take $c = 2, \ldots, 10$). The well-known FCM [1] algorithm can be stated as follows:

$$\begin{aligned} \min \boldsymbol{J}(\boldsymbol{U}, \boldsymbol{V}) &= \sum_{k=1}^{nd} \sum_{i=1}^{c} (u_{ik})^m (\|x_k - v_i\|)_A \\ s.t. \quad & o \leq u_{ik} \leq 1, \forall i, k \\ & \sum_{i=1}^{c} u_{ik} = 1, \forall k \\ & o \leq \sum_{k=1}^{nd} u_{ik} \leq nd, \forall i \end{aligned} ,$$

where J is objective function to be minimized and $\|.\|_A$ is a norm that specifies a distance-based similarity between the data vector x_k and a fuzzy cluster center v_i. In particular, $A = I$ *is the Euclidian norm and* $A = C^{-1}$ *is the Mahalonobis norm, etc.*

Once the optimal pair (m^*, c^*) is determined with the application of FCM algorithm, and a cluster validity index, one next identifies the cluster centers for $m = m^*$ and $c = 1,\ldots,c^*$ as

$$\underset{m*}{v_{X|Y,j}} = ({x_{1,j}}^c, {x_{2,j}}^c, \cdots, {x_{nv,j}}^c, {y_j}^c).$$

From this, we identify the cluster centers of the input space again for $m = m^*$ and $c = 1, \ldots, c^*$ as

$$\underset{m*}{v_{X,j}} = ({x_{1,j}}^c, {x_{2,j}}^c, \cdots, {x_{nv,j}}^c).$$

Next, one computes the normalized membership values of each vector of observations in the training data set with the use of the cluster center values determined in the previous step. There are generally two steps in these calculations.

First, one determines the (local) optimum membership values u_{ik}'s and then determines μ_{ik}'s that are above an α-cut in order to eliminate harmonics generated by FCM as:

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{\|x_k - v_{X,i}\|}{\|x_k - v_{X,j}\|} \right)^{\frac{2}{m-1}} \right)^{-1}, \mu_{ik} \geq \alpha,$$

where μ_{ik} denotes the membership value of the k^{th} vector, $k = 1,\ldots, nd$, in the i^{th} rule, $i = 1,\ldots,c^*$, and x_k denotes the k^{th} vector and for all the input variables $j = 1,\ldots, nv$ in the input space.

Next, we normalize them as

$$\gamma_{ij}(x_j) = \frac{\mu_{ij}(x_j)}{\sum_{i'-1}^{c} \mu_{i'j}(x_j)},$$

where γ_{ij} is the normalized membership value of $x_j, j = 1,\ldots, nv,$ in the i^{th} rule, $i = 1,\ldots,c^*,$ which in turn will indicate the membership value that will constitute a new input variable in our proposed scheme of function identification for the representation of the i^{th} cluster.

Let $\Gamma_i = (\gamma_{ij} | i = 1,\ldots,c*; j = 1,\ldots,nv)$ be the membership values of X in the i^{th} cluster, i.e., rule. Next, we determine a new augmented input matrix X for each of the clusters which could take on several forms depending on which *transformations* of membership values we want to or need to include in our system structure identification for our intended system study. Let

$$X_i{}' = [1, \Gamma_i, X],$$
$$X_i{}'' = [1, \Gamma_i{}^2, X],$$
$$X_i{}''' = \left[1, \Gamma_i{}^2, \Gamma_i^m, \exp(\Gamma_i), X\right],$$

etc., where $X_i{}'$, $X_i{}''$, $X_i{}'''$ are the new input matrices to be used in least-squares estimation of a new system structure identification where

$$\Gamma_i = (\gamma_{ij} | i = 1,\ldots,c*; j = 1,\ldots,nv).$$

The choice depends on whether we want to or need to include just the membership values or some of their transformations as new input variables in order to obtain the best representation of a system behavior. In particular, this is done in order to get a higher value of R2 to show that a better model is obtained for an application. A new augmented input matrix, say $\grave{X_i}$, would look as shown below for the special case of $X = X_j$, i.e., the matrix X is just a vector of a single variable, $X_j = (x_{jk} \mid \mathrm{k} = 1,\ldots, nd)$, for the j^{th} variable:

$$\grave{X_i}_j = [1, \Gamma_i, X_{ij}] = \begin{bmatrix} 1 & \gamma_{i1} & x_{ij_1} \\ \vdots & \vdots & \vdots \\ 1 & \gamma_{ind} & x_{ijnd} \end{bmatrix}.$$

Thus, the fuzzy regression function, $Y_i = \beta_{i0} + \beta_{i1}\Gamma_i + \beta_{i2}X_{ij}$, that represents the i^{th} rule corresponding to the i^{th} interactive (joint) cluster in space (Y_i, Γ_i, X_j), $\beta_i{}^* = \left(\grave{X_{ij}}^T \grave{X_{ij}}\right)^{-1} \left(\grave{X_{ij}}^T Y_i\right)$,

$$X_{ij}{}' = [1, \Gamma_i, X_{ij}],$$

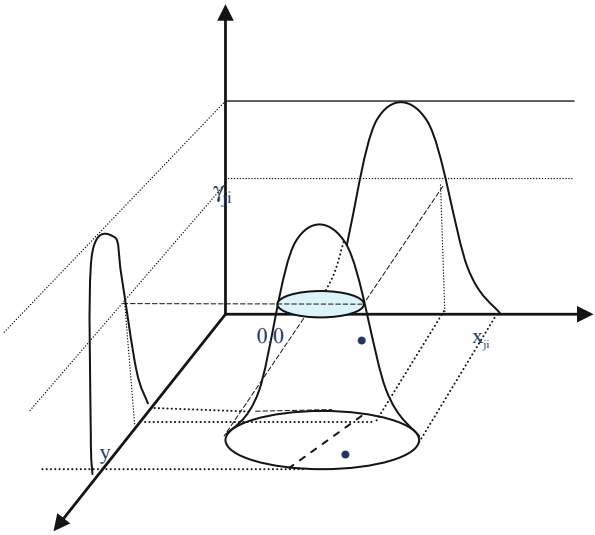


Fig. 4.1 A fuzzy cluster in $U \times X \times Y$ space

such that $\beta_i{}^* = (\beta_{i0}{}^*, \beta_{i1}{}^*, \beta_{i2}{}^*)$ and the estimate of Y_i would be obtained as $Y^*{}_i = \beta^*_{i0} + \beta^*_{i1}\Gamma_i + \beta^*_{i2}X_{ij}$.

Within the proposed framework, the general form of the shape of a cluster can be conceptually captured by a second order (cone) in the space of $U \times X \times Y$ which can be illustrated with a prototype shown in Fig. 4.1.

One usually determines Type 1 membership values with an application of FCM algorithm shown below:

ALGORITHM 1. Fuzzy *c*-means Clustering Algorithm (FCM) Given data vectors, $X = \{\boldsymbol{x}_1,..,\boldsymbol{x}_n\}$, number of clusters, $\boldsymbol{c}$, degree of fuzziness, m, and a termination constant, ε (maximum iteration number in this case). Initialize the partition matrix, U, randomly.

Step 1: Find initial cluster centers using Eq. 1 shown below using membership values of initial partition matrix as inputs.

Step 2: Start iteration $t = 1\ldots$ max-iteration value:

Step 2.1. Calculate membership values of each input data object k in cluster i, $\mu_{ik}^{(t)}$, using the membership value calculation equation in via Eq. 1 below, where x_k are input data objects as vectors and $\upsilon_i^{(t-1)}$are cluster centers from the $(t-1)^{\text{th}}$ iteration.

Step 2.2. Calculate cluster center of each cluster i at iteration t, $\upsilon_i^{(t)}$, using the cluster center function in Eq. 2 shown below, where the inputs are the input data matrix, $\boldsymbol{x}_k$, and the membership values of iteration t, $\mu_{ik}^{(t)}$.

Step 2.3. Stop if termination condition satisfied, e.g., $\left|\upsilon_i^{(t)} - \upsilon_i^{(t-1)}\right| \leq \varepsilon$. Otherwise, go to step 1.

Where Eq. 1 stated in the algorithm above is

$$\mu_{ik}^{(t)} = \left[\sum_{j=1}^{c} \left(\frac{d\left(x_k, \upsilon_i^{(t-1)}\right)}{d\left(x_k, \upsilon_j^{(t-1)}\right)} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{4.1}$$

and Eq. 2 is

$$\upsilon_i^{(t)} = \left(\sum_{k=1}^{n} \left(\mu_{ik}^{(t)}\right)^m x_k \right) / \sum_{k=1}^{n} \left(\mu_{ik}^{(t)}\right)^m, \forall i = 1, ..., c. \tag{4.2}$$

4.3 Generation of Full Type 2 Membership Values

For this purpose, we propose and hence introduce a new algorithm in order to generate Full Type 2 membership value distribution from the results obtained with an application of FCM which produce a Type 1 membership value distribution for our studies of Full Type 2 investigations.

4.3.1 Full Type 2 Fuzziness i.e., Membership of Membership

Here, we want to show how one determines the second-order degree of fuzziness in order to develop Full Type 2 fuzzy system models.

It should be noted that depending on where $x \in X$, there may be more than one second-order membership value distribution.

4.3.1.1 Full Type 2 Fuzzy Set Extraction Algorithm

We propose the following Full Type 2 fuzzy set extraction algorithm from a given data set called FT2FCM (Türkşen [7]):

Full Type 2 Fuzzy Clustering Algorithm

$$Min\ J\grave{}\left(U^{'}(U), W\right) =$$

$$= \sum_{k=1}^{nd} \sum_{i=1}^{c^{'}} \sum_{l=0}^{1} \left(\mu_{\mu_i(x_k)}(z)\right) \left(||\mu_{\mu_i(x_k)}(z_l) - \overline{\mu}_{(x_k)}(z_l)||A\right),$$

$$k = 1, \ldots, nd; i = 1, \ldots, c^{'}$$

$$st.\ 0 \leq \mu_{\mu_i(x_k)}(z) \leq 1$$

$$0 \leq \mu_i(x_k) \leq 1$$

$$0 \leq \sum_{k=1}^{nd} \mu_i\,(x_k) \leq nd$$

$$\mu_i\,(x_k) \in [0,1];\;\; \mu_{\mu_i(x_k)}\,(z) \in [0,1];\;\; l \in [0,1]\,,$$

where $J^{`}$ is the objective function to be minimized for a given $x_k \in X$; $||.||_A$ is a norm, i.e., Euclidian or Mahalanobis, that specifies a distance measure based on membership values for a given $x_k \in X$ and its second-order fuzzy cluster center $\overline{\mu}_i(x_k)$.

Next, one computes the normalized membership values of these Full Type 2 membership values for each vector of membership values obtained in an initial application of the original FCM or IFC algorithm in the first stage.

There are generally two steps in these calculations:

We first determine (local) optimum membership of membership values $\mu_{\mu_i}(x_k)$'s and then apply an α-cut in order to eliminate the second-order harmonics generated by an application of FT2FCM as

$$\begin{matrix}\mu_{\mu_i}\,(x_k) \\ x_k \in X\end{matrix} = \left[\left(\sum_{i=1}^{c'} \frac{||\mu_{\mu_i}\,(x_k) - \overline{\mu}_i\,(x_k)||}{||\mu_{\mu_i}\,(x_k) - \mu_j\,(x_k)||}\right)^{\frac{2}{m-1}}\right]^{-1}$$

$$\gamma'_{\mu_{\mu_i}(x_k)} = \frac{\mu_{\mu_i}\,(x_k)\,|_{x_k \in X}}{\sum_{i=1}^{c'} \mu_{\mu_i}\,(x_k)},\;\; \mu_{\mu_i}\,(x_k) \geq \alpha, \gamma'_{\mu_{\mu_i}(x_k)} \geq \alpha,$$

where $\gamma'_{\mu_{\mu_i}(x_k)}$ denotes the membership values of the membership values of the k^{th} vector $k = 1,..,nd$ in the i^{th} rule, or i^{th} fuzzy regression function (Türkşen 2012), and $x_k \in X$ denotes the k^{th} vector and for all the input variables, $k = 1,..,\, nd$ in the input space.

Recall that we are able to obtain the membership value distribution as

$$X'_{ij} = [1, \Gamma_{\text{i}}, X_{ij}] = \begin{bmatrix} 1\;\gamma_{i1} & x_{i1} \\ \vdots\;\;\vdots & \vdots \\ 1\gamma_{ind} & x_{ind} \end{bmatrix}$$

$$\Gamma_i = (\gamma_{ik} | i = 1, \ldots, c*; k = 1, \ldots, nd)$$

$$\Gamma_i = (\gamma_{ij} | i = 1, \ldots, c^*; j = 1, \ldots, nd) = \begin{bmatrix} \gamma_{11} & \gamma_{21} & \cdots & \gamma_{c^*1} \\ \vdots & \vdots & & \vdots \\ \gamma_{1nd} & \gamma_{2nd} & \cdots & \gamma_{c^*nd} \end{bmatrix}.$$

We process each Γ_i via our Full Type 2 clustering algorithm given above, called FT2FCM, to determine Full Type 2 distribution for each cluster *i,* $\Gamma_i = (\gamma_{ij} | i = 1, \ldots, c^*; j = 1, \ldots, nd)$. Thus, we apply to each Γ_i ALGORITHM 2 given below to generate Full Type 2 membership values, i.e., membership of membership.

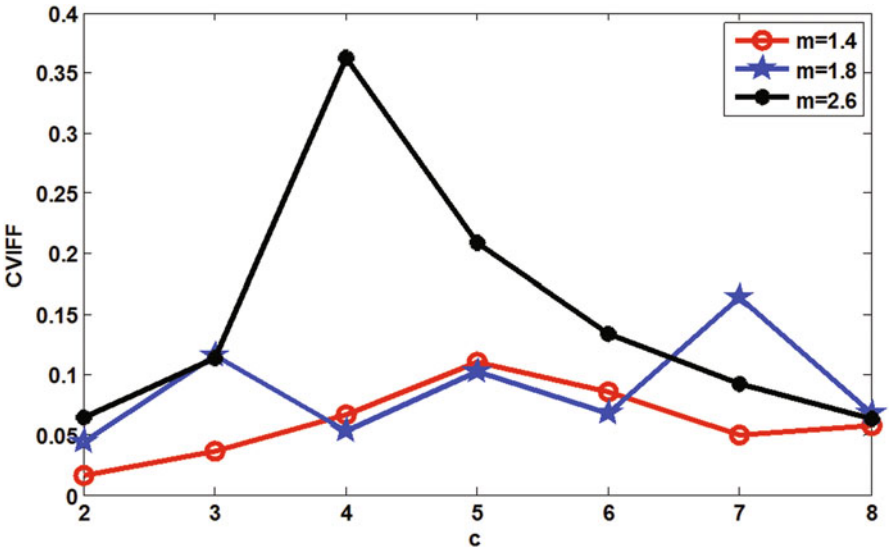


Fig. 4.2 Çelikyılmaz–Türkşen's validity index results for TD_Stockprice data

ALGORITHM 2 Given data vectors, $\gamma_i = (\gamma_{ij}|i = 1, \ldots, c^*; j = 1, \ldots, nd)$ the number of clusters, $\boldsymbol{c}$`, degree of fuzziness, m, and termination constant, ε (maximum iteration number in this case). Initialize a partition matrix, Γ, randomly:

Step 1: Find initial cluster centers using Eq. 4.3 shown below using membership of membership values of initial partition matrix as inputs.
Step 2: Start iteration $t = 1 \ldots$ max-iteration value;
Step 2.1. Calculate Type 2 membership values of a given Γ vector of each input data object k in cluster i, $\mu_{\mu_i}(x_k)$, using each Γ vector of the membership values where x_k are input data objects as vectors and $\overline{\mu}_i(x_k)$ are Type 2 cluster centers from the $(t-1)^{\text{th}}$ iteration.
Step 2.2. Calculate Type 2 cluster center w_{ik} of each cluster l at iteration t, the t-th $\overline{\mu}_i(x_k)$ he cluster center function of Type 2 membership values in Eq. 4.4 shown below, where the inputs are the input data matrix, , , and the membership of the membership values of iteration t, $\mu_{\mu_i}(x_{k,t})$
Step 2.3. Stop if termination condition is satisfied, e.g., $|\overline{\mu}_i(x_k)\,@t - \overline{\mu}_i(x_k)\,@t-1| \leq \varepsilon$. Otherwise, go to step 1.

4.4 Experimental Results

We present here our experimental results for TD_Stockprice data set that is available for all researchers on the Internet (Figs. 4.2, 4.3, 4.4, and 4.5).

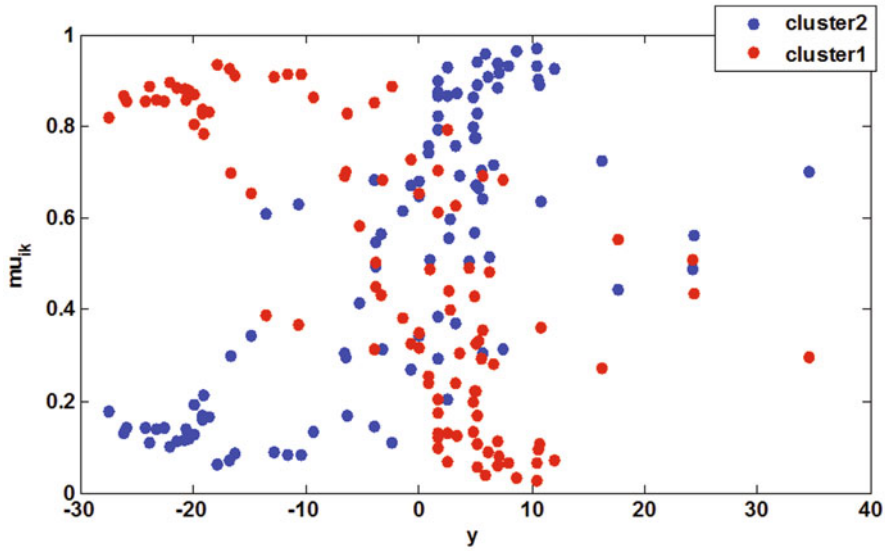


Fig. 4.3 Fuzzy classification of TD_Stockprice data: ($c^* = 2$, $m^* = 1.8$)

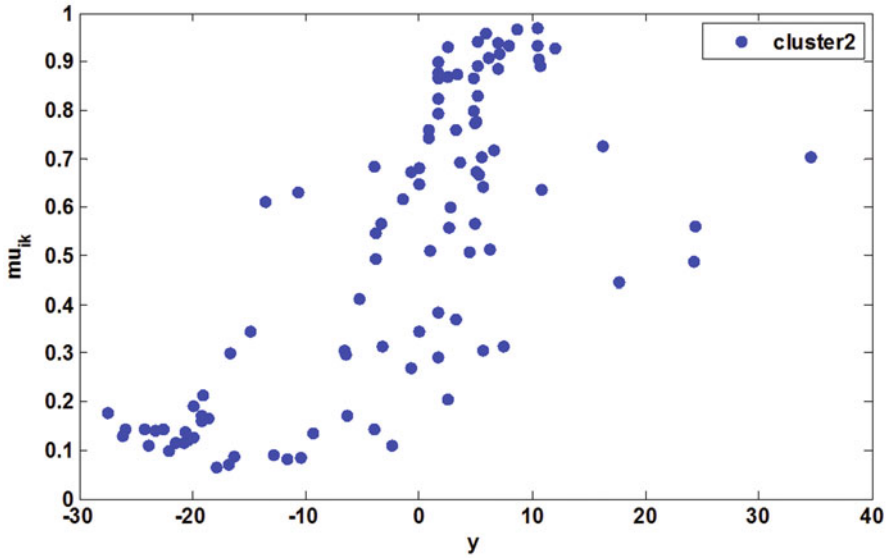


Fig. 4.4 Cluster-2 view for TD_Stockprice data

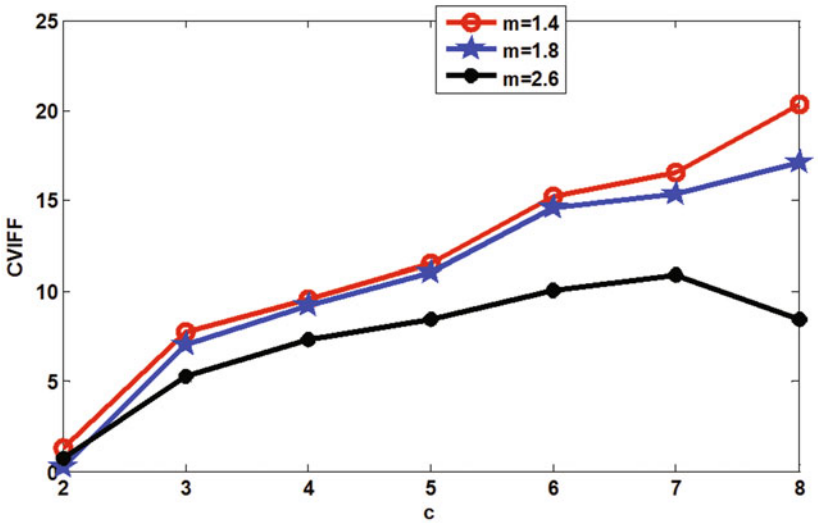


Fig. 4.5 Çelikyılmaz–Türkşen's validity index for μ_{ik} data

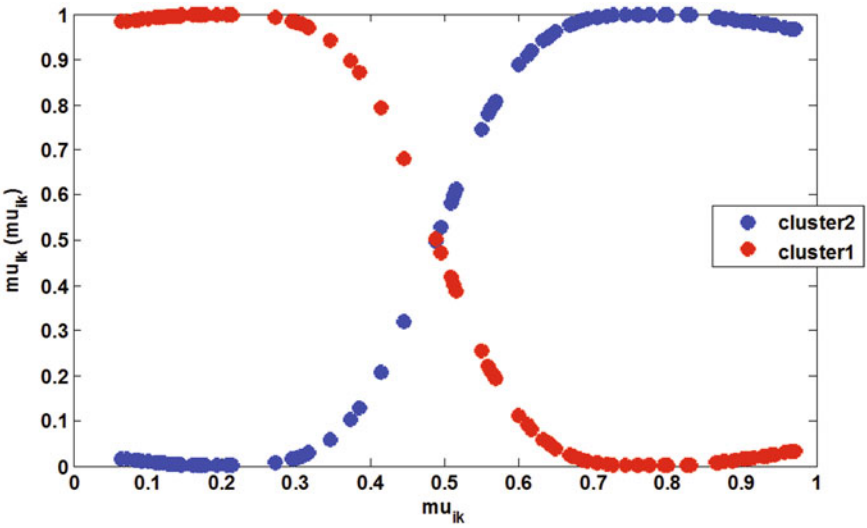


Fig. 4.6 $\mu_{lk}(\mu_{ik})$ for cluster 1 and 2 are shown above

Cluster-2 Results of TD_Stockprice Data (c* = 2, m* = 1.8) According to the Çelikyılmaz–Türkşen index, the suitable number of cluster should be chosen as $c' = 2$ (μ_{ik} data are the membership values of the first study's cluster-2), where $c' = 2$, $m' = 1.8$ (Figs. 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, and 4.13).

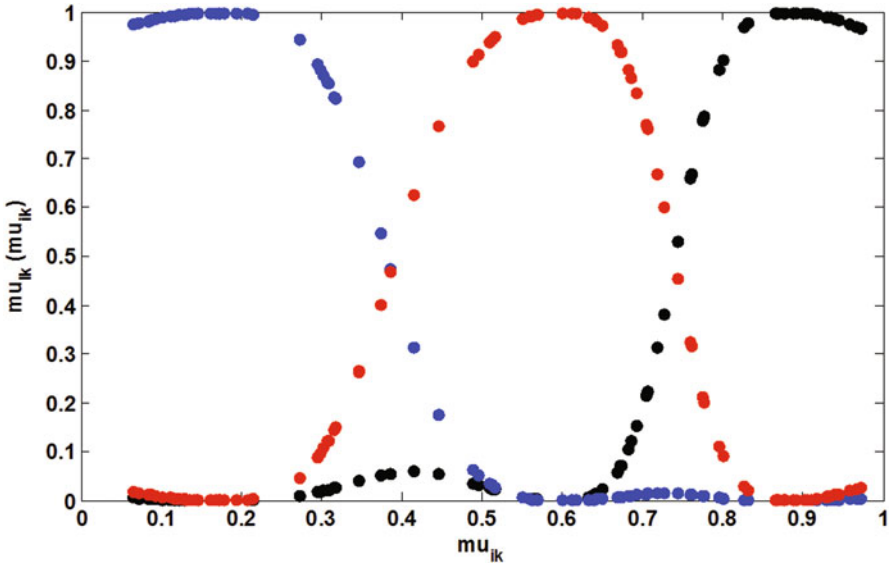


Fig. 4.7 A possible three-cluster view

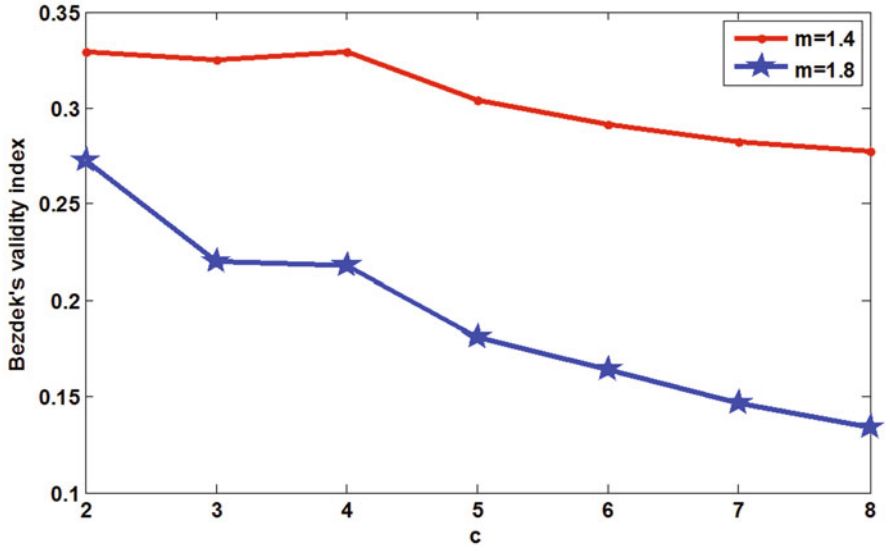


Fig. 4.8 TD_Stockprice data set: According to Bezdek's validity index results (shown as follows), the suitable number of cluster was chosen as $c^* = 3$

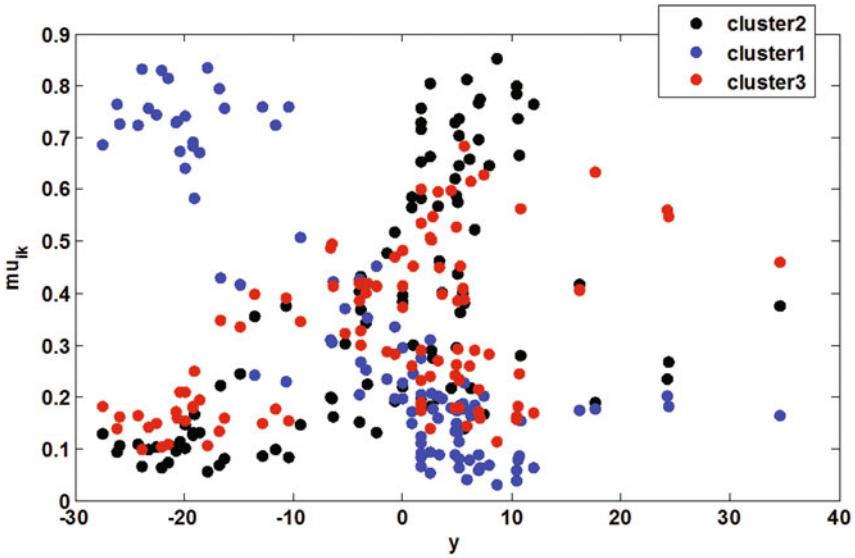


Fig. 4.9 Fuzzy classification of TD_Stockprice data: ($c^* = 3$, $m^* = 2.0$)

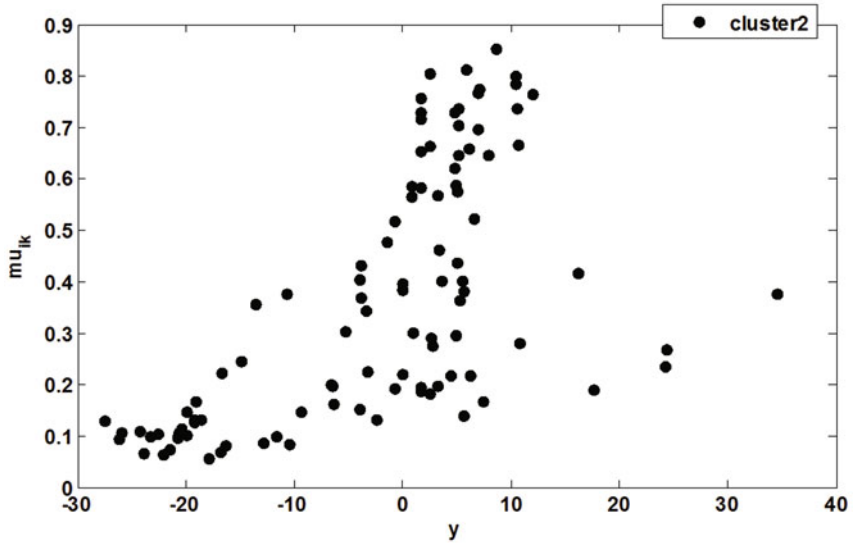


Fig. 4.10 Cluster-2 view for TD_Stockprice data

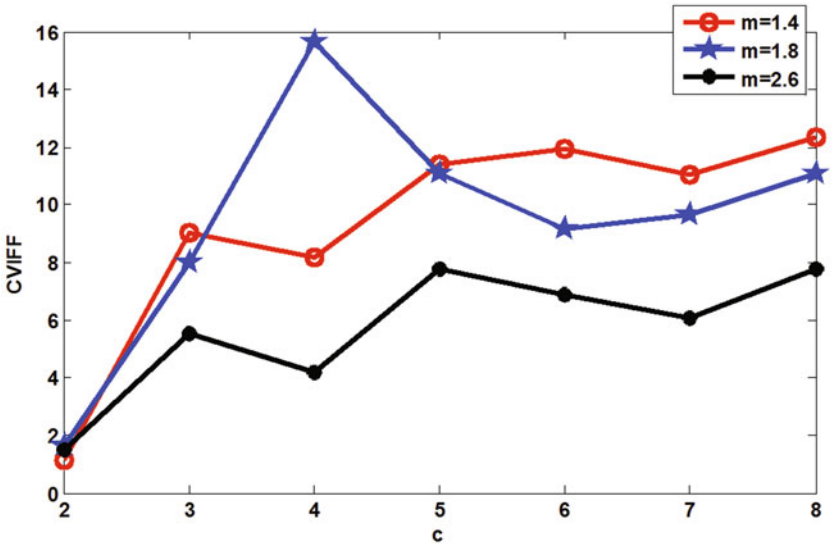


Fig. 4.11 Çelikyılmaz–Türkşen's validity index for μ_{ik} data

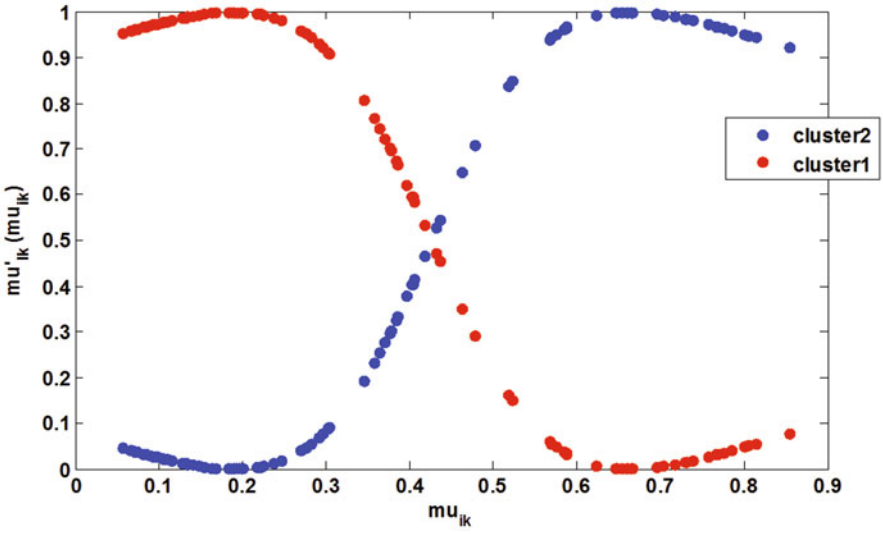


Fig. 4.12 According to Çelikyılmaz–Türkşen index, the suitable number of cluster should be chosen as $c' = 2$ (the μ_{ik} data are the membership values of the first study's cluster-2), where $c' = 2, m' = 2.0$

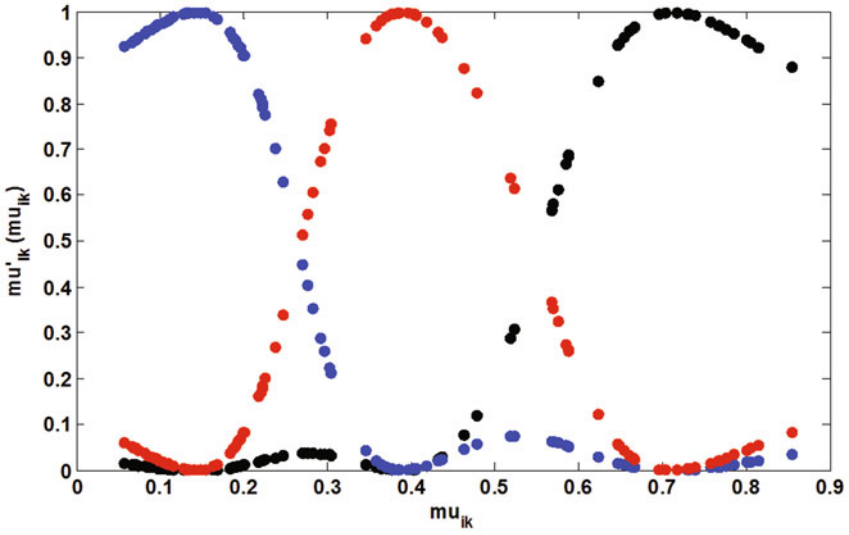


Fig. 4.13 A possible three-cluster view

Cluster-2 Results of TD_Stockprice Data (c* = 3, m* = 2.0) for Membership of Membership

4.5 Conclusions

In this chapter, we first briefly provide a framework for the review of the essentials of fuzzy system models, namely: (1) Zadeh's rule base model, (2) Takagi and Sugeno's model which is partly a rule base and partly a regression function, and (3) Türkşen's model of fuzzy regression functions where a fuzzy regression function corresponds to each fuzzy rule in a fuzzy rule base model. In this framework, a fuzzy rule base is replaced with "fuzzy regression function" models. Next, we review the well-known FCM algorithm which lets one to extract Type 1 membership values from a given data set for the development of "Type 1" fuzzy system models as a foundation for the development of "Full Type 2" fuzzy system models. For this purpose, we provide a new algorithm which lets one to generate Full Type 2 membership value distributions for a development of second-order fuzzy system models with our proposed second-order data analysis. If required, one can generate Full Type 3, . . . , Full Type *n* fuzzy system models with an iterative execution of our algorithm. Finally, we present our results graphically for TD_Stockprice data with respect to two validity indices, namely: (1) Çelikyılmaz–Türkşen and (2) Bezdek indices. Based on our development, we expect in the future new results would be obtained in "Full Type 3, . . . , Full Type *n*" fuzzy system model analyses.

References

1. J. C. Bezdek, Cluster validity with fuzzy sets. J. Cybernet. **3**, 58–72 (1974)
2. A. Çelikyılmaz, I.B. Türkşen, Validation criteria for enhanced fuzzy clustering. Pattern Recogn. Lett. **29**(2), 97–10 (2008)
3. L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-I, II, III. Inf. Sci. **8**, 199–251 (1975)
4. M. Sugeno, T. Yasukawa, A fuzzy logic based approach to qualitative modelling. IEEE Trans. Fuzzy Syst. **1**(1), 7–31 (1993)
5. T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **SMC-15**(1), 116–132 (1985)
6. I. B. Türkşen, Fuzzy functions with LSE. Int. J. Appl. Soft Comput. **8**(3), 78–88 (2008)
7. İ. B. Türkşen, From type 1 to full type N fuzzy system models, J. of Mult.-Valued Logic & Soft Computing, **22**, 543–560 (2014)
8. I. Ozkan, I.B. Turksen, Upper and lower values for the level of fuzziness in FCM. Inf. Sci. **177**(23), 5143–5152 (2007)

Chapter 5
On the Use of Participatory Genetic Fuzzy System Approach to Develop Fuzzy Models

Yi Ling Liu and Fernando Gomide

Abstract Genetic fuzzy systems constitute an essential approach to build fuzzy models. There is an increasing interest to develop fuzzy models in the realm of complex, large-scale, multiobjective, and high-dimensional systems. Nowadays, fast and scalable evolutionary algorithms to handle complex fuzzy modeling is a major need. Procedures to learn rule bases and tune their parameters are being shaped with the purpose to produce parsimonious and accurate models. Approaches to develop distinct types of fuzzy models such as type one and higher types, or higher order fuzzy trees, fuzzy relations, fuzzy cognitive maps, and neural fuzzy networks are rare. This chapter introduces participatory evolutionary learning as a framework for data driven fuzzy modeling. The participatory evolutionary learning approach is a population-based paradigm in which the population itself defines the fitness of the individuals as evolution progress. The approach uses compatibility between population individuals during selection and recombination. A mechanism for information exchange in recombination based on selective transfer is introduced. Combination of participatory learning and selective transfer offers a new class of genetic fuzzy systems. Despite the focus on participatory learning and the selective transfer to build first order fuzzy rule-based models, the use of the genetic fuzzy systems to develop higher order fuzzy rule-based models is also discussed. An electric system maintenance data modeling problem is explored to illustrate the usefulness of the participatory genetic fuzzy systems approach in practice. The performance of participatory evolutionary learning is evaluated using the mean squared modeling error and number of fuzzy rules to measure model accuracy and complexity, respectively. The results suggest that the participatory evolutionary learning develops high quality models and is highly competitive with current state of the art approaches.

Y. L. Liu (✉) · F. Gomide
School of Electrical and Computer Engineering,
University of Campinas, Sao Paulo, Brazil
e-mail: yilingli@dca.fee.unicamp.br

F. Gomide
e-mail: gomide@dca.fee.unicamp.br

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_5

5.1 Introduction

During the last decade, genetic fuzzy systems (GFS) have been shown to be instrumental as a tool to develop fuzzy classifiers, process models, control systems, robotics, and economy applications. From the soft computing point of view, genetic fuzzy systems combine two major archetypes, evolutionary computation and fuzzy systems theory. Basically, a GFS is a fuzzy system together with a learning mechanism based on evolutionary computation. GFS use genetic algorithms (GA) and genetic programming mostly, but evolutionary strategies, particle swarms, and their hybridizations with classic modeling and optimization methods have also been fruitful [14].

Nowadays, there is a renewed interest in methodologies and approaches to develop fuzzy models for industrial processes, economic systems, and large data mining and knowledge acquisition systems. In general, these systems are multifarious and require sophisticated modeling procedures. Intuition and expert knowledge are not enough, and modeling should benefit from data-driven approaches to complete knowledge-based design and development. Multiobjective evolutionary algorithms have been constructed to build linguistic fuzzy rule-based systems from data. Many successful GFS use embedded genetic database learning to learn simple and accurate models fast [3].

Genetic algorithm is one of the most important components of evolutionary computation. GA is a heuristic search that uses mechanisms inspired by the principles of natural evolution. The purpose of a GA is to serve as a metaphor to solve complex problems that can be put within a system optimization framework. Commonly, in GA an objective function is given externally, and a selection process together with recombination and mutation operators work simultaneously through generations to search for the best solution. From this point of view, the objective function can be seen as the representation of some skill required by nature, the better an agent at this skill, the more it is preferred. The objective function is the main way to account for fitness in genetic algorithms. Fitness measures the reproductive suitability of the individuals that form a population.

In the real world survival of the fittest saga there appears to be an additional process going on. In particular, the objective function, besides being determined by some external requirement, is always strongly effected by the population itself. The population is always involved in determining the desired properties of the reproductive fitness of each individual that assembles itself. This observation suggests the possibility to build a GA in which the current population plays a role in determining or modifying reproduction and evolution. This can be viewed as a kind of fitness learning. The real world survival of the fittest suggests evolutionary algorithms in which fitness shapes the reproductive suitability of individuals through an evaluation that combines external objective function with the influence of the population itself. One further observation is that, in natural environments, often the effect of the population participation in the crafting of the fitness function is to make fitness more like itself [33].

During the 1990s, a recombination mechanism called selective transfer was suggested as an alternative for crossover of genetic algorithms. Selective transfer was inspired in modular technologies. Here an agent may adopt a part or all of another agent's action if the first agent estimates that this increases his/her own fitness [31].

The purpose of this chapter is to introduce the participatory evolutionary learning algorithm (PELA), a learning procedure in which compatibility between individuals, in addition to objective function, influences the evolution of a population. Together, objective function values and compatibility degrees shape fitness of the individuals. PELA uses selective transfer to retain the similarity information of good individuals in offspring. It forms the kernel of a genetic fuzzy system framework aiming at complex, large-scale, and high-dimensional fuzzy system modeling. It produces parsimonious and accurate models within reasonable computer processing times. The chapter discusses representation issues for first and higher order fuzzy rule-based models.

After this introduction, the next section overviews the area of genetic fuzzy systems. Section 5.3 details the participatory evolutionary learning algorithm emphasizing selective transfer and its properties. Computational experiments and comparisons with a state of the art approach [3] is done in Sect. 5.4 using the actual data to develop a linguistic fuzzy model of electrical maintenance systems. Section 5.5 concludes the chapter summarizing its contribution and issues for further development.

5.2 Genetic Fuzzy Systems

This section overviews the GFS and the applications whose development has benefited from it. As a development framework, GFS spans a wide class of fuzzy models, as shown in Fig. 5.1. Despite the rich body of fuzzy models available today, for the clarity of exposition we will focus on the most visible instance of GFS, that is, genetic fuzzy rule-based systems (GFRBS). Mutatis mutandi, the concepts and main ideas behind GFS are similar for the different instances of fuzzy models and generalizations of the notion of fuzzy sets. More specifically, the focus is on ordinary (equivalently, type 1, first order) fuzzy set-based model development.

Given a set X, recall that an ordinary fuzzy set A is defined by a function of the form

$$A : X \to [0, 1] \tag{5.1}$$

Fuzzy sets of this type are the most prevalent in the literature and applications of fuzzy set theory. In this chapter, ordinary fuzzy sets are called just fuzzy sets, for short. More general types of fuzzy sets have been proposed in the literature. For instance, interval-valued fuzzy sets are sets defined as functions of the form

$$A : X \to I([0, 1]) \tag{5.2}$$

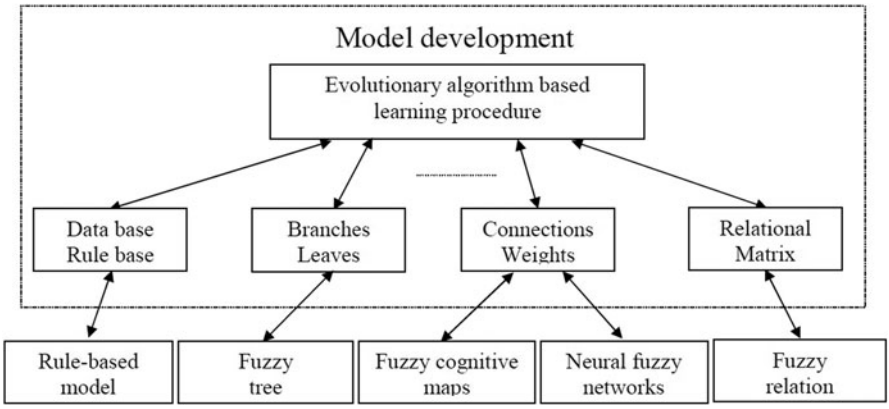


Fig. 5.1 Evolutionary fuzzy modeling process

where $I([0,1])$ denotes the family of all closed intervals of $[0,1]$. More generally, the intervals of $I([0,1])$ themselves can be replaced by fuzzy sets of $F([0,1])$, the set of all ordinary fuzzy sets on $[0,1]$, as follows:

$$A : X \rightarrow F([0,1]) \tag{5.3}$$

These sets are called fuzzy sets of type 2. Thus, fuzzy sets are type 1 fuzzy sets. Fuzzy sets of type 2 own higher expressive power, but demand greater computational effort that interval-valued and fuzzy sets. Higher types of fuzzy sets can be obtained recursively in a similar way as type 2 sets. L-fuzzy sets relax the requirement that membership values should be in $[0,1]$; let them be elements of a partially ordered set L. L-fuzzy set A has as membership function

$$A : X \rightarrow L \tag{5.4}$$

A different generalization is when a fuzzy set is defined by functions whose domain itself has fuzzy sets as elements. If $F(X)$ denotes the set of all fuzzy sets on domain X, then order 2 fuzzy set A (alternatively, level 2 fuzzy sets) has membership function of the form

$$A : F(X) \rightarrow [0,1] \tag{5.5}$$

Order 2 fuzzy sets can be generalized into higher order fuzzy sets recursively, similarly as type 2.

Except for this brief remind, the properties, operations, and procedures to computate with generalized fuzzy sets will not be addressed in this chapter. See [13, 20–22] for further explanation and details.

GFRBS is a fuzzy rule-based system enhanced by a learning procedure based on genetic algorithms [14]. A fuzzy rule-based system (FRBS) is expressed in terms of a knowledge base (KB) and an appropriate inference engine. Often, input interfaces

are required to convert measurements into fuzzy sets to capture data imprecision. Input interfaces serves as a fuzzification module to allow fuzzy reasoning. Conversely, a fuzzy output may be converted by output interfaces into a single value that, in some sense, is the best representative of the fuzzy set. The KB has two main components, the data base and the fuzzy rule base. The data base (DB) contains definition of the linguistic variables, scaling functions of variables, membership functions, and parameters of the model. The fuzzy rule base (RB) is a collection of fuzzy rules. Fuzzy rules can be either linguistic or functional. Linguistic fuzzy rules have linguistic variables and associated fuzzy sets in both, rule antecedent and consequent. Functional fuzzy rules have linguistic variables in the rule antecedent only. The consequents are functions of the base input variable, or a function approximator such as a multilayer feedforward neural network.

A GFS must perform two major tasks: tuning and learning [9]. Tuning concerns optimization of an existing FRBS, whereas, learning constitutes an automatic design procedure to develop collections of fuzzy rule-bases. Tuning assumes that a RB is available. The aim is to find optimal values for parameters of the membership, scaling functions, gains, that is, to find appropriate values of the entries of the DB. Learning performs a search in the space of the rule bases, data bases, or both.

Scaling functions are used to normalize the domains of the input and output variables. The scaling functions are parameterized either by a single scale factor, or a lower and upper bound factor in case of linear scaling. Several contraction or dilation parameters are needed in nonlinear scaling. Parameter tuning involves adaptation of one to four parameters per variable, depending on the scaling strategy. For instance, linear scaling requires two and nonlinear scaling three or four. Real and binary representations can be used to encode parameters. Because the number of input and output variables and the number of scaling parameters are known a priori, fixed length code is the simplest.

Tuning of membership functions often requires an individual representing the entire DB because its chromosome encodes parameterised membership functions associated to the linguistic terms of all fuzzy partitions. Common shapes for the membership functions are triangular, trapezoidal, or Gaussian. In these cases, the number of parameters per membership function ranges from one to four. The parameters can be encoded using either real or binary values. The structures of the chromosome are slightly different depending if the rules are linguistic or functional. Tuning the membership functions of the linguistic models requires that the entire fuzzy partitions be encoded into the chromosome. The chromosome must be globally adapted to maintain the semantics of the RB, that is, the meaning of the model. If we assume a predefined number of linguistic terms for each variable, then fixed length chromosomes can be adopted. Notice that, even when the length is fixed, the number of linguistic terms of a variable can be found using and extra gene to encode the maximum number of variables. This strategy can also be adopted to evolve the parameters of the consequents of functional fuzzy rules. The maximum number of parameters defines the length of the chromosome.

An approach to learn the KB of an FRBS is to develop the DB and the RB in two separate steps within the same process. An example is the embedded GFRBS

which learns the DB using simultaneously a simple method to develop a RB for each DB [14]. The embedded GFRBS, however, does not necessarily provide simple, transparent, and competitive models in terms of the generalization capability. They do not scale well in terms of processing time and memory [3]. A way to reduce the search space in an embedded genetic DB learning framework is suggested in [3] together with a fast multiobjective evolutionary algorithm. Lateral displacement of fuzzy partitions using a unique parameter for all membership functions of each linguistic variable is one of the mechanisms adopted to reduce the dimension of the search space. The idea is to prescreen promising partitions to avoid overfitting, and to maintain coverage and semantic soundness of the fuzzy partitions. The algorithm also includes incest prevention, restarting, and rule-cropping in the RB generation process to improve convergence. Despite the mechanisms introduced to manage dimensionality, the algorithm does not scale-up on the number of data in datasets. A strategy to deal with scalability is to avoid large percentage of samples and error estimation using a subset of the original dataset. A postprocessing step refines the model further.

Evaluation of complete solutions in isolation is one feature of evolutionary approaches. In general, interactions between the individuals of a population are not accounted for and there is no evolutionary pressure for coadaptation, which is inadequate to develop complex models. One way to approach this issue is trough cooperative coevolution [23]. An alternative to cooperative coevolution is the participatory evolution approach, detailed in Sect. 5.3. The architecture of a general cooperative coevolutionary fuzzy systems is shown in Fig. 5.2. In principle, any evolutionary algorithm could be adopted, but in what follows we emphasize genetic algorithm [10].

Individuals of the four populations in Fig. 5.2 represent four distinct species and encode different design parameters. Partitions of the domain of the variables and corresponding membership functions, called partition set level for short, are at the first level. Rules are the individuals of the second level, sets of fuzzy rules are the individuals of the third level, and rule semantics and inference compose the population of the fourth level. The encoding scheme uses real and integer encoding, summarized in Fig. 5.3. The populations coevolve via repeated application of the evolutionary operators selection, crossover, and mutation.

Note that higher order fuzzy modeling can be pursued extending the encoding of the individuals accordingly.

Beyond GFRBS, the use of GFS and genetic programming approaches to develop fuzzy decision trees is addressed in [11, 18] and references therein. A survey of fuzzy decision tree classifier methodology is given in [30]. Fuzzy relational modeling in the realm of evolutionary computation is discussed in [24], and neural network design in [4]. For a recent account of evolutionary and genetic algorithms to design fuzzy models, see [1].

The application examples of the GFS are many. For instance, [27] develops a multiobjective approach in which a fuzzy controller regulates the selection procedure and fitness function of genetic algorithms. They developed timetables of railway networks to reduce passenger waiting time when switching trains while, at the same

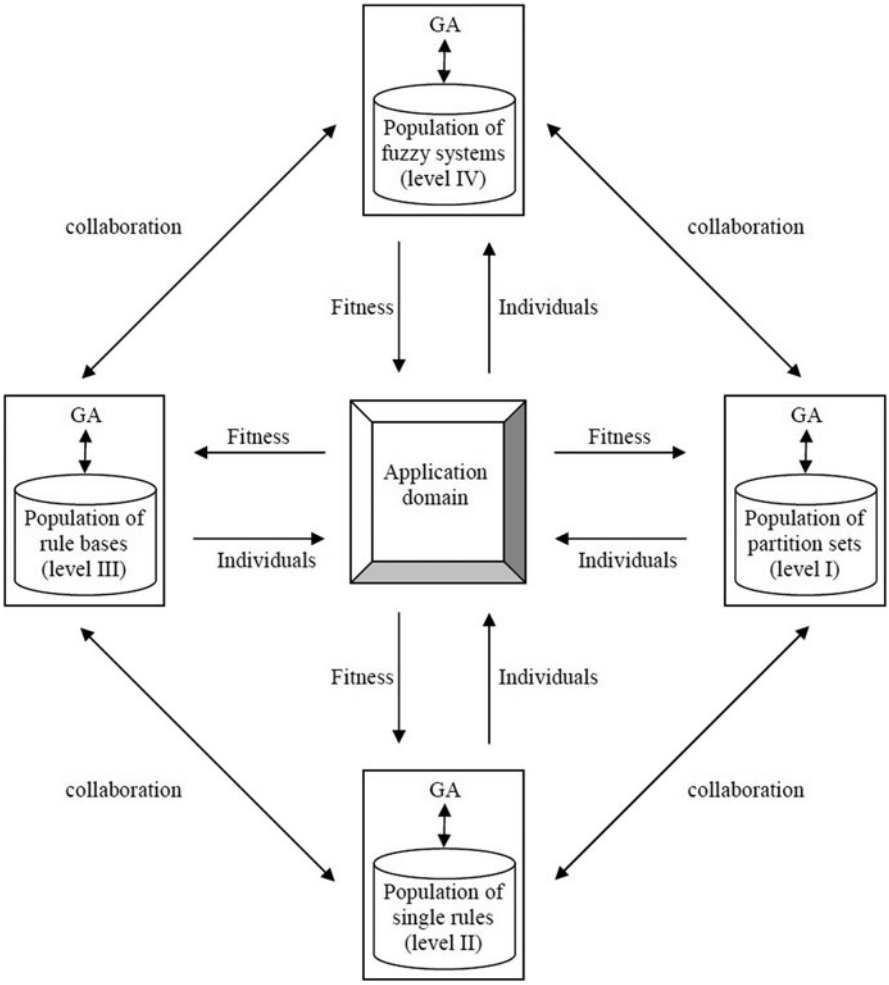


Fig. 5.2 Coevolutionary genetic fuzzy system architecture

time, minimizing the cost of new investments to increase capacity. The result of the genetic optimization is a cost–benefit curve that shows the effect of investments on the accumulated passenger waiting time and trade-offs between criteria [9]. In [17] the aim was to optimize trip time and energy consumption of a high-speed railway using fuzzy c-means clustering and a genetic algorithm. The method was used to derive a control strategy for a high-speed train line. An economical train run with a trip time margin of less than 7 % and an energy saving of 5 % was reported [9]. A model to relate the total length of low voltage line installed in a rural town with the number of people in the town and the mean of the distances from the center of the town to the three furthest clients in it was addressed in [7]. The authors compare the

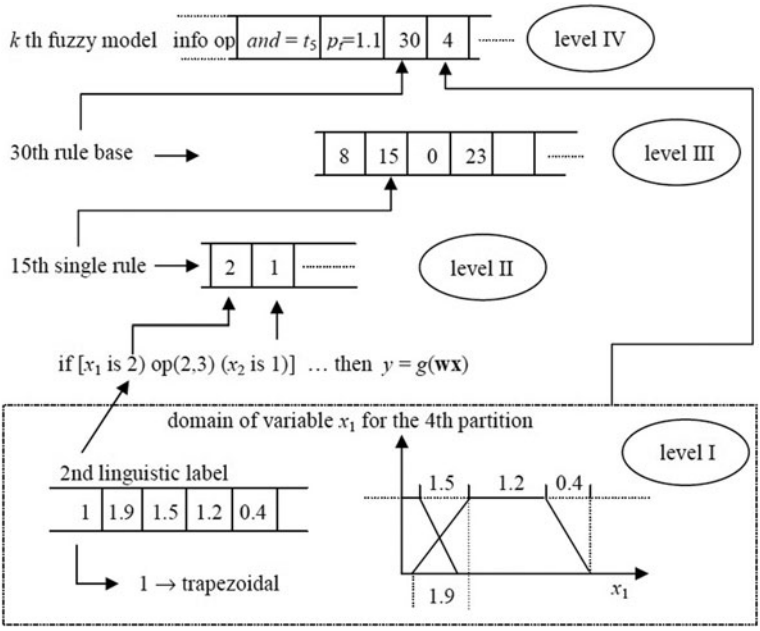


Fig. 5.3 Encoding in coevolutionary genetic fuzzy systems

training and test errors achieved by different modeling techniques. See also [1] for further techniques and applications of evolutionary intelligent systems.

5.3 Participatory Evolutionary Learning Algorithm

This section describes the participatory evolutionary learning algorithm (PELA). We first review the main concepts and ideas of the participatory learning paradigm. Next the selective transfer recombination mechanism is detailed and explained, and the difference between the selective transfer and classic crossover is highlighted.

5.3.1 Participatory Learning

During the 1990s, Yager suggested the concept of participatory learning, a scheme in which the process learning depends on what is already known or believed [32]. A central characteristic of the idea of participatory learning is that an observation has the greatest impact in causing learning or knowledge revision when it is compatible with the current knowledge. Learning occurs in an environment in which the current knowledge participates in the process of learning about itself. A fundamental part of

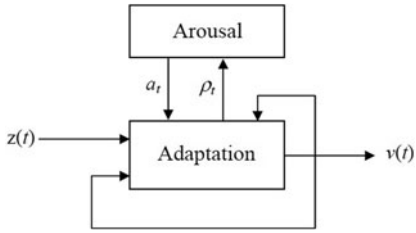


Fig. 5.4 Participatory learning

this learning scheme is the compatibility between observation and knowledge. This is depicted in Fig. 5.4. What is important to note is that the current knowledge in addition to providing, via the lower loop, a standard against which the observations are compared, they directly affect the process used for learning via the upper loop. The upper loop indicates that current knowledge affects how the system accepts and processes input information. The upper loop corresponds to the participatory nature of the learning scheme addressed in this chapter.

Participatory learning uses the notion of compatibility degree which is computed as the similarity measure between current knowledge and current observation. When an observation is far from the current knowledge, it is filtered [32]. That is, if observations are very conflicting with the current knowledge, then they are discounted.

A formal mechanism to update knowledge is a smoothing algorithm like the following

$$v(t+1) = v(t) + \alpha\rho_t(z(t) - v(t)) \tag{5.6}$$

where $v(t+1)$, $v(t)$ and $z(t)$ are n-dimensional vectors corresponding to the new knowledge, the old knowledge, and the current observation at step t, respectively. The basic learning rate is $\alpha \in [0,1]$, and $\rho_t \in [0,1]$ is the compatibility degree at t. This smoothing mechanism corresponds to the lower loop of Fig. 5.4 and will not be emphasized here. See [19] for details. Essential for the purpose of this chapter is the notion of compatibility degree. One way to compute the compatibility degree ρ_t is as follows:

$$\rho_t = 1 - \frac{1}{n}\sum_{k=1}^{n} |z_k(t) - v_k(t)|. \tag{5.7}$$

Note that if $\rho_t = 0$, then $v(t+1) = v(t)$ which means that the current observation $z(t)$ is completely incompatible with the current knowledge $v(t)$. This condition implies that the system is not open to any learning from the current observation. On the other hand, if $\rho_t = 1$, then $v(t+1) = z(t)$. In this case the observation is in complete agreement with the current belief system and thus the system is fully open for learning. Also, notice that the basic learning rate α is modulated by the compatibility degree. This helps to attenuate swings due to values of z which are far from v because it smooths the effect of conflicting observations such as outliers.

5.3.2 Selection

Selection of PELA uses (5.7) to compute the compatibility degree between individuals z and v of a population, that is, to measure how similar individuals z and v are.

Assume a set $S^t = S^t_{old} \cup S^t_{new}$ of strings of fixed length n at step t. Let $S^t_{old} \subset S^t$ denote an old population, and $S^t_{new} \subset S^t$ denote a new population at step t. Let $s^* \in S^t$ be the best individual from the objective function point of view, and $s \in S^t_{new}$ and $s^{'} \in S^t_{old}$ be two individuals chosen randomly. In PELA selection proceeds computing compatibility degrees between s and $s^{'}$ and choosing the one that is closest to the best individual, s^* . More precisely, the compatibility degrees $\rho^s(s, s^*)$ and $\rho^{s^{'}}(s^{'}, s^*)$ are found as follows

$$\rho^s = 1 - \frac{1}{n}\sum_{k=1}^{n} |s_k - s_k^*|, \tag{5.8}$$

and

$$\rho^{s^{'}} = 1 - \frac{1}{n}\sum_{k=1}^{n} |s_k^{'} - s_k^*|, \tag{5.9}$$

and the individual whose compatibility degree with s^* as the largest is selected. Important in this step is to notice that selection depends on both, the objective function which identifies s^*, and on ρ which measures the compatibility degree between s^* and the candidate individuals. Combination of objective function values with compatibility degrees defines the fitness of the individuals. Jointly, s^* and $\rho(p, s^*)$, $p \in S^t$, decide which individual is selected. We see that the individuals themselves are a part of the selection procedure, which is a manifestation of the participatory nature of PELA.

5.3.3 Selective Transfer

During the last few years, we have witnessed a growing interest to use economic principles and models of learning in the genetic algorithms. For instance, evolutionary processes have been used to model the adaptive behavior of a population of economic agents [5]. Here agents develop models of fitness to their environment in conjunction with the corresponding economic activities. Economists believe that behavior acquired through individual experience can be transmitted to future generations, and that learning changes the way to search the space in which evolution operates. This is an argument in favor of the interaction between the processes of evolution and learning. Since technical knowledge is distributed across the economic population, technological change can be viewed as a process of distributed learning.

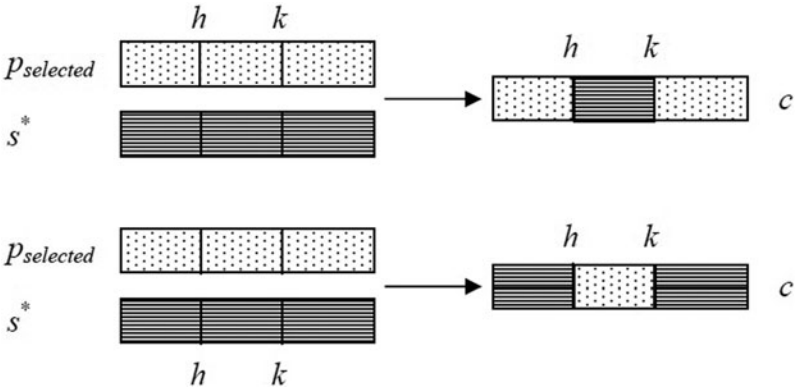


Fig. 5.5 Selective transfer

Here, the term learning is used in a broad sense, that is, there is no distinction between learning as propagation of knowledge through the population and the process of innovation, creation, and discovery. The distributed learning perspective helps to understand technological change and focus on the population suggests that an evolutionary perspective may be appropriate.

Birchenhall et al. [5] claim that our knowledge and technology are modular, that is, they can be decomposed into several components or modules. From the evolutionary computation point of view, they suggest that the crossover operator of genetic algorithms could be seen as a representative of modular imitation. To bring these ideas together, they advocate an algorithm that replaces selection and crossover operators by an operator based on selective transfer. Essentially, selective transfer is a filtered replacement of substrings from one string to another, without excluding the possibility that the entire sequence is copied [6]. Clearly, the selective transfer is similar to Holland crossover, but it is one-way transfer of strings, not an exchange of strings. The behavior selective transfer is likely to be very different from the combination of selection and crossover. PELA translates the selective transfer idea into a recombination procedure as follows.

Assume that an individual $p_{selected}$ is selected using the objective function and compatibility as described in the Sect. 5.3.2. Two positions $h \leq k$ in the $p_{selected}$ string are chosen randomly, and a fair coin is tossed. If the coin turns head, then the substring from $p_{selected}(h)$ to $p_{selected}(k)$ of $p_{selected}$ is replaced by the corresponding substring from $s^*(h)$ to $s^*(k)$ of s^*. If the coin turns up tail, then the substring from $p_{selected}(1)$ to $p_{selected}(h-1)$ and from $p_{selected}(k+1)$ to $p_{selected}(n)$ are replaced by the corresponding substrings of s^*. Figure 5.5 illustrates the idea of selective transfer.

Despite similarity with crossover of the standard genetic algorithms, there are some differences. The most important one is that selective transfer uses one-way relocation of substrings, from the best individual to the individual selected, and hence it is not a crossover. This is important because selective transfer is much more schemata destructive than the standard crossover.

In what follows, an analysis of the selective transfer behavior is developed. Since, recombination aims at mixing individuals to combine their meaningful properties, it should fulfill appropriate requirements to produce offspring with such desirable properties. For instance, an important requirement uses the metric adopted as follows [25]. Given two parent solutions x^{p1} and x^{p2} and an offspring x^0, proper recombination operators should fulfill [26]

$$d(x^{p1}, x^{p2}) \geq \max(d(x^{p1}, x^0), d(x^{p2}, x^0)). \tag{5.10}$$

Inequality (5.10) expresses the requirement on the distances of offspring and their parents to be equal to or smaller than the distance between the parents.

When the distance $d(x^{p1}, x^{p2})$ between x^{p1} and x^{p2} is viewed as a measurement of dissimilarity, (5.10) guarantees that offspring are similar to parents. In an extreme case, when parents are the same, $x^{p1} = x^{p2}$, the offspring are equal to the respective parents, $x^0 = x^{p1} = x^{p2}$.

The PELA performs selective transfer as follows.

1. Choose h, k, $h \leq k$, and $r \in [0, 1]$ randomly.

2. If $r \leq 1/2$ then $c = [s^*(1:h), p_{selected}(h+1:k), s^*(k+1:n)]$;
   else $c = [p_{selected}(1, 1:h), s^*(h+1:k), p_{selected}(k+1:n)]$.

Notice that s^*, $p_{seleted}$, and c play the role of x^{p1}, x^{p2} and x^0 in (5.10). Notation $c = [s^*(1:h), p_{selected}(h+1:k), s^*(k+1:n)]$ means that the string c is assembled merging the substrings $s^*(1:h)$, $p_{selected}(h+1:k)$, and $s^*(k+1:n)$ of s^*, $p_{selected}$, and s^*], respectively. Substring $s^*(1:h)$ is a copy of the first h components of string s, $p_{selected}(h+1:k)$ is a copy of $p_{seleccted}$ from component $h+1$ up to k. Similarly $s^*(k+1:n)$ is a copy of components from $h+1$ to n of string s^*. To analyze the behavior of selective transfer, we must verify if

$$d(s^*, p_{selected}) \geq \max(d(s^*, c), d(p_{selected}, c)). \tag{5.11}$$

Because $p_{selected}$ is the individual whose compatibility degree of either s or $s^{'}$ with s^* is the largest, we have two options to be verified, respectively.

1. If $p_{selected} = s$, then

$$d(s^*, s) \geq \max(d(s^*, c), d(s, c)). \tag{5.12}$$

2. If $p_{selected} = s'$, then

$$d(s^*, s^{'}) \geq \max(d(s^*, c), d(s^{'}, c)). \tag{5.13}$$

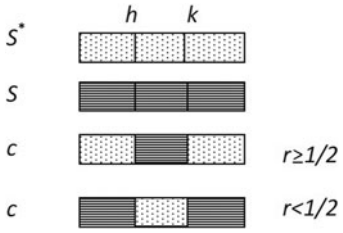


Fig. 5.6 Analysis of selective transfer

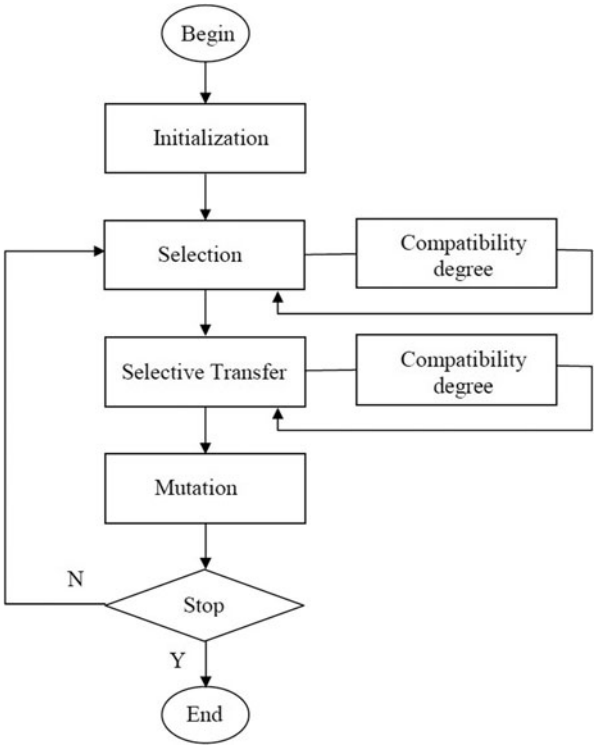


Fig. 5.7 Participatory evolutionary learning algorithm

First, to verify (5.12), we need to check if $d(s^*, s) \geq d(s^*, c)$ and $d(s^*, s) \geq d(s, c)$ hold simultaneously. From Fig. 5.6, if $r \geq 1/2$, then $d(s^*(1:h), c(1:h)) = 0$, $d(s^*(k+1:n), c(k+1:n)) = 0$, and hence $d(s^*, s) \geq d(s^*, c)$. Similarly, since $d(s(h:k), c(h:k)) = 0$, $d(s^*, s) \geq d(s, c)$. If $r < 1/2$, from Fig. 5.6 we see that $d(s^*(h:k), c(h:k)) = 0$, and consequently $d(s^*, s) \geq d(s^*, c)$. Also, because $d(s(1:h), c(1:h)) = 0$ and $d(s(k+1:n), c(k+1:n)) = 0$, $d(s^*, s) \geq d(s, c)$. Thus, (5.12) holds. It can be shown similarly that (5.13) also holds. Therefore, both (5.12) and (5.13) hold. Intuitively speaking, this result suggests that, because the distance of offspring and parents are equal to or smaller than the distance between the parents, selective transfer essentially is an exploitation mechanism.

Figure 5.7 summarizes the participatory evolutionary learning algorithm. The aim of this chapter is to introduce a participatory evolutionary learning algorithm to develop fuzzy models. PELA is an instance of participatory evolutionary algorithm useful to construct a participatory genetic fuzzy system (PGFS) approach. The PGFS assumes that an appropriate encoding scheme for the intended class of fuzzy models has been chosen by the user. It works as follows.

As before, let $S^t = S^t_{old} \cup S^t_{new}$ be a set of strings of length n at step t. $S^t_{old} \subset S^t$ denotes the population at step t called old population, and $S^t_{new} \subset S^t$ denotes the population at step t, called new population. Let $s^* \in S^t$ be the best individual from the objective function point of view. The directive $last(S^{t+1})$ denotes the last individual of the population S^{t+1}. The detailed steps of the PGFS are given in Fig. 5.8.

5.4 Application Example and Performance Evaluation

This section illustrates the use of PGFS to develop a linguistic fuzzy rule-based model of electrical maintenance using actual data. Performance of PGFS is evaluated and compared with a state of the art GFRBS. The data and encoding methods adopted here are the same as reported in [3]. They will be briefly reviewed for sake of completeness. For comparison and performance evaluation purposes, we also rely on the results achieved in [3]. The PGFS adopts the embedded GFRBS approach. It learns the DB using Wang and Mendel [28, 29] method to construct the RB for each DB.

1. Database encoding uses a double-encoding scheme $C = C_1 + C_2$. First, equidistant strong fuzzy partitions are defined considering the granularity of C_1. Second, the membership functions of each variable are uniformly displaced to a new position considering the displacement values of C_2.
   a. Number of linguistic labels C_1: It is a vector of integers of size N representing the number of linguistic variables.

$$C_1 = (L^1, \ldots, L^N). \tag{5.14}$$

   Gene L^i is the number of labels of the ith linguistic variable, $L^i \in \{2, \ldots, 7\}$.
   b. Lateral displacements C_2: It is a vector of real numbers of size N. It encodes displacements α^i of the different variables, $\alpha^i \in [-0.1, 0.1]$.

$$C_2 = (\alpha^1, \ldots, \alpha^N). \tag{5.15}$$

   For a more detailed description of the linguistic 2-tuple representation scheme, please see [2, 15].
2. Rule bases are produced using the Wang and Mendel algorithm as follows:
   a. Partition the input and output spaces
   b. Generate fuzzy rules using data
   c. Assign a confidence degree to each rule generated to solve conflicts

Fig. 5.8 Participatory genetic fuzzy system procedure

procedure PGFS
  f objective function
  generate S^0 randomly
  set $a_0 = 0$; $t \leftarrow 0$
  while $t \leq t_{max}$ **do**
    compute s^*
    Selection
    choose $s \in S^t_{new}$ and $s' \in S^t_{old}$ randomly
    $best \leftarrow s^*$
    compute $\rho^s(s, best)$ and $\rho^{s'}(s', best)$
    if $\rho^s \geq \rho^{s'}$ **then**
      $p_{selected} \leftarrow s$
    else
      $p_{selected} \leftarrow s'$
    end if
    Selective Transfer
    choose $\alpha, \beta \in [0, 1]$ randomly
    choose h, k, $h \leq k$, and $r \in [0, 1]$ randomly
    if $r \leq 1/2$ **then**
      $c = [best(1, 1:h), p_{selected}(1, h+1:k), best(1, k+1:n)]$
    else
      $c = [p_{selected}(1, 1:h), best(1, h+1:k), p_{selected}(1, k+1:n)]$
    end if
    Compute compatibility degrees
    $\rho^c(c, best)$ and $\rho^{p_{selected}}(p_{selected}, best)$
    if $\rho^c \geq \rho^{s^*}$ **then**
      $p_r \leftarrow c$
    else
      $p_r \leftarrow p_{selected}$
    end if
    Mutation
    compute $E = n \cdot m$, m is the mutation probability
    choose $\gamma \in [0, 1]$ randomly
    if $\gamma < E$ **then**
      $p_m \leftarrow mutate(p_r)$
    end if
    if $f(p_m)$ better than $f(best)$ **then**
      $best \leftarrow p_m$
    end if
    generate S^{t+1}
    $last(S^{t+1}) \leftarrow best$
    $t \rightarrow t + 1$
  end while
  return *best*
end procedure

d. Create a fuzzy rule base combining the rules generated from data with, if available, rules given by experts
e. Find the input–output mapping using the fuzzy rule base, an inference engine, and a defuzzification method

3. Objective function: This is the mean-squared error (MSE)

$$MSE = \frac{1}{2|D|} \sum_{l=1}^{|D|} (F(x^l) - y^l)^2 \tag{5.16}$$

Table 5.1 ELE modeling methods considered in the experiments [3]

Method	Type of learning
WM(3)	Rule base by Wang and Mendel algorithm (WM) with 3 labels
WM(5)	Rule base by WM with 5 labels
WM(7)	Rule base by WM with 7 labels
WM	Rule base by WM with different labels
FSMOGFS	Gr [8], lateral partition parameters and rule base by WM
FSMOGFS+TUN	FSMOGFS + (Tuning of MF parameters and rule selection by SPEA2$_{E/E}$ [12])
FSMOGFS^e	FSMOGFS including fast error estimation
FSMOGFS^e+TUN^e	FSMOGFS + TUN including fast error estimation

where $|D|$ is the size of the dataset, $F(x)$ is the output of the FRBS model, and y the actual output produced by input x.

This section uses max–min fuzzy inference and center of gravity defuzification.

4. Initial population: This is generated as follows. Each chromosome has the same number of labels, from two to seven labels for each input variable. For each number of input labels, all possible combinations are used to assign respective rules consequents. Additionally, for each combination, two copies are included with different values in the C_2 part. The first has values randomly chosen in $[-0.1, 0]$ and the second random values in $[0, 0.1]$.
5. Selective transfer: This assumes that $p_{selected} = (x_1 \ldots x_N)$ and $s^* = (y_1 \ldots y_N)$ with $x_i, y_i \in [a_i, b_i]$ and $i = 1, .., N$, are two real-coded chromosomes in the part C_2. The parent-centric BLX [16], which is based on BLX-α, is applied to the C_2 part. The selective transfer operator generates the following individuals:

$$c_2 = (c_{11} \ldots c_{1\,N}) \tag{5.17}$$

where

a. If $r \leq 1/2$ then c_{1i} is randomly generated in the interval $[l_i^1, u_i^1]$, with $l_i^1 = max\{a_i, x_i - I_i\}, u_i^1 = min\{b_i, x_i + I_i\}$, and $I_i = |x_i - y_i| \cdot \alpha$ with $\alpha = 0.3$.
b. If $r > 1/2$ then c_{1i} is randomly generated in the interval $[l_i^2, u_i^2]$, with $l_i^2 = max\{a_i, y_i - I_i\}, u_i^2 = min\{b_i, y_i + I_i\}$.

6. Mutation: This uses the expected value E. It decreases the granularity by 1 in a randomly selected gene g, that is, $L^g = L^g - 1$.

The electric maintenance model has four input variablesand one output variable. The dataset (ELE) has 1056 samples. The nature of data suggests learning methods to develop a considerable number of rules. Therefore, the problem involves a large search space [3]. The ELE data is available at http://sci2s.ugr.es/keel/index.php.

Table 5.1 summarizes the methods considered in [3]. WM [29] is a reference because all methods addressed in this chapter use it to evolve rule bases.

Table 5.2 shows the average of 12 runs of the participatory genetic fuzzy system (PGFS). The population size was 60 and evolution stops after 1000 generations.

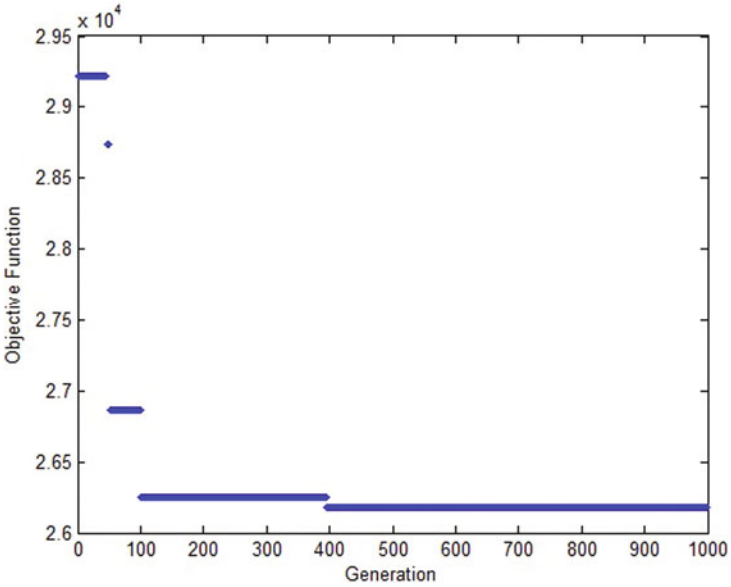


Fig. 5.9 Convergence of PGFS

The selection of two inputs in the method WM was done as suggested in [3]. The results of Tables 5.2 and 5.3 highlight the following. The average mean-squared error (MSE) achieved by the different models evolved by the participatory genetic fuzzy system (PGFS) were lower than the average values of Table 5.3. The four inputs model evolved by the PGFS using WM with different labels is more accurate than the PGFS using WM with a fixed number of linguistic labels. The standard deviation (SD) of the MSE of PGFS with WM is smaller. In both, Tables 5.2 and 5.3, we notice that the MSE decreases as the number of linguistic labels adopted by WM increases. The lowest value is achieved for seven labels, WM(7). For the two inputs modeling case, the PGFS using WM performs best among FSMOGFS and FSMOGFS + TUN from the point of view of the MSE and SD. In this case, the processing time of the PGFS was 105.3 s while FSMOGFS + TUN took 125 s. Thus PGFS is faster.

Figure 5.9 depicts the values of the objective function (MSE) during a typical run. Note the fast convergence of the participatory genetic fuzzy procedure.

5.5 Conclusion

This chapter has introduced a new evolutionary learning approach based on participatory learning and selective transfer. The participatory nature means that current population plays an important role in shaping evolution. Exchange of information during recombination is done using selective transfer instead of selection and

Table 5.2 ELE data modeling: MSE values and averages of PGFS with different labels

	PGFS algorithm									
Method Inputs Rules	WM(3) 4 27		WM(5) 4 65		WM(7) 4 103		WM 4 80		WM 2 8	
Run	MSE	Time	MSE	Time	MSE	Time	MSE	Time	MSE	Time
1	41,135	206	23,460	392	22,209	610	17,191	455	6349.8	104
2	44,878	204	23,542	409	15,598	613	13,007	476	6633.7	106
3	55,793	193	23,531	391	15,683	616	17,191	465	6637.2	104
4	54,108	194	22,197	417	7683.3	610	13,007	481	6637.6	102
5	38,388	193	23,531	392	15,683	611	12,995	451	6633	104
6	43,175	196	28,365	388	15,693	623	17,191	474	6352.7	107
7	63,158	193	23,460	399	7683.3	615	12,995	505	6330.5	104
8	59,713	196	28,379	391	15,682	613	12,995	471	6632.2	105
9	40,615	194	36,036	417	22,053	613	12,995	483	6625	105
10	41,906	197	26,445	391	29,544	614	12,995	476	6631.8	106
11	50,074	192	28,433	409	22,275	614	13,007	465	6628.5	106
12	41,906	202	35,034	391	7683.3	612	13,007	475	6668.6	102
Mean	47,904.08	197	26,867.75	399.39	16,455.83	614.07	14,048	473.55	6563.38	105.3
SD	8369.57	4.62	4634.54	10.38	6766.05	3.42	1895.31	13.93	132.63	1.57

Table 5.3 Average MSE values when training with different algorithms for ELE dataset [3]

Method	WM(3)	WM(5)	WM(7)	FSMOGFS	FSMOGFS^e	FSMOGFS +TUN	FSMOGFS^e +TUN^e
Input	4	4	4	2	2	2	2
Rules	27	65	103	10	9	9	8
Mean	192,241	56,135	53,092	16,018	16,153	8803	9665
SD	9658	1498	1955	314	450	739	823

crossover of the canonical genetic algorithm. Participatory learning and selective transfer helps to improve the performance of evolutionary learning because it decreases complexity and increases search power, producing high quality models with less computational effort. The approach is an instance of participatory genetic fuzzy system whose main purpose is to serve as a tool to develop fuzzy models. Computational experiments using actual data suggest that the participatory genetic fuzzy system performs better than state of the art genetic fuzzy systems approaches. Future work shall address convergence analysis formally, new forms of participatory evolutionary computation, and applications in modeling and optimization.

Acknowledgment The second author is grateful to CNPq, the Brazilian National Research Council, for grant 304596/2009-4.

References

1. A. Abraham, C. Grosan, W. Pedrycz, *Engineering Evolutionary Intelligent Systems*, vol. 82 (Springer, Berlin, 2008)
2. R. Alcalá, J. Alcalá-Fdez, F. Herrera, J. Otero, Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation. Int. J. Approx. Reason **44**(1), 45–64 (2007)
3. R. Alcalá, M. Gacto, F. Herrera, A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. IEEE Trans. Fuzzy Syst. **19**(4), 666–681 (2011)
4. A. Azzini, A.G.B. Tettamanzi, A new genetic approach for neural network design, in *Engineering Evolutionary Intelligent Systems*, ed. by A. Abraham, C. Grosan, W. Pedrycz. (Springer, Berlin, 2008) pp. 289–323
5. C. Birchenhall, L. Jie-Shin, *Learning Adaptive Artificial Agents: Analysis of an Evolutionary Economic Model*. 6th International Conference Society for Computational Economics, Spain, July 2000
6. C. Birchenhall, N. Kastrinos, S. Metcalfe, Genetic algorithms in evolutionary modelling. J. Evol. Econ. **7**, 375–393 (1997)
7. O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Base* (World Scientific, Singapore, 2001)
8. O. Cordón, F. Herrera, P. Villar, Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. IEEE Trans. Fuzzy Syst. **9**(4), 667–674 (2001)
9. O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Sets Syst. **41**(1), 5–31 (2004)

10. M. Delgado, F. Von Zubenb, F. Gomide, Coevolutionary genetic fuzzy systems: a hierarchical collaborative approach. Fuzzy Sets Syst. **141**(1), 89–106 (2004)
11. J. Eggermont, Evolving fuzzy decision trees with genetic programming and clustering, in *Genetic Programming*, ed. by J.A. Foster, E. Lutton, J.F. Miller, C. Ryan, A.G.B. Tezzamanzi. (Springer, Berlin, 2002), pp. 71–82
12. M. Gacto, R. Alcalá, F. Herrera, An improved multi-objective genetic algorithm for tuning linguistic fuzzy systems. *Proceedings of International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Spain, pp. 1121–1128, 2008
13. S. Gottwald, Set theory for fuzzy sets of higher level. Fuzzy Sets Syst. **2**, 125–151 (1979)
14. F. Herrera, Genetic fuzzy systems: taxonomy, current research trends and prospects, Evol. Intell. **1**, 27–46 (2008)
15. F. Herrera, L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words. IEEE Trans. Fuzzy Syst. **8**(6), 746–752 (2000)
16. F. Herrera, M. Lozano, A. Sánchez, A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study. Int. J. Intell. Syst. **18**, 309–338 (2003)
17. H. Hwang, Control strategy for optimal compromise between trip time and energy consumption in a high-speed railway. IEEE Trans. Syst. Man Cybern. 28(6), 791–802 (1998)
18. M.W. Kim, J.W. Ryu, Optimized fuzzy decision tree using genetic algorithm, in *Neural Information Processing*, ed. by I. King, J. Wang, L-W. Chan, D. Wang. (Springer, Berlin, 2006), pp. 797–806
19. Y.L. Liu, F. Gomide, Evolutionary Participatory Learning and Fuzzy Systems Modeling, Internal Report UNICAMP-FEEC-DCA, (submitted), 2014
20. J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions* (Prentice Hall, Upper-Saddle River, 2001)
21. H. Nguyen, E. Walker, *A First Course in Fuzzy Logic*, 3rd edn. (Chapman Hall, Boca Raton 2006)
22. W. Pedrycz, F. Gomide, *Fuzzy Systems Engineering: Toward Human-Centric Computing*. (Wiley Interscience/IEEE Press, Hoboken, 2007)
23. M. Porter, K. De Jong, Cooperative coevolution: an architecture for evolving coadapted subcomponents. Evol. Comput. **8**, 1–29 (2000)
24. M. Prosperi, G. Ulivi, Evolutionary fuzzy modelling for drug resistant HIV-1 treatment optimization, in *Engineering Evolutionary Intelligent Systems*, ed. by A. Abraham, C. Grosan, W. Pedrycz. (Springer, Berlin, 2008), pp. 251–287
25. N.J. Radcliffe, P.D. Surry, Evolutionary computing, vol. 865, in *Lecture Notes in Computer Science, Formal Memetic Algorithms*, ed. by Fogarty, T.C. (Springer, Berlin, 1994)
26. F. Rothlauf, *Design of Modern Heuristics: Principles and Application* (Springer, Berlin, 2011)
27. S. Voget, M. Kolonko, Multidimensional optimization with a fuzzy genetic algorithm. J. Heuristics **4**(3), 221–244 (1998)
28. L. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis* (Prentice Hall, Upper-Saddle River, 1994)
29. L. Wang, J. Mendel, Generation fuzzy rules by learning from examples. IEEE Trans. Syst. Man Cybern. **22**(6), 1414–1427 (1992)
30. T. Wang, Z. Li, Y. Yan, H. Chen, A survey of fuzzy decision tree classifier methodology. Fuzzy Info. Eng. **40**, 959–968 (2007)
31. P. Windrum, C. Birchenhall, Is product life cycle theory a special case? Dominant designs and the emergence of market niches through coevolutionary-learning. Struct. Change Econ. Dynam. **9**, 109–134 (1998)
32. R. Yager, A model of participatory learning. IEEE Trans. Syst. Man Cybern. **20**(5), 1229–1234 (1990)
33. R. Yager, Participatory Genetic Algorithms. BISC Group List, Message posted on 29 Aug 2000.

Chapter 6
Fuzzy Modeling of Economic Institutional Rules

Christopher Frantz, Martin K. Purvis, Maryam A. Purvis, Mariusz Nowostawski and Nathan D. Lewis

Abstract Modeling collective social action is challenging not only because of the opacity of the underlying social processes, but even more because of the insufficient information detail concerning the activities under investigation. Such information gaps are customarily filled using the modeler's intuition or randomization techniques. A promising alternative is to employ fuzzy reasoning. We have built on this potential to employ fuzzy methods as an alternative mechanism to integrate numerous opinions in order to model the establishment of economic institutional rules. Our empirical application domain is based on a historic trade scenario in which traders established rules and shared information in order to prevent the sellers of their goods from cheating them. We address this modeling problem by employing two different group decision-making mechanisms—majority-based voting on the one hand (which follows the original historical case) and preference aggregation using interval type-2 fuzzy sets on the other hand. We compare the outcomes of these two approaches and identify significantly lower sensitivity of the outcomes (i.e., instability of the outcomes to small changes in parameter settings) using fuzzy-set-based approaches in contrast to majority votes. The results suggest that the use of abstract decision-making mechanisms (such as preference aggregation) may be more useful in scenarios that prescribe a decision-making mechanism, but do not provide information to model this process in its entirety. Based on our finding, the potential for a wider use of fuzzy logic in the context of social simulation is discussed and pointers for future investigations are provided.

6.1 Introduction

Collective decision making is the building block for cooperation among social animals, including human beings. However, to provide generalizable and explicit behavioral prescriptions in human societies is a fundamental characteristic that enabled sustained cooperative behavior. Individuals can move between different open societies and, in many cases, rely on the existence of similar institutional settings

C. Frantz (✉) · M. K. Purvis · M. A. Purvis · M. Nowostawski · N. D. Lewis
Department of Information Science, University of Otago, Dunedin, New Zealand
e-mail: christopher.frantz@otago.ac.nz

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_6

and legal environments, such as the security of individual property and basic human rights. A historic precursor of more unified legal systems is the Lex Mercatoria [44] which enabled merchants to trade their goods safely across different jurisdictions in medieval Europe and brought great prosperity to cities (by means of taxation) and traders. An ongoing quest is the analysis of how those institutions, or as North put it "the rules of the game" [47] [1], have come about. The method of simulation is one approach to address the complexity of decision-making processes in an experimental fashion. Modeling decision-making processes in social systems is a complex task that is not only challenging because of its influence from a wide range of social-scientific disciplines such as sociology, psychology, political science, economics, etc., but even more so as it is characterized by uncertainty and subjectivity of the individuals involved in the process.

Under the conceptual roof of group decision making that builds on social choice theory [1], modelers find a range of decision-making operations that they can apply to specific decision problems.

- One well-known modeling technique that lies at the heart of modeling decision problems is *game theory*, which follows utilitarian principles to map a problem scenario on a set of players that face one or more potentially interdependent decision choices which have associated payoffs. Specifying decisions in a game-theoretical fashion facilitates the identification of Nash equilibria that may indicate stable forms of social interaction that become institutionalized by means of institutional rules governing the payoffs (e.g., [22]). Probably the best known example for games that explore cooperative behavior is the Prisoner's dilemma game [2].
- For approaches that require stronger attention to contextual detail, *social simulation* is an alternative that intersects the social sciences with computer science by modeling artificial societies, and thus provides the environment to explore social problems in silica, be it by system-level modeling (e.g., system dynamics [13]), or individual-based modeling using the notion of agents (agent-based social simulation [3, 12]).
- *Fuzzy logic* [59] is another approach that looks at decision making based on the important assumption that individual actors may bear uncertainty about their own opinion or preferences (in contrast to game theory).

In this chapter, we look at the application of fuzzy sets in the context of social simulation to model the establishment of economic institutional rules. A key challenge of social simulation scenarios is the lack of consistent information at a given detailed level to provide a complete model specification. When filling those gaps with crisp data, experimenters must perform careful sensitivity analysis to investigate the impact of their approximations (or the data generated by applied randomization or parameter optimization mechanisms, such as Monte Carlo Methods). An alternative or complementary approach is to utilize representations of uncertainty in their models. However, depending on the context and knowledge level of detail,

[1] We will refine the definition of institutions in Sect. 6.3.

this may even lead to fundamental adaptations of the model by replacing a known component (e.g., voting-based decision-making mechanism) with another that can accommodate a higher level of abstraction and does not require such detailed contextual knowledge (e.g., preference aggregation). We suggest that simulation outcomes that operate on higher abstraction levels are less sensitive to individual parameter inputs, which is particularly important for scenarios that should be explored from multiple perspectives and under different objectives.

To investigate these ideas, we borrow a prototypical scenario from the area of Comparative Economics tagged as the "Maghribi Traders Coalition" [21, 22] that has been used to postulate the emergence of institutional rules in individualistic societies. This historical scenario uses the Italian city state of Genoa as an example, where economic institutional rules emerged, in contrast to a Jewish traders collective (the actual "Maghribi Traders") that operated along the North African coast and could rely on social norms to sustain cooperative behavior. A compelling argument about *why* such rules have come about was given by Avner Greif, who offers a convincing account backed by game-theoretical exploration [22].

In this chapter, we use an agent-based approach to accommodate more individual detail and enable social ability, i.e., direct interaction between individuals, which is crucial for the realistic modeling of social scenarios. Ultimately this allows us to look at the "how" that extends Greif's utilitarian perspective on "why" institutional rules emerged, respectively, 'what' motivated their establishment.

In order to achieve this, in this chapter, we rely on a model that incorporates the metaphor of social forces. Individuals experience their social environment by means of social forces based on individual attraction and repulsion (e.g., by similarity and/or difference, respectively), which drive them in a "social proximity." Being united by a common concern, we assume that individuals take some sort of collective action, which, in this case, is the formulation of rules they will collectively follow.

In the next section, we provide a brief introduction to fuzzy sets that outlines why they are attractive in the context of this work and the field of group decision making in general. We further contextualize our approach in this field.

In Sect. 6.3, we introduce the Maghribi Traders Coalition scenario in more detail, and in Sect. 6.4, we embed it into a simulation concept that takes inspiration from social psychology and models social influence as "forces" that drive the shaping of groups as a precursor for collective action. Section 6.4.2 is dedicated to the analysis of a baseline scenario that uses prescriptions from Greif's approach to establish economic institutional rules by means of the majority vote. Extending this, we use the more inclusive and differentiated approach of preference aggregation by means of interval type-2 fuzzy sets (in Sect. 6.5) and contrast the effects of the different decision-making mechanisms.

Section 6.6 discusses the overall outcomes and the significance of the application of fuzzy sets for the concrete simulation case as well as for the wider context of social simulation.

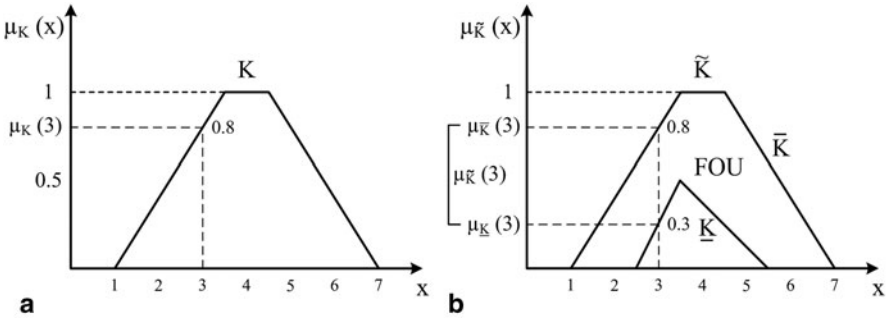


Fig. 6.1 Type-1 vs. type-2 fuzzy sets. **a** Example of type-1 fuzzy set. **b** Example of type-2 fuzzy set

6.2 Fuzzy Sets and Their Application in Group Decision Making

6.2.1 Fuzzy Sets

In this section, we offer a brief nontechnical introduction into the basic idea of Zadeh's fuzzy sets, and in particular, the difference between type-1 (T1FS) and type-2 (T2FS) fuzzy sets [59, 61] to identify properties which make the latter particularly attractive for the application in the context of social simulation.

In contrast to classic set theory that models the belonging of an element to a set using bivalent logic, *fuzzy sets* are characterized by the ability to model uncertainty about the assignment of an element to one or more given sets. Fuzzy sets are modeled as membership functions (MFs) within a universe of discourse. For any input value in a given domain x, a T1FS MF returns the *degree of membership* of that input with the fuzzy set as a value between 0 and 1. Referring to Fig. 6.1a as an example, the degree of membership with the fuzzy set K for the input value 3 is $\mu_K(3)$ which resolves to a certainty of 0.8.

As for any other fuzzy set type, the purpose of the original T1FS is to represent uncertainty. However, the fact that T1FS MF represents uncertainty as crisp values against which input values are evaluated is considered a limitation for their application. When specifying membership functions, modelers need to be "certain about the uncertainty" those functions represent, which operates against the objective to model uncertainty. Klir and Folger [29] offer a good account on this paradox, and Mendel [41, 42] explores its philosophical implications for the scientific applicability of T1FS. The limitations of building systems using T1FS are further explored by Hagras [24, 25], Mendel [40], as well as Wu and Tan [58].

To overcome this problem, Zadeh generalized the concept of fuzzy sets to T2FS and eventually type-n sets [61]. The specialization of *interval type-2 fuzzy sets* [40] (IT2FS) has found wider popularity, mostly because of the benefits derived from their trade-off of expressing uncertainty in membership functions, their comparatively low computational costs, and the reduced complexity that makes them accessible to

a wider community [43]. We refer to Fig. 6.1b to explain the central characteristics following [40] and [57]. To introduce uncertainty into membership functions, IT2FS express the membership function for a given value as intervals that are bounded by an upper ($\overline{\mathrm{K}}$) and a lower ($\underline{\mathrm{K}}$) type-1 membership function. This allows the introduction of varying degrees of uncertainty across the membership function. The area between upper and lower membership function is referred to as the *Footprint of Uncertainty* (FOU) of an interval type-2 MF. For a given input value 3, we thus receive an interval [0.3, 0.8] that describes the degree of membership $\mu_{\widetilde{\mathrm{K}}}(3)$ for that input with fuzzy set $\widetilde{\mathrm{K}}$.

The already wide adoption of fuzzy sets across a range of areas (e.g., medical systems, controller design, decision making) has been broadened by IT2FS, based on their ability to express second-order uncertainty, and it reaches from the optimization of supply chains [45] in Logistics via modeling of age-structured bird populations in the context of Ecology [51] to stock price prediction [37].

Another area that greatly benefits from the application of fuzziness is that of complex social systems which are characterized by the inherent connectedness of individuals (and by the vagueness and ambiguity of shared ontologies). Their inherent collective complexity irreducibly conflicts with the desire for precision (Zadeh's Incompatibility Principle [60]). An example that shows the challenges of analyzing shared understanding across human subjects is Zadeh's, Mendel's and others' work on *Computing with Words* [43, 63], which gears toward a generalized interpretation of subjective qualitative categorization of given words. So it is not surprising that moving to a wider scope, uncertainty plays a central role when building artificial societies that aim to replicate human behavior, be it by the differentiated opinions among artificial individuals, or insufficient or vague information on social relationships, or simulation models in general as analyzed by Hassan et al. [26].

The application of fuzzy sets in the context of group decision making has a long-standing tradition reaching back to the 1970s[2]. To contextualize approaches, a brief overview of group decision making is thus indicated.

6.2.2 Group Decision Making

The field of group decision making (GDM) originally emerged from social choice theory [1], which integrates aspects from decision theory and Economics. A central theme of GDM research is the investigation of decision-making methods and their adequate selection based on the nature of a social choice problem. A selection of commonly applied decision-making mechanisms based on social choice theory include [7]:

[2] See [48] for an overview.

- Preference Aggregation: Preference aggregation maps sets of individual preferences onto a collective set of preferences and yields toward the highest possible degree of consensus achievable based on the applied aggregation rules.
- Voting: Voting is concerned with the achievement of commonly accepted decisions based on agreed voting rules. The abundance of possible voting rules have made this an extensively researched field [5]. Popular voting rules include simple majority vote, and qualified majorities (e.g., 2/3 majority for constitutional amendments) [8, 20, 54]. Likewise decisions could be driven by submajorities [54][3] or even a minority, such as given in the case of dictatorial rule (i.e., one vote) or a given *quorum threshold* needed to effectuate collective behavior ("quorum response") [53].
- Resource Allocation and Fair Division: Resource allocation is associated with the distribution of limited resources across one or more sets of individuals taking into account the individuals' preferences as well as efficiency and fairness measures [46].
- Coalition Formation: Coalition formation is concerned with modeling the formation of cooperative behavior among individuals to increase the net benefit per individual leveraged from that cooperation. It is interrelated with fairness problems regarding the distribution of added value. An active subfield of investigation is cooperative game theory [34].
- Belief Merging and Judgement Aggregation: These are two closely related fields that apply strategies to integrate individual judgements into matching collective ones [10]. In contrast to voting, consensus is the desired outcome of that process, which can be facilitated by a moderating party or by prescriptions for selecting opinions. Examples for this mechanism are expert panels or juries. It bears dynamic characteristics, as participants may be required to modify or trade opinions during the process.

Many of the mentioned methods are interrelated, i.e., judgement aggregation is a specialization of voting and extends it with the iterative refinement process to reach consensus.

We can broadly structure those group decision-making mechanisms into two categories based on the respective *decision-making objective*. Preference aggregation, voting, as well as belief merging and judgement aggregation, have the objective to *integrate the opinions* of the participants. Resource allocation problems and coalition formation apply a *negotiation approach to satisfy the interests* of respective stakeholders. The iterative nature of judgement aggregation bears negotiation elements, but nevertheless operates along the lines of unifying opinions approximating the highest possible degree of consensus. Figure 6.2 structures these decision-making mechanisms by their objectives.

GDM is an active research area for fuzzy sets. A central theme of work in this area aligns with the achievements in the context of *Computing with Words* [63, 43] and

[3] An example is the choice of new nesting sites in social insects driven by a submajority [55].

Fig. 6.2 Group decision-making mechanisms by objective

uses linguistic quantifiers [62] to derive thresholds for different consensus measures (e.g., "more than 80 %," "most," etc.) [6]. Examples of these include the modeling of majority decisions [50] or consensus-based models that incorporate qualitative aspects and rely on the analysis of linguistic interval preference [16]. GDM in this context is generally interpreted as what we earlier described as judgement aggregation in expert groups and involves two processes (see e.g., [28]):

- The *consensus process* that describes how a highest possible degree of consensus can be achieved and is generally supported by a moderator [11].
- As a second step the *selection process* describes how to isolate a set of solutions from opinions developed in the consensus process.

The work we describe in this chapter applies fuzzy sets for a different purpose and concentrates on the two other opinion integration methods, namely, preference aggregation and voting to develop regulated behavior. Instead of exploiting fuzzy sets for the interpretation of imprecise linguistic expressions, we investigate their potential to integrate differing opinions across large numbers of agents. To achieve this, we exploit the ability of IT2 MFs to express second-order uncertainty, which makes them attractive for the aggregation of individual opinions. To achieve that, we refer to the a historical case that concentrates on the establishment of institutional rules in order to regulate deviant behavior in the context of long-distance trade, which is introduced in the upcoming section.

6.3 Maghribi Traders Coalition

The problem of the Maghribi Traders' Coalition is taken from the area of Comparative Economics and was introduced by Avner Greif [21, 22], who highlighted it as an example of the establishment of institutional rules in eleventh and twelfth century Italy, which was then dominated by influential city states such as Genoa and Venice. They existed in contrast to a Jewish trader community (known as the 'Maghribi

Traders') that operated along the North African Mediterranean coast between the tenth and twelfth centuries. The prominence of this scenario is of interest for comparative purposes for two reasons. As a first aspect, Genoese traders and Maghribis both engaged in long-distance trade around the Mediterranean Sea, their main trade centered around a roughly comparable region; Genoese traders focused on the Southern European coast of the Mediterranean sea, and Maghribi traders dominated trade along the North African shore. However, both trader communities hardly had any interaction with each other and thus developed sustainable cooperation mechanisms independently from each other.

In both cases, long-distance traders relied on "agents" they endowed with their goods to sell them overseas at a commission. However, given the long distance between trading base (here, Genoa) and market place (e.g., Spain), the owners of the goods (traders) were not able to completely observe if the selling agents (which we call "sellers" from here on to avoid confusion with the notion of software agents) truthfully reported gains and losses. Beyond the poor detection of cheating, the harsh competition and lack of interaction among traders (who maintained information about cheaters as a business secret) was to the cheaters' benefit, so they could exploit a large number of traders, even if individual traders memorized and dismissed cheaters individually. For that reason, the Italian traders established institutional rules to control cheating of sellers which was an economic innovation in the context of long-distance trade at that time. Its key benefit was to recover losses, but fore-mostly to identify cheaters publicly in order to hinder their reemployment.

Maghribis, in contrast, relied on an extensive network of relationships that ensured a constant information flow. Using that mechanism, cheating behavior was collectively monitored, communicated, and sanctioned in a decentralized fashion. A partial answer to the question as to why the decentralized model prevailed for Maghribis lies in the homogeneity and the size of the respective group. Maghribis (which is derived from the Arab word for "West") were Jews that had been driven out of the Iraq and established themselves as traders along the North African shore, thereby adopting Muslim customs without adopting their religion or actually fully integrating into their social environment. Their own religion and kinship-based background acted as a social binding link among them, and this social coherence was achievable by the virtue of the group's limited size—the total number of Maghribi traders ranged around 330 to 550. In contrast, Genoa's size (and with this the trader community's size) exploded from 30,000 to 100,000 between 1200 and 1300 AD [22]. Its primary purpose to serve as a base for pirating activities and subsequently long-distance trade (and its constant competition with the superior city state of Venice) suggests that a significant proportion of that population operated as either traders or agents acting on behalf of traders to sell their goods (which we call "sellers" for the remainder of the chapter).

A second reason for the importance of this model is the reasonable level[4] of information maintained about both communities. Avner Greif [22] undertook an extensive game-theoretic analysis of economic, political, and social circumstances as well as characteristics of both groups. His conclusions are that a variety of situational circumstances (e.g., Genoa as a city state) and societal changes (e.g., the notion of nuclear family promoted by church) have motivated the establishment of depersonalized centralized-institutional rules in contrast to the decentralized-control mechanisms Maghribis relied on. Greif went beyond this scenario and generalized the Genoese traders' behavior as individualistic and used Maghribis as an archetype of a collective culture. An interest of Greif's was to understand why a widely distributed collection of self-interested agents could arrive at stable institutions, which he calls " ... a system of rules, beliefs, norms, and organizations that together generate a regularity of (social) behaviour" [22].

For his analyses, Greif put an emphasis on a rational motivation to explain *why* institutional rules came about, but he hardly addressed the situational drivers, the *how*, that drove individual traders into cooperation; nor did he suggest more differentiated institutional rules to govern deviant behavior, other than excluding cheating partners from future business. For this purpose, he applied game-theoretical approaches to derive Nash equilibria for given scenarios, which he saw as symptomatic for the existence of institutional rules.

Our work shares Greif's focus on the "Genoese case," namely, the establishment of institutional rules, but it extends beyond his achievements in that the objective lies in the application of modeling mechanisms that might help to understand how the Genoese might have arrived at given institutional rules. We also wish to consider more differentiated rule sets. This thought is motivated by an interest in the contrasts between collective and individualistic societies. Durkheim [9] characterized collective societies as linked by mechanical solidarity and put them in contrast to societies that were interacting based on individual interest and specialization but also stronger detachment from the social surrounding, and, in accordance with this, reducing the devotion to the "commons." However, Durkheim also outlined patterns for the different society types. He suggested that judicial systems driven by the interest of the group with a strong hierarchical structure generally exhibit more repressive and categorical sanctions, while organic solidarity promotes restitutive sanctions and differentiated legal systems (e.g., commercial law, administrative law, etc.). Applying this dichotomy to our case, we suggest that legal sanctions that the Genoese established by institutional rule were more differentiated than simply prescribing the exclusion of deviants from future commercial interaction, a pattern that could be more suitably attributed to the normatively regulated Maghribis.

[4] Based on their contractual nature, the information about the Genoese society is mostly derived from systematically maintained cartularies [22]. Information about the Maghribis, on the other hand, is largely derived from individual business letters (see [19, 22]), providing the model foundation based on a less systematic and comprehensive but rather anecdotal account.

6.4 Simulation Scenario

To investigate the effects of a differentiated system of sanctions, we conceptualize an agent-based simulation that represents the Genoese case. It consists of a parameterized number of traders and sellers. Traders hold goods of predefined value. Both traders and sellers can only have a parameterized maximum number of concurrent employment relationships.

At the beginning of the simulation, traders are initialized with a given offer wage. Sellers are likewise set up with a minimum wage they accept. Traders send employment requests to sellers. Traders can employ a parameterized maximum number of sellers, sellers can likewise have a parameterized maximum number of employers. Upon employment acceptance by sellers, traders send goods (of fixed value) to sellers. Sellers then realize the market transaction which, with a given probability, results in profit or loss[5]. Once realizing the trade, sellers send the profit back to their traders who can then again send goods during the next round. If a trader suspects cheating, he has a range of three means to sanction the respective seller. He also memorizes this particular suspected cheater and tries to re-employ another seller during the next round. The sanctions traders can apply range from a low sanction which a trader would apply for negligible cheating behavior and merely warns the respective cheater (Sanction: Warning). A stronger sanction would be to recover the loss (Sanction: Loss Recovery). In this case, the trader would recover exactly the amount of loss the cheater had generated. The strongest sanction is a dismissal (Sanction: Dismissal). In this case, the trader quits the employment relationship with the seller after recovering the loss. If a trader does not have sufficient funds to employ sellers, he is forced to quit any economic transaction and is rendered unemployed. His sellers can then wait for employment by another trader.

Throughout the course of the simulation traders memorize the trading behavior of their business partners. They keep records of the truthful behavior of individual sellers. This is incremented for compliant behavior (here, noncheating) and decremented for deviant behavior. To reflect the concept of "negativity bias" [4], impact of negative behavior is valued significantly stronger than of compliant behavior. In this given context, negative experience is elevated by adding the incurred loss to the cheating seller's (negative) reputation. This memory of past interactions is discounted throughout the runtime of the simulation (γ_{forget}), and converges toward zero. We include this mechanism to simulate forgetfulness and to introduce a recency bias to support the idea of a strong situational embedding of agents. Once memories about sellers fall below a specified threshold ($\theta_{discard}$), they are discarded. The agent thus forgets whether this seller acted in a compliant or deviant manner. Maintaining the mean of the past experiences with sellers allows traders to derive an indicator for the emotional and economic "pressure" from cheating they experience (*cheaterPressure*), which is discussed further below.

[5] Realizing losses during market transactions was realistic, as the price for commodities could fluctuate considerably, given the extended travel times in long-distance trade [22].

Given that Genoese traders operated in fierce competition with each other, they did not directly communicate their experience with particular cheaters. So the question arises: If the traders did not communicate, how then were they able to shape institutional rules?

In order to address this issue, we first need to provide the reader with a brief background of our social-psychologically inspired modeling approach concerning the formation of interest groups. Given that it is secondary to the objective to show the benefit of T2FS in our system, we limit this introduction to bare essentials that suffice to understand how rule shaping in the system can come about.

6.4.1 Social Forces as a Pressure Metaphor

To express the Genoese ability to establish institutional rules without direct communication, we rely on the notion of "social forces" introduced by Kurt Lewin [33], which will be used to bring agents with a common purpose together (on a social plane) which is a precursor to establishing shared norms and rules. Lewin's notion of social forces was motivated by his attempt to provide a comprehensive formalization of psychological constructs. We now know that this objective has not been met, but the theories developed in its context have found wide-spread adoption beyond their metaphorical value (e.g., force-field analysis [32], approach-avoidance conflict [31]). The core essence of his work was to suggest that an individual's behavior is fundamentally influenced by its own personality but likewise by its environment, and in particular, social environment. Following this idea, we model our (software) agents to experience attraction to and/or repulsion from other agents in their environment based on their situational relationship. Under these circumstances they generally oscillate between prioritization of cooperation relationships within the in-group (e.g., attraction by sharing similar concerns) and competition among the in-group leading toward attraction to complementary out-group members (e.g., employer–worker relationship).

To operationalize this model, individual agents are randomly positioned in a multidimensional environment, and potential relationship types (social planes) between them are specified, some of which can be static (e.g., belonging to an ethnic group, fixed role assignment) or dynamic (e.g., employment relationships). These are associated with the weights dynamically assigned and modified by the individual agent. Figure 6.3 visualizes the concept schematically.

In addition, individuals can express their attitudes and opinions using tags, which include aspects such as dissatisfaction with given situations. Depending on the specification, tags can be associated with mutual attraction or repulsion. The aggregated effects of tag-based attraction (or repulsion) and relationship-based attraction can lead to agents "approaching" each other to form social constellations which can act as a precursor for collective action, such as the establishment of institutional rules. If individual situational prioritization for concerns (and thus, relationships)

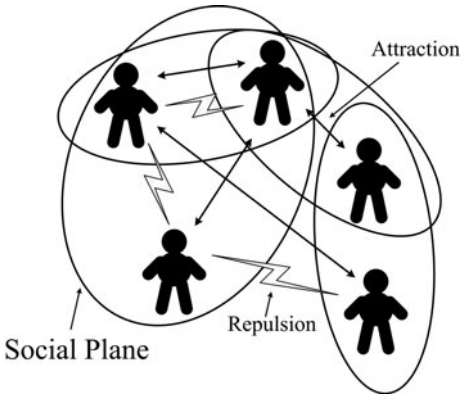


Fig. 6.3 Social forces model

is derived from the agent internals and observing attraction (i.e., social forces) operate on a global level will provide the experimenter with a visual and conceptual understanding of the global social constellation.

Figure 6.4 provides an overview of the different operation stages of the social forces model. The action of *social forces* in the *Attraction* stage leads to *Association* behavior with *social constellations* as its outcome, which provide the basis for the collective to engage in decision making (*rule formation*). The *forces* that lead to association of individuals and even rule formation can then be causally associated with the reaction of those groups. Established rules are thus a behavior that addresses issues that caused individuals to associate in the first place. This applies the metaphor of collective action rooted in social pressure.

We think that this modeling approach has generic value in describing situational social pressures and arising patterns of social constellations. It is applicable to effectively any scenario that incorporates social influence and operates based on the assumption that perceived social pressure, if sustained over longer periods of time, leads to collective action.

However, with its general nature come various challenges regarding its operationalization, such as the choice of independent social planes which, in the context of social sciences, are difficult to specify even though they are intuitively understandable (For example, consider the interdependence of individual demographic factors such as affluence and education level, etc.). Another challenge is the choice of parameters, such as the attraction values on social planes. These aspects are significant empirical issues, but they are not the focus of this work here. We can avoid the discussion at this stage, as this model itself does not have confounding influence on the objective of interest of this work. Instead, our key objective is to present work in the evaluation of rule formation using majority-based decision-making (using votes) in contrast to mechanisms that aggregate individual opinions based on fuzzy sets. It thus concentrates on the last stage (*action stage* in Fig. 6.4) of the social forces model.

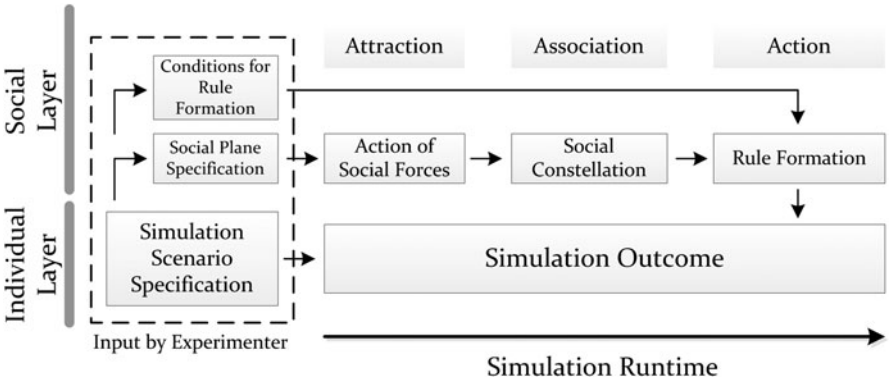


Fig. 6.4 Operation of social forces model

We use this mechanism as a basis to perform collective action, which is achieved by letting individuals engage in decision-making processes based on decision-making strategies provided by the experimenter. This is where the formation of institutional rules becomes meaningful. This is also the starting point for the performance comparison of different decision-making mechanisms (or *rule formation conditions*).

Applying this idea to the Genoese case, which had the characteristics of an individualist society and had established citizen rights based on elementary democratic principles (i.e., participation of citizens with voting rights [22]). In this social context, traders do not directly communicate their experiences with each other. Instead traders privately evaluate their individual experience with sellers. If the overall experience (the mean across all individual experiences) is negative, traders interpret that as being exposed to social "pressure" from cheating and assign themselves a tag displaying the fact that they experience pressure. Tags in this model carry heavier semantics than the conventional tag that acts as a mere syntactic marker. Additional to an action or state description, tags allow the association of a valence with that state. Utilizing this mechanism, individuals can indirectly express their opinion about a condition (e.g., be neutral or negative about the fact that they are affected by cheating).

If this shared experience is of dominant concern (which is measured as an aggregate of the tag weights), individuals are socially attracted to each other and thus move into closer social proximity. If their attraction is sustained, they are clustered and can suggest general measures that address the issue of concern without actually sharing detailed information (such as names of cheaters) that would be to the potential benefit of their fellow trader competitors. In our case, this means that cheated traders can agree upon consequences they impose on detected cheaters. In our model, where clustering signifies a common objective, any suggested countermeasure would be associated with that common purpose. In this case, traders are united by their perceived pressure by cheaters and can then suggest a sanction they would prefer to impose on cheaters, which is generally the sanction they apply individually in the first place.

Using the cluster metrics, we can then specify conditions and means according to which suggested consequences are processed and activated. An example for *rule formation conditions* is the majority-based voting mechanism in order to identify a dominant rule suggestion. Note at this stage that the choice of a decision mechanism along with its parameterization is strongly context-dependent. However, in our case, the City of Genoa was ruled by an elected consulate [22] in which traders' interests were (though not perfectly) represented in a democratic fashion[6].

6.4.2 Baseline Scenario for the Establishment of Institutional Rules

Corresponding to the manner in which Greif employed game theory to suggest a plausible solution to cheating, i.e., the establishment of institutional rules that banned cheaters from further participation in commercial interactions, we base our initial approach on social choice theory and suggest the use of the egalitarian weighted majority rule [39]:

> If the majority of actively employing traders are united in the concern to punish cheaters for their actions, the dominating sanction suggestion is chosen as a permanent rule and adopted by all individuals participating in the rule formation.

This *rule formation condition* still leaves open how individual traders derive rules from their internals. In the preceding section, we introduced the dynamic measure of "social pressure," and more concrete "cheater pressure," as a determinant of a trader's overall perception of emotional and economic pressure. To operationalize it, the trader maintains a memory about the most negative cheating experience ($cheaterPressure_{min}$) he has encountered during the entire simulation runtime. This memory is discounted at a different (generally lesser) rate (represented as $\gamma_{mostNegative}$) due to its greater distinctiveness in comparison to other instances of cheating experience (which are discounted using γ_{forget}). Potential sanctions, "warning," "loss recovery," and "dismissal" are ordered by severity. Starting from the most severe sanction, "dismissal," traders apply and suggest this sanction if their overall situational cheating pressure drops below $cheaterPressure_{min}/2$ (recall that cheating pressure is a negative number). Otherwise, traders apply "recovery of loss" as long as the situational cheater pressure is smaller than $cheaterPressure_{min}/4$. Any lesser cheater pressure (i.e., remaining value range below zero) carries a "warning" as a consequence. Algorithm 1 summarizes this rule set.

[6] The consulate itself, although democratic in its nature, was imperfect as it needed to balance interests of competing clans which, over time, had varying degree of influence (see Greif [22] for details).

Algorithm 1 – Cheater Pressure Algorithm

```
if cheaterPressure < (cheaterPressure_min / 2) then
    consider as HIGH_CHEATING
else
    if cheaterPressure < (cheaterPressure_min / 4) then
        consider as MEDIUM_CHEATING
    else
        if cheaterPressure < 0 then
            consider as LOW_CHEATING
        end if
    end if
end if
```

Given this rule set, traders develop a dynamic and differentiated perception of the cheating behavior of individual traders. We use this simulation as an initial baseline scenario whose results serve as a comparison for further rule formation conditions we will introduce below. In Table 6.1, we highlight simulation parameters that are central to the observations described here. The choice of the selected parameters was driven by sensitivity analysis of the parameter set and focused on stable simulation behavior and identification of tipping points in sanction choice, rather than being based on historically authenticated market behavior[7]. Beyond that, an important variable is the decision-making strategy used to drive collective action, which, at that stage, is the majority rule provided above. The alternative approach based on fuzzy sets will be introduced in Sect. 6.5.

Dependent variables in this simulation are employment levels for traders and sellers, the subjectively perceived pressure from cheating, as well as employment levels for both groups. In addition, we take note of metrics that are particular to the applied decision-making mechanisms. For our simulation, the choice of the cheater memory discount factor (γ) plays a critical role, since the traders' sanction boundaries are directly derived from memory about past interactions. We executed the simulation for each parameter setup (15 times, initially for 1000 rounds but later reduced to 500 rounds as simulations stabilize within this number of rounds) on Intel Quad-core desktop computers (2.66 GHz, 4 GB RAM) running Windows XP/7 and Java 1.7. The tables provided here show the mean and standard deviation value of these runs. The figures provided in the chapter are chosen from a simulation run that is in closest alignment with the mean values.

To show the range of possible simulation outcomes using the weighted majority rule, we concentrate on the manipulation of the memory discount factor γ_{forget}. We show employment figures for γ_{forget} values 0.96 (Fig. 6.5a), 0.97 (Fig. 6.5b) as well as 0.98 (Fig. 6.6a), and 0.99 (Fig. 6.6b) and discuss characteristics of the respective outcomes. All figures consist of four series, two of which represent the trader and seller employment levels. Along with those, the cheater pressure is represented by two complementary series. Although seemingly redundant, the symmetry of those graphs support their visual differentiation from the other (employment-related) data series.

[7] For this reason, commissions that sellers can achieve in the system have been set to fixed values in order to encourage maximum employment levels.

Table 6.1 Simulation parameters

Parameter	Value
Number of traders	150
Number of sellers	200
Maximum number of employers per seller	2
Maximum number of sellers per employer	5
Quota of cheating sellers	0.3
Probability of cheating	random value between 0.2 and 0.8 (initialized on per-seller basis)
Fraction of trade withheld by cheater	random value between 0.3 and 0.9 (determined on per-trade basis)
Cheater memory discount factor (γ_{forget})	0.96–0.99
Discount factor for highest cheating extent ($\gamma_{mostNegative}$)	0.99
Fraction of transactions with suspected cheating	0.05
Threshold for discarding memory entries ($\theta_{discard}$)	0.01
Goods value	200
Probability that selling goods results in profit	0.8
Maximum win/loss from market transaction	0.3

If initializing γ_{forget} with 0.96, traders converge on the categorization of cheating behavior as weak, and thus on the weakest sanction, after around 44 rounds. This is caused by their relatively high degree of forgetting. As a consequence, the majority of economically pressured traders opts for the weakest sanction ("Warning"). Given that the rule is collectively adopted, traders that actually employ cheaters have no harsher means of sanctioning that would allow them to recover their losses or exclude deviants from further transactions. This presents the worst possible outcome, in as much as it results in an economic decline caused by the inability of most traders to sustain a sufficient level of income that allows continued employment of sellers. This effect can be clearly observed by the sudden drops in seller employment caused by traders that let off all sellers (potentially up to five at a time) upon exiting the market. Consequently, the cheater pressure declines to a low level maintained by traders that employ a mix of cheaters and noncheaters. In sum, to model the Genoese case from an economic perspective, converging to a soft sanction is not desirable.

Looking at the baseline scenario with γ_{forget} at 0.97 (Fig. 6.5b), traders maintain a memory of negative trading experience for longer and agree on the recovery of loss as their unified reaction to cheater's behavior. Given that the recovery of loss is sufficient to sustain the traders' income (traders recover any loss from cheating from the cheater while keeping them in employment), no firing occurs, i.e., employment levels are high, and income levels are stable. However, cheaters never leave the system, so traders continue to perceive the emotional pressure by cheating, which is only reduced by forgetting.

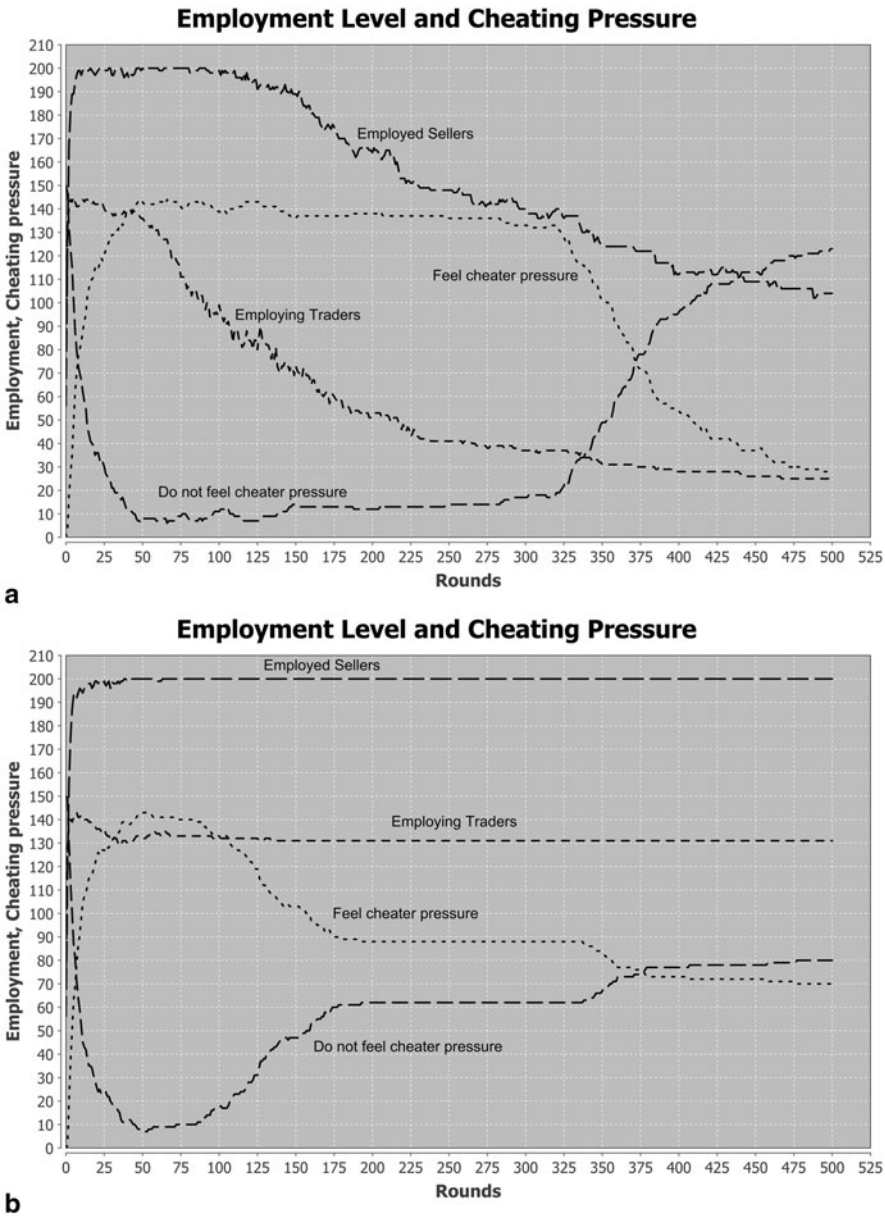


Fig. 6.5 Employment and cheater pressure (1/2). **a** Employment and cheater pressure for $\gamma_{forget} = 0.96$. **b** Employment and cheater pressure for $\gamma_{forget} = 0.97$

We extend our view to the third case, a γ_{forget} value of 0.98. Traders maintain an even longer memory about cheating experiences, which ultimately lets them converge to DISMISSAL as a general sanction. As a result of this sanction choice, cheater pressure levels off more rapidly.

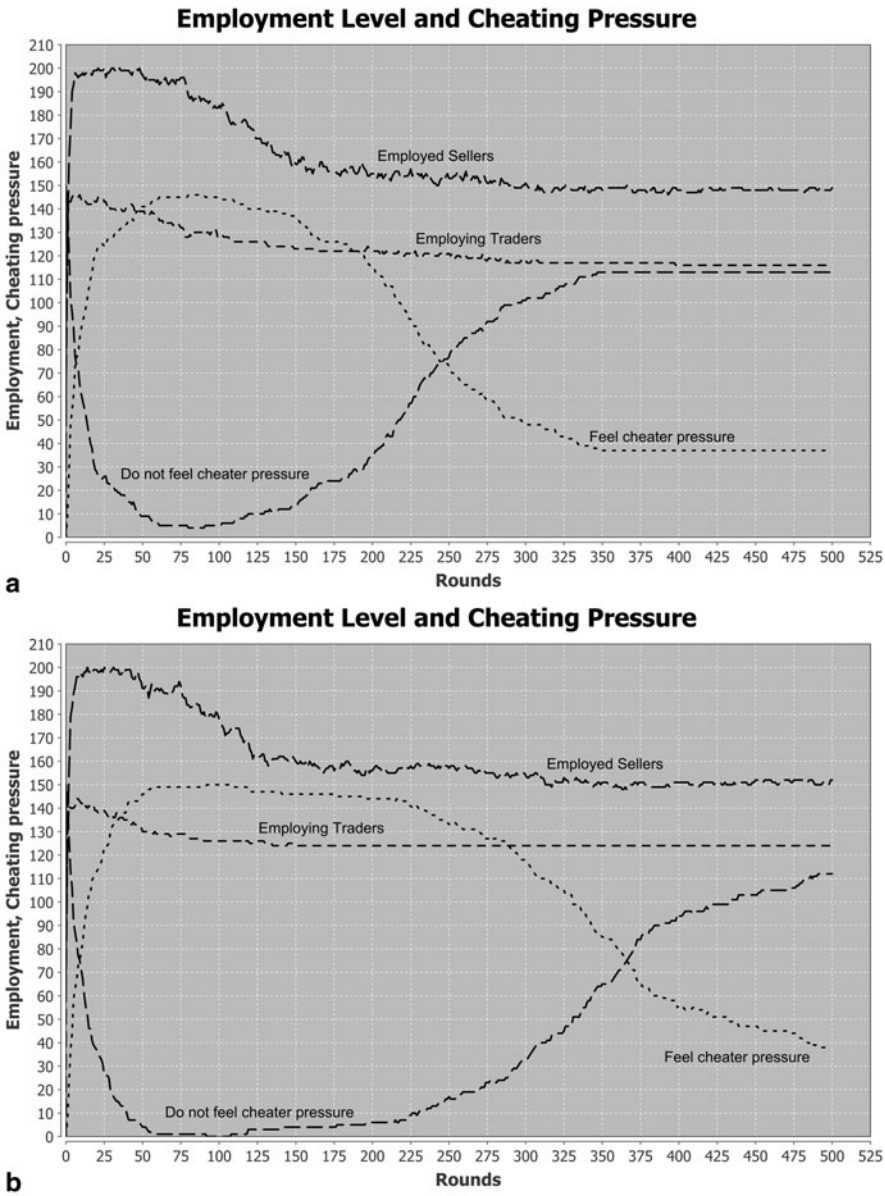


Fig. 6.6 Employment and cheater pressure (2/2). **a** Employment and cheater pressure for $\gamma_{forget} = 0.98$. **b** Employment and cheater pressure for $\gamma_{forget} = 0.99$

The last case, with γ_{forget} set to 0.99, equally helps drive traders toward agreement on the harshest sanction, which is the firing of individuals and public announcement to prevent (or at least delay) re-employment. Given their better memory, they

consequently apply this sanction in the first place, which leads to rapid adjustment of employment levels. The effect of this sanction is an employment level of sellers slightly above 150.

Table 6.2 provides us with further metrics that describe when the majority rule condition (described at the beginning of this subsection) of this voting-based approach is met for the particular configurations, how many individuals (employing traders) participated in the agreement on a rule, but also how many of those individuals' opinions are directly represented in the codified rule.

An important characteristic for the evaluation of aggregation mechanisms is the representativeness, i.e., the fraction of individuals whose choice is directly represented in the outcome (e.g., all individuals that opted for DISMISSAL if this sanction was chosen). In all configurations, the quota of individuals whose opinion is directly represented, lies between 38 and 60 %, mostly leading to outcomes that represent submajority rule. Note the stronger the convergence the more unified the perceived cheater pressure is. For γ_{forget} at 0.97, we see a low representativeness. Many participants opted for the bordering alternative sanctions. For γ_{forget} at 0.99, we see a particularly strong advocation of a sanction as cheater pressure was dominantly perceived as "HIGH."

This current model has some clear limitations. Converging to a single rule seems unrealistic, given the largely varying extent of cheating and the available sanctions set to address this. The sensitivity of the simulation toward the γ_{forget} parameter (based on the largely differing simulation outcomes) suggest that a more differentiated selection of sanctions seems not only appropriate but necessary in the cases where sufficient detail information is not available.

6.5 Modeling Differentiated Institutional Rules with Interval T2FS

IT2FS offer attractive options for representing the negotiation of a differentiated categorization and sanctioning of cheaters. Given that traders' sanction choices are driven by their individual experiences with cheaters (see Algorithm 1), they develop an independent understanding of cheating categories. Instead of having individuals put forth their preferred sanction choices (their classification of situational cheater pressure, as done in our initial model), IT2FS enable individuals to provide the *boundaries (intervals) for all cheating categories*. From a functional perspective in this context, the use of IT2FS can be interpreted in terms of a concurrent negotiation of a shared understanding on positions, which is ultimately used as a model for subsequent decision-making processes. Given the universal domain of applicability along with the potential anonymity of "voters" and a neutral treatment of all decision options, the use of fuzzy sets complies with characteristics for generalizable aggregation rules in the context of social choice theory [39]. In contrast to the majority-based approach, however, opinions can be excluded by statistical means. Second, from the standpoint of computation, we require a notion of global observer

Table 6.2 Established rules in the majority-based rule formation scenario

γ_{forget}	Round of rule establishment	σ	Individuals participating in rule establishment	σ	Individuals who are represented in rule	σ	Ratio of represented individuals	Chosen sanction
0.96	44	4.55	69	9.95	36	6.21	0.522	WARNING
0.97	43	11.07	74	10.19	28	5.67	0.378	LOSS_RECOVERY
0.98	46	11.78	78	7.66	37	4.90	0.474	DISMISSAL
0.99	46	22.78	70	5.97	41	5.69	0.586	DISMISSAL

that collects and integrates the different opinions. In majority-based approaches, on the other hand, it can be organized in a decentralized fashion.

Nevertheless, given the intent to include diverse opinions within statistical boundaries, IT2FS may offer a better approximation of decision outcomes than majority-based models. Given the suppression of information of the unrepresented fraction of a quorum, a majority-based voting model can react very sensitively to minor input adjustments and is prone to display distinct tipping points (as shown for the memory discount factor in Sect. 6.4.2), at least as long as it relies on the conventional use of crisp votes[8]. As we show below, fuzzy modeling may be a better approach to avoid high sensitivity to adjusted input parameters, especially for cases in which precise information is not available. Moreover, as mentioned in the previous paragraph, inclusion of individual opinions in this scheme is not limited to identifying the sanction with maximum support as in the majority-based approach. Instead inclusion is determined by the statistical similarity to others' opinions about the category boundaries of the input set "cheating."

In order to extend the simulation with fuzzy logic functionality, we ported Mendel and Liu's membership function generator (whose algorithm is described in the context of their application of IT2FS to *Computing with Words* [36]) to Java, which is the implementation language of our simulation framework. Membership functions generated from individual intervals are then used to generate an interval type-2 fuzzy logic system (IT2FLS) using a ported (and modified) version of Dongrui Wu's IT2FLS Processor [57][9]. Both components have been integrated, modified, and extended with evaluation utilities and visualization capabilities to provide mechanisms for analyzing the creation of membership functions and the exploration of generated IT2FLS instances. Note that we interpret an IT2FLS instance as a IT2FLS that has been generated at a given point in time, incorporating the collective membership functions and rule set valid at that time. The architecture of our IT2FLS Module is depicted in Fig. 6.7. We briefly describe the components as follows.

Upon reaching into close social proximity, agents send their individual intervals (boundaries of cheater pressure) for each cheating category into the interval preprocessor, which performs a basic validation of numeric values, and organizes intervals by input set (here, we only use a single input set), and respective categories (here, cheating categories HIGH, MEDIUM, LOW). Beyond this, the preprocessor ensures that only inputs from the current simulation step are processed (i.e., it clears and rebuilds the input sets during each step), and prevents multiple inputs by individual agents. Once all intervals for a given simulation step have been added, the collected input intervals are used to generate a nine point membership function[10] for each input interval set. Generated MFs can be inspected using a visualizer extension (see Fig. 6.8 for an example).

---

[8] See Li et al. [35] for an approach to model fuzzy majority votes.

[9] In both cases, the original code is written in MATLAB.

[10] In IT2FS, a membership function is described by a nine point vector that describes its shape in a two-dimensional representation with the input assigned to the domain axis and upper and lower boundaries of the function assigned to the range axis [57].

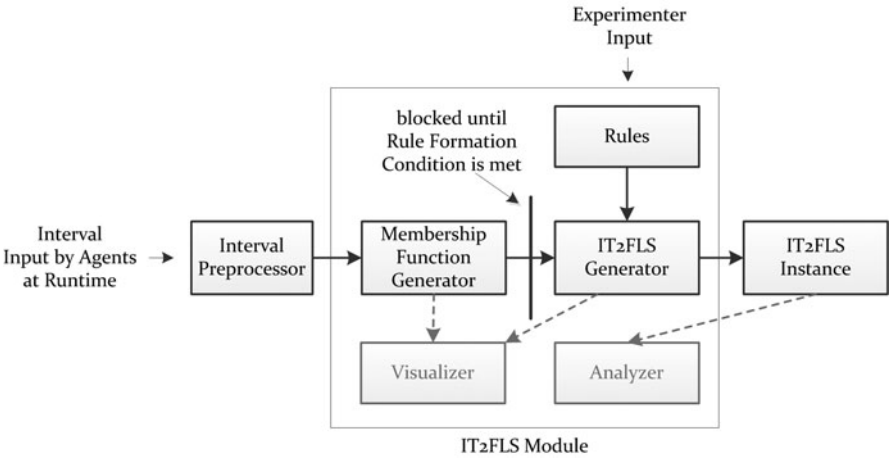


Fig. 6.7 IT2FLS module

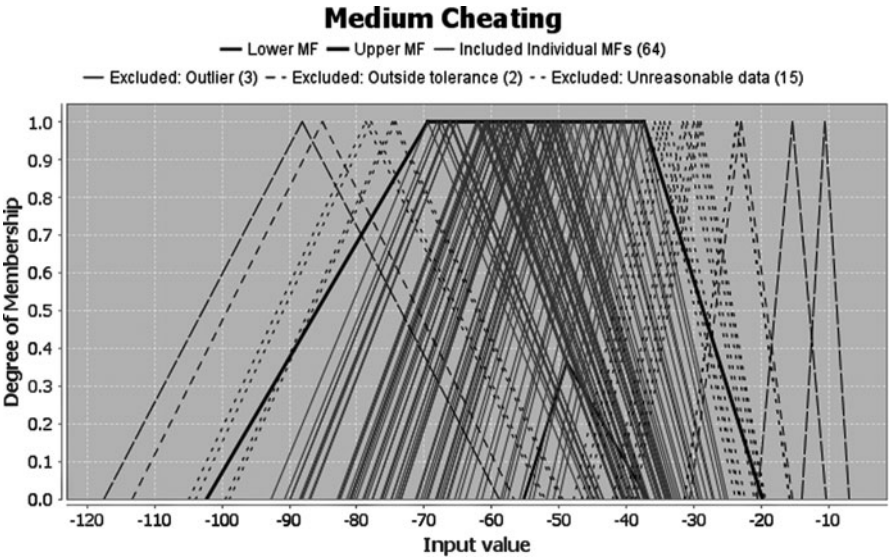


Fig. 6.8 Example of generated nine-point membership function

Liu and Mendel's MF generation algorithm is particularly useful in our context, since it introduces extensive data preprocessing to support the generation of robust membership functions. Given that their work is based on questionnaires of human subjects, the proposed preprocessing steps seem equally applicable to data generated by an artificial entity (that builds on the "human" metaphor). Without exploring

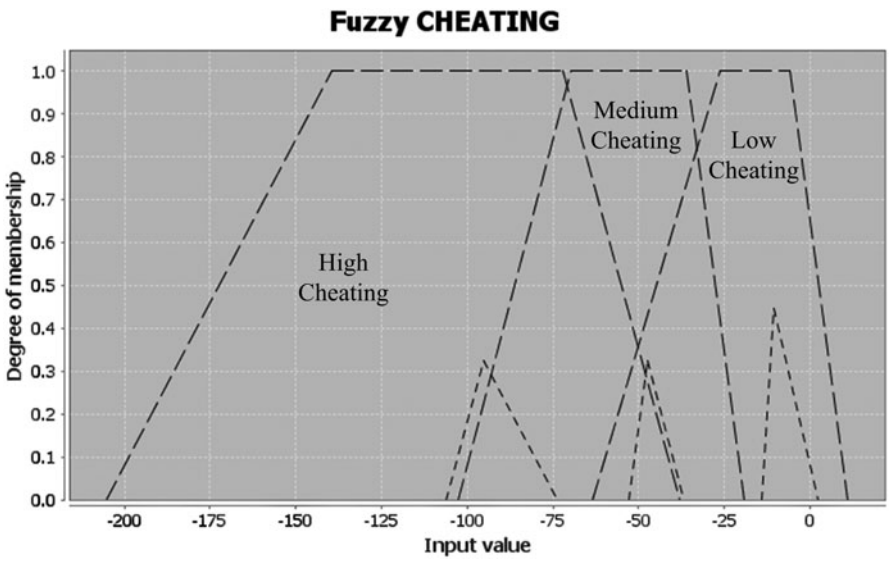


Fig. 6.9 Example of membership functions to categorize cheating levels derived from collective trader input

the entire MF generation in detail[11], Liu and Mendel's approach [36] eliminates useless results (*bad data*), e.g., caused by inverted values for upper and lower interval boundaries, and data that are unreasonably high or low, i.e., outside 1.5 * Interquartile_Range (*outliers*). As further steps, data outside a confidence boundary of 95 % of the distribution of the collected intervals for a given set are excluded (*outside tolerance limits*). Finally, their approach eliminates data that is not sufficiently overlapping with other intervals (*unreasonable data*). The MF shown in Fig. 6.8 is an extreme example for an MF generated from agent interval inputs. The visualization also provides information on intervals that have been excluded during any data preprocessing steps (annotated with the reason for exclusion). In contrast to input generated by human subjects, the production of nonplausible "bad data" can be controlled by design. However, all other processing steps that rely on statistical measures for exclusion equally apply to data produced by artificial entities. In the figure, the upper MF and lower MF are plotted with a strong stroke. Intervals that have been included in the generation appear dark gray (and are by principle of MF generation framed by the upper MF). The remaining intervals have line patterns that indicate the reason for elimination (see the legend in Fig. 6.8) with the outer left and right ones being outliers, followed by intervals that are outside the tolerance zone, and ultimately intervals that are considered "unreasonable data" according to the last condition in the data preprocessing step referred to before.

[11] For full details, please refer to [36].

Observing the potentially extensive exclusion of data, the role of data preprocessing in generating meaningful MFs becomes clear. This is especially important for larger numbers of input intervals, an aspect that is relevant for agent-based simulations that operate with a large numbers of agents.

Calculating the MFs for all inputs provides a complete picture of potential MFs, an example of which is given in Fig. 6.9 to show the characteristic functions produced by the system based on the determination of sanctions by individuals (see Algorithm 1). The deviation among input intervals is directly reflected in the size of the *footprint of uncertainty* of the respective MF, which offers a high inclusion of different opinions but naturally trades this for precision. The mentioned data preprocessing steps provide a means to adjust this trade-off into either direction, which is an aspect beyond the scope of this chapter.

Note the widely varying input on the left-most membership function (*high cheating*) in Fig. 6.9. This is ultimately caused by the randomized cheating behavior of sellers and is amplified by the cheater pressure algorithm (Algorithm 1), which has a low threshold for the categorization of cheater perception as "HIGH."

In order to put established rule sets into action using a generated IT2FLS instance, we require the specification of rules along with consequences that are associated with evaluated degrees of membership of input values with different fuzzy sets. In cases that operate on multiple input sets (e.g., cheating, employment status), a rule must be specified for each input set-fuzzy set combination.

In our current example, which at this stage, works with a single input set, the rule consequences are directly associated with the input value's (k) membership in an associated fuzzy set. In IT2FSs, the rule consequent is an interval of upper and lower boundary. For our rule consequences, we use crisp values and thus set upper and lower boundary to the same value. This rule consequent directly maps onto a given sanction (highlighted in parentheses):

IF k is HIGH_CHEATING, THEN: [0.0, 0.0] (DISMISSAL)
IF k is MEDIUM_CHEATING, THEN: [0.5, 0.5] (RECOVER_LOSS)
IF k is LOW_CHEATING, THEN: [1.0, 1.0] (WARNING)

Based on this information, i.e., all generated type-2 membership functions and rules, the IT2FLS module (see Fig. 6.7) can create an IT2FLS instance.

To operate the IT2FLS, agents invoke it using their current cheater pressure and expect a return value between 0 and 1, which they classify and associate with sanctions. Values in the range from $0 <= x < 1/3$ are interpreted as DISMISSAL (i.e., firing of cheaters), $1/3 <= x <= 2/3$ as RECOVER_LOSS, and the remainder $2/3 < x <= 1$ as WARNING.

In our simulation, the generation and operation of IT2FLSs is integrated into the overall simulation architecture and triggered by a specified rule formation condition (see Sect. 6.4.1). Derived from the consensus-based rule formation condition, codified rules (that traders are bound to) are only established once the *majority of employing traders participate in the collective decision process* (i.e., feed their interval sets into the preprocessor). The resulting IT2FLS instance is then fixed and is invoked by traders to react to any future cheating. In order to address the fact that the range of the input values is unknown beforehand and can also change after the

establishment of a rule, the IT2FLS implementation for this simulation treats left- and right-most MFs as left and right shoulders, even if boundaries are not specified. This way the range of input values for the ordinally scaled cheating categories can vary while ensuring that they are still captured by the left or right outer MF.

6.5.1 Results

Running the simulation with the fuzzy set extension, with the cheater memory discount factor γ_{forget} set to 0.96, 0.97, 0.98, and 0.99, we receive the results shown in Figs. 6.10a, 6.11a, 6.12a, and 6.13a. To ease the comparison with previously shown results for rule choice based on majority vote, the previous results are provided just below the respective outcomes of their IT2FLS equivalent (Figs. 6.10b, 6.11b, 6.12b, and 6.13b). The representation of all generated membership functions for this simulation setup (which is, given the unchanged individual categorization of cheating, constant across all simulation runs) is the one shown in Fig. 6.9.

It can be immediately observed that the sensitivity of the simulation to γ_{forget} for all parameter choices is reduced. This result is afforded by the differentiated sanction set with which traders can operate. The differences in simulation outcomes across γ_{forget} values are essentially reduced to the speed of convergence and comparatively minor shifts in employment levels.

For $\gamma_{forget} = 0.96$ (Fig. 6.10a), we can observe the slow convergence of seller employment toward 145 (employed sellers) after about 450 rounds, a level at which most cheaters have been identified and excluded from further transactions. Trader employment levels remain around 110, which is in stark contrast to the continuous drop in employment levels for the previously described voting approach (Fig. 6.10b). The perceived cheater pressure is neutralized after about 160 rounds, which is facilitated by the relatively high forgetfulness of traders.

For $\gamma_{forget} = 0.97$ (Fig. 6.11a), the fuzzy selection of sanctions leads to very similar employment levels as for γ_{forget} at 0.96. The central difference is the difference in convergence time which is caused by the stronger influence of cheater memory that drives the establishment of common rules. As shown earlier for the nonfuzzy approach (Fig. 6.11b), traders agree on the second-harshest sanction, which maintains very high employment levels for both traders and sellers at the cost of continued pressure caused by cheating (see Sect. 6.4.2).

The effect of preference aggregation using IT2FS for γ_{forget} at 0.98 (Fig. 6.12a) follows a similar pattern as for 0.97 and converges roughly in the same fashion. For this setup, we observe an effect that is not directly visible from the employment levels. Employment levels do not decline in the gradual fashion as for previous values, but produce a step pattern until they eventually reach a stable plateau. This pattern directly reflects the situationally preferred sanction. After establishment of the fuzzy rule set, most sellers respond to their high cheating pressure by dismissing cheaters which, once the pressure starts to decline, is replaced by the recovery of loss as a dominantly applied sanction. This pattern finally subsides, and traders fall back to a mixed application of sanctions. Cheating levels ultimately balance at around 235 rounds.

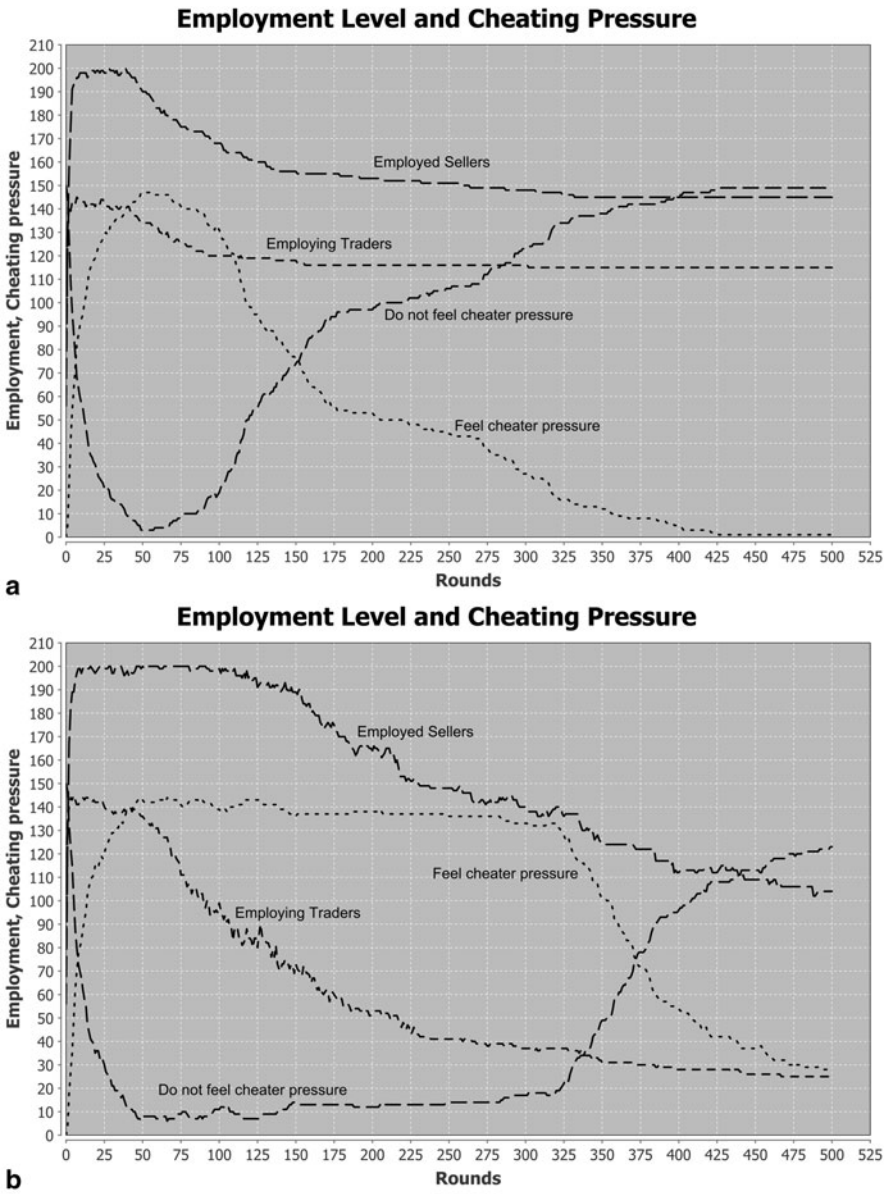


Fig. 6.10 Employment and cheating levels in fuzzy and nonfuzzy rule formation. **a** Fuzzy rule formation for $\gamma_{forget} = 0.96$. **b** Nonfuzzy rule formation for $\gamma_{forget} = 0.96$

For γ_{forget} at 0.99 (Fig. 6.13a), we can observe a similar pattern as for the value 0.98. The core difference is a more consistent application of the DISMISSAL sanction, which leads to an initial steep decline in employment levels that bottoms out just

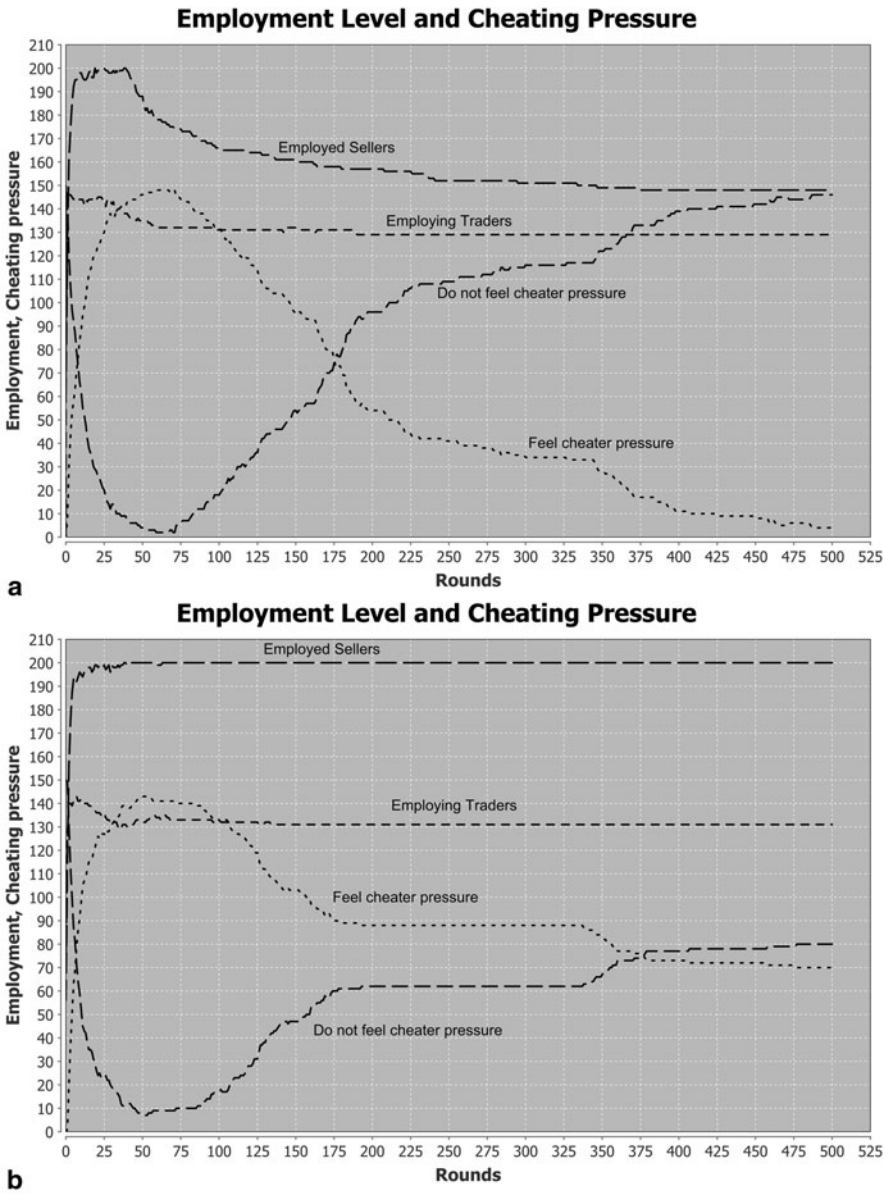


Fig. 6.11 Employment and cheating levels in fuzzy and nonfuzzy rule formation. **a** Fuzzy rule formation for $\gamma_{forget} = 0.97$. **b** Nonfuzzy rule formation for $\gamma_{forget} = 0.97$

below 150 employed sellers after 180 rounds. This leads to a slightly better outcome for the fuzzy version as compared to the nonfuzzy one, since it manages to reduce the cheater pressure after around 335 rounds in comparison to 380 for the nonfuzzy version (Fig. 6.13b).

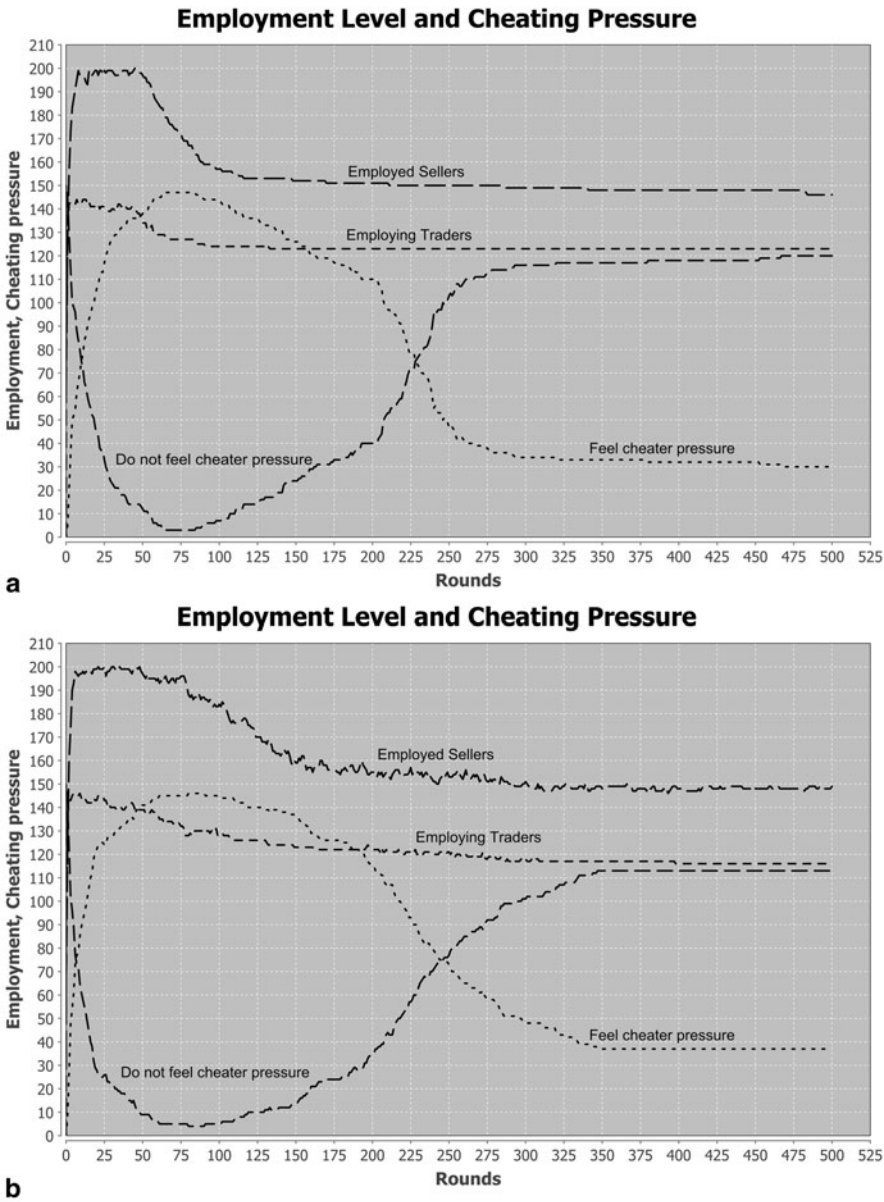


Fig. 6.12 Employment and cheating levels in fuzzy and nonfuzzy rule formation. **a** Fuzzy rule formation for $\gamma_{forget} = 0.98$. **b** Nonfuzzy rule formation for $\gamma_{forget} = 0.98$

Differences in employment levels and rule efficiency are reflected by various measures of rule establishment, such as round of rule establishment (before that point, traders carry out sanctions according to their own preferences) and numbers of participating traders (higher numbers of traders lead to more consistent rule application).

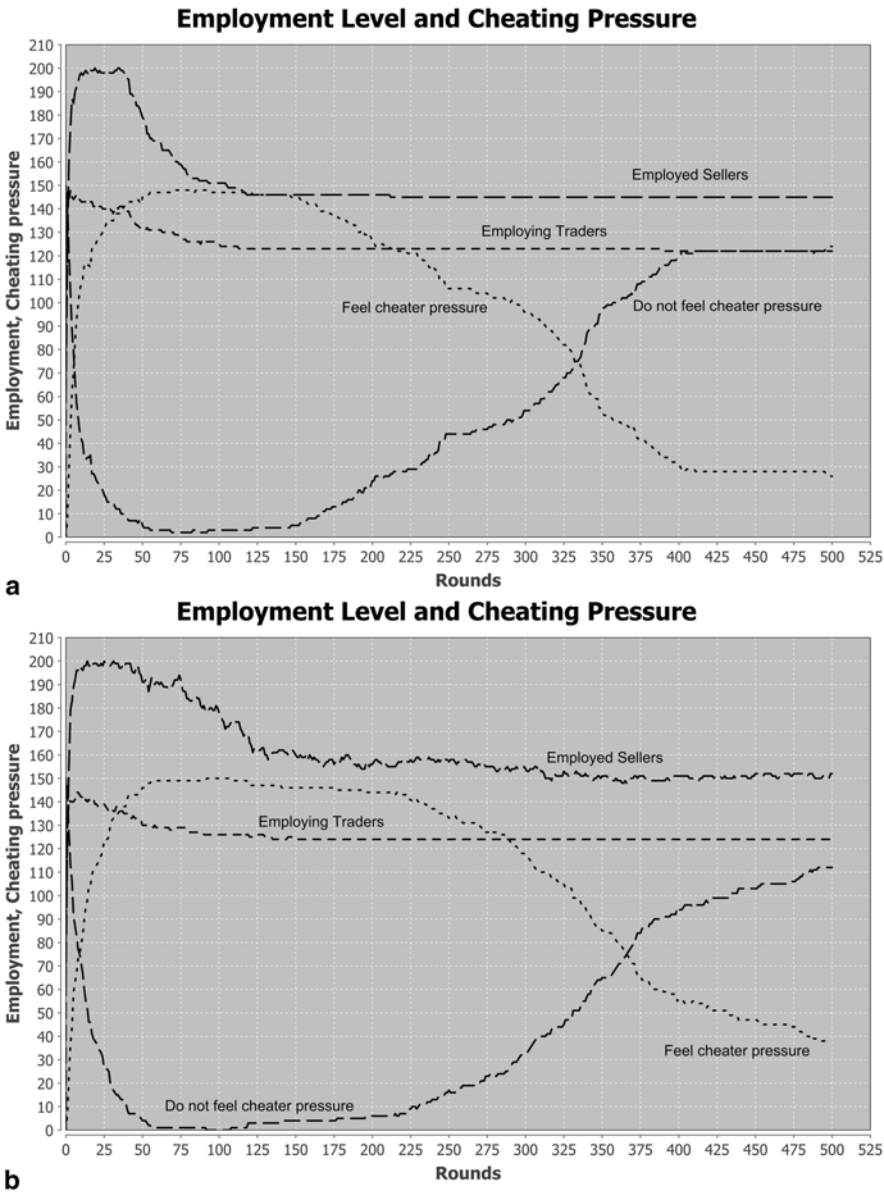


Fig. 6.13 Employment and cheating levels in fuzzy and nonfuzzy rule formation. **a** Fuzzy rule formation for $\gamma_{forget} = 0.99$. **b** Nonfuzzy rule formation for $\gamma_{forget} = 0.99$

We will thus have a look at various metrics of the fuzzy preference aggregation approach as shown in Table 6.3 and compare them to the voting-based approach (Table 6.2).

Table 6.3 Established rules using IT2FS-based preference aggregation

γ_{forget}	Round of rule establishment	σ	Individuals participating in rule establishment	σ	Individuals who are represented in rule	σ	Ratio of represented individuals
0.96	46	3.72	79	6.62	72	7.29	0.911
0.97	42	2.47	81	5.85	76	6.91	0.938
0.98	44	3.30	78	8.37	71	7.59	0.910
0.99	49	4.04	95	10.16	84	11.21	0.884

As a summary, in contrast to the voting-based decision making, the fuzzy approach using preference aggregation bears a set of differences. It allows the definition of a set of sanctions which traders converge on, instead of focusing on a fixed sanction or complex means to enable differentiated mechanisms of argumentation to establish a sanction set. A key characteristic of preference aggregation is the high level of inclusiveness and thus representation. Exclusion of individuals from aggregation only occurs based on statistical evaluation of the distribution (see Sect. 6.5). In general about 90 % of all opinions are considered and thus represented in the rule outcome. By nature, (nonqualified) majority-based voting yields lower levels of representation. Our results for that scenario range between 38 % to just above 60 % (see Table 6.2).

However, a central observation is that the introduction of fuzzy sets considerably reduces the sensitivity of the simulation to a specific parameter, especially in contrast to mechanisms with strong tipping point, such as majority-based voting. This may not necessarily be wrong per se, but for cases in which detailed information is lacking, this may lead to high parametric sensitivity, which can be spurious and misleading. This was evident from the nonfuzzy (voting) outcome for $\gamma_{forget} = 0.96$ and 0.97.

Our application of fuzzy logic up to this stage is still relatively rudimentary as we only focus on a single input set, the extent of cheating. However, a trader's sanctioning behavior may not only depend on his perception of cheating extent but, as a further influence factor, may depend on his affluence. As shown in Fig. 6.12a (fuzzy setup with $\gamma_{forget} = 0.98$), traders' behavior showed linearity with regard to sanction cheating categorization by transitioning from dismissing to the recovery of loss. We, therefore, proceed to incorporate traders' wealth information into the sanction choice to represent a more differentiated behavior.

6.5.2 Adding Wealth as Input Set for Differentiated Rule Sets

To make traders' sanctioning behavior not only dependent on perceived cheating pressure, we add the perception of wealth as a second dimension which is a starting point for the consideration of further (here, demographic) factors.

In order to include wealth self-perception into the gradual sanctioning of cheaters, traders not only provide their intervals for different cheating categories but likewise intervals for wealth categories to the interval preprocessor (see Fig. 6.7). Similar to the cheating categories, the boundaries for wealth levels are derived based on individual experience. The rules to derive categories are described in Algorithm 2. If the evaluated wealth level is within the first quartile of the range of experienced wealth, wealth is considered low. Wealth values ranging within the second quartiles indicate medium wealth. Higher values are considered to indicate high wealth.

Algorithm 2 – Wealth Level Algorithm

```
if wealth < wealth_min + ((wealth_max - wealth_min) / 4) then
    consider as LOW_WEALTH
else
    if wealth < wealth_min + ((wealth_max - wealth_min) / 4) * 2 then
        consider as MEDIUM_WEALTH
    else
        consider as HIGH_WEALTH
    end if
end if
```

We thereby introduce the assumption that the harshness of a trader's reaction is negatively correlated with his perceived wealth status. To integrate the new input set, we need to adapt the rule specification for the IT2FLS generation by assigning consequents for all possible input set combinations:

IF k is HIGH_CHEATING AND LOW_WEALTH, THEN: [0.0, 0.0] (DISMISSAL)
IF k is HIGH_CHEATING AND MEDIUM_WEALTH, THEN: [0.0, 0.0] (DISMISSAL)
IF k is HIGH_CHEATING AND HIGH_WEALTH, THEN: [0.0, 0.0] (DISMISSAL)

IF k is MEDIUM_CHEATING AND LOW_WEALTH, THEN: [0.0, 0.0] (DISMISSAL)
IF k is MEDIUM_CHEATING AND MEDIUM_WEALTH, THEN: [0.0, 0.0] (DISMISSAL)
IF k is MEDIUM_CHEATING AND HIGH_WEALTH, THEN: [0.5, 0.5] (LOSS_RECOVERY)

IF k is LOW_CHEATING AND LOW_WEALTH, THEN: [0.0, 0.0] (DISMISSAL)
IF k is LOW_CHEATING AND MEDIUM_WEALTH, THEN: [0.5, 0.5] (LOSS_RECOVERY)
IF k is LOW_CHEATING AND HIGH_WEALTH, THEN: [1.0, 1.0] (WARNING)

With increasing wealth, traders thus react more leniently when facing deviant behavior. The consideration of wealth for rule suggestions has equally been integrated into the nonfuzzy majority-based voting approach to provide direct comparison. The generated membership functions for the categorization of wealth are shown in Fig. 6.14.

In contrast to cheating (see Fig. 6.9), in this conception, wealth is generally of positive value, thus ordered from low to high wealth level. Another difference to cheating is that wealth self-perception is much more certain (as visible by the relatively small FOU), as it is only influenced by extreme wealth changes that extend the boundaries of wealth categories (see Algorithm 2), which hardly occur based on an individual transaction, but stabilize over the run of the simulation. The perception of cheating, in contrast, can be strongly influenced by an individual transaction with a strong cheater. To compare the impact of introducing wealth into sanction considerations, we pair figures showing the fuzzy approach along with the nonfuzzy

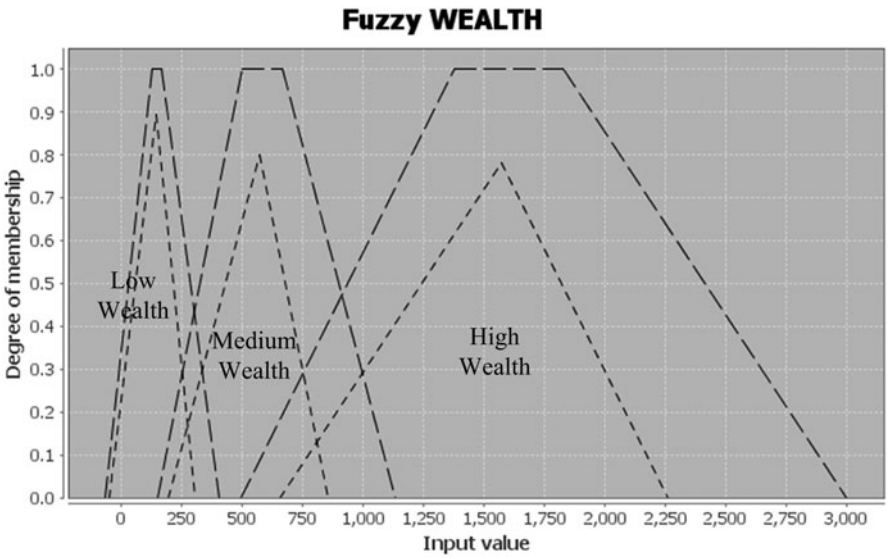


Fig. 6.14 Example of membership functions to categorize cheating levels as well as wealth levels derived from collective trader input

approach to extend our argument of the smoothening impact of integrating fuzziness into our rule formation system.

For the nonfuzzy version with $\gamma_{forget} = 0.96$, incorporating wealth (Fig. 6.15b), we see similar behavior as in the initial nonfuzzy setup (Fig. 6.10b). Employment levels surge because of the convergence to a weak sanction. Employment levels, in fact, drop faster in the wealth-including version as individuals that consider themselves well off act more leniently, even in the case of strong cheating. We see a similar pattern for the fuzzy version (Fig. 6.15a) where lenient sanctioning effects more rapidly dropping employment levels of traders.

The nonfuzzy version of that model, which includes wealth for γ_{forget} of 0.97, is a good example of the significant impact the inclusion of wealth can have on the convergence behavior for a majority-based voting system. Integrating wealth into the decision-making for the nonfuzzy setup, traders converge toward the weakest sanction, which is in stark contrast to the nonfuzzy setup that does not consider wealth (Fig. 6.11b) and in which traders concentrate on the recovery of losses. This is reflected in the relatively poor representativeness (compared to all other values for γ_{forget}) of the resulting sanction (see Table 6.4) for the overall opinion. We can see that the overall majority opts for the two harsher sanctions; only a submajority of around 48 % defines the rule (compared to 65 % for γ_{forget} at 0.96 for the same sanction). In this context, $\gamma_{forget} = 0.97$ is just below the tipping point toward a harsher sanction.

For γ_{forget} at 0.98 and 0.99, we can see the sanction shift, which is relatively extreme for the wealth-enhanced nonfuzzy approach and directly shifts to the harshest sanction (DISMISSAL) (see Table 6.4). The patterns of those two graph pairs are comparable in that they show similar outcomes. The fuzzy approaches show

Table 6.4 Established rules in the majority-based rule formation scenario (including wealth)

γ_{forget}	Round of rule establishment	σ	Individuals participating in rule establishment	σ	Individuals who are represented in rule	σ	Ratio of represented individuals	Chosen sanction
0.96	40	3.10	68	4.87	44	5.09	0.647	WARNING
0.97	40	2.99	71	4.08	34	2.69	0.478	WARNING
0.98	37	13.65	68	8.69	48	5.51	0.706	DISMISSAL
0.99	41	11.90	71	6.32	58	6.19	0.817	DISMISSAL

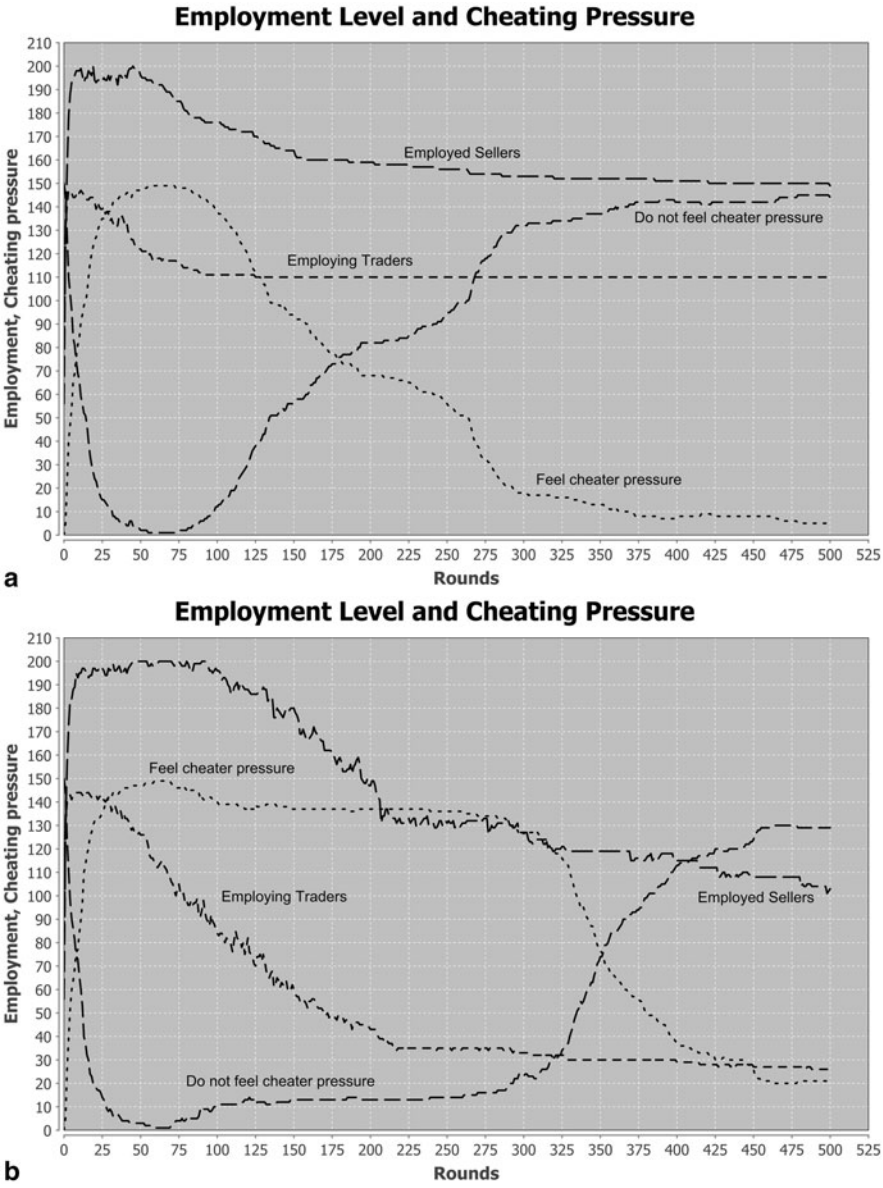


Fig. 6.15 Employment and cheating levels in fuzzy and nonfuzzy rule formation. **a** Fuzzy rule formation (using cheating and wealth as input sets) for $\gamma_{forget} = 0.96$. **b** Nonfuzzy rule formation (using cheating and wealth as input sets) for $\gamma_{forget} = 0.96$

smoothened results because of their differentiated sanction application. However, both approaches ultimately arrive at the same employment levels. The only difference for γ_{forget} at 0.99 (compared to γ_{forget} at 0.98) is the slightly lower employment rate for sellers as of the harsher cheating reinforcement by lower forgetfulness.

Table 6.5 Established rules using IT2FS-based preference aggregation (including wealth)

γ_{forget}	Round of rule establishment	σ	Individuals participating in rule establishment	σ	Individuals who are represented in rule	σ	Ratio of represented individuals
0.96	37	2.27	79	5.73	71	7.32	0.898
0.97	36	5.19	80	11.40	76	12.29	0.950
0.98	37	2.21	68	9.53	65	8.26	0.956
0.99	44	7.79	92	12.43	81	14.45	0.880

Compared to the fuzzy approaches without consideration of wealth (see e.g., Fig. 6.13a in comparison to Fig. 6.18a), the wealth inclusion effects more moderate sanctioning behavior. This is ultimately to the disadvantage of traders; their employment levels shrink. Table 6.5 shows the representativeness of wealth-incorporating decision outcomes, which have similar ranges as the previous simulation sets that excluded wealth aspects (see Table 6.3) while displaying slightly faster convergence.

6.6 Discussion and Conclusions

To this stage, we have shown a set of experiments that compare a majority-based voting approach with preference aggregation using IT2FSs. Although for the given scenario of rule generation, the majority-based approach is more aligned with the scenario's background, we suggest that the alternative approach using fuzzy sets might in fact be more useful to model this case. For simulations for which consistent knowledge at a given abstraction level is not available, choosing alternative mechanisms on a higher abstraction level and the reduced precision might in fact be more suitable to model subtle dynamics of social simulations which would otherwise be lost (see the sanction choice shift in the wealth-enhanced version between $\gamma_{forget} = 0.97$ and 0.98 (Figs. 6.16a and 6.17a) for example). Beyond that, introducing fuzziness in such cases makes simulation behavior less dependent on the experimenter's specifications, which is a core criticism of the social simulation approach in general [23]. The experiments shown in this work provided a clear account of a simulation scenario that was strongly sensitive to a single parameter which was ultimately rooted in the choice of a seemingly correct decision-making mechanism (majority-based voting). Knowledge about this in combination with lacking detailed knowledge resulted a widely varying behavior (unstable to small parameter variation). In contrast, the use of preference aggregation provided more believable behavior along with a mechanism to enable differentiated sanction choices.

This chapter provides an instance that shows the strong potential of IT2FS to establish a comparatively simple aggregation of opinions into a fuzzy set. This approach is generalizable and could find application across different simulation scenarios. It also shows that fuzzy mechanisms can be applied as an alternative to conventional

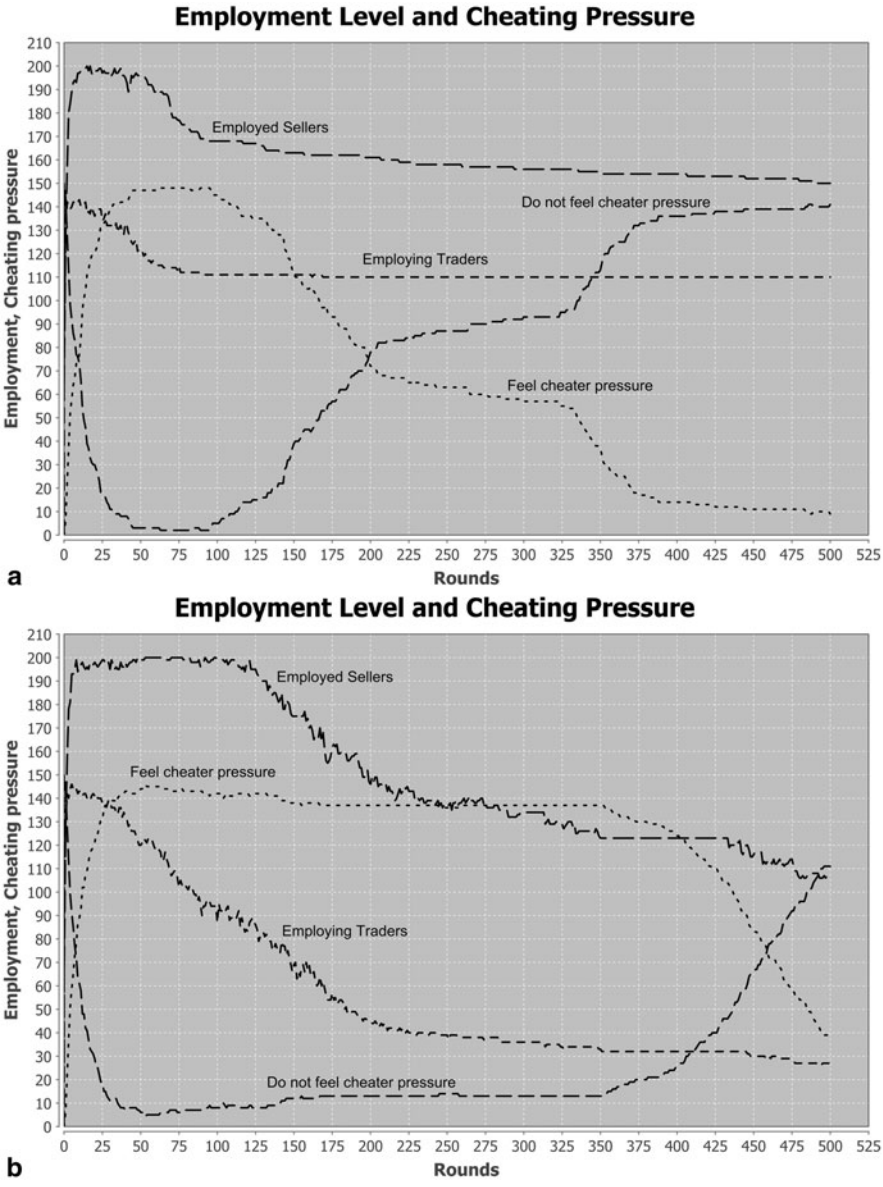


Fig. 6.16 Employment and cheating levels in fuzzy and nonfuzzy rule formation. **a** Fuzzy rule formation (using cheating and wealth as input sets) $\gamma_{forget} = 0.97$. **b** Nonfuzzy rule formation (using cheating and wealth as input sets) for $\gamma_{forget} = 0.97$

decision-making approaches, which can open exploration options from different angles and easily introduce new influence factors (in the shape of additional input sets and corresponding additional rule combinations as shown for the inclusion of wealth in addition to the consideration of cheating categories in Sect. 6.5.2).

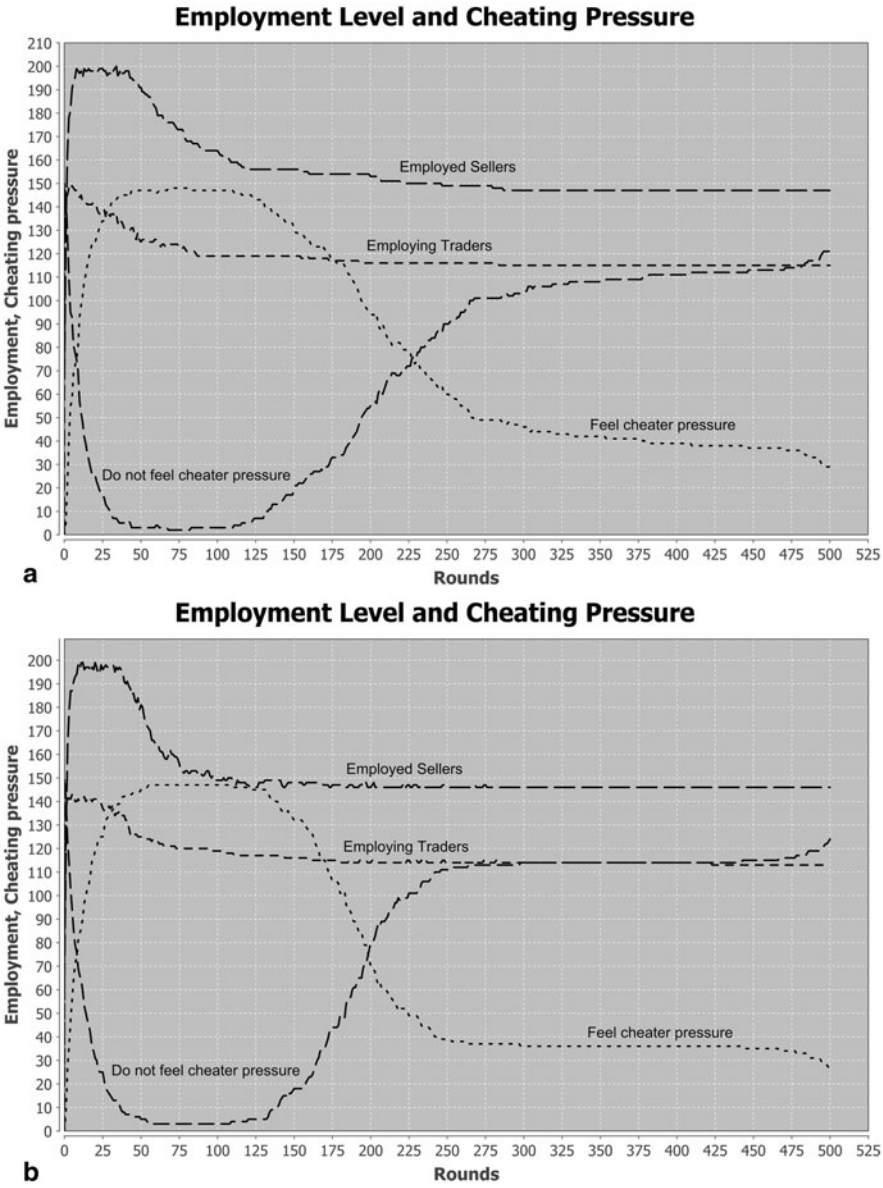


Fig. 6.17 Employment and cheating levels in fuzzy and nonfuzzy rule formation. **a** Fuzzy rule formation (using cheating and wealth as input sets) for $\gamma_{forget} = 0.98$. **b** Nonfuzzy rule formation (using cheating and wealth as input sets) for $\gamma_{forget} = 0.98$

The potential of IT2FS for social simulation lies in the degree to which they support the modeling of collective opinions. In addition, the specialization of *interval type-2* FS (in contrast to general type-2 FS) makes the creation of fuzzy logic systems more transparent to the modeler and eases their adoption. Type-1 fuzzy sets

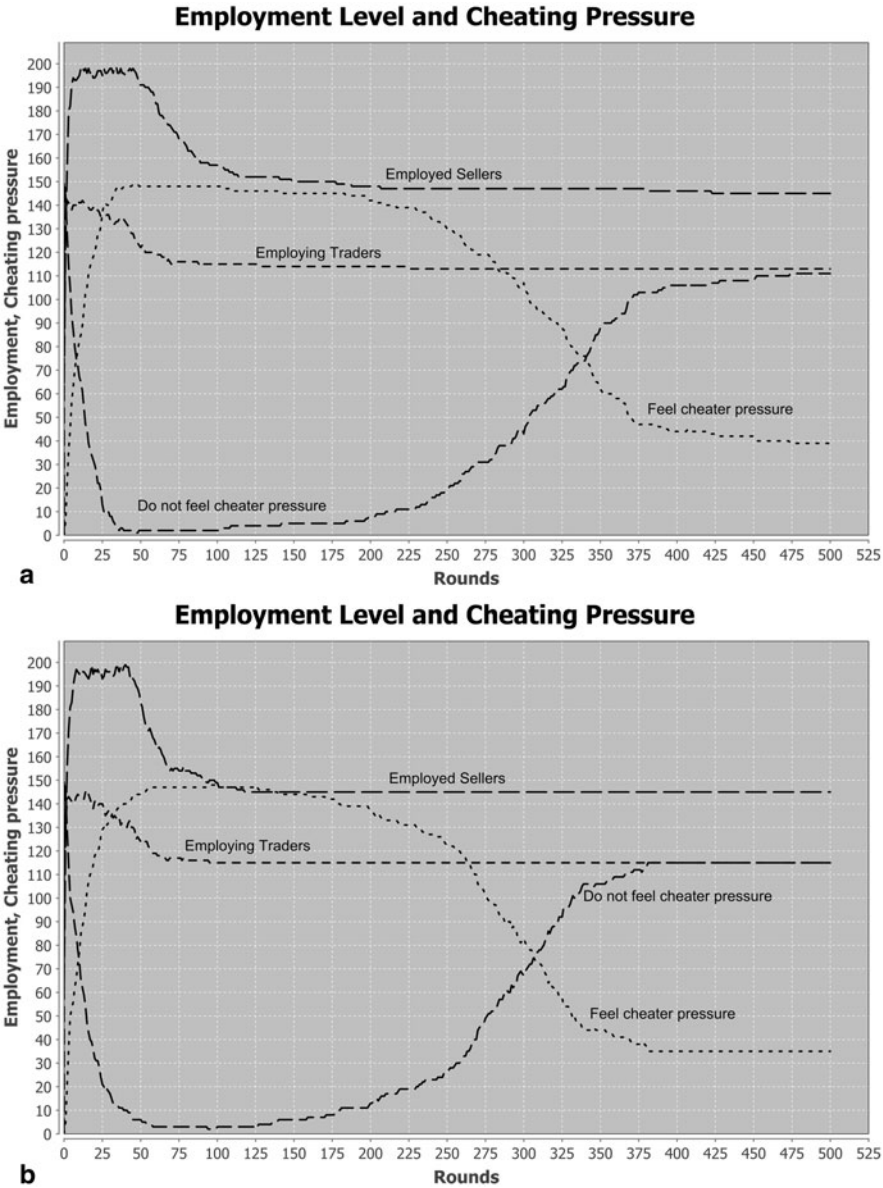


Fig. 6.18 Employment and cheating levels in fuzy and nonfuzzy rule formation. **a** Fuzzy rule formation (using cheating and wealth as input sets) for $\gamma_{forget} = 0.99$. **b** Nonfuzzy rule formation (using cheating and wealth as input sets) for $\gamma_{forget} = 0.99$

do not offer the potential of integrating individual opinions into a fuzzy set that bears uncertainty in itself. This particular characteristic, the second-order uncertainty, makes the adoption of fuzzy approaches an alternative to the inclusion of randomness (which modelers would otherwise incorporate by using random number

distributions or Monte Carlo Simulations in cases where they want to express uncertainty or vague knowledge about behavior they intend to simulate). The use of those approaches shows that uncertainty is often a necessary element for social simulations and thus requires an appropriate representation. However, this also implies that those simulations generally support a conceptual understanding of a particular social constellation, not precise specification or specific outcome, which is in line with Zadeh's own observation that precision is in inherent conflict with uncertainty [60] (see Sect. 6.2.1).

We note that our work is not isolated in its advocation of fuzzy sets. In the context of agent-based modeling and social simulations, fuzzy sets (of type 1) have found application for the modeling of personality traits [18, 49], trust [30] as well as social relationships [26, 27, 52]. An explicit comparison for the incorporation of crisp and fuzzy sets in the BDI architecture [17] is explored by Vu et al. [56] for decision making using the example of soccer penalties. Hassan et al. [26] further provided a powerful general argument for the adoption of fuzzy modeling in the area of social simulation. To date, IT2FS have hardly been applied to agent-based modeling of social phenomena, one rare exception being Márquez et al.'s [38] approach to model poverty levels for the city of Tijuana representing the demographics of individual areas as agents.

Our work extends the argument for the adoption of IT2FS for agent-based modeling in general, and institutional modeling in particular. We have proposed a mechanism to represent collective decision making by integrating individual microperspectives with a collective macroperspective based on configurable strategies, an application of IT2FS that we have not observed in the literature so far.

For our particular simulation discussed here, there are multiple future avenues to explore. Making the assumption that wealthier individuals are more lenient to cheating behavior, one could also adopt the opposing stance, claiming their behavior to be driven by an orthodox understanding of normative behavior. In contrast, leniency could apply for less affluent members of the societies which are more likely to be socialized in environments in which cheating thrives, thus being more "used to it," and thus more accepting or even expecting.

From a more conceptual perspective, a strong potential lies in the exploitation of T2FS to extend an agent's awareness. As individuals, we often experience situations that have clear normative or even institutionalized prescriptions, such as to wait in front of a red traffic light as a pedestrian. However, depending on the situation (which country are we in, if there are cars around, or other people, small children, etc.) we might decide differently whether or not to comply with this prescription. Thus, individuals have a situational understanding of the degree of freedom their actions underlie, which from a modelers perspective, can be interpreted as the footprint of uncertainty of a type-2 membership function, ranging between the extreme cases of strict compliance and ignorance of such rule. Thus, IT2FS can stand in a direct relationship to the modeling of collective perception, and in consequence, to the modeling of social systems in general. To achieve this, we plan to integrate IT2FS with the concept of dynamic deontics [15] that allows the representation of the dynamic and fluid nature of norms, in addition to the nADICO grammar [14]

that offers a general syntax to express an individual's or a collective's norm and rule understanding in a comprehensive manner. This would enable the modeler to provide agents with insight into the membership function details (or its abstraction) for particular norms, both on individual and collective level, thus, giving them awareness of the individual boundary levels as well as the boundaries of a generated membership function for a given fuzzy set (such as exemplified in Fig. 6.8). We think that this can directly aid the modeling of cultural and social aspects of societies and their understanding of rules, aspects which are otherwise inaccessible to individuals or require artificial entities for evaluation (e.g., impersonalization of normative authorities in artificial societies).

Complex systems and fuzzy sets share a commonality in their ability to incorporate plurality in behavior and opinion. Social simulation can rely on the formal backing of fuzzy logic and capitalize on capabilities fuzzy logic provides at this stage (and discussed in this section). An argument for the intellectual investment into the incorporation of fuzzy methods can also be supported by the active research that defines the ecosystem of fuzzy logic in general.

We believe that a more systematic consideration of fuzzy approaches in the social modeler's toolbox is a natural step in the progression of social modeling and simulation toward a more realistic representation of the social environment.

References

1. J.K. Arrow, *Social Choice and Individual Values*, 2nd edn. (Wiley, New York, 1963)
2. R. Axelrod, *The Evolution of Cooperation* (Basic Books, New York, 1984)
3. R. Axelrod, Advancing the art of simulation in the social sciences, in *Simulating Social Phenomena*, ed. by R. Conte, R. Hegselmann, P. Terna (Springer, Berlin, 1997), pp. 21–40
4. R.F. Baumeister, E. Bratslavsky, C. Finkenauer, Bad is stronger than good. Rev. Gen. Psychol. **5**(4), 323–370 (2001)
5. S. Brams, P. Fishburn, Voting procedures, in *Handbook of Social Choice and Welfare*, ed. by K.J. Arrow, A.K. Sen, K. Suzumura (Elsevier, Amsterdam, 2004)
6. F.J. Cabrerizo, S. Alonso, I.J. Pérez, E. Herrera-Viedma, On consensus measures in fuzzy group decision making, in *Modeling Decisions for Artificial Intelligence*, ed. by V. Torra, Y. Narukawa, vol. 5285 of Lecture Notes in Computer Science (Springer, Sabadell, 2008), pp. 86–97
7. Y. Chevaleyre, U. Endriss, J. Lang, N. Maudet, A short introduction to computational social choice, in *33rd Conference on Current Trends in Theory and Practice of Computer Science*, ed. by J. van Leeuwen, G.F. Italiano, W. van der Hoek, C. Meinel, H. Sack, F. Plasil (Springer, Berlin, 2007)
8. L. Conradt, C. List, Group decisions in humans and animals: A survey. Philos. Trans. R. Soc. B **364**, 719–742 (2009)
9. E. Durkheim, *The Division of Labour in Society* (Free Press 1933, New York, 1893)
10. D. Eckert, G. Pigozzi, Judgment aggregation , and some links with social choice theory, in *Belief Change in Rational Agents: Perspectives from Artificial Intelligence, Philosophy and Economics, Dagstuhl Seminar Proceedings*, ed. by J. Delgrande, J. Lang, J. Rott, J.-M. Tallon (IBFI, Germany 2005)
11. E. Ephrati, J.S. Rosenschein, Deriving consensus in multiagent systems. Artif. Intell. **87**(1–2), 21–74 (1996)

12. J. Epstein, R. Axtell, *Growing Artificial Societies—Social Science from the Bottom Up* (Brookings Institution/MIT Press, Cambridge, 1996)
13. J. Forrester, Counterintuitive behavior of social systems. Technol. Rev. **73**(3), 52–68 (1971)
14. C. Frantz, M.K. Purvis, M. Nowostawski, B.T.R. Savarimuthu, nADICO: A nested grammar of institutions, in *PRIMA 2013: Principles and Practice of Multi-Agent Systems*, ed. by G. Boella, E. Elkind, B.T.R. Savarimuthu, F. Dignum, M.K. Purvis, vol. 8291 of LNAI (Springer, Dunedin, 2013), pp. 429–436
15. C. Frantz, M.K. Purvis, M. Nowostawski, B.T.R. Savarimuthu, Modelling institutions using dynamic deontics, in *Coordination, Organizations, Institutions and Norms in Agent Systems IX*, ed. by T. Balke, A. Chopra, F. Dignum, B. van Riemsdijk (Springer, Dunedin 2014)
16. J.M.T. García, M.J. del Moral, M.A. Martínez, E. Herrera-Viedma, A consensus model for group decision making problems with linguistic interval fuzzy preference relations. Expert Syst. Appl. **39**(11), 10022–10030 (2012)
17. M.P. Georgeff, B. Pell, M.E. Pollack, M. Tambe, M. Wooldridge, The belief-desire-intention model of agency, in *Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages, ATAL '98*, ed. by J. Muller, M.P. Singh, A.S. Rao (Springer, London, 1999), pp. 1–10
18. N. Ghasem-Aghaee, T. Ören, Towards fuzzy agents with dynamic personality for human behavior simulation. *Proceedings of the 2003 Summer Computer Simulation Conference*, Montreal, 20–24 July 2003, pp. 3–10 (SCS)
19. L.J. Goldberg, *Trade and Institutions in the Medieval Mediterranean: The Geniza Merchants and their Business World* (Cambridge University Press, Cambridge, 2012)
20. R. Goodin, C. List, A conditional defense of plurality rule: Generalizing May's theorem in a restricted informational environment. Am. J. Polit. Sci. **50**, 940–949 (2006)
21. A. Greif, Contract enforceability and economic institution in early trade: The Maghribi traders' coalition. Am. Econ. Rev. **83**(3), 525–548, (1993)
22. A. Greif, *Institutions and the Path to the Modern Economy* (Cambridge University Press, New York, 2006)
23. V. Grimm, Ten years of individual-based modelling in ecology: What have we learned and what could we learn in the future? Ecol. Model. **115**(2–5), 129–148 (1999)
24. H. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Trans. Fuzzy Syst. **12**, 524–539 (2004)
25. H. Hagras, Type-2 FLCs: A new generation of fuzzy controllers. IEEE Comput. Intell. Mag. **2**(1), 30–43 (2007)
26. S. Hassan, L. Garmendia, J. Pavón, Agent-based social modeling and simulation with fuzzy sets, in *Innovations in Hybrid Intelligent Systems, Advances in Soft Computing* (Springer Berlin, Heidelberg, 2008), pp. 40–47
27. S. Hassan, M. Salgado, J. Pavón, Friendship dynamics: Modelling social relationships through a fuzzy agent-based simulation. Discret. Dyn. Nat. Soc. 2011, 19 p, Article ID 765640, (2011). doi:10.1155/2011/765640
28. E. Herrera-Viedma, F. Herrera, F. Chiclana, A consensus model for multiperson decision making with different preference structures. IEEE Trans. Syst. Man Cybern. Part A **32**(3), 394–402 (2002)
29. G. Klir, T. Folger, *Fuzzy Sets, Uncertainty and Information* (Prentice Hall, Englewood Cliffs, 1988)
30. M. Lesani, N. Montazeri, Fuzzy trust aggregation and personalized trust inference in virtual social networks. Comput. Intell. **25**(2), 51–83 (2009)
31. K. Lewin, *A Dynamic Theory of Personality* (McGraw-Hill, New York, 1935)
32. K. Lewin, Defining the "Field at a Given Time". Psychol. Rev. **50**, 292–310 (1943)
33. K. Lewin, *Field Theory in Social Science* (Tavistock, London, 1952)
34. K. Leyton-Brown, Y. Shoham, *Essentials of Game Theory: A Concise Multidisciplinary Introduction*. Synthesis Lectures on Artificial Intelligence and Machine Learning. (Morgan & Claypool, San Rafael, CA, USA, 2008)

35. C. Li, Y. Wang, H.D. Yang, Combining fuzzy partitions using fuzzy majority vote and KNN. JCP **5**(5), 791–798 (2010)
36. F. Liu, J.M. Mendel, Encoding words into interval type-2 fuzzy sets using an interval approach. IEEE Trans. Fuzzy Syst. **16**(6), 1503–1521 (2008)
37. C.-F. Liu, C.-Y. Yeh, S.-J. Lee, Application of type-2 neuro-fuzzy modeling in stock price prediction. Appl. Soft. Comput. **12**(4), 1348–1358 (2012)
38. B. Márquez, M. Castanon-Puga, J. Castro, E. Suarez, S. Magdaleno-Palencia, Fuzzy models for complex social systems using distributed agencies in poverty studies, in *Software Engineering and Computer Systems, Communications in Computer and Information Science*, vol. 179, ed. by J. Mohamad Zain, W. Wan Mohd, E. El-Qawasmeh (Springer, Berlin, 2011), pp. 391–400
39. K. May, A set of independent, necessary and sufficient conditions for simple majority decision. Econometrica **20**, 680–684 (1952)
40. J. Mendel, *Uncertain Rule-based Fuzzy Logic Systems: Introduction and New Directions* (Prentice Hall, Upper Saddle River, 2001)
41. J.M. Mendel, Fuzzy sets for words: A new beginning. *IEEE FUZZ Conference*, St. Louis, 26–28 May 2003, pp. 37–42
42. J.M. Mendel, Computing with words: Zadeh, Turing, Popper and Occam. IEEE Comput. Intell. Mag. **2**, 10–17 (2007)
43. J.M. Mendel, R.I. John, F. Liu, Interval type-2 fuzzy logic systems made simple. IEEE Trans. Fuzzy Syst. **14**, 808–821 (2006)
44. P.R. Milgrom, D.C. North, B.R. Weingast, The role of institutions in the revival of the trade: The law merchant, private Judges, and the champagne fairs. Econ. Polit. **2**, 1954–1985 (1990)
45. S. Miller, M.A. Góngora, J.M. Garibaldi, R. John, Interval type-2 fuzzy modelling and stochastic search for real-world inventory management. Soft. Comput. **16**(8), 1447–1459 (2012)
46. H. Moulin. *Axioms of Cooperative Decision Making (Econometric Society Monographs)*. (Cambridge University Press, Cambridge, 1991)
47. D.C. North, *Institutions, Institutional Change, and Economic Performance* (Cambridge University Press, Cambridge, 1990)
48. H. Nurmi, Fuzzy social choice: A selective retrospect. Soft. Comput. **12**, 281–288 (2008)
49. T. Ören, N. Ghasem-Aghaee, Personality representation processable in fuzzy logic for human behavior simulation. *Proceedings of the 2003 Summer Computer Simulation Conference*, Montreal, 20–24 July 2003, pp. 11–18 (SCS)
50. J.I. Peláez, M.J. Doña, A majority model in group decision making using QMA–OWA operators. Int. J. Intell. Syst. **21**(2), 193–208 (2006)
51. C.L. Ramírez, O. Castillo, P. Melin, A.R. Díaz, Simulation of the bird age-structured population growth based on an interval type-2 fuzzy cellular structure. Inf. Sci. **181**(3), 519–535 (2011)
52. E. Sabeur, G. Denis, Human behavior and social network simulation: Fuzzy sets/logic and agents-based approach, in *SpringSim 2007*, ed. by M.J. Ades, vol. 2 (SCS/ACM, San Diego, 2007), pp. 102–109
53. D. Sumpter, S. Pratt, Quorum responses and concensus decision making. Philos. Trans. R. Soc. B **364**, 743–753 (2009)
54. A. Vermeule, Submajority rules: Forcing accountability upon majorities. J. Polit. Philos. **13**, 74–98 (2005)
55. P. Visscher, Group decision making in nest-site selection among social insects. Annu. Rev. Entomol. **52**, 255–275 (2007)
56. T.M. Vu, P.-O. Siebers, C. Wagner, Comparison of crisp systems and fuzzy systems in agent-based simulation. *13th UK Workshop on Computational Intelligence (UKCI)*, University of Surrey, pp. 54–61, 2013
57. D. Wu, A brief tutorial on interval type-2 fuzzy sets and systems (2012), http://sites.google.com/site/drwu09/publications/A_Brief_Tutorial_on_Interval_Type-2_Fuzzy_Sets_and_Systems.pdf. Accessed 10 Aug 2014
58. D. Wu, W. Tan, Genetic learning and performation evaluation of type-2 fuzzy logic controllers. Eng. Appl. Artif. Intell. **19**(8), 829–841 (2006)

59. L.A. Zadeh, Fuzzy sets. Inf. Comput. **8**, 338–353 (1965)
60. L. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst. Man Cybern. **1**, 28–44 (1973)
61. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-I. Inf. Sci. **8**, 199–249 (1975)
62. L.A. Zadeh, A computational approach to fuzzy quantifiers in natural language. Comput. Math. Appl. **9**(1), 149–184 (1983)
63. L.A. Zadeh, Fuzzy logic = computing with words. IEEE Trans. Fuzzy Syst. **4**(2), 103–111 (1996)

Chapter 7
Modeling the Uncertainty of a Set of Graphs Using Higher-Order Fuzzy Sets

Lorenzo Livi and Antonello Rizzi

Abstract Recent advances in type-2 fuzzy sets (T2FS) have attracted considerable attention for applications in data mining and pattern recognition. In particular, there is an effort in designing granulation procedures able to generate, from raw input measurements, data, granules of information modeled as T2FS. From our viewpoint, the principal aim of those procedures is to embed into the generated T2FS model the key uncertainty characterizing the input data. However, to date there is no formal principle or guideline for the formal evaluation of such granulation procedures in these terms. In this paper, our aim is to define a framework to design and evaluate what we called uncertainty-preserving transformation procedures, which are basically computational procedures that generate, from raw input measurements, information granules modeled as T2FS. In particular, in this chapter, we deal with input measurements that are represented as graphs; hence, a set of graphs $\mathcal{G}$ is seen as a set of raw input measurements sampled from an unknown data generating process P. The framework is, however, meant to be general and thus applicable to any input type. We motivate and explain the proposed framework by performing experimental evaluations on ad hoc synthetically generated datasets.

7.1 Introduction

The concepts of uncertainty and information pervade our lives ubiquitously, from the perceptions of the world to the formal interpretation of experiments and systems. The two concepts are intimately related. In fact, the uncertainty is usually related to some action, which may involve a prediction, a decision making, or a representation—conceptualization of a formal system. Gaining information instead

L. Livi (✉) · A. Rizzi
Department of Information Engineering, Electronics, and Telecommunications,
SAPIENZA University of Rome, Via Eudossiana 18, 00184 Rome, Italy
e-mail: llivi@scs.ryerson.ca

A. Rizzi
e-mail: antonello.rizzi@uniroma1.it

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_7

can be described as the act of reducing the uncertainty of a particular situation, such as one of those previously described [15]. As claimed by Klir [15] "The nature of uncertainty depends on the mathematical theory within which problem situations are formalized." This intuitive fact suggests that the mathematical description of the uncertainty, and accordingly of the information, pertaining to a specific situation may change the *quantification* of those two observed concepts. The well-known mathematical frameworks for this purpose are probability theory, fuzzy set theory, possibility theory, and the Dempster–Shafer theory [15, 16, 48, 49]. Beyond such a technical problem, there is the intuitive fact that a particular data system should have also an absolute description in terms of uncertainty, meaning that if the data system is uncertain, it should remain "adequately" uncertain regardless the particular mathematical framework adopted for the description.

Research on data-driven modeling has defined several automatic systems able to cope with dataset of $\mathbb{R}^n$ feature vectors [45]. However, many interesting practical applications deal directly with *structured patterns*, such as images [9, 31], audio/video signals [36], biochemical compounds [4], and metabolic networks [46]. In this context, labeled graphs are general and widely adopted structures able to represent the topology and the characterizing attributes of data. Consequently, the graph-based representation has been adopted extensively in different contexts [11, 21, 27]. The design of effective pattern recognition and data mining systems is, however, harder on a structured domain, since the common *metric* structure underlying feature-based representations is usually missing or nontrivial. Consequently, researchers in this context focus on the so-called dissimilarity measures and related *embedding strategies* [21, 34].

Recent advances in type-2 fuzzy logic [28, 29] are gathering considerable attention for the application in data mining and pattern recognition systems. In particular, there is an effort in designing *granulation procedures* able to generate type-2 fuzzy set (T2FS) granules of information [1] from input raw measurements data [7, 18, 26, 30, 40, 44]. The main aim of those procedures is to model the uncertainty characterizing the input dataset by means of a proper T2FS model. To our knowledge, the only formalized framework to design such granulation procedures is the so-called principle of justifiable granularity [33]. Although this principle has been widely applied, it is not conceived to provide a formal and objective measure to express the "quality" of the granulation itself.

In this paper, we propose a novel framework that is based on the well-known *principles of uncertainty* [15, 16]. We elaborate such principles in an experimental setting. We develop a framework with the aim of providing the groundwork for future theoretical and experimental contextualization of such procedures. As a result, procedures for generating T2FS can be evaluated also by measurable and, hence, controllable criteria. Here, we focus on input data represented in terms of labeled graphs and granules of information modeled as T2FS. The framework is, however, general and thus, it is applicable to virtually any type of input data and any suitable setting of information granulation. The paper is structured as follows. Section 7.2 briefly reviews the T2FS basic definitions. In Sect. 7.3, we portray the application

of (labeled) graphs as data models. In Sect. 7.4, we present and motivate the framework for the design of uncertainty-preserving transformations. Section 7.6 describes the experiments performed on a batch of synthetically generated dataset of graphs. Finally, conclusions and future directions are drawn in Sect. 7.7.

7.2 Brief Review of T2FS

A T2FS $\widetilde{\mathcal{A}}$ defined on the universe of discourse $\mathcal{X}$ is represented as in [28, 29, 39, 41–43]

$$\begin{aligned} \widetilde{\mathcal{A}} &= \{(x, \mu_{\widetilde{\mathcal{A}}}(x)) \mid x \in \mathcal{X}, \\ \mu_{\widetilde{\mathcal{A}}}(x) &= \{(u, f_x(u)) \mid u \in \mathcal{J}_x \subseteq [0,1], f_x(u) \in [0,1]\}. \end{aligned} \tag{7.1}$$

We refer to $\mu_{\widetilde{\mathcal{A}}}(x)$ as the *fuzzy membership value* of x in $\widetilde{\mathcal{A}}$—it is also known as *secondary membership function* or *secondary set*. Moreover, in Eq. (7.1), x is called *primary variable*, the set $\mathcal{J}_x$ represents the *primary membership values* of x, and $f_x(u)$ is named *secondary grade*. $\mathcal{X}$, as well as $\mathcal{J}_x, \forall x \in \mathcal{X}$, can be continuous or discrete, defining, respectively, continuous or discrete T2FS.

A T2FS in which $\forall u \in \mathcal{J}_x$, $f_x(u) = 1$ holds, reduces to the so-called interval type-2 fuzzy set (IT2FS). In case of IT2FSs, $\mu_{\widetilde{\mathcal{A}}}(x)$ is referred to as *interval membership value*, since the membership degree of an input element is an interval in $[0, 1]$. An IT2FS is fully characterized by its so-called footprint Of uncertainty (FOU) [28, 40], which is defined as:

$$\text{FOU}(\widetilde{\mathcal{A}}) = \bigcup_{x \in \mathcal{X}} \left[\underline{\mu}_{\widetilde{\mathcal{A}}}(x), \overline{\mu}_{\widetilde{\mathcal{A}}}(x) \right]. \tag{7.2}$$

As can be observed from Eq. (7.2), FOU is a bounded region depicting the uncertainties associated with the membership grades of IT2FS $\widetilde{\mathcal{A}}$. Notably, the FOU can be characterized by two type-1 fuzzy sets (T1FSs) only, which are called *upper membership function* (UMF) and *lower membership function* (LMF), respectively, and are defined as follows:

$$\text{UMF}(\widetilde{\mathcal{A}}) = \overline{\text{FOU}}(\widetilde{\mathcal{A}}) = \left\{ (x, \overline{\mu}_{\widetilde{\mathcal{A}}}(x)) \mid x \in \mathcal{X} \right\}, \tag{7.3}$$

$$\text{LMF}(\widetilde{\mathcal{A}}) = \underline{\text{FOU}}(\widetilde{\mathcal{A}}) = \left\{ (x, \underline{\mu}_{\widetilde{\mathcal{A}}}(x)) \mid x \in \mathcal{X} \right\}. \tag{7.4}$$

7.3 Graphs as Data Patterns

A labeled graph is defined as a tuple $G = (\mathcal{V}, \mathcal{E}, \mu, \nu)$, where $\mathcal{V}$ is the (finite) set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, $\mu : \mathcal{V} \to \mathcal{L}_{\mathcal{V}}$ is the vertex labeling (total) function, with $\mathcal{L}_{\mathcal{V}}$ denoting the vertex-labels set, and finally $\nu : \mathcal{E} \to \mathcal{L}_{\mathcal{E}}$ is the edge

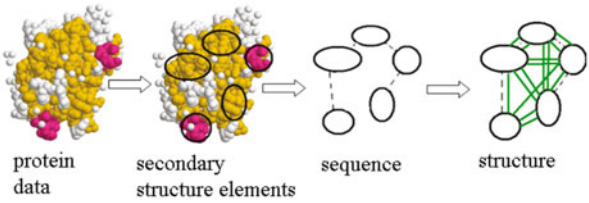


Fig. 7.1 Graph representation of proteins data

Fig. 7.2 Example of drawings of the letter "A"

(total) labeling function, with $\mathcal{L}_\mathcal{E}$ denoting the edge-labels set [2, 19, 21–23, 25]. The topology of a graph enables the description of patterns that are characterized in terms of interacting elements, by describing their spatiotemporal relations. Moreover, the generality of both $\mathcal{L}_\mathcal{V}$ and $\mathcal{L}_\mathcal{E}$ allows fitting a broad range of patterns.

There are many fields of application where labeled graphs can be, and have been, applied as a powerful and general representation framework [11, 21, 27]. For example, we can cite applications to web content-based information retrieval [37], smart grids modeling [10], and complex networks analysis [3, 6]. The graph-based representation is very effective when dealing with biochemical molecules. For example, in [13], the recognition of *mutagenic* compounds is performed by using labeled graphs as models of the data. The representation of molecules as graphs is straightforward: the atoms are the vertices and the covalent bonds are the (undirected) edges. Vertices are labeled with the corresponding chemical symbol and edges by the valence of the linkage. In [4], labeled graphs modeling proteins are considered for recognition. The graphs are constructed considering the secondary structure elements of the proteins and their spatial postfolding relations. In fact, each vertex is connected to the three nearest neighbors in the 3-D space. Both vertices and edges are equipped with complex composite type labels, describing both biological and spatial information. Figure 7.1, taken from [4, Fig. 2], shows a simple illustration of the graph's elaboration process. A further example of graph-based representation comes from the characters recognition problem, largely described in [31]. Graphs are employed to represent distorted letter drawings. For example, Fig. 7.2 shows different level of distortions applied to the "A" letter. Labeled graphs are constructed representing straight lines by undirected and unlabeled edges and ending points of lines by vertices. Each vertex is labeled with a 2-D attribute giving its position relative to a reference coordinate system (usually the 2-D plane). Figure 7.3 shows different graph representations of the (distorted) "A" letter.

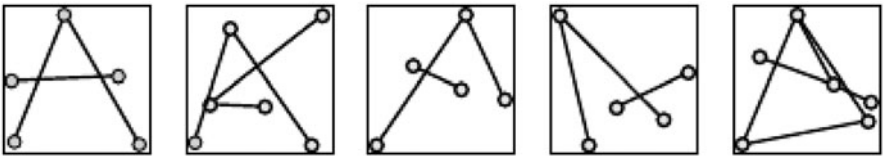


Fig. 7.3 Graph representations of the letter in Fig. 7.2

7.4 Uncertainty Invariance of the IT2FS Transformation

Motivated by the recent works on (interval) T2FSs data granulation [7, 18, 30, 40, 44], in this section, we formalize a framework, forged into an experimental and conjectural setting, for the design of what we called *uncertainty-preserving transformations*. Those procedures, which we denote as $\phi : \mathcal{X} \rightarrow \mathcal{F}_2(\mathcal{X})$, are computational procedures aimed at generating (interval) T2FSs starting from a set of raw input measurements, such as points in $\mathbb{R}^d$, sequences, interval endpoints, or labeled graphs—$\mathcal{F}_2(\mathcal{X})$ denotes the set of all (interval) T2FSs generated over the domain $\mathcal{X}$. To date, the performance, and more generally, the effectiveness of those procedures has been evaluated based on user-centered criteria, or considering conventions established by the scientific community (e.g., reasonability of the derived T2FS shape and positioning in the input domain, etc.). While one of the ultimate aims of data granulation is to allow a semantic interpretation of the extracted information granules, we believe that a formal, quantifiable, and objective evaluation criteria can be established in this case.

The main aim of such a data granulation procedure, $\phi(\cdot)$, is to embed into the generated T2FS model $\widetilde{\mathcal{A}}$, the (relevant) uncertainty that characterizes the input measurements. Accordingly, our claim is that any procedure $\phi(\cdot)$ conceived for this purpose should be designed and, hence, evaluated also on the base of this fact. In the following, and throughout this paper, we use the concept of entropy as an universal measure of uncertainty, since there is correspondence between mainstream uncertainty theories (i.e., probability theory, possibility theory, and fuzzy set theory) [8, 15, 20, 47]. Moreover, without loss of generality, we focus on reconstructing IT2FS granules of information only.

7.4.1 Preserving the Uncertainty of the Observed Measurements

A process P generates data according to some analytical model, which may be deterministic or not; it may also be completely known, partially known, or completely unknown in its closed form. For instance, assuming that P is a stochastic process, we can characterize the uncertainty of P by computing its *entropy rate* [8], which is

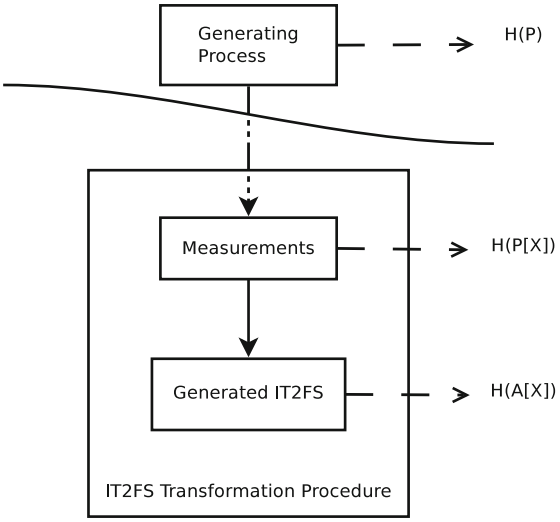


Fig. 7.4 Overview of the uncertainty modeling and calculation stages

defined as the limit (when exists) of the joint entropy:

$$H(P) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n). \tag{7.5}$$

Note that in Eq. (7.5), $X_i, i = 1, 2, \ldots, n$, is a random variable describing the ith observed variable of P. However, in real-world data mining and pattern recognition problems, the true underlying model of P is most of the time unknown, and the only available evidence is a finite sampling $P[\mathcal{X}]$. As a consequence, a computational procedure, $\phi(\cdot)$, whose aim is to model the uncertainty of the observed measurements $P[\mathcal{X}]$ through an IT2FS $\widetilde{\mathcal{A}}[\mathcal{X}]$, i.e., $\phi(P[\mathcal{X}]) = \widetilde{\mathcal{A}}[\mathcal{X}]$, must consider the fact that the *intrinsic* uncertainty of P, denoted as $H(P)$, characterizes directly the diversity of the available measurements, $P[\mathcal{X}]$. For example, if $P[\mathcal{X}]$ is constituted by measurements that are all equal, we deduce that the generating process P is certain, meaning that it generates certainly a specific measurement. This fact must be reflected in turn by the uncertainty of the generated IT2FS, $\widetilde{\mathcal{A}}[\mathcal{X}]$, modeling these measurements.

Formalizing this intuitive idea, we write the following expression:

$$H(P) \simeq H(P[\mathcal{X}]) = g(H(\widetilde{\mathcal{A}}[\mathcal{X}])). \tag{7.6}$$

The uncertainty $H(P[\mathcal{X}])$ estimated from the observed measurements is assumed to be sufficiently descriptive of the uncertainty characterizing the data generating process—$H(P) \simeq H(P[\mathcal{X}])$. The degree of validity of this assumption is, however, dependent on the number of available measurements, and to the complexity (nonlinearity) of P; in this paper, we assume this hypothesis to be true. The last part of Eq. (7.6), that is, $H(P[\mathcal{X}]) = g(H(\widetilde{\mathcal{A}}[\mathcal{X}]))$, will be elaborated in the following. Figure 7.4 shows an high-level diagram depicting the, herein, introduced relation

among the uncertainty at the three different stages: the data generating process, the available set of samples, and the reconstructed IT2FS model.

Since our aim is to provide a framework to guide the design of computational procedures to model the uncertainty of a set of input data using an IT2FS, we focus our analysis completely on $P[\mathcal{X}]$. Let $\mathbf{D}$ be the dissimilarity matrix of $P[\mathcal{X}]$, which is defined as

$$D_{ij} = d(x_i, x_j), \forall x_i, x_j \in P[\mathcal{X}], \tag{7.7}$$

where $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ is a suitable dissimilarity measure [21, 34, 38]. $H(P[\mathcal{X}])$ can be computed by considering the estimated entropy $\hat{H}(\mathbf{D})$ calculated from the dissimilarity matrix $\mathbf{D}^{n\times n}$ of $P[\mathcal{X}], n = |P[\mathcal{X}]|$:

$$H(P[\mathcal{X}]) = \hat{H}(\mathbf{D}). \tag{7.8}$$

This estimation is performed by interpreting the n rows (columns) of $\mathbf{D}$ as n-dimensional random vectors, generated by an unknown random process. If $\mathbf{D}$ denotes degenerate dissimilarity values, i.e., it contains numerical values concentrated around a single point, the estimated entropy is close to zero. Conversely, the higher the variability of the dissimilarity values, the higher is the estimated entropy [24, 25, 35]. This mechanism permits to approximate the uncertainty of the unknown data generating process by analyzing only a finite set of samples. It is worth to stress that the definition $H(P[\mathcal{X}]) = \hat{H}(\mathbf{D})$ works regardless the nature of P. The only requirement is the definition of a proper dissimilarity measure, $d(\cdot, \cdot)$, operating in the input space $\mathcal{X}$.

Equation (7.6) aims to provide an operative expression for the well-known *principle of uncertainty invariance* [15, 16, 47, 50]. This principle states that "the uncertainty of a modeled system should be invariant with respect to the particular mathematical framework adopted for the description." According to Eq. (7.6), (artificial) variations of the uncertainty related to the generating process P induce a direct variation of the uncertainty estimated from $\hat{H}(\mathbf{D})$, which in turn must be monotonically related with the variation of the uncertainty observed in the generated IT2FS—$\phi(P[\mathcal{X}]) = \widetilde{\mathcal{A}}[\mathcal{X}]$. We can describe the second part of this causality relation by introducing a monotonic function $g : \mathbb{R}^+ \to \mathbb{R}^+$ such that:

$$\hat{H}(\mathbf{D}) = g(H(\phi(P[\mathcal{X}]))). \tag{7.9}$$

The function $g(\cdot)$ (7.9) is introduced to handle the fact that the underlying model of $P[\mathcal{X}]$ and the one of the final IT2FS $\widetilde{\mathcal{A}}[\mathcal{X}]$ may be based on different mathematical frameworks, which may result in a different practical quantification of the uncertainty. Moreover, the numerical quantification may also differ assuming the same mathematical framework (e.g., fuzzy sets) since there exist different formulations of the entropy [47].

The procedure $\phi(\cdot)$ which generates the IT2FSs—$\phi(P[\mathcal{X}]) = \widetilde{\mathcal{A}}[\mathcal{X}]$—is usually implemented through a complex computational procedure, which may be composed of different parametric subroutines. Moreover, since we rely on the available measurements $P[\mathcal{X}]$ only, the function $g(\cdot)$ can be determined only experimentally.

To this end, let us assume a suitable artificial scenario in which we are able to control the uncertainty of the generating process P, producing m independent samplings $P[\mathcal{X}]_j$, $j = 1, \ldots, m$, characterized by a monotone uncertainty variation (e.g., monotonically decreasing). To approximate $g(\cdot)$, we put in relation the results obtained on the batch of m different (and independent) evaluations of $x_j = \hat{H}(\mathbf{D}_j)$ and $y_j = H(\phi(P[\mathcal{X}]_j))$, $j = 1, 2, \ldots, m$, by using a suitable *best-fitting* algorithm [14] on the collection of $[x_j, y_j]^T \in \mathbb{R}^2$, $j = 1, 2, \ldots, m$. As a consequence, if we denote with $f(\cdot)$ the obtained approximation of $g(\cdot)$ on such a batch, the equality (7.9) most likely will not be satisfied perfectly by $f(\cdot)$, since we may introduce errors during the best-fitting procedure. Indeed, the best fit will be most likely imprecise and/or the batch size m may be insufficiently large to describe the process with complete accuracy.

Once $f(\cdot)$ has been experimentally derived, according to Eqs. (7.6) and (7.9), $\phi(\cdot)$ must be defined such that the derived IT2FS loses the minimum amount of uncertainty with respect to the observed input measurements. By defining

$$\delta = e\left(f(\cdot), \hat{H}(\mathbf{D}_j), H(\phi(P[\mathcal{X}]_j)); \{P[\mathcal{X}]_j\}_{j=1}^m\right) \tag{7.10}$$

as a measure quantifying the error introduced by $f(\cdot)$ on the batch of m samplings $\{P[\mathcal{X}]_j\}_{j=1}^m$, we say that the transformation function $\phi(\cdot)$ is δ-divergent, meaning that the procedure $\phi(\cdot)$ diverges of an amount of input uncertainty experimentally quantifiable as δ. Of course, the best possible result is obtained when $\delta = 0$, i.e., when the uncertainty is perfectly preserved during the transformation. It is worth underlying that δ is intended either as an excess or deficiency of uncertainty, i.e., δ may be either positive or negative. Therefore, the module of δ, $|\delta|$, can be used as an absolute quality indicator for $\phi(\cdot)$.

In essence, we claim that the procedure $\phi(\cdot)$ should be designed and evaluated, among with the other context-dependent requirements, with the minimization of $|\delta|$ in mind:

$$\min_{\phi(\cdot)} |\delta|. \tag{7.11}$$

Equation (7.11), contextualized in our experimental setting, takes into account also another principle of uncertainty, called *principle of minimum uncertainty* [15]. This principle in fact states that "one should consider solutions from a given solution set such that the resulting information reduction is as small as possible," which is in accord with our framework.

7.5 Modeling the Uncertainty of a Set of Graphs

Let P be a data generating process that generates raw measurements described by means of graphs, G_i. Let $P[\mathcal{G}] = \{G_1, G_2, \ldots, G_n\}$ be the finite set of measurements observed from P. By using $\phi(\cdot)$, we model $P[\mathcal{G}]$ (and hence by assumption also

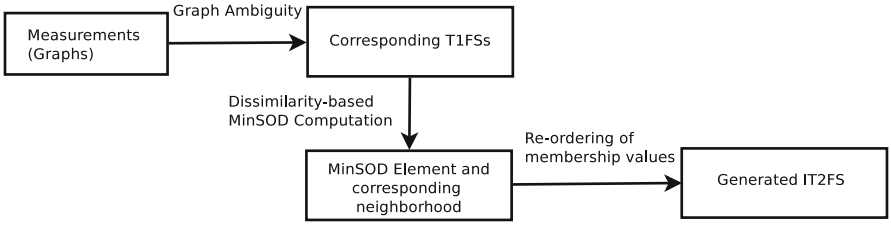


Fig. 7.5 Main steps of the proposed IT2FS transformation procedure, $\phi(\cdot)$

P) as a single IT2FS, $\widetilde{\mathcal{A}}[\mathcal{G}]$. As claimed in Sect. 7.4, the model of $\widetilde{\mathcal{A}}[\mathcal{G}]$ must allow a plausible quantification of the uncertainty of the observed measurements, $P[\mathcal{G}]$, according to the intrinsic uncertainty of the (unknown) data generating process, P. In this section, we propose an instance of such a computational procedure, $\phi(\cdot)$, able to process set of input graphs, yielding the corresponding IT2FS model as output.

Figure 7.5 gives the block diagram describing the fundamental steps characterizing the proposed computational procedure $\phi(\cdot)$. The first step consists of inducing the collection of T1FSs $\mathcal{D}$, corresponding to $P[\mathcal{G}]$. In this stage, each input graph G_i is mapped into a suitable T1FS $\mathcal{F}_i$ by means of the graph ambiguity [20]; this concept is discussed in Sect. 7.5.1. Successively, the set $\mathcal{D}$ is analyzed using a dissimilarity-based algorithm [30, 44] that first identifies a core subset $\mathcal{N}$ of $\mathcal{D}$ characterized by a single element (i.e., a single T1FS) called minimum sum-of-distances (MinSOD) element ν. By using ν and its neighbors $\mathcal{N}$, the final step of the procedure generates the output IT2FS.

7.5.1 *Membership Function Elicitation from a Graph*

Recently, Livi and Rizzi [20] have proposed what they have called *graph ambiguity*. The graph ambiguity is a new theoretical development that defines the concept of fuzzy entropy for a graph—the fuzzy entropy is effectively considered as a measure of ambiguity/uncertainty for a graph. To this end, a graph $G = (\mathcal{V}, \mathcal{E})$, of order $|\mathcal{V}| = N$, is first mapped into a T1FS $\mathcal{F}_i$ defined on a finite universe of discourse $\mathcal{U}$ of $|\mathcal{U}| = N$ elements, each referring to a specific vertex of the graph. This mapping is performed by means of the following membership function:

$$\mu_{\mathcal{F}_i}(v) = \alpha(v) \cdot \tau(v), \forall v \in \mathcal{V}. \tag{7.12}$$

The factor $\alpha(v) \in [0, 1]$ takes into account a measure of *degree concentration* of the vertex v, while $\tau(v) \in [0, 1]$ is a measure of *centrality* referred to that vertex in the graph G. Once the corresponding T1FS has been derived, the input graph can be effectively manipulated according to the T1FSs mathematics; hence, well-known fuzzy entropy formulations can be used to derive the uncertainty–ambiguity of the (fuzzified) graph. Since the fuzzy entropy is a function that depends on the

membership values only, and not on their order with respect to the elements of the input domain, Livi and Rizzi [20] proved that isomorphic graphs have the same fuzzy entropy. This particular property will be used in this paper during the final IT2FS model generation.

Therefore, we deal with the first task shown in Fig. 7.5 by using the herein briefly introduced graph ambiguity embedding mechanism, obtaining eventually the set $\mathcal{D} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n\}$ of n T1FSs corresponding to the n input graphs.

7.5.2 Reconstruction of the IT2FS Model

We generate the corresponding IT2FS by analyzing the set $\mathcal{D} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n\}$ only. In this chapter, we make use of a variation of the recently proposed procedure outlined in [30, 44]. The procedure is a dissimilarity-based algorithm that generates the output IT2FS in two stages. In the first stage, the algorithm individuates a subset $\mathcal{N} \subseteq \mathcal{D}$ of T1FSs represented by a single element $\nu \in \mathcal{N}$: the MinSOD element. Accordingly, ν is characterized by the following equation,

$$\nu = \underset{\mathcal{F}_i \in \mathcal{D}}{\arg\min} \sum_{j=1}^{n} d(\mathcal{F}_i, \mathcal{F}_j), \tag{7.13}$$

where $d(\cdot, \cdot)$ is a proper dissimilarity measure for T1FSs, such as one of the measures discussed in [30, 44]. The set $\mathcal{N}$ is defined considering only the closest neighbors of ν in $\mathcal{D}$; the number of neighbors is, however, determined automatically by the algorithm, on the base of the dissimilarity values distribution among the elements in $\mathcal{D}$ with respect to ν.

The second stage of the IT2FS transformation procedure makes use of both ν and $\mathcal{N}$ to generate the output IT2FS $\widetilde{\mathcal{A}}$ (dropping for simplicity, the additional notation involving $\mathcal{G}$). This stage of the elaboration is based on the principle of justifiable granularity developed by Pedrycz [32]. In few words, the algorithm tries to cover the maximum number of membership values keeping the generated interval memberships as narrow as possible. However, in this paper, we follow a different (and simplified) approach, since we deal with controlled synthetic experiments only (see Sect. 7.6), avoiding thus the possibility of observing outliers. Let N be the maximum order of the input graphs:

$$N = \max_{G_i \in P[\mathcal{G}]} |\mathcal{V}(G_i)|. \tag{7.14}$$

The corresponding T1FSs of $\mathcal{D}$ are thus defined on a domain with N elements, $|\mathcal{U}| = N$. Accordingly, we generate the corresponding IT2FS $\widetilde{\mathcal{A}}$ as a finite IT2FS whose interval membership function describes those N elements. First, we order the membership values of each $\mathcal{F}_i \in \mathcal{N}$ in a nondecreasing order. That is, we make sure that $j \leq k \Leftrightarrow \mathcal{F}_i(v_j) \leq \mathcal{F}_i(v_k)$ holds $\forall v_j, v_k \in \mathcal{U}$ and $\forall \mathcal{F}_i \in \mathcal{N}$. Reordering the elements describing the T1FSs (i.e., the vertices of the embedded graphs) does not

alter the representation of a graph in terms of fuzzy set, since its fuzzy entropy is invariant with respect to this type of transformations (i.e., permutations) [20]. Let $\mathcal{N}'$ be the set of T1FSs with reordered membership values. The (interval) membership function of the output IT2FS is hence generated according to the following formula:

$$\widetilde{\mathcal{A}}(v_j) = \left[\min_{\mathcal{F}_k \in \mathcal{N}'} \mathcal{F}_k(v_j), \max_{\mathcal{F}_h \in \mathcal{N}'} \mathcal{F}_h(v_j) \right], j = 1, 2, \ldots, N. \tag{7.15}$$

7.6 Experiments

7.6.1 Synthetic Dataset Generation

The benchmarking dataset of input graphs is generated using a discrete, time-homogeneous, Markov chain, P. The corresponding transition matrix $\mathbf{T}$ describes entirely the generation process. Labeled (simple) graphs can be generated performing a random walk on a suitable transition matrix $\mathbf{T}^{N \times N}$. The number of states of the chain determines the order of the generated graph, $G = (\mathcal{V}, \mathcal{E})$, $N = |\mathcal{V}|$. During the transition from state i to state j, we add an edge between the corresponding vertices, $v_i, v_j \in \mathcal{V}$ of the graph, i.e., $e_{ij} = (v_i, v_j)$ is added to $\mathcal{E}$. The length l of the random walk determines the size of the graph, that is, the number of edges. Both vertices and edges are labeled using unique integer-valued identifiers, which are generated progressively during the walks.

A transition matrix $\mathbf{T}$ is characterized by "certain" transition probabilities if at each state there is a certain probability (i.e., equal to 1) of transition through a specific state. Performing a random walk on such a transition matrix, will generate only isomorphic graphs, yielding in turn equivalent representations in terms of T1FSs (see Sect. 7.5.1). Consequently, we expect to generate an IT2FS characterized by zero-width interval membership values. Instead, if we use more uncertain transition probabilities in $\mathbf{T}$, we would likely generate different measurements/graphs. This fact should be reflected in turn on the generated IT2FS, which should denote more uncertain membership values (i.e., a wider interval).

We consider in our experiments 10 different dataset instances, each containing 200 graphs generated according to the herein described scheme based on Markov chains. In particular, we use chains defined by 50 states, generating thus graphs with order 50. The size of the graphs, i.e., the length of the random walks, varies from 200 to 250. The similarity, and thus the certainty, of the generated graphs of a specific dataset instance is strictly controlled, producing a sequence of dataset instances of decreasing uncertainty, which corresponds to sample ten different process instances characterized by decreasing uncertainty.

7.6.2 Entropy Estimation Methods

In this chapter, the uncertainty of the generated IT2FSs is computed by evaluating the (normalized) fuzzy entropy proposed by Burillo and Bustince [5], which reads as:

$$H(\widetilde{\mathcal{A}}) = \left(\frac{1}{N} \sum_{i=1}^{N} \overline{\mu}_{\widetilde{\mathcal{A}}}(x_i) - \underline{\mu}_{\widetilde{\mathcal{A}}}(x_i) \right) \in [0, 1]. \tag{7.16}$$

We also provide the computation of the the average fuzzy entropy measured from the set of T1FSs $\mathcal{D}$. To this end, for calculating the fuzzy entropy of a T1FS, we use the expression proposed by Kosko [17],

$$H(\mathcal{F}) = \frac{\Sigma(\mathcal{F} \top \mathcal{F}^c)}{\Sigma(\mathcal{F} \bot \mathcal{F}^c)} \in [0, 1], \tag{7.17}$$

where $\top$ and $\bot$ implement, respectively, a t-norm and t-conorm, and $\Sigma(\cdot)$ is a measure of the derived fuzzy set cardinality [1]. In this paper, the t-norm and t-conorm are implemented as the minimum and maximum operators, respectively, while the cardinality is computed taking the sum of the membership values of the T1FS.

Finally, the dissimilarity matrices $\mathbf{D}_i, i = 1, 2, \ldots, 10$, have been calculated by means of the *graph coverage* dissimilarity measure for labeled graphs [19, 23]. The corresponding entropy is estimated making use of the recently-proposed quadratic Rényi entropy estimator [35], elaborated in the dissimilarity representation context by Livi et al. [24, 25].

7.6.3 Results

Table 7.1 depicts the experimental results obtained on the batch of the datasets $P[\mathcal{G}]_i, i = 1, 2, \ldots, 10$. The column "*DM entropy*" shows the entropy estimated from the dissimilarity matrix $\mathbf{D}_i$ of the input graphs. Column "*IT2FS entropy*" shows the entropy computed on the generated IT2FS $\widetilde{\mathcal{A}}[\mathcal{G}]_i$. Finally, the last column of the table, i.e., "*Average T1FSs entropy,*" contains the average fuzzy entropy computed on the set $\mathcal{D}_i$ of T1FSs, from which we generate $\widetilde{\mathcal{A}}[\mathcal{G}]_i$. Figure 7.6 shows a 2-D plot that correlates the entropy estimated from the dissimilarity matrix with the one estimated from the IT2FS (depicted in blue in the figure), and with the one computed as the average of the T1FSs entropy (depicted in orange). In the figure, we also report the corresponding linear best-fitting functions (i.e., the best fit $f(\cdot)$ introduced in Sect. 7.4.1), together with the coefficient of determination, $R^2 \in [0, 1]$ [12]. A coefficient close to zero means that the linear regression does not fit the data. Translated into our scenario, it means that the procedure $\phi(\cdot)$ does not generate IT2FSs that preserve adequately the uncertainty of the input data. Conversely, when R^2 is close

Table 7.1 Entropy estimation results corresponding to the ten synthetic datasets

Dataset	DM entropy	IT2FS entropy	Average T1FSs entropy
1	0.00930967	0.163009	0.384406
2	0.00893943	0.157133	0.357200
3	0.00828627	0.155459	0.353313
4	0.00769669	0.150279	0.347275
5	0.00589712	0.143724	0.341297
6	0.00553871	0.13446	0.336101
7	0.00525712	0.145131	0.331705
8	0.00518623	0.148126	0.322282
9	0.00495156	0.143333	0.314029
10	0.00490041	0.145925	0.312859

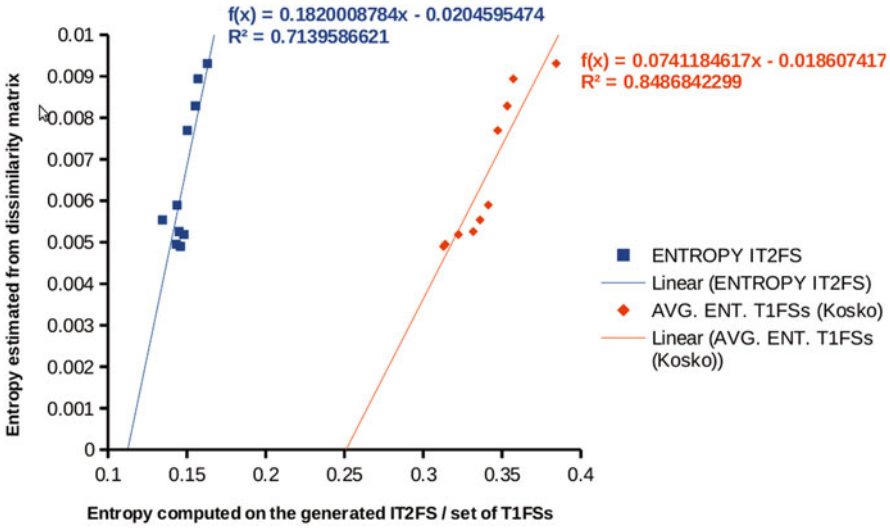


Fig. 7.6 Linear best-fitting results considering the entropy estimated from $\mathbf{D}_i$ with respect to $\widetilde{\mathcal{A}}[\mathcal{G}_i]$ and $\mathcal{D}_i$, for $i = 1, 2, \ldots, 10$

to one, $\phi(\cdot)$ is generating IT2FSs characterized by a suitable uncertainty trend with respect to one of the input data. Note that R^2 can be related straightforwardly to the error δ introduced in Eq. (7.10) by taking $|\delta| = 1 - R^2$. From the (linear) best-fitting equations shown in Fig. 7.6, it is possible to observe that there is more correlation when considering the average entropy computed on the set of T1FSs, $\mathcal{D}_i$, instead of the one taken on $\widetilde{\mathcal{A}}[\mathcal{G}]_i$, $i = 1, 2, \ldots, 10$. Notwithstanding, in both cases, we obtain monotonic best-fitting functions, characterized by reasonable errors.

7.7 Conclusions

In this chapter, we have described a framework, forged into an experimental and conjectural setting, for the design of what we have called uncertainty-preserving transformations. Those transformations are basically computational procedures aimed at generating T2FS granules of information from raw input measurements data. From our viewpoint, these procedures share the same common underlying objective: they are designed to model the input data uncertainty by means of a properly synthesized (interval) T2FS model. Accordingly, we have grounded the proposed framework over the guidelines provided by the so-called principles of uncertainty. We have discussed a specific case study involving graphs as the input measurements (patterns); however, the framework is meant to be valid in general, regardless of the particular input data type and the specific information granule model. Experimental results on synthetically generated problems show encouraging preliminary results.

Future research efforts will be devoted to the theoretical consolidation of the proposed framework. Furthermore, more experimental results on several data types and contexts that confirm the validity of the herein proposed framework are of course of considerable interest.

References

1. A. Bargiela, W. Pedrycz, *Granular Computing: An Introduction*. Kluwer International Series in Engineering and Computer Science (Kluwer Academic, Dordrecht, 2003) (ISBN 9781402072734)
2. F.M. Bianchi, L. Livi, A. Rizzi, A. Sadeghian, A granular computing approach to the design of optimized graph classification systems. Soft. Comput. **18**(2), 393–412 (2014). doi:10.1007/s00500-013-1065-z (ISSN 1432-7643)
3. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D. Hwang, Structure and dynamics. Phys. Rep. **424**(4–5), 175–308 (2006). doi:10.1016/j.physrep.2005.10.009 (ISSN 03701573)
4. K.M. Borgwardt, C.S. Ong, S. Schönauer, S.V.N. Vishwanathan, A.J. Smola, H.-P. Kriegel, Protein function prediction via graph kernels. Bioinformatics **21**, 47–56 (2005). doi:10.1093/bioinformatics (ISSN 1367-4803)
5. P. Burillo, H. Bustince, Entropy on intuitionistic fuzzy sets and on interval-valued fuzzy sets. Fuzzy Sets Syst. **78**(3), 305–316 (1996). doi:10.1016/0165-0114(96)84611-2 (ISSN 0165-0114)
6. L.S. Buriol, C. Castillo, D. Donato, S. Leonardi, S. Millozzi, Temporal analysis of the wikigraph. *Web Intelligence Conference*, Dec 2006, pp. 45–51 (IEEE CS Press)
7. S. Coupland, J. Mendel, D. Wu, Enhanced interval approach for encoding words into interval type-2 fuzzy sets and convergence of the word FOUs. *Proceedings of the IEEE International Conference on Fuzzy Systems*, July 2010, pp. 1–8. doi:10.1109/FUZZY.2010.5584725
8. T.M. Cover, J.A. Thomas, *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing (Wiley, New York, 2006) (ISBN 9780471241959)
9. G. Del Vescovo, A. Rizzi,*Online handwriting recognition by the symbolic histograms approach. Proceedings of the 2007 IEEE International Conference on Granular Computing*, GRC '07, Washington, DC, 2007, pp. 686–700 (IEEE Computer Society). doi:10.1109/GRC.2007.116 (ISBN 0-7695-3032-X)
10. F. Dörfler, F. Bullo, Kron reduction of graphs with applications to electrical networks. IEEE Trans. Circuits Syst. **60-I**(1), 150–163 (2013). doi:10.1109/TCSI.2012.2215780

11. X. Gao, B. Xiao, D. Tao, X. Li, A survey of graph edit distance. Pattern Anal. Appl. **13**(1), 113–129 (2010). doi:10.1007/s10044-008-0141-y (ISSN 1433-7541)
12. K.-I. Goh, M.E. Cusick, D. Valle, B. Childs, M. Vidal, A.-L. Barabási, The human disease network. Proc. Nat. Acad. Sci. **104**(21), 8685–8690 (2007). doi:10.1073/pnas.0701361104
13. J. Kazius, R. McGuire, R. Bursi, Derivation and validation of toxicophores for mutagenicity prediction. J. Med. Chem. **48**(1), 312–320 (2005). doi:10.1021/jm040835a
14. D. Kincaid, E. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*. Pure and Applied Undergraduate Texts (American Mathematical Society, Providence, 2002) (ISBN 9780821847886)
15. G.J. Klir, Principles of uncertainty: What are they? Why do we need them? Fuzzy Sets Syst. **74**(1), 15–31 (1995). doi:10.1016/0165-0114(95)00032-G (ISSN 0165-0114)
16. J.G. Klir, *Uncertainty and Information: Foundations of Generalized Information Theory* (Wiley-Interscience, New York, 2006) (ISBN 9780471748670)
17. B. Kosko, Fuzzy entropy and conditioning. Inf. Sci. **40**(2), 165–174 (1986). doi:10.1016/0020-0255(86)90006-X (ISSN 0020-0255)
18. F. Liu, J. Mendel, Encoding words into interval type-2 fuzzy sets using an interval approach. IEEE Trans. Fuzzy Syst. **16**(6), 1503–1521 (2008). doi:10.1109/TFUZZ.2008.2005002 (ISSN 1063-6706)
19. L. Livi, A. Rizzi, Parallel algorithms for tensor product-based inexact graph matching, in *Proceedings of the 2012 International Joint Conference on Neural Networks*, pp. 2276–2283, June 2012. doi:10.1109/IJCNN.2012.6252681 (ISBN 978-1-4673-1489-3)
20. L. Livi, A. Rizzi, Graph ambiguity. Fuzzy Sets Syst. **221**, 24–47 (2013). doi:10.1016/j.fss.2013.01.001 (ISSN 0165-0114)
21. L. Livi, A. Rizzi, The graph matching problem. Pattern. Anal. Appl. **16**(3), 253–283 (2013). doi:10.1007/s10044-012-0284-8 (ISSN 1433-7541)
22. L. Livi, G. Del Vescovo, A. Rizzi, in *Proceedings of the First International Conference on Pattern Recognition Applications and Methods*, vol. 1, Feb 2012, pp. 186–191. doi:10.5220/0003733201860191 (ISBN 978-989-8425-98-0)
23. L. Livi, G. Del Vescovo, A. Rizzi, in *Proceedings of the First International Conference on Pattern Recognition Applications and Methods*, vol. 1, Feb 2012, pp. 269–272. doi:10.5220/0003732802690272 (ISBN 978-989-8425-98-0)
24. L. Livi, F.M. Bianchi, A. Rizzi, A. Sadeghian, Dissimilarity space embedding of labeled graphs by a clustering-based compression procedure, in *Proceedings of the 2013 International Joint Conference on Neural Networks*, Aug 2013, pp. 1646–1653. doi:10.1109/IJCNN.2013.6706937 (ISBN 978-1-4673-6129-3)
25. L. Livi, A. Rizzi, A. Sadeghian, Optimized dissimilarity space embedding for labeled graphs. Inf. Sci. **266**, 47–64 (2014). doi:10.1016/j.ins.2014.01.005 (ISSN 0020-0255)
26. L. Livi, H. Tahayori, A. Sadeghian, A. Rizzi, Distinguishability of interval type-2 fuzzy sets data by analyzing upper and lower membership functions. Appl. Soft Comput. **17**, 79–89 (2014). doi:10.1016/j.asoc.2013.12.020 (ISSN 1568-4946)
27. R. Marfil, F. Escolano, A. Bandera, Graph-based representations in pattern recognition and computational intelligence, in *Bio-Inspired Systems: Computational and Ambient Intelligence*, ed. by J. Cabestany, F. Sandoval, A. Prieto, J. Corchado. Lecture Notes in Computer Science, vol. 5517 (Springer, Berlin, 2009). pp. 399–406. doi:10.1007/978-3-642-02478-8_50 (ISBN 978-3-642-02477-1)
28. J. Mendel, R. John, Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002). doi:10.1109/91.995115 (ISSN 1063-6706)
29. J. Mendel, R. John, F. Liu, Interval type-2 fuzzy logic systems made simple. IEEE Trans. Fuzzy Syst. **14**(6), 808–821 (2006). doi:10.1109/TFUZZ.2006.879986 (ISSN 1063-6706)
30. M. Moharrer, H. Tahayori, L. Livi, A. Sadeghian, A. Rizzi, Interval type-2 fuzzy sets to model linguistic label perception in online services satisfaction. Soft Comput. (2014). doi:10.1007/s00500-014-1246-4 (ISSN 1432-7643)
31. M. Neuhaus, H. Bunke, *Bridging the Gap Between Graph Edit Distance and Kernel Machines*. Series in Machine Perception and Artificial Intelligence, vol. 8 (World Scientific, Singapore, 2007) (ISBN 9789812708175)

32. W. Pedrycz, Human centricity in computing with fuzzy sets: An interpretability quest for higher order granular constructs. J. Ambient Intell. Hum. Comput. **1**, 65–74, (2010). doi:10.1007/s12652-009-0008-0 (ISSN 1868-5137)
33. W. Pedrycz, W. Homenda, Building the fundamentals of granular computing: A principle of justifiable granularity. Appl. Soft Comput. **13**(10), 4209–4218 (2013). doi:10.1016/j.asoc.2013.06.017 (ISSN 1568-4946)
34. E. Pękalska, R.P.W. Duin, *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. Series in Machine Perception and Artificial Intelligence, vol. 64 (World Scientific, Singapore, 2005) (ISBN 9789812565303)
35. C.J. Príncipe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Information Science and Statistics. (Springer, Berlin, 2010) (ISBN 9781441915696)
36. A. Rizzi, G. Del Vescovo, Automatic image classification by a granular computing approach. *Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, Sept 2006, pp. 33–38. doi:10.1109/MLSP.2006.275517
37. A. Schenker, *Graph-Theoretic Techniques for Web Content Mining*. Series in Machine Perception and Artificial Intelligence, vol. 62. (World Scientific, Singapore, 2005) (ISBN 9789812563392)
38. T. Skopal, Unified framework for fast exact and approximate search in dissimilarity spaces. ACM Trans. Database Syst. 32(4), (2007). doi:10.1145/1292609.1292619 (ISSN 0362-5915)
39. H. Tahayori, G. Degli Antoni, Operations on concavoconvex type-2 fuzzy Sets. Int. J. Fuzzy Syst. 10(4), 276–286 (2008) (ISSN 1562-2479)
40. H. Tahayori, A. Sadeghian, Median interval approach to model words with interval type-2 fuzzy sets. Int. J. Adv. Intell. Paradig. **4**(3), 313–336 (2013)
41. H. Tahayori, A. Tettamanzi, G.D. Antoni, Approximated type-2 fuzzy set operations, in *Proceedings of the IEEE International Conference on Fuzzy Systems*, 2006, pp. 1910–1917
42. H. Tahayori, A.G.B. Tettamanzi, G.D. Antoni, A. Visconti, 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). doi:10.1016/j.fss.2009.06.005 (ISSN 0165-0114)
43. H. Tahayori, A. Tettamanzi, G. Degli Antoni, A. Visconti, M. Moharrer, Concave type-2 fuzzy sets: Properties and operations. Soft Comput. **14**, 749–756 (2010). doi:10.1007/s00500-009-0462-9 (ISSN 1432-7643)
44. H. Tahayori, L. Livi, A. Sadeghian, A. Rizzi, Interval type-2 fuzzy sets reconstruction based on fuzzy information-theoretic kernels. IEEE Trans. Fuzzy Syst. (2014). Under review, Manuscript ID: TFS-2013-0660.R1. doi:10.1109/TFUZZ.2014.2336673
45. S. Theodoridis, K. Koutroumbas. *Pattern Recognition*, 4th edn. (Elsevier/Academic, Amsterdam/New York, 2008) (ISBN 9780123695314)
46. K. Tun, P. Dhar, M. Palumbo, A. Giuliani, Metabolic pathways variability and sequence/networks comparisons. BMC Bioinform. **7**(1), 24 (2006). doi:10.1186/1471-2105-7-24 (ISSN 1471-2105)
47. D. Wu, J.M. Mendel, Uncertainty measures for interval type-2 fuzzy sets. Inf. Sci. **177**(23), 5378–5393 (2007). doi:10.1016/j.ins.2007.07.012 (ISSN 0020-0255)
48. L.A. Zadeh, Toward a generalized theory of uncertainty (GTU)—An outline. Inf. Sci. **172**(1–2), 1–40 (2005). doi:10.1016/j.ins.2005.01.017 (ISSN 0020-0255)
49. L.A. Zadeh, Generalized theory of uncertainty (GTU)—Principal concepts and ideas. Comput. Stat. Data Anal. **51**(1), 15–46 (2006). doi:10.1016/j.csda.2006.04.029 (ISSN 0167-9473)
50. D. Zhai, J.M. Mendel, Uncertainty measures for general type-2 fuzzy sets. Inf. Sci. **181**(3), 503–518 (2011). doi:10.1016/j.ins.2010.09.020 (ISSN 0020-0255)

Chapter 8
Time-Series Forecasting via Complex Fuzzy Logic

Omolbanin Yazdanbakhsh and Scott Dick

Abstract Adaptive neuro-complex-fuzzy inference system (ANCFIS) is a neuro-fuzzy system that employs complex fuzzy sets for time-series forecasting. One of the particular advantages of this architecture is that each input to the network is a windowed segment of the time series, rather than a single lag as in most other neural networks. This allows ANCFIS to predict even chaotic time series very accurately, using a small number of rules. Some recent findings, however, indicate that published results on ANCFIS are suboptimal; they could be improved by changing how the length of an input window is determined, and/or subsampling the input window.

We compare the performance of ANCFIS using three different approaches to defining an input window, across six time-series datasets. These include chaotic datasets and time series up to 20,000 observations in length. We found that the optimal choice of input formats was dataset dependent, and may be influenced by the size of the dataset. We finally develop a recommended approach to determining input windows that balances the twin concerns of accuracy and computation time.

8.1 Introduction

Time-series forecasting has emerged as the first major application of complex fuzzy sets and logic, which were first described by Ramot in [1]. Beginning in 2007, complex-valued neuro-fuzzy systems were developed to inductively learn forecasting models; these include the adaptive neuro-complex-fuzzy inference system (ANCFIS) architecture [2], and the family of complex neuro-fuzzy system (CNFS) architectures [3]. Both ANCFIS and CNFS are modifications of the well-known ANFIS architecture, in which complex fuzzy sets and complex-valued network signals are used. These architectures showed that complex fuzzy sets were naturally useful in creating very accurate forecasting models. ANCFIS in particular is also very parsimonious;

S. Dick (✉) · O. Yazdanbakhsh
Department of Electrical and Computer Engineering, University of Alberta, 2nd Floor ECERF Building, 9107–116 Street, Edmonton, AB T6G 2V4, Canada
e-mail: dick@ece.ualberta.ca

O. Yazdanbakhsh
e-mail: yazdanba@ualberta.ca

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_8

experiments in [2] showed that this architecture could forecast even chaotic systems with no more than three complex fuzzy rules.

One of the key reasons why ANCFIS is so parsimonious is its input format. Most generic machine-learning algorithms must use lagged inputs in order to create a forecasting model. While this approach is mathematically sound, it means that the number of inputs to the learning algorithm has to be equal to the number of lags required to reconstruct the state space of the system that generated the time series (i.e., to form a delay reconstruction in the sense of Takens [4]). This directly leads to a combinatorial explosion in the complexity of the model. However, due to the nature of complex fuzzy sets, ANCFIS does not use lagged inputs; rather, an entire windowed segment of the time series is taken as a single input to the network, greatly reducing the curse of dimensionality. Recent experiments reported in [5] indicated that we might be able to further improve the accuracy of ANCFIS by subsampling the input windows. This is possible because in ANCFIS, we use sinusoidal membership functions for the complex fuzzy sets, which are sampled and convolved with the input window. Subsampling the input window simply implies that we also sample the complex fuzzy sets at a lower rate. Our goal in the current chapter is to determine if such subsampling generally leads to improved accuracy, or if this was a dataset-specific effect.

We compare the forecast accuracy of ANCFIS using three different approaches to identifying and sampling input windows on six time-series datasets. Two of these (a realization of the Mackey–Glass map and the Santa Fe Laser A dataset) are known to be chaotic; the remainder are observations of physical processes (sunspots, stellar brightness, waves, solar power production). All but the last one have been previously studied in the forecasting community and in [2]. The solar power dataset was developed in our laboratory and is discussed in depth in [5]. In [2], the length of the input windows for each of the five datasets was set at one "period" in the dataset, as determined by ad-hoc inspection. We explore the use of heuristics from [4] to construct two different delay embeddings of the time series: one with an "optimal" delay between each lag and one with unit delays between each lag. These input lags are concatenated together in chronological order to form our input windows.

The remainder of this chapter is organized as follows. Section 8.2 provides an essential background on complex fuzzy sets and logic, as well as the ANCFIS architecture. Section 8.3 describes our datasets and experimental methodology. We provide our experimental results in Sect. 8.4 and close with a summary and discussion of future work in Sect. 8.5.

8.2 Literature Review

8.2.1 Complex Fuzzy Sets and Logic

Ramot et al. in 2002 proposed the *complex fuzzy set* (CFS) as a fuzzy set whose membership function takes complex-valued grades, bounded by the unit circle [1]:

$$\mu_s(x) = r_s(x).\mathrm{e}^{(jw_s(x))}\, j = \sqrt{-1}, \tag{8.1}$$

where $r_s(x) \in [0,1]$ is the magnitude and $w_s(x)$ is the phase of the *complex fuzzy set* S. Ramot defined complex fuzzy intersection and union to act solely on the magnitude of complex fuzzy sets; the phase was treated as a means of adding application-dependent context to the CFS. In 2003, Ramot et al. proposed an isomorphic *complex fuzzy logic* (CFL) based on the generalized modus ponens inference rule. To implement complex fuzzy implication, he suggested the complex product [6]:

$$\mu_{A\to B}(x,y) = \mu_A(x) \cdot \mu_B(y), \tag{8.2}$$

where $\mu_{A\to B}(x,y)$ is the complex-valued membership function of the implication, and $\mu_A(x), \mu_B(y)$ are both CFS. To aggregate multiple rules, Ramot proposed a complex-valued weighted sum called vector aggregation as given below [6]:

$$v: \ \{a|a \in C, |a| \le 1\}^n \to \ \{b|b \in C, \ |b| \le 1\} \tag{8.3}$$

$$\mu_A(x) = v(\mu_{A_1}(x), \ \mu_{A_2}(x), \ \ldots, \ \mu_{A_n}(x)) = \sum_{i=1}^{n} w_i \mu_{A_i}(x), \tag{8.4}$$

where $w_i \in \{a|a \in C, \ |a| \le 1\}$ for all i, and $\sum_{i=1}^{n} |w_i| = 1$. In vector aggregation, rules can interfere constructively or destructively with each other.

Dick in 2005 showed that considering the phase as a relative quantity in Ramot et al.'s papers [1, 6] can be interpreted as rotational invariance, meaning that if two vectors undergo rotation by φ radians about the origin, their union, intersection, or complement will be rotated by the same amount [7]. It was shown that the algebraic product and the traditional complement, $f(x) = -x$, are not rotationally invariant. He then proposed a new formulation of membership degree by considering amplitude and phase simultaneously. The algebraic product was shown to be a conjunction operator, and the existence of a dual disjunction operator was proved. He then argued that capturing the behavior of approximately periodic phenomena was a possible application for CFL, and sinusoidal functions were suggested as appropriate complex fuzzy membership functions.

Tamir et al. in [8] proposed a new definition for complex fuzzy degrees using the Cartesian representation of complex numbers where both real and imaginary parts may vary from 0 to 1. These are called "pure" complex fuzzy sets and are defined as [8]:

$$\begin{aligned} \mu(V,z) &= \mu_r(V) + j\mu_i(z) \\ \mu_r, \mu_i &\in [0,1], \end{aligned} \tag{8.5}$$

where $\mu_r(V)$ and $\mu_i(Z)$ are the real and imaginary part of the pure complex fuzzy membership grade, $\mu(V,z)$. An interpretation of pure complex fuzzy grades was proposed based on the pure fuzzy class of order 1; "a fuzzy class is a finite or infinite collection of objects and fuzzy sets which can be unambiguously defined and complies with the class theory; a pure fuzzy class of order 1 can have only fuzzy sets" [8]. Consider the pure complex fuzzy membership function, $\mu_\Gamma(V,z)$:

$$\mu_\Gamma(V,z) = \mu_r(V) + j\mu_i(z). \tag{8.6}$$

Let Γ be a complex fuzzy class, V a fuzzy set, and z a variable in the universe of discourse U. $\mu_\Gamma(V, z)$ can be interpreted as the degree of membership of z in V and the degree of membership of Vin Γ. Complement, intersection, and union operations were also proposed for the complex fuzzy classes in the paper [8].

Tamir et al. proposed a first-order predicate CFL in [9]. In the CFL system, based on ŁΠ system (propositional logic system) by Běhounek et al. [10], a complex fuzzy proposition, $\Gamma = \Gamma_r + j\Gamma_i$, is considered as a composition of two propositions each with truth value in the interval [0, 1]. The proposed CFL was extended to generalized propositional CFL, applicable in multidimensional fuzzy propositional and predicate logic, through definitions in [10] based on fuzzy Łukasiewicz logical system [11]. Tamir et al. [12] proposed an extended complex post-logical system (ECPS) based on the extended Post system (EPS) of order $p > 2$ by DiZenzo [13]. One of the possible applications of the proposed system is in discrete processes such as digital signal processing (DSP), real-time applications, and embedded systems.

Salleh [14] defined complex Atanassov's intuitionistic fuzzy sets (CAIFS). Intuitionistic fuzzy sets were introduced by Atanasov [15]; they record the degree of membership and nonmembership of an element in a set, each indicated by a value in [0, 1]. In complex intuitionistic fuzzy sets, the degrees of membership and nonmembership are each drawn from the unit circle in the complex plane. Basic operations on CAIFS, including complement, union, and intersection, were also presented in the paper. Yager and Abbasov [16] defined Pythagorean membership grades as a subset of complex fuzzy grades, $\mu = re^{\theta}$ with the properties $r \in [0,1]$ and $\theta \in [0, \frac{\pi}{2}]$.

Zhang et al. [17] studied different operations and their properties on the complex fuzzy sets introduced by Ramot et al. [1] when phases are restricted to $[0, 2\pi]$. A new definition for distance of complex fuzzy sets was introduced in the paper:

$$d(A, B) = \max\left(sup_{X\in U}\left|r_A(X) - r_B(X)\right|, \frac{1}{2\pi} sup_{X\in U}\left|arg_A(X) - arg_B(X)\right|\right), \tag{8.7}$$

where $d(A, B)$ is the distance of two complex fuzzy sets $A = r_A(X)e^{j\cdot arg_A(X)}$ and $B = r_B(X)e^{j\cdot arg_B(X)}$. Then, based on this definition, δ-equalities of complex fuzzy sets were proposed; two complex fuzzy sets, A and B, are δ-equal if and only if $d(A, B) \leq 1-\delta, 0 \leq \delta \leq 1$. Zhang et al. also defined δ-equalities for complex fuzzy relations [18]. Alkouri and Salleh [19] introduced a distance between two CAIFS, and proposed complex intuitionistic fuzzy relations by extending operations defined in [17, 18]. They also proposed projection and cylindrical extensions for CAIFS.

A few authors investigated complex-valued membership grades before Ramot's work. Moses et al. [20] proposed a CFS with memberships drawn from the unit square $U \rightarrow [0,1] \times [0,1]$. These complex fuzzy sets contain two fuzzy sets represented by the real and imaginary domains. Linguistic coordinate transformations (i.e., the relationship between linguistic variables when transforming from Cartesian to polar coordinates or vice versa) were also studied. Nguyen et al. [21] proposed an optimization approach to select the best representation of complex fuzzy sets for a particular application.

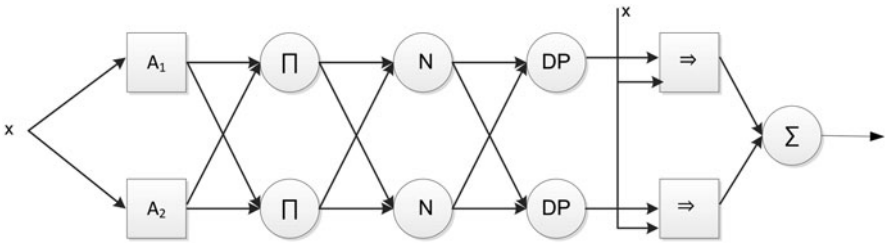


Fig. 8.1 Two-rule ANCFIS architecture for univariate time-series problems [2]

8.2.2 Adaptive Neuro-Complex-Fuzzy Inference System

Chen et al. [2, 22] proposed the first inductive machine-learning realization of the CFL proposed by Dick [7] and Ramot [6]. The ANCFIS architecture is a relative of Jang's well-known ANFIS. The main differences are (1) ANCFIS uses a sinusoidal membership function as follows:

$$r(\theta) = d \sin\left(a(\theta = x) + b\right) + c, \tag{8.8}$$

where $r(\theta)$ is the amplitude and θ is the phase of the membership grade of element x; (2) an additional layer implements rule interference, inspired by Ramot's vector aggregation in [6]; (3) the network signals are complex-valued up through this rule interference layer; and (4) the learning algorithm incorporates a derivative-free optimization component (Fig. 8.1).

As suggested by Dick [7], one possible application of CFL is capturing approximately periodic behavior of phenomena; Chen et al. [2] suggested that time-series forecasting is a good example of such behaviors. Thus, sinusoidal functions are candidate complex fuzzy membership functions since a periodic function can be represented by a Fourier series, i.e., a sum of sin and cosine functions. In Eq. (8.8), the four parameters $\{a, b, c, d\}$ act as follows: a changes the frequency of the sine wave, b gives a phase shift whereas c shifts the wave vertically, and d changes the amplitude of the sine wave. Since the amplitude of complex fuzzy memberships is limited to [0,1], the parameters must satisfy the following conditions:

$$0 \leq d + c \leq 1,\ 1 \geq c \geq d \geq 0. \tag{8.9}$$

The use of a sinusoidal CFS in ANCFIS also implies an important operational difference between ANCFIS and ANFIS (and indeed most other machine-learning algorithms). In using ANFIS and other algorithms for time-series forecasting, input vectors containing lagged values of a variate are presented to the network to predict the next value of the variate. The components of the input vectors are considered orthogonal; thus, to predict $f(t)$, the components $f(t-1)$, $f(t-2), \ldots, f(t-n)$ of an input vector are presented as separate inputs to the system. However, this cannot work in ANCFIS, because matching a sinusoidal membership function to an observation requires that we keep the phase information in our inputs. Orthogonal lagged inputs

destroy this phase information by definition. Instead, in ANCFIS, we take a sliding window of the variate as a single input, $[f(t-1), f(t-2), \ldots, f(t-n)]$, and then match that window to the membership functions. This implies that ANCFIS requires only a single input for each variate of a time series, whereas systems using a lagged input require $\prod_{i=1}^{n} r_i$ inputs, where r_i is the number of lags for a given variate, and n is the number of variates. Thus, ANCFIS significantly reduces the combinatorial explosion inherent in time-series forecasting [2].

As the network signals in ANCFIS are complex-valued, the backward-pass computations in the network must also be different from ANFIS. Like ANFIS, ANCFIS uses a hybrid learning rule where consequent parameters are updated on the forward pass, and antecedent parameters on the backward pass. Indeed, as network signals are real-valued at the consequent layer (layer 5 in ANCFIS), we employ the same least-squares algorithm as ANFIS. However, the backward pass requires back-propagation of complex-valued signals; and ultimately, there is no closed-form expression for the partial derivative of network error with respect to the CFS parameters in Eq. (8.8). As described in [2], we use gradient descent to determine the back-propagating error signals until layer 1, and then use a derivative-free optimization technique (a variant of simulated annealing) to determine the update to the CFS parameters. This technique is the variable neighbourhood chaotic simulated annealing (VNCSA) algorithm.

In chaotic simulated annealing (CSA), the generation of new candidate solutions is governed by a chaotic map instead of a random number generator. This potentially makes the algorithm faster, as we only search a fractal subset of the total solution space. In VNCSA, we create an initial population of solutions by iterating the logistic map starting from a random point. We create new solutions by iterating the Ulam-von Neumann map and adding that value to the existing best solution, weighted by a neighborhood size factor. As with other simulated-annealing algorithms, we will allow a new solution with a worse objective function to replace the current solution with a probability that depends on the current temperature T. As the routine continues, T is gradually reduced by a constant factor at each iteration. When T is updated, the neighborhood size is also updated, depending in part on how much the objective function changed from the last iteration to the current one (thus limiting the neighborhood in which new candidate solutions are generated in the next iteration). For additional details, please see [2].

The ANCFIS architecture has six layers as follows [2]:

- *Layer 1*: In this layer, the input vector is convolved with the membership function. First, the membership function is sampled over one period by

$$r_k(\theta_k) = d\sin(a\theta_k + b) + c, \quad \theta_k = \frac{2\pi}{n}k, \qquad k = 1, 2, \ldots, n \tag{8.10}$$

  where n is the length of the input vector. Then, the sampled membership functions are convolved with the input vector:

$$conv = \sum_{k=1}^{2n-1} \sum_{j=\max(1,k+1-n)}^{\min(k,n)} f(j)g(k+1-j), \tag{8.11}$$

where $f(.)$ is the input vector and $g(.)$ is the sampled membership function (in Cartesian coordinates). To ensure that the convolution sum remains within the unit disc, it is normalized using the Eliot function:

$$O_{1,i} = \frac{conv}{1 + |conv|}. \tag{8.12}$$

- *Layer 2*: In this layer, the firing strength of a fuzzy rule is calculated as follows:

$$O_{2,i} = \prod_i O_{1,i}, \quad i = 1, 2, \ldots, |O_1|, \tag{8.13}$$

where $|O_1|$ is the number of nodes in layer 1. For univariate time series, neurons in this layer reduce to the identity function.

- *Layer 3*: The output of each node represents the normalized firing strength of a rule:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{\sum_{j=1}^{|O_2|} |w_j|}, \quad i = 1,2, \ldots, |O_2|, \tag{8.14}$$

where $|O_2|$ is the number of rules. This layer only normalizes the magnitude whereas phases are unchanged.

- *Layer 4*: This layer realizes the property of "rule interference" from [6], using the dot product:

$$O_{4,i} = w_i^{DP} = \overline{w_i} \cdot \sum_{i=1}^{|O_3|} \overline{w_i}, \tag{8.15}$$

where $|O_3|$ is the number of nodes in layer 3 and $\sum_{i=1}^{|O_3|} \overline{w_i}$ is the complex sum. Both constructive and destructive interference are possible.

- *Layer 5*: This layer implements the linear consequent function:

$$O_{5,i} = w_i^{DP} \left[\sum_{j=1}^{n} p_{i,j} x_j + r_i \right], \tag{8.16}$$

where w_i^{DP} is the output of layer 4, x_j is the jth data point if the input vector, n is the length of the input vector, and $p_{i,j}$, r_i are the parameters of a linear function of x_j. $\{p_{i,j}, r_i\}$ are obtained in the forward pass by least-squares estimation.

- *Layer 6:* This layer sums all incoming signals.

The ANCFIS architecture is related to the more general area of complex-valued neural networks (CVNN) where inputs, outputs, biases, and weights can take on complex values [23–26]. In the specific domain of neuro-fuzzy systems, a few other studies have investigated complex-valued network signals in the ANFIS framework. Li and Jang [27] proposed CANFIS, which accepts the real and imaginary parts of a complex number as separate inputs in ANFIS. Malekzadeh-A and Akbarzadeh-T

[28] designed a different CANFIS, using CFS in the first layer—but preprocessing them through a CVNN first. This architecture is limited to complex-valued inputs and outputs. A complex steepest descent algorithm and a complex least-square estimator were used to update complex membership function parameters and complex bias and weight parameters, respectively.

Other complex fuzzy machine-learning architectures have also been proposed; Aghakhani and Dick [29] proposed an online learning algorithm for ANCFIS. The architecture uses recursive least squares instead of least-square algorithm in the forward pass, and applies the downhill simplex algorithm instead of VNCSA in the backward pass. Li and Chiang [3] proposed a different variation of ANFIS called the CNFS. Their learning algorithm is a hybrid of particle swarm optimization (PSO) and recursive least-square estimator (RLSE) algorithm. Li and Chiang [30] extended CNFS by using a Gaussian-type CFS as:

$$c\text{Gaussian}\,(h,m,\sigma) = Re(c\text{Gaussian}\,(h,m,\sigma)) + jIm(c\text{Gaussian}\,(h,m,\sigma)) \tag{8.17}$$

$$Re(c\text{Gaussian}\,(h,m,\sigma)) = \exp\left[-0.5\left(\frac{h-m}{\sigma}\right)^2\right] \tag{8.18}$$

$$Im(c\text{Gaussian}\,(h,m,\sigma)) = -\exp\left[-0.5\left(\frac{h-m}{\sigma}\right)^2\right] \times \left(\frac{h-m}{\sigma^2}\right), \tag{8.19}$$

where $\{m,\ \sigma\}$ are the mean and spread of the Gaussian function, and h is the input. This paper also proposed the "dual-output property," which refers to treating the real and imaginary components of the output as separate variates (and thus, the network can naturally handle bivariate time series). Several other papers from this research group explore other variations on the CNFS architecture; Li and Chan [31] used the artificial bee colony (ABC) algorithm instead of PSO, and applied CNFS to image restoration (Li et al. [33] used the CNFS from [30] for image restoration as well). Li and Chan [32] used CNFS with the ABC-learning algorithm for knowledge discovery. Li and Chiang [35] replaced PSO in the CNFS proposed by [30] with a multiple-swarm variation called hierarchical multi-swarm PSO, while also proposing a new Gaussian-type complex fuzzy set:

$$c\text{Gaussian}\,(h,m,\sigma,\lambda) = r_s(h,m,\sigma)\exp\left(jw_s(h,m,\sigma,\lambda)\right) \tag{8.20}$$

$$r_s(h,m,\sigma) = \text{Gaussian}\,(h,m,\sigma) = \exp\left[-0.5\left(\frac{h-m}{\sigma}\right)^2\right] \tag{8.21}$$

$$w_s(h,m,\sigma,\lambda) = -\exp\left[-0.5\left(\frac{h-m}{\sigma}\right)^2\right] \times \left(\frac{h-m}{\sigma^2}\right) \times \lambda, \tag{8.22}$$

where $\{m,\sigma,\lambda\}$ are the mean, spread, and phase frequency factor for the complex fuzzy set, and h is the input. Li and Chiang [36] developed a model based on CNFS

proposed in [35] where the consequent layer is an ARIMA model, CNFS-ARIMA. Li and Chiang [34] applied the system proposed in [35] for financial time-series forecasting. Li et al. [37] proposed a CNFS using the Gaussian membership function introduced in [35], which updates the premise and consequent parameters based on PSO-RLSE-learning algorithm. To minimize the rule base of CNFS, a clustering algorithm called FCM-based splitting algorithm (FBSA) is employed [38].

Ma et al. [39] proposed a product–sum aggregation operator for the complex fuzzy sets introduced by Ramot et al. [1]. Based on this operator, a prediction method was developed to solve multiple periodic factor prediction problems in multisensory data fusion applications containing semantic uncertainty and periodicity. We can view this as the first CFL inferential system whose rules are declaratively specified rather than inductively learned. Alkouri et al. [40] defined linguistic variables of complex fuzzy sets based on the ideas in [39]. Linguistic hedges (as introduced by Zadeh [41]) were also extended to complex fuzzy sets. Hamming, Euclidean, normalized Hamming, and normalized Euclidean distances and their boundaries were also obtained for complex fuzzy sets. Deshmukh et al. [42] designed a hardware implementation for the CFL proposed in [6].

8.2.3 *Delay Embedding of a Time Series*

Machine-learning algorithms are usually applied to a delay embedding of a time-series dataset (also called the lagged representation), rather than the raw dataset itself. Each lag is a previous observation of the time series; a delay vector is a chronologically ordered sequence of lags [43]:

$$S_n = \left(s_{n-(m-1)\tau},\ s_{n-(m-2)\tau},\ \ldots,\ s_n\right), \tag{8.23}$$

where S_n is the delay vector with dimension m, and τ specifies the delay between successive observations. Each delay vector can be considered as a point in a state space, whose dimensions are the chosen lags. According to Takens' embedding theorem [4], if a sufficient number of lags are taken (i.e., a sufficiently large delay vector constructed), the resulting state space is equivalent to the original state space of the system that gave rise to the time series. Thus, the evolution of the time series can be predicted from its trajectory through the embedding space, because this trajectory is equivalent to the original system's trajectory in its state space.

Takens' embedding theorem does not, however, provide a constructive method for determining the parameters m and τ; instead, we need to use heuristics to determine *adequate* values for both parameters. Mathematically, embeddings with different τ are equivalent to each other; however, in real-world data, the choice of the delay parameter has a significant influence on the utility of an embedding. Small values of τ generally lead to higher correlations between observations in each delay vector; and thus the distribution of delay vectors (and hence the apparent state-space trajectory) tend to be concentrated in a small region of the embedding space, potentially

obscuring important features of the trajectory. On the other hand, large values of τ tend to make observations in a delay vector poorly correlated. This tends to result in the delay vectors becoming a weakly differentiated cloud of points, with little apparent structure. Heuristics for determining a "best" value for τ include taking the first zero of the autocorrelation function, or the first minimum of the time-delayed mutual information. These are given in Eqs. (8.24) and (8.25), respectively [4]:

$$c_\tau = \frac{1}{\sigma^2} \langle (s_n - \langle s \rangle)(s_{n-\tau} - \langle s \rangle) \rangle , \tag{8.24}$$

where c_τ is the autocorrelation between values of s_n and $s_{n-\tau}$ where there is a time lag of τ between them. $\langle . \rangle$ indicates average over time, and σ^2 denotes the variance.

$$I(\tau) = \sum_{i,j} p_{ij}(\tau) \ln p_{ij}(\tau) - 2 \sum_i p_i \ln p_i, \tag{8.25}$$

where $I(\tau)$ is the mutual information between s_n and $s_{n-\tau}$. To compute this value, consider a histogram of s_n. p_i is the probability that s_n lies in the i-th interval, and p_{ij} is the joint probability that s_n has values in the i-th interval and $s_{n-\tau}$ has values in the j-th interval.

To determine the dimensionality of the embedding space, Kennel et al. [44] proposed applying the false nearest neighbors technique. The Euclidean distance between one delay vector and its rth nearest neighbor in the embedding space of dimension m is given by [44]:

$$R_d^2(n,r) = \sum_{k=0}^{m-1} \left[s_{n-k\tau} - s^r_{n-k\tau} \right]^2, \tag{8.26}$$

where R_d is the Euclidean distance and $s_{n-k\tau}$ are elements of the delay vector in the embedding space. When the dimension of the embedding space increases to $m+1$, the delay vectors have one more coordinate which is $s_{n-m\tau}$. The Euclidean distance between the delay vectors in the new embedding space is calculated as [44]:

$$R_{d+1}^2(n,r) = R_d^2(n,r) + \left[s_{n-m\tau} - s^r_{n-m\tau} \right]^2. \tag{8.27}$$

Thus, the false nearest neighbor method can be stated as the following criterion [44]:

$$\frac{\left| s_{n-m\tau} - s^r_{n-m\tau} \right|}{R_d(n,r)} > R_{\text{tol}}, \tag{8.28}$$

where R_{tol} is a threshold. That means increasing the embedding dimensionality must not increase the distance between two neighbors more than the given threshold. The estimated number of dimensions is determined by plotting the fraction of false nearest neighbors in the dataset against the number of dimensions, for several different values of the threshold R_{tol}. When all of the curves saturate at a low value, we consider that to be the best estimate of the necessary embedding dimensionality for the dataset. All three heuristics can be computed using the TISEAN software package [43].

8.3 Methodology

8.3.1 Experimental Design

The goal of the current chapter is to evaluate alternative time-series representations in forecasting with ANCFIS. We explore three different approaches for setting the length of the input windows and subsampling them. The first is the approach used in [2], the second is the delay embedding technique from Sect. 8.2.3, and the third is a hybrid of the two. Specifically:

- Method 1 is to make an ad hoc determination of the length of one "period" in the dataset. The input window is set to this length and is not subsampled. As this method was used in [2] for five of the six datasets, we will use the same period lengths as in that paper. For the sixth dataset (solar power forecasting), the length of a period is clearly 1 day (see our discussion in Sect. 8.3.2.1 for further details).
- Method 2 is to form a delay embedding, relying on the heuristics from Sect. 8.2.3 to guide our selection of the embedding dimensionality and delay. We will use the mutual-information heuristic to select the delay, and the false nearest neighbors technique to select the dimensionality. We can consider this a subsampling of an input window; for dimensionality m and delay τ, we select every τ-th sample from an input window of length $((m+1)\cdot\tau)+1$.
- Method 3 is to assume that the delay is always equal to 1 and to employ the false nearest neighbors technique for selecting the embedding dimensionality under that assumption. This will mean that the input window is again not subsampled.

Our experiments follow a common design in the time-series forecasting literature. We use a single-split design, with all elements of the training set chronologically earlier than elements of the testing set. The embedding dimension and delay are determined from the training set only, and are then applied to both the training and testing sets. The results of the three input representations are compared in terms of root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y_1})^2}{n}}, \tag{8.29}$$

where y_i is the expected value, $\hat{y}_1$ is the predicted value, and n is the number of inputs.

8.3.2 Datasets

8.3.2.1 Solar Power Dataset

This dataset was created in [5], as very few high-resolution solar power datasets are publicly available. It was developed from a public dataset recording air temperature

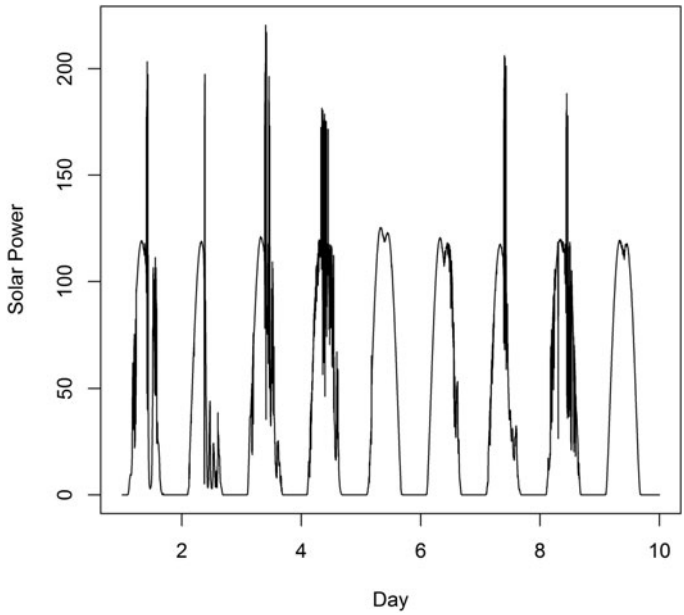


Fig. 8.2 Training set for the solar power dataset

(°C) and total solar radiation (W/m^2) measured every 1 min from May 30, 2008, to August 12, 2012, at the Lowry Range Solar Station. Total solar radiation is the sum of direct irradiance, diffuse irradiance, and ground-reflected radiation, and is measured by a LICOR LI-200 Pyranometer mounted 7 ft above ground level on a Rotating Shadow Band Radiometer (RSR). Air temperature is measured by a thermometer mounted 5 ft above ground level inside a naturally aspirated radiation shield [45]. These two measurements are the principal variables affecting power output from a photovoltaic cell; we convert them to an estimated power output using the model proposed in [46], following the specifications of a Photowatt PW 2650-24V panel. The result is a new time series recording solar power production at 1-min intervals over a period of 5 years, giving 2,212,520 observations.[1] For our experiments in the current chapter, we use data from July 31, 2012, to August 13, 2012, giving us 20,000 measurements. The dataset is split into 2/3 and 1/3 for the training and testing sets, respectively. A plot of the training set is given in Fig. 8.2.

In this dataset, the length of one period is clearly 1 day, or 1440 observations. When we attempted to take this entire period as an input window following Method 1, we found that the computation time is infeasibly long on our computer system (Intel(R) Core^{TM} 2 Duo CPU E8400 @ 3.00 GHz, 4 GB of memory). We were thus

[1] Available online at http://www.ualberta.ca/yazdanba/SolarData.txt.

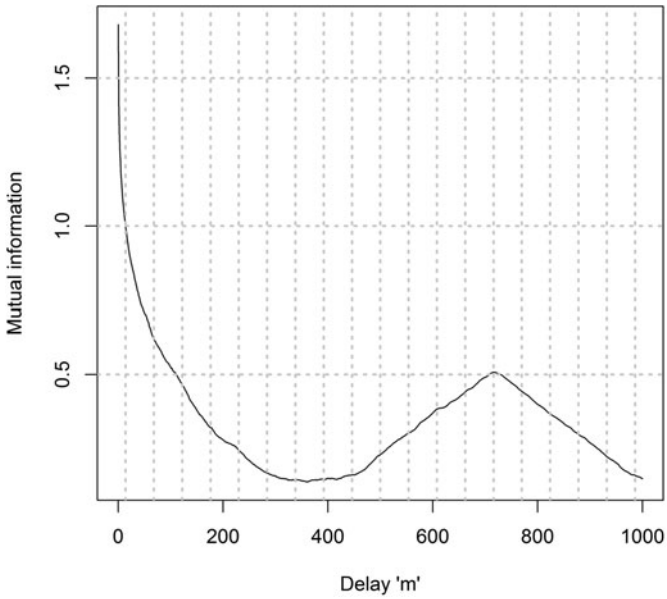


Fig. 8.3 Mutual information versus delay

forced to subsample this window; we take every tenth measurement, giving us 145 observations, which we concatenate together in chronological order. For Method 2, we plot the mutual information statistic versus delay in Fig. 8.3. The first minimum of the curve occurs at $\tau = 370$, and so, we adopt this value as our delay parameter. With this delay, the false nearest neighbor plot is given in Fig. 8.4. For all values of R_{tol} examined, the curves saturate at $m = 12$, and so, we adopt this value as our embedding dimension. We can also view this input as a subsampled window of length $(m-1)\tau = 4070$ data points. For Method 3, we set the delay $\tau = 1$, and rerun the false nearest neighbor procedure. This time, the apparent minimum embedding dimension is 15, and so, we adopt this as our window length.

8.3.2.2 Mackey–Glass Dataset

This dataset is a realization of the Mackey–Glass differential equation, a frequently used benchmark for testing time-series forecasting algorithms. The equation is given by [47]:

$$\dot{x}(t) = \frac{0.2x(t-\delta)}{1+x^{10}(t-\delta)} - 0.1x(t). \tag{8.30}$$

This equation is useful because it exhibits chaotic behavior for appropriate choices of the parameter δ. In particular, we follow the design in [47], where $124 < t < 1123$, $x(0) = 1.2$, $\tau = 17$, and the time step is 0.1. The first 500 data points are used as the

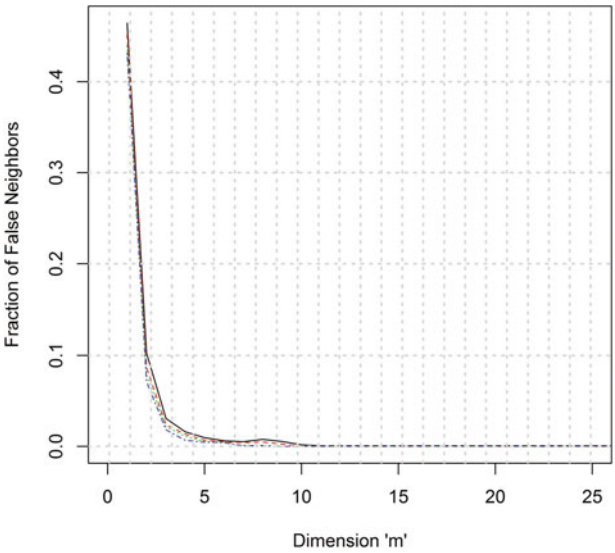


Fig. 8.4 Fraction of false nearest neighbors versus dimensionality

training set and the remaining data points form the testing set; this is also the design that was used in [2]. We omit the details of how the embedding dimension and delay are determined in the interest of brevity; we simply note that they followed the same process as in Sect. 8.3.2.1.

8.3.2.3 Santa Fe Laser A

The Santa Fe time-series forecasting competition held in 1991 has left us six datasets for use as benchmarks. The "Laser A" dataset is frequently used, as it also exhibits chaotic behavior. This dataset records the amplitude of an 81.5-micron 14NH3 cw (FIR) laser being controlled by the Lorenz system of equations for modeling turbulent flow; with appropriate choices of parameters, the Lorenz system is also chaotic. We normalized the data to the range [0, 1] (matching [2]). The dataset has 1000 data points; the first 900 data points are the training set, and the last 100 are the testing set (again, matching [2]).

8.3.2.4 Sunspot Dataset

This dataset consists of observations of a physical system: it is the average number of sunspots observed for each day in a calendar year, recorded from 1700–1979 (280 observations) [48]. The first 220-year measurements, years 1700–1920, are used as the training set and the remaining points form the testing set; this again matches [2].

Table 8.1 Results for the solar power dataset

Method	Input vector design		RMSE
	Dimension	Delay	
1	145	10	5.293
2	12	370	3.1057
3	15	1	4.846

8.3.2.5 Stellar Brightness

This dataset is also made up of observations of a physical system. The time-series records the daily observed brightness of a variable star on 600 successive midnights. The first 480 night measurements are used as the training set, and the remainder form the testing set. This is again the same split used in [2].

8.3.2.6 Waves

This dataset also records observations of a physical system, but this time in a laboratory setting. The time-series measures the oscillation of a cylinder suspended in a tank of water every 0.15 s. There are a total of 320 data points of which the first 256 data points form the training set and the last 64 points are the testing set. This also matches the split in [2].

8.4 Experimental Results

The following tables record the input window parameters and out-of-sample error for each of our input representations in Sect. 8.3.1, over each of the datasets described in Sect. 8.3.2. Table 8.1 presents our results for the solar power dataset.

Quite plainly, in this dataset, the traditional delay embedding was superior to the down-sampled "one period" input window, even though the smaller dimensionality provides far fewer tunable parameters in the consequent layer. It was also superior to the unit-delay input window created from Method 3, even though Method 2 again resulted in fewer dimensions.

Table 8.2 presents our results on the Mackey–Glass dataset; the results for Method 1 are taken from [2]. Interestingly, this dataset presents a completely different picture than the solar power dataset. The traditional delay embedding gave—by two orders of magnitude—the least accurate predictions, even though the most accurate approach (Method 3) differed only in the delay length (reflecting what was stated in Sect. 8.2.3; all values of the delay parameter are theoretically equivalent, but in practice a good choice of the delay parameter can significantly impact the performance of a forecasting algorithm).

Table 8.3 presents our results for the Santa Fe Laser A dataset. This time, Method 1 appears to be the best, while Method 2 yields the worst results. Table 8.4 presents

Table 8.2 Results for the Mackey-Glass dataset

Method	Input vector design		RMSE
	Dimension	Delay	
1 [2]	44	1	5.57e-4
2	9	11	0.015
3	9	1	5.29e-4

Table 8.3 Results for Santa Fe Laser A

Method	Input vector design		RMSE
	# Lags	Delay	
1 [2]	8	1	0.033
2	9	2	0.114
3	9	1	0.067

Table 8.4 Results for sunspot

Method	Input vector design		RMSE
	# Lags	Delay	
1 [2]	12	1	0.091
2	5	4	0.103
3	6	1	0.089

Table 8.5 Results for stellar brightness

Method	Input vector design		RMSE
	# Lags	Delay	
1 [2]	27	1	7.49e-3
2	6	7	1.4e-2
3	6	1	1.3e-2

Table 8.6 Results for Waves

Method	Input vector design		RMSE
	# Lags	Delay	
1 [2]	12	1	0.0032
2	4	4	0.00866
3	4	1	0.0104

our results for the sunspot dataset. This time, the traditional delay embedding is somewhat worse than Methods 1 and 3; however, the difference is not very large. In addition, Methods 1 and 3 are nearly indistinguishable from one another.

Table 8.5 presents our results for the Stellar Brightness dataset. This time, Method 1 is substantially better than Methods 2 or 3; furthermore, Method 2 is slightly less accurate than Method 3.

Table 8.6 presents our results for the Waves dataset. Once again, Method 1 proved to be the most accurate. However, this time Method 3 was less accurate than Method 2.

8.4.1 Discussion

As with many other experiments in pattern recognition, our general finding is that the "best" input representation for ANCFIS is dataset dependent. In five of our six datasets, the traditional delay embedding was clearly outperformed by the window-based approaches, but in our single largest dataset, Method 2 was clearly the best. Four times, Method 1 was either the best approach or virtually identical to Method 3 and superior to Method 2.

Our findings do, however, suggest which methods seem more likely to succeed in future experiments. Method 3 was the better approach once, and essentially tied with Method 1 on a second dataset. This method also seems to lead to lower embedding dimensionalities than Method 1. This still matters in ANCFIS, even though we have reduced the combinatorial explosion of rules seen in other machine-learning methods using orthogonal lags. A complexity analysis carried out in [2] indicates that the running time of both the least-squares estimate of the consequent parameters, and the VNCSA optimization of the CFS parameters, depend linearly on the length of the input vector (this explains why running ANCFIS on the full input window for the solar power dataset took an infeasibly long time). Thus, with Method 3 often providing strong results, and usually resulting in a significantly smaller input window, this seems to be the most effective initial approach to modeling a time series with ANCFIS. We would recommend that Method 1 be tried next, and finally, the traditional delay embedding.

8.5 Conclusions

In this study, we have explored three different approaches for representing time-series inputs for the ANCFIS machine-learning algorithm. We compared input windows based on an ad hoc determination of what constitutes one "period" in the dataset; the traditional delay embedding, guided by the mutual-information and false-nearest-neighbor heuristics; and the use of only the false-nearest-neighbor heuristic, across six time-series datasets. While the "best" method appears to be dataset dependent, we found enough evidence that we recommend Method 3 as the best combination of accuracy and expected computation time.

In future work, we intend to explore the use of ANCFIS as a stream data mining algorithm. An online-learning version of ANCFIS was developed in [29], meaning that incremental learning is possible in the ANCFIS framework. This characteristic, along with the demonstrated success of ANCFIS as a time-series forecasting algorithm, indicates that it may be an appropriate algorithm for modeling data streams. One of the datasets we will use in evaluating this possibility is the full solar power dataset.

Acknowledgments This research was supported in part by the Natural Science and Engineering Research Council of Canada under grant no. RGPIN 262151, and in part by Transport Canada under grant no. RES0017834.

References

1. D. Ramot et al., Complex fuzzy sets. IEEE Trans. Fuzzy Syst. **10**(2), 171–186 (2002)
2. Z. Chen et al., ANCFIS: A neurofuzzy architecture employing complex fuzzy sets. IEEE Trans. Fuzzy Syst. **19**(2), 305–322 (2011)
3. C. Li, T.-W. Chiang, Complex neuro-fuzzy self-learning approach to function approximation, in *Intelligent Information and Database Systems*. (Springer, Berlin, 2010), pp. 289–299
4. H. Kantz, T. Schreiber, *Nonlinear Time Series Analysis. Cambridge Nonlinear Science Series,* vol. xvi (Cambridge University Press, Cambridge, 1997), p. 304
5. O. Yazdanbaksh, A. Krahn, S. Dick, Predicting solar power output using complex fuzzy logic, in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, IEEE, 2013
6. D. Ramot et al., Complex fuzzy logic. IEEE Trans. Fuzzy Syst. **11**(4), 450–461 (2003)
7. S. Dick, Toward complex fuzzy logic. IEEE Trans. Fuzzy Syst. **13**(3), 405–414 (2005)
8. D.E. Tamir, L. Jin, A. Kandel, A new interpretation of complex membership grade. Int. J. Intelli. Syst. **26**(4), 285–312 (2011)
9. D.E. Tamir, A. Kandel, Axiomatic theory of complex fuzzy logic and complex fuzzy classes. Int. J. Comp. Commun. Control **VI**(3), (2011)
10. L. Běhounek, P. Cintula, Fuzzy class theory. Fuzzy Set. Syst. **154**(1), 34–55 (2005)
11. D.E. Tamir, M. Last, A. Kandel, The theory and applications of generalized complex fuzzy propositional logic, in *Soft Computing: State of the Art Theory and Novel Applications* (Springer, Berlin, 2013), pp. 177–192
12. D.E. Tamir et al., Discrete complex fuzzy logic, in Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American, IEEE, 2012
13. S.D. Zenzo, A many-valued logic for approximate reasoning. IBM J. Res. Dev. **32**(4), 552–565 (1988)
14. A.R. Salleh, Complex intuitionistic fuzzy sets, in *AIP Conference Proceedings*, 2012
15. K.T. Atanassov, Intuitionistic fuzzy sets. Fuzzy Set. Syst. **20**(1), 87–96 (1986)
16. R.R. Yager, A.M. Abbasov, Pythagorean membership grades, complex numbers, and decision making. Int. J. intell. Syst. **28**, 436–452 (2013)
17. G. Zhang et al., Operation properties and δ-equalities of complex fuzzy sets. Int. J. Approx. Reason. **50**(8), 1227–1249 (2009)
18. G. Zhang et al., Delta-equalities of Complex Fuzzy Relations, in Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, IEEE, 2010
19. A.U.M. Alkouri, A.R. Salleh, Complex Atanassov's intuitionistic Fuzzy relation, in *Abstract and Applied Analysis* (Hindawi Publishing Corporation, Nasr City, 2013)
20. D. Moses et al., Linguistic coordinate transformations for complex fuzzy sets, in *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE'99. 1999 IEEE International*, IEEE, 1999
21. H.T. Nguyen, A. Kandel, V. Kreinovich, Complex fuzzy sets: towards new foundations, in *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, IEEE, 2000
22. J. Man, Z. Chen, S. Dick, Towards inductive learning of complex fuzzy inference systems, in *Fuzzy Information Processing Society, 2007. NAFIPS'07. Annual Meeting of the North American*, IEEE, 2007
23. A. Hirose, *Complex-Valued Neural Networks*, vol. 400 (Springer, Berlin, 2012)
24. H.E. Michel, A.A.S. Awwal, Artificial neural networks using complex numbers and phase encoded weights. Appl. Opt. **49**(10), B71–B82 (2010)

25. A.J. Noest, Discrete-state phasor neural networks. Phys. Rev. A **38**(4), 2196 (1988)
26. I. Nishikawa, T. Iritani, K. Sakakibara, Improvements of the traffic signal control by complex-valued Hopfield networks, in *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, IEEE, 2006
27. Y. Li, Y.-T. Jang, Complex adaptive fuzzy inference systems, in Fuzzy Systems Symposium, 1996. 'Soft Computing in Intelligent Systems and Information Processing', Proceedings of the 1996 Asian, IEEE, 1996
28. A. Malekzadeh-A, M.-R. Akbarzadeh-T, Complex-valued adaptive neuro fuzzy inference system-CANFIS. Proc. World Aut. Congr. **17**, 477–482 (2004)
29. S. Aghakhani, S. Dick, An on-line learning algorithm for complex fuzzy logic, in *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, IEEE, 2010
30. C. Li, T.-W. Chiang, Function approximation with complex neuro-fuzzy system using complex fuzzy sets-a new approach. New Generat. Comput. **29**(3), 261–276 (2011)
31. C. Li, F. Chan, Complex-fuzzy adaptive image restoration-an artificial-bee-colony-based learning approach, in *Intelligent Information and Database Systems* (Springer, Berlin, 2011), pp. 90–99
32. Li, C. and F.-T. Chan, Knowledge discovery by an intelligent approach using complex fuzzy sets, in *Intelligent Information and Database Systems*. Springer, 320–329 (2012)
33. C. Li, T. Wu, F.-T. Chan, Self-learning complex neuro-fuzzy system with complex fuzzy sets and its application to adaptive image noise canceling. Neurocomputing **94**, 121–139 (2012)
34. C. Li, T.-W. Chiang, Intelligent financial time series forecasting: A complex neuro-fuzzy approach with multi-swarm intelligence. Int. J. Appl. Math. Comp. Sci. **22**(4), 787–800 (2012)
35. C. Li, T.-W. Chiang, Complex fuzzy computing to time series prediction a multi-swarm PSO learning approach, in *Intelligent Information and Database Systems* (Springer, Berlin, 2011), pp. 242–251
36. C. Li, T. Chiang, Complex Neurofuzzy ARIMA Forecasting—A New Approach Using Complex Fuzzy Sets, Fuzzy Systems, IEEE Transactions on **21**(3), 567–584 (2013)
37. C. Li, T.-W. Chiang, L.-C. Yeh, A novel self-organizing complex neuro-fuzzy approach to the problem of time series forecasting. Neurocomputing **99**, 467–476 (2013)
38. H. Sun, S. Wang, Q. Jiang, FCM-based model selection algorithms for determining the number of clusters. Pattern Recogn. **37**(10), 2027–2037 (2004)
39. J. Ma, G. Zhang, J. Lu, A method for multiple periodic factor prediction problems using complex fuzzy sets. IEEE Trans. Fuzzy Syst. **20**(1), 32–45 (2012)
40. A.U.M. Alkouri, A.R. Salleh, Linguistic variables, hedges and several distances on complex fuzzy sets. J. Intell. Fuzzy Syst. **26**(5), 2527–2535 (2014)
41. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-I. Inform. Sci. **8**(3), 199–249 (1975)
42. A. Deshmukh et al., Implementation of complex fuzzy logic modules with VLSI approach. Int. J. Comp. Sci. Netw. Security **8**, 172–178 (2008)
43. R. Hegger, H. Kantz, T. Schreiber, Practical implementation of nonlinear time series methods: The TISEAN package. Chaos **9**(2), 413–435 (1999)
44. M.B. Kennel, R. Brown, H.D. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys. Rev. A **45**(6), 3403 (1992)
45. NREL Staff, Lowry Range Solar Station (LRSS), Colarado State Land Board, http://www.nrel.gov/midc/lrss
46. A. Bellini et al., Simplified model of a photovoltaic module, in *Applied Electronics, 2009. AE 2009*, IEEE, 2009
47. J.S.R. Jang, ANFIS: Adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cyb. **23**(3), 665–685 (1993)
48. M. Li et al., Sunspot numbers forecasting using neural networks, in *Intelligent Control, 1990. Proceedings., 5th IEEE International Symposium on*, IEEE, 1990

Chapter 9
Multi-Subject Type-2 Linguistic Summaries of Relational Databases

Adam Niewiadomski and Izabela Superson

Abstract It is almost trivial to say that fuzzy sets and fuzzy logic are one of the most powerful computing methods for natural language-driven representation of information. Despite, in this chapter, we focus on extensions of Zadeh's concepts, and these extensions are called "general type-2 fuzzy sets" or "higher-order fuzzy sets." They cover some more specific groups, like interval-valued (interval type-2) fuzzy set, triangular or Gaussian type-2 fuzzy sets, and even the traditional fuzzy sets. To be more precise, here we are interested in applying type-2 fuzzy sets to relational database exploration. The point of departure is the concept of a *linguistic summary of a database* by Yager (1982).

In this chapter, basics of type-2 linguistic summarization of data (Niewiadomski 2008) are enhanced. In particular, we introduce new forms of linguistic summaries that use type-2 fuzzy sets as representations of linguistic information, and, so far, original methods for evaluating the degrees of truth based on cardinalities of type-2 fuzzy sets. The new forms are named "multi-subject" linguistic summaries, because they can describe more than one table or more than one set of records/objects collected in a database, for example, *More boys than girls play football well*. Thanks to that, the generated linguistic summaries—quasi-natural language sentences—are more interesting and human-oriented. Besides, using higher-order fuzzy sets helps us to deal with situations when membership degrees in models of linguistic terms are imprecise or too vague to use traditional fuzzy sets.

A. Niewiadomski (✉) · I. Superson
Institute of Information Technology, Łódź University of Technology,
Wólczańska 215, 90-924 Łódź, Poland
e-mail: Adam.Niewiadomski@p.lodz.pl

I. Superson
e-mail: 143104@edu.p.lodz.pl

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_9

9.1 Linguistic Summarization of Data: Essential Ideas and Goals

More than 30 years ago, R. R. Yager proposed the idea of *a linguistic summary of a (relational) database* [28], for example, *More than half of basketball players are very tall*. This simple concept appeared to be a direct answer to people's needs for quick and friendly receiving of large amounts of data and/or information. Most important is that the idea does not refer to any terse statistical method for aggregating data (mean, variation, standard deviation, etc.) but to fuzzy models of natural language expressions. Even if these expressions are less precise than numbers, for example, *more than half of objects* instead of 55.6 % *of objects* or *a very tall boy* instead of $6'\ 5''$*-tall boy*, they are commonly understood and provide knowledge on what the summarized data mean.

To be more precise, the concept of a linguistic summary is based on Zadeh's calculus of linguistically quantified propositions (statements) [31]. There are two basic forms of linguistic summaries (based on two forms of linguistically quantified propositions, respectively) presented in the literature [2, 5, 7, 8, 28, 30]:

$$Q\,P \text{ are/have } S[T] \tag{9.1}$$

for example, *Many boys are tall* [0.83], and

$$Q\,P \text{ being } W \text{ are/have } S[T] \tag{9.2}$$

for example, *Many boys who are teenagers, are tall* [0.63]. In both forms (9.1) and (9.2), Q is a *quantity in agreement*, for example, *Many, more than 900*, represented by an aggregation operator, for example, fuzzy quantifier or an ordered weighted averaging (OWA) operator [29]; P is the subject of the summary, for example, men, cars, or any other objects described in the summarized database, and S is a *summarizer*—a linguistic expression for properties of the objects—represented by a fuzzy set. The W symbol, appearing only in form (9.2), is *a qualifier*, represented by a fuzzy set, that determines additional and/or specific properties of the objects that the summary deals with. $T \in [0, 1]$ is *a degree of truth* and it determines how good (how informative, how true) the summary is; values of T are evaluated according to the Zadeh calculus of linguistically quantified propositions and/or to another different methods of evaluating [5, 14].

Obviously, this chapter is too short to present or even mention all methods and applications of linguistic summarization of relational databases (e.g., [6, 12, 32]). Moreover, we are not able to enumerate all the concepts for data summarization that are based on fuzzy sets but take into account assumptions different than the Yager originals (e.g., [1, 20, 21, 24]). What is to be done here is to introduce a *multi-subject linguistic summary* of a relational database. That means that a summary contains more than one subject P_1, for example, P_1 and P_2, and models of imprecise linguistic expressions (summarizers, quantifiers, etc.) are built using higher-order fuzzy sets.

Hence, the rest of the chapter is organized as follows: In Sect. 9.2, we make an overview on using higher-order fuzzy sets in data linguistic representation and summarization. In particular, we intend to enumerate papers that are of essential meaning for computing with words (CWW) techniques and type-2-fuzzy-set-based methods applied to linguistic representation and summarization of data collected in relational databases.

The new concept called a *multi-subject linguistic summary* of a relational database is presented in Sect. 9.3 of this chapter. We intend to construct and evaluate summaries, including type-2 linguistic summaries, related to more than one subject P that is represented by tuples in the summarized database $\mathbb{D}$, for example, to P_1 *and* P_2 or to P_1 *in comparison to* P_2. These two or more subjects are represented by nonfuzzy sets of tuples collected in separated tables in $\mathbb{D}$, or can be, if necessary, results of some other selecting, querying and/or filtering tuples, with respect to chosen values and/or attributes, for example, male and female. Obviously, these general explanations are explained in details and exemplified in Sect. 9.3 of the chapter.

Section 9.4 contains a brief description of the experiment (developing and exploring software) that helped us to present, determine and evaluate usefulness and performance of multi-subject linguistic summaries of relational databases. We show sample outputs of the program produced for a chosen database, and how users (intermediate and advanced) may affect summaries that are generated by the software.

Finally, there are conclusions on the usefulness of the concepts and the methods presented, drawn in Sect. 9.5.

9.2 An Overview of Higher-Order Fuzzy Sets in Data Representation and Linguistic Summarization

In general, type-2 fuzzy sets are adapted in fuzzy logic systems and in fuzzy representations of information and knowledge, if only linguistic terms applied in these systems are too vague to be represented by traditional fuzzy sets. Obviously, there is no point here to argue more for using type-2 fuzzy sets, because all the pros, as far as cons, are well and intuitively clarified by Jerry Mendel in [9] or in the web-published essay *Why We Need Type-2 Fuzzy Logic Systems*.[1]

It seems natural that capabilities of type-2 fuzzy sets, especially their flexibility and imprecise memberships, have been noticed soon as a very intuitive means to increase the power of methods for data linguistic representation. To be more specific, in linguistic summarization, type-2 fuzzy sets are applied as models of linguistic quantifiers Q, summarizers S, and qualifiers W, by notations of (9.1) and (9.2). To keep consistency with commonly used symbols for type-2 fuzzy sets, we denote

[1] E-version at `http://www.informit.com/articles/article.aspx?p=21312`, access: March 6, 2013. A comparative translation of this essay into Polish is elaborated by I. Superson and published at:`http://ics.p.lodz.pl/~aniewiadomski/ksr/WHY.pdf`.

type-2 quantifiers, summarizers, and qualifiers as $\widetilde{Q}$, $\widetilde{S}$, and $\widetilde{W}$, respectively. Higher-order fuzzy sets helps us to deal with situations when membership degrees of fuzzy models of statements are imprecise or too vague to use traditional fuzzy sets with real membership degrees. For instance, a method for dealing with imprecise (linguistic) memberships used to build type-2 summarizers for database of papers contributed to a conference is shown in [17].

One of the first proposals of extending the linguistic summarization methods via higher-order fuzzy sets has been given by Niewiadomski [10, 19]. In particular, the authors have used interval-valued fuzzy sets[2] [22, 25] to represent linguistic expressions (i.e., quantifiers, summarizers, and qualifiers) appearing in both (9.1) and (9.2) forms of data linguistic summaries. The paper [19] introduces also an extension of Zadeh's calculus of linguistically quantified propositions and properties of interval-valued fuzzy sets representing linguistic quantifiers. The direct consequence of such an approach is that degrees of truth, quality measures, and other characteristics of summaries and of sets applied (e.g., supports) are expressed with intervals in $\mathbb{R}$. Moreover, it requires to define and use extended arithmetic operations (addition, subtraction, power, etc., cf. [23]), for example, for $a = [\underline{a}, \overline{a}]$, $b = [\underline{b}, \overline{b}]$ in $\mathbb{R}$, $[\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] = [\underline{a} + \underline{b}, \overline{a} + \overline{b}]$ and $[\underline{a}, \overline{a}] - [\underline{b}, \overline{b}] = [\underline{a} - \overline{b}, \overline{a} - \underline{b}]$ and some partial order binary relations, for example, $a \leq b \Leftrightarrow \underline{a} \leq \underline{b} \wedge \overline{a} \leq \overline{b}$, to evaluate and compare interval-valued degrees of truth or other characteristics and measures (cf. [3, 23]). Besides, what is very important, it is necessary to define and apply such arithmetic and ordering operations that cover analogous operations for real numbers, looking at them as at "degenerated intervals," that is, $[\underline{a}, \overline{a}] = a \in \mathbb{R} \leftrightarrow \underline{a} = \overline{a}$. Imprecision measures for interval-valued fuzzy sets and quality measures for interval-valued linguistic summaries have been proposed and discussed in [11, 13, 16].

However, a question arises on how general type-2 fuzzy sets can be employed as elements of linguistic summaries and means for data linguistic representation. On one hand, this seems to be quite simple just to replace interval-valued quantifiers, summarizers, and qualifiers, with analogous general type-2 fuzzy sets, if only fuzzy membership degrees are necessary to represent one or more linguistic expressions used in a model. What is needed here, is to extend again (with respect to the interval-valued approach) the calculus of linguistically quantified statements, using scalar (!) cardinalities of type-2 fuzzy sets [17, 18]. It is also necessary to define convexity (via embedded type-1 fuzzy sets) and normality (via restrictions put on secondary membership degrees) of type-2 fuzzy sets used as models of linguistic quantifiers [17], having in mind that these new definitions must also apply to type-1 fuzzy sets [17, 15]. But on the other hand, one may see a problem or difficulty: what are proper criteria of choosing a group of type-2 fuzzy sets? Or in other words: if cardinalities (and similar related values) of interval type-2 fuzzy sets are expressed with or

[2] It is worth adding that nowadays interval-valued fuzzy sets are considered as interval type-2 fuzzy sets, and their cardinalities, imprecision measures, and other characteristics can be represented by different values, for example, scalar or fuzzy, not strictly intervals.

related to intervals (it means with/to type-1 fuzzy sets with rectangular membership functions), should cardinalities/characteristics of triangular type-2 fuzzy sets be expressed with or related to triangular membership functions, or cardinalities of Gaussian type-2 fuzzy sets with/to Gaussian membership functions? Obviously, such a choice would imply that only one group ("class") of type-2 fuzzy sets can be taken into account in a model, and it would significantly block the most expected properties of type-2-fuzzy-based representation of information—its versatility and flexibility [17].

Mostly because of the arguments provided above, the type-2-fuzzy-set-based linguistic summarization of relational databases presented in [15] and in [18] relies on scalar cardinalities for finite type-1 and type-2 fuzzy sets and on adequate scalar measures for infinite ones. The proposed scalar cardinalities/measures[3] are evaluated for all type-2 fuzzy sets, and these evaluation methods are independent of shapes of their particular secondary membership functions. The following formula is based on the proposal by Jang and Ralescu [4], and, following the original, called *nonfuzzy sigma-count*, nfσ-count:

$$|\widetilde{A}| = \text{nf}\sigma\text{-count}(\tilde{A}) = \sum_{x \in \mathbb{X}} \sup\{u_{\widetilde{A}} \in J_x : \mu_{\widetilde{A}}(x, u_{\widetilde{A}}) = 1\} \tag{9.3}$$

assuming that $\sup \emptyset = 0$, if there is no $u \in J_{x'}$ such that $\mu_{x'}(u) = 1$ for a given $x' \in \mathbb{X}$. Equation (9.3) is interpreted as the sum of those greatest primary membership degrees for which the secondary membership degrees are 1. Even an interval type-2 fuzzy set may have a scalar cardinality (or an adequate measure,[4] if infinite) evaluated, though the interval-valued approach to the calculus of linguistically quantified propositions assumed that these values are always intervals [19]. The same applies to type-1 fuzzy sets. Some other proposals of evaluating cardinalities of type-2 fuzzy sets are given in [18].

Also a few questions on finity/infinity and countability/uncountability of type-2 fuzzy sets in applications must be asked, since these two properties may exert influence on some other characteristics of sets, measures, and operations on related sets (supports, universes of discourse, domains of attributes in a database, etc.) and, as a consequence, on degrees of truth and other measures of type-2 linguistic summaries. Hence, they are crucial for the expected model, and an attempt of answering the questions is presented in [18]. Finity and countability of a type-2 fuzzy set is proposed to be related to *the support* of a type-2 fuzzy set $\widetilde{A}$ in $\mathbb{X}$, defined as type-1 fuzzy set supp $(\widetilde{A})$ such that $\mu_{\text{supp}(\widetilde{A})}(x) = \sup_{u \in J_x \setminus \{0\}} \mu_x(u)$. Since the properties of finity and countability are defined for type-1 fuzzy sets, we also may define these two properties for higher-order fuzzy sets [18].

---

[3] These measures for infinite type-2 fuzzy sets are based mainly on integrals of their principal or lower/upper membership functions. The intuitions expressed by the measures are very similar to the meaning of a scalar cardinality of a finite type-2 fuzzy set, so the authors of [17, 18] propose to call them *cardinality-like measures*, clm $(\cdot)$, of type-2 fuzzy sets, to underline this analogy.

[4] See Footnote 3.

Such a proposed set of linguistic summarization methods based on type-2 fuzzy sets allows users and/or programmers to use all described types of numbers and sets, that is, type-2 fuzzy sets, interval-valued fuzzy sets, type-1 fuzzy sets, intervals and real numbers, to be applied together in the same model/system. We believe it is a very important conclusion showing that higher-order fuzzy sets do not find any separated group of computing methods, but they can be considered as the most universal (from the practical point of view) and taking into account all "lower-order" sets and fuzzy sets as special cases, it is undeniably necessary to keep consistency between traditional fuzzy methods and higher-order fuzzy methods (not only from the point of view of data summarization, but also in fuzzy logic systems, etc.).

An important extension to linguistic summarization methods is proposed by Wu and Mendel [27]: the authors provide us with algorithms that make it possible to extract knowledge and formulate user-friendly messages using IF–THEN expressions, instead of commonly used in summaries aggregating data via linguistic quantifiers, OWA operators, etc. It must be said that actually new forms of data linguistic summaries and methods of generating them, as far as quality measures adequate for these new forms, are proposed in this chapter. To the best knowledge of the authors, no application combining both aggregation and IF-THEN approaches is proposed till now. Another interesting approach is presented by the same authors [26]; it is aggregation of data using interval type-2 fuzzy sets. Though it is not strictly data linguistic summarization in terms introduced by Yager and followed in this paper, using averaging operators and higher-order fuzzy sets to aggregate information, and (implicitly) to extract and represent knowledge from sets of data is, for sure, worth noticing in the context of this paper. A very similar criterion applies to the paper by Zhou et al. [33]; aggregation of uncertain information is really close to data summarization using fuzzy sets and higher-order fuzzy sets, especially, if it is directly related to the use of type-2 linguistic quantifiers, that is, linguistic quantifiers represented by type-2 fuzzy sets. All the three papers quoted in this paragraph [26, 27, 33] present a real applicational potential of higher-order fuzzy sets in data aggregation, summarization, and representation.

9.3 New Forms of Summaries: Multi-Subject Type-2 Linguistic Summaries

We are sure that the reader is well familiar with relational databases by Codd (1970), hence, the first paragraph of this section is only to set the denotations used in the next parts of the chapter. We assume that a summarized database consists of tables which are sets of tuples (usually called "records"), and one tuple is a representation of one real object (a child, person, car, transaction, etc.). This set of objects is here denoted as $\mathbb{Y} = \{y_1, \ldots, y_m\}$. Obviously, not all possible parameters of the objects are stored in the database, but only those relevant to a model are created. Table $\mathbb{D}$ consists of tuples d_i, $i = 1, 2, \ldots, m$ which are "rows" in the table: $\mathbb{D} = \{d_1, \ldots, d_m\}^T$, $m \in \mathbb{N}$ is the number of tuples in $\mathbb{D}$. Each tuple d_i consists of $n \in \mathbb{N}$ values of

Table 9.1 A sample database D collecting information on children in school age

ID	Gender	Age	Height
1	Girl	7	130
2	Boy	8	120
3	Boy	13	150
4	Girl	8	140
5	Girl	18	160

Table 9.2 The part of dataset D presented in Table 9.1 filtered with respect to the "boy" value of the "gender" attribute

ID	Gender	Age	Height
2	Boy	8	120
3	Boy	13	150

Table 9.3 The part of dataset D presented in Table 9.1 filtered with respect to the "girl" value of the "gender" attribute

ID	Gender	Age	Height
1	Girl	7	130
4	Girl	8	140
5	Girl	18	160

attributes $V_1, \ldots, V_n$ and the domains of the attributes are $\mathbb{X}_1, \ldots, \mathbb{X}_n$, respectively. The values of attributes express properties of objects, for example, height, salary, price, date, etc., and they are treated as "columns" of the table. Domains of the attributes are just sets of possible values taken into account in the database, for example, set $\mathbb{X}_1 = [50, 250]$ is the domain of V_1="height of person in centimeters." The value of attribute V_j for object y_i, is denoted as $V_j(y_i) \in \mathbb{X}_j$, $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$. Hence, the database D collecting information on elements from $\mathbb{Y} = \{y_1, \ldots, y_m\}$ is in the form:

$$D = \begin{Bmatrix} \langle V_1(y_1), & \ldots, & V_n(y_1)\rangle \\ \vdots & & \vdots \\ \langle V_1(y_m), & \ldots, & V_n(y_m)\rangle \end{Bmatrix} = \begin{Bmatrix} d_1 \\ \vdots \\ d_m \end{Bmatrix} \tag{9.4}$$

A sample database in the form of (9.4) is shown by Table 9.1. It is a part of the larger database summarized in the experiment described in Sect. 9.4. The table presents also the possibility of extracting two sets of subjects for multi-subject summaries; in this case, it is attribute "gender" that allows us to "split" the set of data into two subsets, represented by Tables 9.2 and 9.3.

It must be underlined that Tables 9.2 and 9.3 do not represent real database tables stored separately in a database management system; such a storage could appear inefficient and nonoptimal, especially, with respect to normal forms of relational database tables, popular optimization criteria for databases. The presented tables are only results of filtering operations performed on $\mathbb{D}$ (represented by Table 9.1) with respect to values of a chosen attribute, here: "gender," for both "boys" and "girls" values.

What is crucial for the main idea of the chapter, is that *two separated sets of objects*, previously stored as one set in $\mathbb{D}$, are now distinguished. These two sets represent *different subjects P_1 and P_2, of multi-subject linguistic summaries* that are now presented.

The first form of a multi-subject type-2 linguistic summary is proposed:

$$\widetilde{Q}\ P_1\ \textit{relatively to}\ P_2\ \textit{are}\ \widetilde{S}_1 \tag{9.5}$$

where $\widetilde{Q}$ is a type-2 fuzzy quantifier, P_1 and P_2 are the subjects of the summary and $\widetilde{S}_1$ is a summarizer, represented by a type-2 fuzzy set. The degree of truth of summary (9.5) is evaluated with formula (9.6):

$$T(\widetilde{Q}\ P_1\ \textit{relatively to}\ P_2\ \textit{are}\ \widetilde{S}_1) = \mu_{\widetilde{Q}}\left(\frac{\frac{1}{M_{P_1}}\text{card}(\widetilde{S}_{1_{P_1}})}{\frac{1}{M_{P_1}}\text{card}(\widetilde{S}_{1_{P_1}}) + \frac{1}{M_{P_2}}\text{card}(\widetilde{S}_{1_{P_2}})}\right) \tag{9.6}$$

where

$$\text{card}(\widetilde{S}_{1_{P_1}}) = \sum_{i=1}^{m} \max\{u_{\widetilde{S}_1} : \mu_{\widetilde{S}_1}(d_i, u_{\widetilde{S}_1}) = 1 \wedge d_i \in^* P_1\} \tag{9.7}$$

and $\text{card}(\widetilde{S}_{1_{P_2}})$, analogously. The notation $d_i \in^* P_1$ means that d_i is a tuple representing P_1 subject. M_{P_1} and M_{P_2} are numbers of tuples representing subjects P_1 and P_2, respectively,

$$M_{P_1} = \sum_{i=1}^{m} t_i \textit{oneof} \tag{9.8}$$

where t_i,

$$t_{i_{P_1}} = \begin{cases} 1, & \text{if } d_i \in^* P_1 \\ 0, & \text{otherwise} \end{cases} \tag{9.9}$$

For instance,

$$t_{i_{\text{boys}}} = \begin{cases} 1, & \text{if } V_j(d_i) = \text{“boy”} \\ 0, & \text{if } V_j(d_i) = \text{“girls”} \end{cases} \tag{9.10}$$

and $V_j = Gender$. An example of a summary in the form of (9.5) is now given below:

$$\textit{Most of boys relatively to girls are tall}\ [0.456] \tag{9.11}$$

where $\widetilde{Q} = \textit{most of}$, $P_1 = \textit{boys}$, $P_2 = \textit{girls}$, $\widetilde{S}_1 = \textit{tall}$.

The second form of a multi-subject summary proposed here is given below:

$$\widetilde{Q}\ P_1\ \textit{relatively to}\ P_2\ \textit{being}\ \widetilde{S}_2\ \textit{are}\ \widetilde{S}_1 \tag{9.12}$$

where $\widetilde{S}_2$ is a qualifier, cf. (9.2). The degree of truth of the summary is evaluated via formula (9.13).

$$T(\widetilde{Q}\ P_1 \textit{ relatively to } P_2 \textit{ being } \widetilde{S}_2 \textit{ are } \widetilde{S}_1) = \mu_{\widetilde{Q}}\left(\frac{\frac{1}{M_{P_1}}\text{card}(\widetilde{S}_{1_{P_1}} \cap \widetilde{S}_{2_{P_1}})}{\frac{1}{M_{P_1}}\text{card}(\widetilde{S}_{2_{P_1}}) + \frac{1}{M_{P_2}}\text{card}(\widetilde{S}_{2_{P_2}})}\right) \tag{9.13}$$

where $\widetilde{Q}$ is a relative quantifier, P_1 and P_2 are the subjects of the summary, $\widetilde{S}_2$ is a qualifier related to both P_1 and P_2 subjects, and $\widetilde{S}_1$ is a summarizer.

$$\text{card}(\widetilde{S}_{1_{P_1}} \cap \widetilde{S}_{2_{P_1}}) = \sum_{i=1}^{m} \min\big\{\max\{u_{\widetilde{S}_1} : \mu_{\widetilde{S}_1}(d_i, u_{\widetilde{S}_1}) = 1 \wedge d_i \in^* P_1\}, \max\{u_{\widetilde{S}_2} : \mu_{\widetilde{S}_2}(d_i, u_{\widetilde{S}_2}) = 1 \wedge d_i \in^* P_1\}\big\} \tag{9.14}$$

and card $(\widetilde{S}_{2_{P_1}})$, card $(\widetilde{S}_{2_{P_2}})$, $d_i \in^* P_1$, analogously to (9.5). An example of a summary in the form of (9.12) is now presented as follows:

About two-third of boys relatively to girls being teenagers, are tall [0.390] (9.15)

where $\widetilde{Q} =$ *about two-third*, $P_1 =$ *boys*, $P_2 =$ *girls*, $\widetilde{S}_1 =$ *tall*, $\widetilde{S}_2 =$ *teenagers*.

Summaries in form (9.12) allow us to retrieve information about selected subjects' features $\widetilde{S}_1$, according to other subjects conditions (specific features that both subjects must possess). It means that in this case, the tuples taken into account represent boys and girls who are qualified by S_2 as teenagers.

The third form of a multi-subject linguistic summary is proposed as:

$$\widetilde{Q}\ P_1 \textit{ being } \widetilde{S}_2 \textit{ relatively to } P_2 \textit{ are } \widetilde{S}_1 \tag{9.16}$$

and its degree of truth is evaluated with formula (9.17).

$$T(\widetilde{Q}\ P_1 \textit{ being } \widetilde{S}_2 \textit{ relatively to } P_2 \textit{ is } \widetilde{S}_1) = \mu_{\widetilde{Q}}\left(\frac{\frac{1}{M_{P_1}}\text{card}(\widetilde{S}_{1_{P_1}} \cap \widetilde{S}_{2_{P_1}})}{\frac{1}{M_{P_1}}\text{card}(\widetilde{S}_{1_{P_1}}) + \frac{1}{M_{P_2}}\text{card}(\widetilde{S}_{1_{P_2}})}\right) \tag{9.17}$$

where $\widetilde{Q}$ is a relative quantifier, P_1 and P_2 are the subjects of the summary, $\widetilde{S}_2$ is a qualifier referring only to subject P_1, and $\widetilde{S}_1$ is a summarizer. An example of such a summary is given (9.16) as:

About half of boys being teenagers relatively to girls, are tall [0.256] (9.18)

where $\widetilde{Q} =$ *about half*, $P_1 =$ *boys*, $P_2 =$ *girls*, $\widetilde{S}_1 =$ *tall*, $\widetilde{S}_2 =$ *teenagers*.

Summaries in the form of (9.16) allow users to retrieve information on some selected features of subjects, according to chosen conditions given for subject P_1 only (i.e., some specific features that only subject P_1 must fulfill). It means that tuples taken into account by the summary represent both P_1 and P_2 subjects, that is, boys and girls, but only P_1 is additionally qualified by $\widetilde{S}_2$ (here, as teenagers).

Note that none of the older forms of linguistic summaries, that is, (9.1) and (9.2), is able to represent the relations between different groups of objects and their properties, for example, boys and girls, and their height, age, etc. On the other hand, these relations can be easily discovered and expressed in an interesting way using multi-subject linguistic summaries. For older, non-multi-subject methods, the only opportunity is to generate summaries that includes the preselected set of objects, for example, boys or girls, as qualifier $\widetilde{W}$, see (9.2), for example, *About half of BOYS are tall*, where *BOYS* is a qualifier $\widetilde{W}$.

Now, in Sect. 9.4, we show results of an experiment: a database containing information is summarized using newly proposed forms of linguistic summaries. The results are finally related to those obtained via non-multi-subject summaries (cf. (9.1) and (9.2)).

9.4 Application Example: Descriptions of Databases Content Using Multi-Subject Type-2 Linguistic Summaries

The application created for testing purposes is based on the Java 1.7 SE Platform. The database used in the experiment contains data of children from the age of 7 up to 18 years old. The data describes factors like children height, mass, date of birth, living conditions such as number of rooms in flat, number of people in family, family financial situation, etc. The database contains data on 13,956 children, including 6991 boys and 6965 girls.

In the experiment, generated summaries are assumed to discover how children age and gender is related to their height. Two subjects taken into account in multi-subject summaries are boys and girls. The process of logical splitting the database into two separated sets of data describing boys and girls, respectively, is exemplified by Tables 9.1, 9.2, and 9.3. The relative quantifiers are used in the experiment called *most of, about two-third*, and *about half* to represent the quantities in agreement for selected subjects, and to evaluate degrees of truth of the multi-subject summaries. The proposed membership functions for the quantifiers used in the experiment are presented in Figs. 9.1 and 9.2.

The generated summaries are based on qualifiers and summarizers represented by Gaussian type-2 fuzzy sets. Sample summarizers and qualifiers are:

- Tall (height)
- Short (height)
- In early school age (age)
- Teenager (age)

For instance, the label *tall* is represented by type-2 fuzzy set

$$\widetilde{A} = \{\langle x, u_{tall}, \mu_x(u_{tall})\rangle : x \in [150, 195], u_{tall} \in [0, 1]\} \tag{9.19}$$

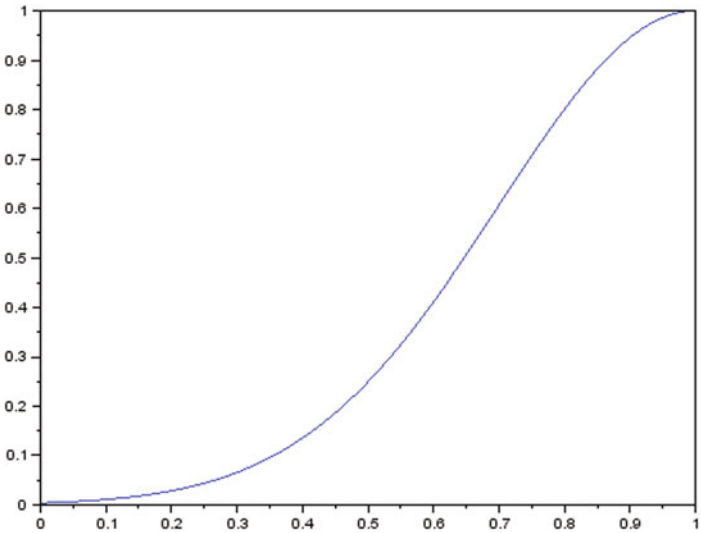


Fig. 9.1 The membership function of the *MOST OF* linguistic quantifier

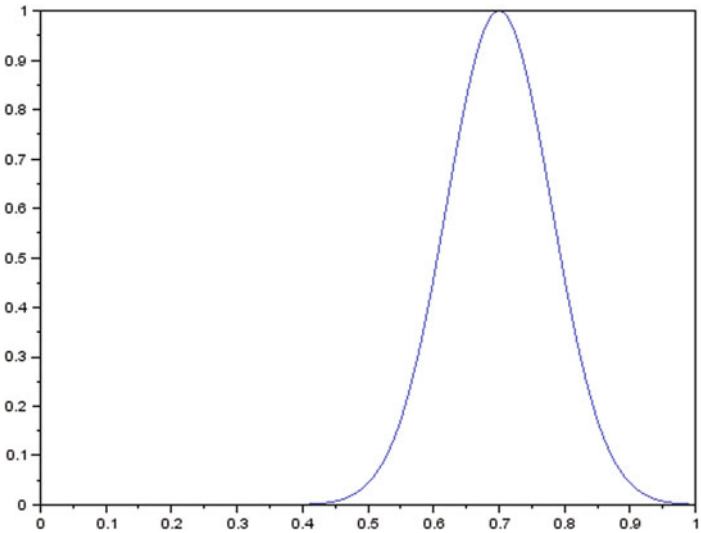


Fig. 9.2 The membership function of the *ABOUT TWO-THIRD* linguistic quantifier

where

$$u_{tall} = \begin{cases} \frac{2(x-150)}{45}, & \text{if} \quad 150 \leq x \leq \frac{150+195}{2} \\ \frac{2(195-x)}{45}, & \text{if} \quad \frac{150+195}{2} \leq x \leq 195 \\ 0, & \text{if} \quad x \leq 150 \ \text{ or } \ x \geq 195 \end{cases} \tag{9.20}$$

and

$$\mu_x(u_{tall}) = \exp\left(-\frac{1}{2}\left(\frac{u_{tall} - m(x)}{0.1}\right)^2\right) \tag{9.21}$$

Analogously, the label *teenage* is represented by type-2 fuzzy set

$$\widetilde{B} = \{\langle x, u_{teenager}, \mu_x(u_{teenager})\rangle x \in [13, 18], u_{teenager} \in [0, 1]\} \tag{9.22}$$

where

$$u_{teenager} = \begin{cases} \frac{2(x-13)}{5}, & \text{if} \quad 13 \le x \le \frac{13+18}{2} \\ \frac{2(18-x)}{5}, & \text{if} \quad \frac{13+18}{2} \le x \le 18 \\ 0, & \text{if} \quad x \le 13 \quad \text{or} \quad x \ge 18 \end{cases} \tag{9.23}$$

and

$$\mu_x(u_{teenager}) = \exp\left(-\frac{1}{2}\left(\frac{u_{teenager} - m(x)}{0.1}\right)^2\right) \tag{9.24}$$

The output of the experimental software, that is, the generated summaries, are collected in Table 9.4. For each summary, the evaluated degree of truth (column T) and the form of the summary (column "Summary form"), are provided. The "Summary form" refers to the number of equation of this chapter, that means (9.5), (9.12), (9.16) refer to the first, the second, and the third form of a multi-subject linguistic summary introduced in this chapter, respectively, and (9.1) and (9.2) refer to the older forms of linguistic summaries.

According to expert opinion, the results are intuitively correct. The first eight summaries 1–8 are constructed according to the first form of a multi-subject linguistic summary (9.5). Analysing their degrees of truth we can see that there is no disproportion between information on boys or girls, for example, summaries 1 and 2 are of very similar degree of truth.

The next summaries, 9–16, lead us to the conclusion that there are more tall girls than tall boys in early school age; for example, summary 9 contains the opposite statement, that is, boys relatively to girls in the early school age are tall, and it is of the very low degree of truth. The situation changes for teenagers; there are more tall teenager boys than teenager girls, summary 10. Also, it cannot be said that in comparison to boys, major part of teenager girls are short, because it would mean that there are many teenager girls from 103 to 150 cm height, summary 10. (The reader must take into consideration that children in the dataset was from 103 to 195 cm tall, so in this circumstances, a short child is more or less between 103 and 150 cm tall.)

Using the older forms of the linguistic summaries, that is, (9.1) and (9.2), with Q, S, W represented by type-2 fuzzy sets gives us extensive information about the analysed dataset. Extending summarizations set from Table 9.4 with summaries in known forms, 17–20, completes our knowledge on the summarized database. For example, information on proportions between boys and girls, amount of tall boys, tall

Table 9.4 Sample multi-subject summaries

No.	Summary	T	Summary form
1.	Most of girls relatively to boys are in early school age	0.495	(9.5)
2.	Most of boys relatively to girls are in early school age	0.505	
3.	Most of girls relatively to boys are teenagers	0.511	
4.	About half of boys relatively to girls are teenagers	0.994	
5.	Most of girls relatively to boys are tall	0.206	
6.	Most of boys relatively to girls are tall	0.298	
7.	Most of girls relatively to boys are short	0.249	
8.	About two-thirds of boys relatively to girls are short	0.043	
9.	Most of boys relatively to girls being in early school age, are tall	0.004	(9.12)
10.	Most of boys relatively to girls being teenagers, are tall	0.129	
11.	Most of girls relatively to boys being in early school age, are short	0.124	
12.	About half of girls relatively to boys being teenagers, are short	0	
13.	Most of girls being in early school age, relatively to boys are short	0.101	(9.16)
14.	Most of girls being teenagers, relatively to boys are short	0.004	
15.	Most of boys being teenagers, relatively to girls are tall	0.098	
16.	About two-thirds of boys in early school age relatively to girls, are tall	0	
17.	About half of children are girls	1	(9.1)
18.	Most of children are in early school age	0.32	
19.	About two-thirds of boys are tall	0	
20.	Most of boys being tall are teenagers	0.031	(9.2)

girls, teenager boys, teenager girls, teenage boys which are tall, early school-aged girls which are short, are provided. The dedicated algorithm can evaluate degrees of truth, select the best (the most informative) summaries and present them in clear and intuitive forms, for example, *About half of children are girls; Most of boys relatively to girls are all; About two-third of girls being in early school age, relatively to boys are tall*. The last conclusion shows in particular, that newly proposed multi-subject summaries of databases do not exclude the older forms, but can be used together with them, to extend and improve the process of extracting and representing knowledge from large datasets.

9.5 Conclusions

The goal of the research is to elaborate fuzzy-based methods that make it possible to describe provided data in as human-friendly a manner as possible, preferably with natural or quasi-natural language. In this chapter, we presented an original

concept that extends the known methods of data linguistic summarization and representation—multi-subject linguistic summaries of relational databases. In particular, we put emphasis on new and more interesting forms of linguistic summaries, that were based on describing one subject P only, until now (for bibliographical references, see Sect. 9.2). The new forms of type-2 linguistic summaries are given by Eqs. (9.5), (9.12), and (9.16), in Sect. 9.3. The details of evaluating degrees of truth of the new forms are presented in Sect. 9.3, too. From the point of view of an average user, the most important detail of the multi-subject linguistic summaries is that the output of the proposed method remains texts/messages composed by a human. Sample application of multi-subject linguistic summaries to a system providing users with natural language information on a chosen set of data, is described in Sect. 9.4. We believe the proposals here introduced, that is, describing more than one subject by a summary, may have potential to extend the summarization methods already known in the scientific literature.

References

1. P. Bosc, O. Pivert, Fuzzy querying in conventional databases, in *Fuzzy Logic for the Management of Uncertainty*, ed. by L.A. Zadeh, J. Kacprzyk (Wiley, New York, 1992), pp. 645–671
2. R. George, R. Srikanth, Data summarization using genetic algorithms and fuzzy logic, in *Genetic Algorithms and Soft Computing*, ed. by F. Herrera, J.L. Verdegay (Physica, Heidelberg, 1996), pp. 599–611
3. H. Ishibuchi, H. Tanaka, Multiobjective programming in optimisation of the interval objective function. Eur. J. Oper. Res. **48**, 219–225 (1990)
4. L.C. Jang, D. Ralescu, Cardinality concept for type-two fuzzy sets. Fuzzy Sets Syst. **118**, 479–487 (2001)
5. J. Kacprzyk, R.R. Yager, Linguistic summaries of data using fuzzy logic. Int. J. Gen. Syst. **30**, 133–154 (2001)
6. J. Kacprzyk, S. Zadrożny, Flexible querying using fuzzy logic: An implementation for Microsoft Access, in *Flexible Query Answering System*, ed. by T. Andreasen, H. Christiansen, H.L. Larsen (Kluwer, Boston, 1997), pp. 247–275
7. J. Kacprzyk, R.R. Yager, S. Zadrożny, A fuzzy logic based approach to linguistic summaries of databases. Int. J. Appl. Math. Comput. Sci. **10**, 813–834 (2000)
8. J. Kacprzyk, R.R. Yager, S. Zadrożny, Fuzzy linguistic summaries of databases for an efficient business data analysis and decision support, in *Knowledge Discovery for Business Information Systems*, ed. by W. Abramowicz, J. Żurada (Kluwer Academic, Boston, 2001)
9. J.M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions* (Prentice-Hall, Upper Saddle River, 2001)
10. A. Niewiadomski, Interval-valued linguistic variables: An application to linguistic summaries, in *Issues in Intelligent Systems: Paradigms*, ed. by O. Hryniewicz, J. Kacprzyk, J. Koronacki, S.T. Wierzchoń (Academic Publishing House EXIT, Warsaw, 2005), pp. 167–183
11. A. Niewiadomski, Interval-valued quality measures for linguistic summaries, in *Issues in Soft Computing: Theory and Applications*, ed. by P. Grzegorzewski, M. Krawczak, S. Zadrożny (Academic Publishing House EXIT, Warsaw, 2005), pp. 211–224
12. A. Niewiadomski, News generating via fuzzy summarization of databases. Lect. Notes Comput. Sci. **3831**, 419–429 (2006)
13. A. Niewiadomski, Interval-valued linguistic summarization of data. New quality measures and applications. Int. J. Inf. Technol. Intell. Comput. **2**(3), 469–488 (2007)

14. A. Niewiadomski, Six new informativeness indices of data linguistic summaries, in *Advances in Intelligent Web Mastering*, ed. by P.S. Szczepaniak, K.M. Węgrzyn-Wolska (Springer, Berlin, 2007), pp. 254–259
15. A. Niewiadomski, A type-2 fuzzy approach to linguistic summarization of data. IEEE Trans. Fuzzy Syst. **16**(1), 198–212 (2008)
16. A. Niewiadomski, Linguistic expressions represented by interval-valued fuzzy sets and their imprecision measures. in *Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume II: Applications*, ed. by K.T. Atanassov, O. Hryniewicz, J. Kacprzyk, M. Krawczak, Z. Nahorski, E. Szmidt, S. Zadrożny (Academic Publishing House EXIT, Warsaw, 2008), pp. 105–119
17. A. Niewiadomski, *Methods for the Linguistic Summarization of Data: Applications of Fuzzy Sets and Their Extensions* (Academic Publishing House EXIT, Warsaw, 2008)
18. A. Niewiadomski, On finity, countability, cardinalities, and cylindric extensions of type-2 fuzzy sets in linguistic summarization of databases. IEEE Trans. Fuzzy Syst. **18**(3), 532–545 (2010)
19. A. Niewiadomski, J. Ochelska, P.S. Szczepaniak, Interval-valued linguistic summaries of databases. Control Cybern. **35**(2), 415–444 (2006)
20. G. Raschia, N. Mouaddib, SAINTETIQ: A fuzzy set-based approach to database summarization. Fuzzy Sets Syst. **129**, 137–162 (2002)
21. D. Rasmussen, R.R. Yager, A fuzzy SQL summary language for data discovery, in *Fuzzy Information Engineering: A Guided Tour of Applications*, ed. by D. Dubois, H. Prade, R.R. Yager (Wiley, New York, 1997), pp. 253–264
22. R. Sambuc, Fonctions ϕ-floues. application à l'aide au diagnostic en pathologie thyroidienne. Ph.D. thesis (in French), Univ. Marseille, France, 1975
23. A. Sengupta, T.K. Pal, D. Chakraborty, Interpretation of inequality constraints involving interval coefficients and a solution to interval linear programming. Fuzzy Sets Syst. **119**, 129–138 (2001)
24. R. Srikanth, R. Agrawal, Mining quantitative association rules in large relational tables. The 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Canada, 4–6 June, 1996, pp. 1–12, 1996
25. I.B. Turksen, Interval-valued fuzzy sets based on normal forms. Fuzzy Sets Syst. **20**, 191–210 (1986)
26. D. Wu, J.M. Mendel, Aggregation using the linguistic weighted average and interval type-2 fuzzy sets. IEEE Trans. Fuzzy Syst. **15**(6), 1145–1161 (2007)
27. D. Wu, J.M. Mendel, Linguistic summarization using if-then rules and interval type-2 fuzzy sets. IEEE Trans. Fuzzy Syst. **19**(1), 136–151 (2011)
28. R.R. Yager, A new approach to the summarization of data. Inf. Sci. **28**, 69–86 (1982)
29. R.R. Yager, On ordered weighted averaging operators in multicriteria decision making. IEEE Trans. Syst. Man Cybern. **18**, 183–190 (1988)
30. R.R. Yager, M. Ford, A.J. Canas, An approach to the linguistic summarization of data. Proceedings of 3rd International Conference, Information Processing and Management of Uncertainty in Knowledge-Based System, Paris, France, 2–6 July, 1990, pp. 456–468, 1990
31. L.A. Zadeh, A computational approach to fuzzy quantifiers in natural languages. Comput. Math. Appl. **9**, 149–184 (1983)
32. S. Zadrożny, *Imprecise Queries and Linguistic Summaries of Databases* (Academic Publishing House EXIT, Warsaw, 2006) (in Polish)
33. S.M. Zhou, F. Chiclana, R.I. John, J.M. Garibaldi, Type-1 OWA operators for aggregating uncertain information with uncertain weights induced by type-2 linguistic quantifiers. Fuzzy Sets Syst. **159**(24), 3281–3296 (2008)

Chapter 10
Bio-Inspired Optimization of Interval Type-2 Fuzzy Controller Design

Oscar Castillo

Abstract This chapter presents a general framework for designing interval type-2 fuzzy controllers based on bio-inspired optimization techniques. The problem of designing optimal type-2 fuzzy controllers for complex nonlinear plants under uncertain environments is of crucial importance in achieving good results for real-world applications. Traditional approaches have been using genetic algorithms or trial and error approaches; however, results tend to be not optimal or require very large design times. More recently, bio-inspired optimization techniques, like ant colony optimization or particle swarm intelligence, have also been applied on optimal design of fuzzy controllers. In this chapter, we show how bio-inspired optimization techniques can be used to obtain results that outperform traditional approaches in the design of optimal type-2 fuzzy controllers.

10.1 Introduction

We describe, in this chapter, new methods for building intelligent systems using type-2 fuzzy logic and bio-inspired optimization techniques. Bio-inspired optimization includes techniques such as particle swarm optimization (PSO), ant colony optimization (ACO) and genetic algorithms (GAs) that have been applied in numerous optimization problems. In this chapter, we are extending the use of fuzzy logic to a higher order, which is called type-2 fuzzy logic [4, 33]. Combining type-2 fuzzy logic with bio-inspired optimization techniques, we can build powerful hybrid intelligent systems that can use the advantages that each technique offers in solving complex control problems.

Fuzzy logic is an area of soft computing that enables a computer system to reason with uncertainty [44]. A fuzzy inference system consists of a set of if–then rules defined over fuzzy sets [4, 23]. Fuzzy sets generalize the concept of a traditional set by allowing the membership degree to be of any value between 0 and 1. This corresponds, in the real world, to many situations where it is difficult to decide in an unambiguous manner if something belongs or not to a specific class. Fuzzy expert

O. Castillo (✉)
Tijuana Institute of Technology, Calzada Tecnologico s/n, 22379 Tijuana, CP, Mexico
e-mail: ocastillo@hafsamx.org

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_10

systems, for example, have been applied with some success to problems of decision, control, diagnosis, and classification, just because they can manage the complex expert reasoning involved in these areas of application. The main disadvantage of fuzzy systems is that they cannot adapt to changing situations. For this reason, it is a good idea to combine fuzzy logic with neural networks or GAs, because either one of these last two methodologies could give adaptability to the fuzzy system [12, 40]. On the other hand, the knowledge that is used to build these fuzzy rules is uncertain. Such uncertainty leads to rules whose antecedents or consequents are uncertain, which translates into uncertain antecedent or consequent membership functions [20, 33]. Type-1 fuzzy systems, like the ones mentioned above, whose membership functions are type-1 fuzzy sets, are unable to directly handle such uncertainties. Type-2 fuzzy sets are fuzzy sets whose membership grades themselves are type-1 fuzzy sets; they are very useful in circumstances where it is difficult to determine an exact membership function for a fuzzy set [2, 6, 7, 31, 32].

Uncertainty is an inherent part of intelligent systems used in real-world applications [3, 5]. The use of new methods for handling incomplete information is of fundamental importance [13, 17, 30, 36, 43]. Type-1 fuzzy sets used in conventional fuzzy systems cannot fully handle the uncertainties present in the intelligent systems [14, 15]. Type-2 fuzzy sets that are used in type-2 fuzzy systems can handle such uncertainties in a better way because they provide us with more parameters [8, 19, 21]. This chapter deals with the design of intelligent systems using interval type-2 fuzzy logic for minimizing the effects of uncertainty produced by the instrumentation elements, environmental noise, etc. Experimental results include simulations of feedback control systems for nonlinear plants using type-1 and type-2 fuzzy logic controllers (FLCs); a comparative analysis of the systems' response is performed, with and without the presence of uncertainty [28, 29]. The main contribution of the chapter is the proposed approach for the design of type-2 FLCs using bio-inspired optimization algorithms [1, 8].

We describe the use of ACO for the problem of finding the optimal intelligent controller for an autonomous wheeled mobile robot, in particular for the problem of tuning a fuzzy controller of the Sugeno type. In our study case, the controller has four inputs, each of them with two membership functions, and we consider the interpolation point for every pair of membership function as the main parameter and their individual shape as secondary ones in order to achieve the tuning of the fuzzy controller by using an ACO algorithm [30]. Simulation results show that using ACO and coding the problem with just three parameters instead of six allow us to find an optimal set of membership function parameters for the fuzzy control system with less computational effort needed [8].

This chapter also describes the application of the optimization algorithm for particle swarm known by its acronym as PSO, used to adjust the parameters of membership functions of an FLC to find the optimal intelligent control for a wheeled autonomous mobile robot. Results of several simulations show that the PSO is able to optimize the type-1 and type-2 FLCs for this specific application [29]. We use the PSO method to find the parameters of the membership functions of a type-2 FLC in order to minimize the state error for nonlinear systems. The PSO is used to find the optimal type-2 FLC

to achieve regulation of the output and stability of the closed-loop system [28]. For this purpose, we change the values of the cognitive, social, and inertia variables in the PSO. The simulation results, with the optimal FLC implemented in SIMULINK, show the feasibility of the proposed approach.

In general, the abovementioned applications of type-2 fuzzy logic in intelligent control are representative of the state-of-the-art in the area. However, we also have to mention that there exist applications of type-2 fuzzy logic in pattern recognition [22], time-series prediction [12], and classification [17], which have been successful in the real world, but are not the main concern in this chapter. There have also been important theoretical advances on type-2 fuzzy logic that have enabled more efficient processing and type-reduction [33, 36, 43], which have helped obtaining solutions to real-world problems [26, 27, 35, 37–39, 41, 42].

10.2 Design of Interval Type-2 Fuzzy Controllers

Uncertainty is an inherent part of intelligent systems used in real-world applications [38, 42]. The use of new methods for handling incomplete information is of fundamental importance [4, 33]. Type-1 fuzzy sets used in conventional fuzzy systems cannot fully handle the uncertainties present in intelligent systems. Type-2 fuzzy sets that are used in type-2 fuzzy systems can handle such uncertainties in a better way because they provide us with more parameters [2, 11]. This section deals with the design of intelligent systems using interval type-2 fuzzy logic for minimizing the effects of uncertainty produced by the instrumentation elements, environmental noise, etc. Experimental results include simulations of feedback control systems for nonlinear plants using type-1 and type-2 FLCs; a comparative analysis of the systems' response is performed, with and without the presence of uncertainty [8].

10.2.1 Introduction to the Design

Uncertainty affects decision making and appears in a number of different forms. The concept of information is fully connected with the concept of uncertainty. The most fundamental aspect of this connection is that the uncertainty involved in any problem-solving situation is a result of some information deficiency, which may be incomplete, imprecise, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way. Uncertainty is an attribute of information. The general framework of fuzzy reasoning allows handling much of this uncertainty; fuzzy systems employ type-1 fuzzy sets, which represent uncertainty by numbers in the range [0, 1]. When something is uncertain, like a measurement, it is difficult to determine its exact value, and of course, the type-1 fuzzy sets make more sense than using sets. However, it is not reasonable to use an accurate membership function for something uncertain, so in this case what we need is another type of fuzzy sets, those that are able to handle

these uncertainties, the so-called type-2 fuzzy sets. So, the amount of uncertainty in a system can be reduced by using type-2 fuzzy logic because it offers better capabilities to handle linguistic uncertainties by modeling vagueness and unreliability of information [4, 33].

In this section, we deal with the application of interval type-2 fuzzy control to nonlinear dynamic systems [24, 25, 34]. It is a well-known fact, that in the control of real systems, the instrumentation elements (instrumentation amplifier, sensors, digital to analog, analog to digital converters, etc.) introduce some sort of unpredictable values in the information that has been collected [8]. So, the controllers designed under idealized conditions tend to behave in an inappropriate manner. Since uncertainty is inherent in the design of controllers for real-world applications, we are presenting how to deal with this problem using type-2 FLC, to reduce the effects of imprecise information [33]. We are supporting this statement with experimental results, qualitative observations, and quantitative measures of errors. For quantifying the errors, we utilized three widely used performance criteria, which are integral of square error (ISE), integral of the absolute value of the error (IAE), and integral of the time multiplied by the absolute value of the error (ITAE).

10.2.2 Fuzzy Logic Systems

In this section, a brief overview of type-1 and type-2 fuzzy systems is presented. This overview is considered as necessary to understand the basic concepts needed to understand the methods and algorithms presented later in the chapter [4, 33].

10.2.2.1 Type-1 Fuzzy Logic Systems

In the 1940s and 1950s, many researchers proved that dynamic systems could be mathematically modeled using differential equations. In these works, we have the foundations of the control theory, which in addition to the transform theory (Laplace's theory) provided an extremely powerful means of analyzing and designing control systems. These theories were developed until the 1970s, when the area was called systems theory to indicate its definitiveness.

Soft computing techniques have become an important research topic, which can be applied in the design of intelligent controllers. These techniques have tried to avoid the abovementioned drawbacks, and they allow us to obtain efficient controllers, which utilize the human experience in a more natural form than the conventional mathematical approach. In the cases in which a mathematical representation of the controlled system is difficult to obtain, the process operator has the knowledge and the experience to express the relationships existing in the process behavior.

An FLS, described completely in terms of type-1 fuzzy sets, is called a type-1 fuzzy logic system (type-1 FLS). It is composed of a knowledge base, which comprises the information given by the process operator in the form of linguistic control rules,

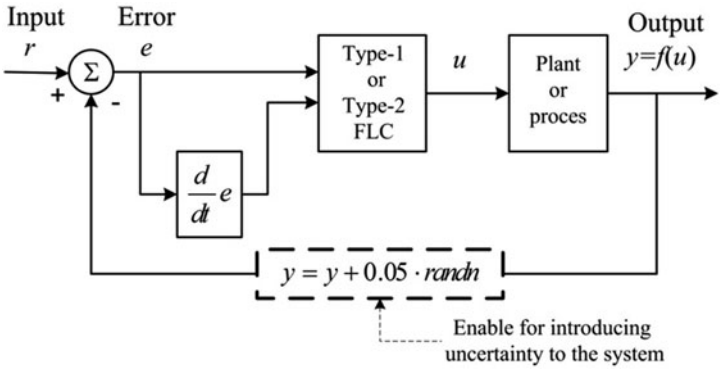


Fig. 10.1 System used for obtaining the experimental results for control

a fuzzification interface, which has the effect of transforming crisp data into fuzzy sets, and an inference system, which uses the fuzzy sets in conjunction with the knowledge base to make inferences by means of a reasoning method. Finally, a defuzzification interface translates the fuzzy control action, so obtained, to a real control action using a defuzzification method.

In this section, the implementation of the fuzzy controller in terms of type-1 fuzzy sets has two input variables, which are the error $e(t)$, the difference between the reference signal and the output of the process, as well as the error variation $\Delta e(t)$:

$$e(t) = r(t) - y(t) \tag{10.1}$$

$$\Delta e(t) = e(t) - e(t-1), \tag{10.2}$$

so the control system can be represented as in Fig. 10.1.

10.2.2.2 Type-2 FLSs

If for a type-1 membership function, as in Fig. 10.2, we blur it to the left and to the right, as illustrated in Fig. 10.3, then a type-2 membership function is obtained. In this case, for a specific value x', the membership function (u') takes on different values, which are not all weighted the same, so we can assign an amplitude distribution to all of those points.

Doing this for all $x \in X$, we create a three-dimensional membership function—a type-2 membership function—that characterizes a type-2 fuzzy set. A type-2 fuzzy set, $\tilde{A}$, is characterized by the membership function:

$$\tilde{A} = \left\{ \left((x,u), \mu_{\tilde{A}}(x,u) \right) | \forall x \in X, \forall u \in J_x \subseteq [0,1] \right\}, \tag{10.3}$$

in which $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$. Another expression for $\tilde{A}$ is

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \qquad J_x \subseteq [0,1], \tag{10.4}$$

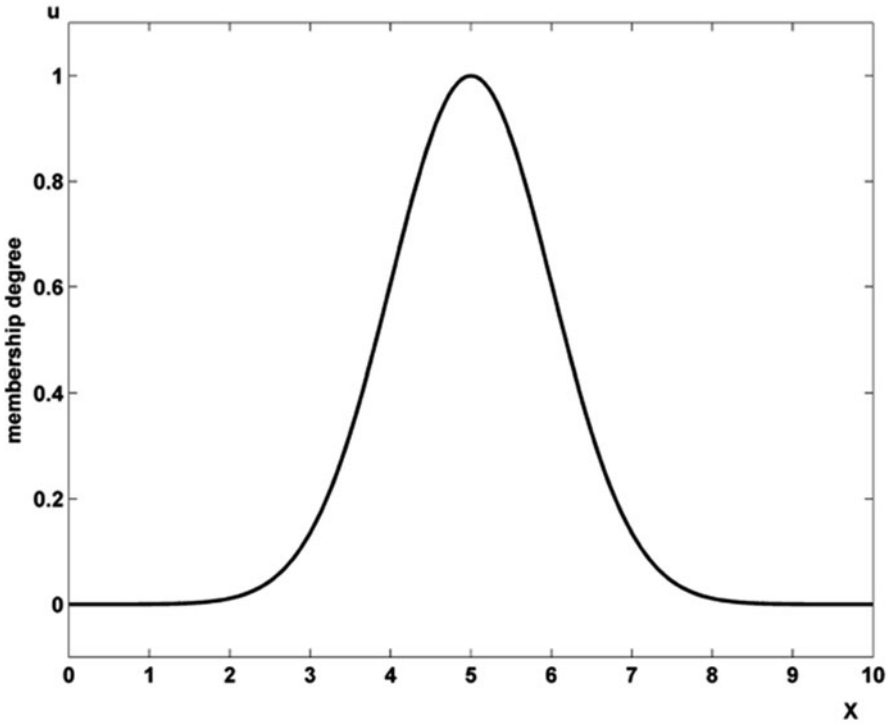


Fig. 10.2 Type-1 membership function

where $\int\int$ denotes the union over all admissible input variables x and u. For discrete universes of discourse, $\int$ is replaced by Σ [33]. In fact, $J_x \subseteq [0,1]$ represents the primary membership of x, and $\mu_{\tilde{A}}(x,u)$ is a type-1 fuzzy set known as the secondary set. Hence, a type-2 membership grade can be any subset in [0, 1], the primary membership, and corresponding to each primary membership, there is a secondary membership (which can also be in [0, 1]) that defines the possibilities for the primary membership. Uncertainty is represented by a region, which is called the footprint of uncertainty (FOU). If $\mu_{\tilde{A}}(x,u) = 1, \forall u \in J_x \subseteq [0,1]$, then we have an interval type-2 membership function, as shown in Fig. 10.4. The uniform shading for the FOU represents the entire interval type-2 fuzzy set, and it can be described in terms of an upper membership function $\bar{\mu}_{\tilde{A}}(x)$ and a lower membership function $\underline{\mu}_{\tilde{A}}(x)$.

An FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain. On the other hand, type-2 FLSs are very useful in circumstances where it is difficult to determine an exact membership function, and there are measurement uncertainties.

It is known that type-2 fuzzy sets enable modeling and minimizing the effects of uncertainties in rule-based FLS. Unfortunately, type-2 fuzzy sets are more difficult

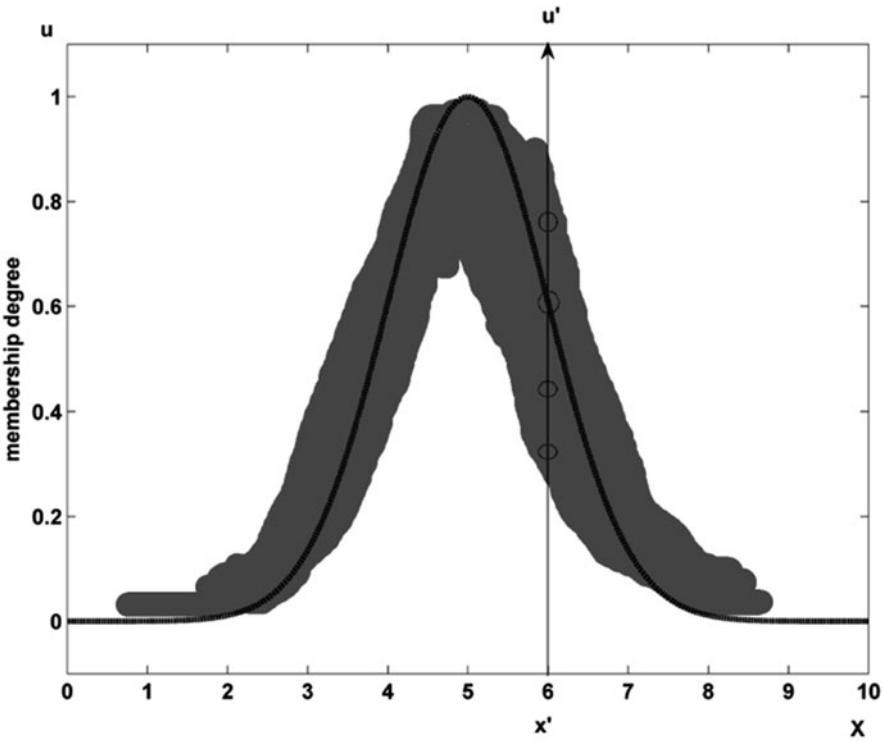


Fig. 10.3 Blurred type-1 membership function

to use and understand than type-1 fuzzy sets; hence, their use is not widespread yet. As a justification for the use of type-2 fuzzy sets, at least four sources of uncertainties not considered in type-1 FLSs are mentioned:

1. The meanings of the words that are used in the antecedents and consequents of rules can be uncertain (words mean different things to different people).
2. Consequents may have histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not agree all.
3. Measurements that activate a type-1 FLS may be noisy and therefore uncertain.
4. The data used to tune the parameters of a type-1 FLS may also be noisy.

All of these uncertainties translate into uncertainties about fuzzy set membership functions. Type-1 fuzzy sets are not able to directly model such uncertainties because their membership functions are totally crisp. On the other hand, type-2 fuzzy sets are able to model such uncertainties because their membership functions are themselves fuzzy. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity.

A type-2 FLS is again characterized by if–then rules, but its antecedent or consequent sets are now of type 2. Type-2 FLSs can be used when the circumstances are

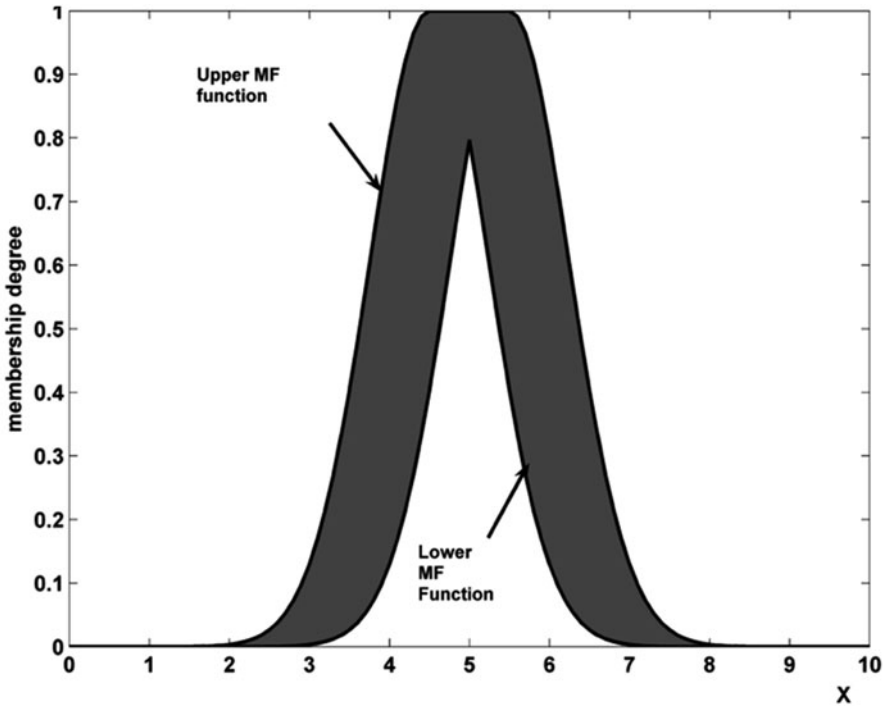


Fig. 10.4 Interval type-2 membership function

too uncertain to determine exact membership grades such as when the training data is corrupted by noise. Similar to a type-1 FLS, a type-2 FLS includes a fuzzifier, a rule base, fuzzy inference engine, and an output processor, as we can see in Fig. 10.5. The output processor includes type reducer and defuzzifier; it generates a type-1 fuzzy set output (from the type reducer) or a crisp number (from the defuzzifier). Now, we will explain each of the blocks of Fig. 10.5.

10.2.2.2.1 Fuzzifier

The fuzzifier maps a crisp point $x = \left(x_1, \ldots, x_p\right)^T \in X_1 \times X_2 \times \ldots \times X_p \equiv X$ into a type-2 fuzzy set $\tilde{A}_x$ in X, interval type-2 fuzzy sets in this case. We will use type-2 singleton fuzzifier; in a singleton fuzzification, the input fuzzy set has only a single point on nonzero membership. $\tilde{A}_x$ is a type-2 fuzzy singleton if $\mu_{\tilde{A}_x}(x) = 1/1$ for $x = x'$ and $\mu_{\tilde{A}_x}(x) = 1/0$ for all other $x \neq x'$.

10.2.2.2.2 Rules

The structure of rules in a type-1 FLS and a type-2 FLS is the same, but in the latter the antecedents and the consequents will be represented by type-2 fuzzy sets. So for

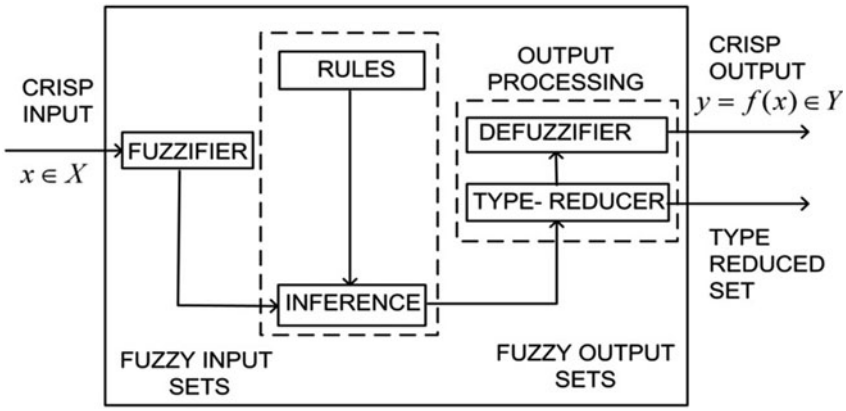


Fig. 10.5 Type-2 fuzzy logic system

a type-2 FLS with p inputs, $x_1 \in X_1, \ldots, x_p \in X_p$ and one output $y \in Y$, multiple input single output (MISO), if we assume there are M rules, the lth rule in the type-2 FLS can be written as follows:

$$R^1 : \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } x_\text{p} \text{ is } \tilde{F}_p^l, \text{ then } y \text{ is } \tilde{G}^l \qquad l = 1, \ldots, M \tag{10.5}$$

10.2.2.2.3 Inference

In the type-2 FLS, the inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. It is necessary to compute the join $\sqcup$ (unions) and the meet Π (intersections), as well as extended sup-star compositions (sup-star compositions) of type-2 relations. If $\tilde{F}_1^l \times \cdots \times \tilde{F}_p^l = \tilde{A}^l$, Eq. (10.5) can be rewritten as

$$R^l : \tilde{F}_1^l \times \cdots \times \tilde{F}_p^l \rightarrow \tilde{G}^l = \tilde{A}^l \rightarrow \tilde{G}^l \qquad l = 1, \ldots, M \tag{10.6}$$

R^l is described by the membership function $\mu_{R^l}(x, y) = \mu_{R^l}(x_1, \ldots, x_p, y)$, where

$$\mu_{R^l}(x, y) = \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(x, y) \tag{10.7}$$

can be written as

$$\begin{aligned} \mu_{R^l}(x, y) &= \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(x, y) = \mu_{\tilde{F}_1^l}(x_1) \Pi \cdots \Pi \mu_{\tilde{F}_p^l}(x_p) \Pi \mu_{\tilde{G}^l}(y) \\ &= [\Pi_{i=1}^p \mu_{\tilde{F}_i^l}(x_i)] \Pi \mu_{\tilde{G}^l}(y). \end{aligned} \tag{10.8}$$

In general, the p-dimensional input to R^l is given by the type-2 fuzzy set $\tilde{A}_x$ whose membership function is

$$\mu_{\tilde{A}_x}(\text{x}) = \mu_{\tilde{x}_1}(x_1) \Pi \cdots \Pi \mu_{\tilde{x}p}(x_p) = \Pi_{i=1}^p \mu_{\tilde{x}i}(x_i) \tag{10.9}$$

where $\tilde{X}_i (i = 1, \ldots, p)$ are the labels of the fuzzy sets describing the inputs. Each rule R^l determines a type-2 fuzzy set $\tilde{B}^l = \tilde{A}_x \cdot R^l$ such that

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{A}_x \cdot R^l} = \sqcup_{x \in X} \left[\mu_{\tilde{A}_x}(x) \Pi \ \mu_{R^l}(x, y) \right] y \in Y \ l = 1, \ldots, M. \quad (10.10)$$

This equation is the input/output relation in Fig. 10.5 between the type-2 fuzzy set that activates one rule in the inference engine and the type-2 fuzzy set at the output of that engine.

In the FLS, we used interval type-2 fuzzy sets and meet under product t-norm, so the result of the input and antecedent operations, which are contained in the firing set $\Pi_{i=1}^{p} \mu_{\tilde{F}_{ii}}(x_i' \equiv F^l(x'))$, is an interval type-1 set:

$$F^l(x') = \left[\underline{f}^l(x'), \bar{f}^l(x') \right] \equiv \left[\underline{f}^l, \bar{f}^l \right], \quad (10.11)$$

where

$$\underline{f}^l(\mathrm{x}') = \underline{\mu}_{\tilde{F}_1^l}(x_1') * \cdots * \underline{\mu}_{\tilde{F}_p^l}(x_p') \quad (10.12)$$

and

$$\bar{f}^l(\mathrm{x}') = \bar{\mu}_{\tilde{F}_1^l}(x_1') * \cdots * \bar{\mu}_{\tilde{F}_p^l}(x_p'), \quad (10.13)$$

where * is the product operation.

10.2.2.2.4 Type-Reducer

The type-reducer generates a type-1 fuzzy set output, which is then converted in a crisp output through the defuzzifier. This type-1 fuzzy set is also an interval set; for the case of our FLS, we used center of sets (cos) type-reduction, $Y_{\cos}$, which is expressed as

$$\begin{aligned} Y_{\cos}(\mathrm{x}) &= [y_l, y_r] \\ &= \int_{y^1 \in [y_l{}^1, y_r{}^1]} \cdots \int_{y^M \in [y_l{}^M, y_r{}^M]} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \cdots \int_{f^M \in [\underline{f}^M, \bar{f}^M]} 1 / \frac{\sum_{i=1}^{M} f^i y^i}{\sum_{i=1}^{M} f^i} \end{aligned} \quad (10.14)$$

This interval set is determined by its two end points, y_l and y_r, which corresponds to the centroid of the type-2 interval consequent set $\tilde{G}^i$,

$$C_{\tilde{G}^i} = \int_{\theta_1 \in J_{y1}} \cdots \int_{\theta_N \in J_{yN}} 1 / \frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i} = [y_l{}^i, y_r{}^i] \quad (10.15)$$

Before the computation of $Y_{\cos}(x)$, we must evaluate Eq. (10.15), and its two end points, y_l and y_r. If the values of f_i and y_i that are associated with y_l are denoted

as f_l^i and y_l^i, respectively, and the values of f_i and y_i that are associated with y_r are denoted as f_r^i and y_r^i, respectively, from Eq. (10.14), we have

$$y_l = \frac{\sum_{i=1}^{M} f_l{}^i y_l{}^i}{\sum_{i=1}^{M} f_l{}^i} \tag{10.16}$$

$$y_r = \frac{\sum_{i=1}^{M} f_r{}^i y_r{}^i}{\sum_{i=1}^{M} f_r{}^i}. \tag{10.17}$$

10.2.2.2.5 Defuzzifier

From the type-reducer, we obtain an interval set $Y_{\cos}$; to defuzzify it, we use the average of y_l and y_r. So, the defuzzified output of an interval singleton type-2 FLS is

$$y(x) = \frac{y_l + y_r}{2}. \tag{10.18}$$

In this chapter, we are simulating the fact that the instrumentation elements (instrumentation amplifier, sensors, digital-to-analog and analog-to-digital converters, etc.) are introducing some sort of unpredictable values in the collected information. In the case of the implementation of the type-2 FLC, we have the same characteristics as in type-1 FLC, but we used type-2 fuzzy sets as membership functions for the inputs and for the output.

10.2.2.3 Performance Criteria

For evaluating the transient closed-loop response of a computer control system, we can use the same criteria that are used normally for adjusting constants in proportional integral derivative (PID) controllers. These are:

1. Integral of square error (ISE):

$$\text{ISE} = \int_0^\infty [e(t)]^2 dt \tag{10.19}$$

2. Integral of the absolute value of the error (IAE):

$$\text{IAE} = \int_0^\infty |e(t)| dt \tag{10.20}$$

3. Integral of the time multiplied by the absolute value of the error (ITAE):

$$\text{ITAE} = \int_0^\infty t|e(t)| dt \tag{10.21}$$

Criteria selection depends on the desired type of control response; the errors will contribute different for each criterion. So, we have those large errors that will increase the value of ISE more heavily than to IAE. ISE will favor responses with smaller overshoot for load changes, but ISE will give longer settling time. In ITAE, the time appears as a factor, and therefore, ITAE will penalize heavily errors that occur late in time, but virtually ignore errors that occur early in time. Designing using ITAE will give us the shortest settling time, but it will produce the largest overshoot among the three criteria considered. Designing considering IAE will give us an intermediate result; in this case, the settling time will not be so large than using ISE or so small than using ITAE, and the same applies for the overshoot response. The selection of a particular criterion is dependent on the type of desired response.

10.3 Optimization of Fuzzy Controllers Using the ACO Metaheuristic

In this section, we describe the application of a simple ACO (S-ACO) as a method of optimization for membership functions' parameters of an FLC in order to find the optimal intelligent controller for an autonomous wheeled mobile robot [8]. The simulation results show that the ACO outperforms a GA in the optimization of FLCs for an autonomous mobile robot [8].

10.3.1 Introduction

Nowadays, fuzzy logic is one of the most used methods of computational intelligence and with the best future. This is possible thanks to the efficiency and simplicity of fuzzy systems since they use linguistic terms similar to those that human beings use [44].

The complexity for developing fuzzy systems can be found at the time of deciding which are the best parameters of the membership functions, the number of rules, or even the best granularity that could give us the best solution for the problem that we want to solve [4].

A solution for the abovementioned problem is the application of evolutionary algorithms for the optimization of fuzzy systems [6, 14, 16, 28, 38, 42]. Evolutionary algorithms can be a useful tool because of its capabilities of solving nonlinear problems, well-constrained or even nondeterministic polynomial-time hard (NP-hard) problems. However, recently, there have also been proposals of new optimization techniques based on biological or natural phenomena that have achieved good results in real-world problems, for example, ACO, the bat algorithm, firefly algorithm, chemical optimization, and others [1].

This section describes the application of one recent bio-inspired algorithm, such as the ACO [24] as a method of optimization of the parameters of the membership functions of the FLC in order to find the best intelligent controller for an autonomous wheeled mobile robot [8, 30].

10.3.2 S-ACO algorithm

ACO is a probabilistic technique that can be used for solving problems that can be reduced to finding a good path along graphs. This method is inspired by the behavior presented by ants in finding paths from the nest or colony to the food source [24].

The S-ACO is an algorithmic implementation that adapts the behavior of real ants to the solutions of minimum-cost path problems on graphs [35]. A number of artificial ants build solutions for a certain optimization problem and exchange information about the quality of these solutions making allusion to the communication systems of the real ants.

Let us define the graph $G = (V, E)$, where V is the set of nodes and E is the matrix of the links between nodes. G has $n_G = |V|$ nodes. Let us define L^K as the number of hops in the path built by the ant k from the origin node to the destiny node. Therefore, it is necessary to find:

$$Q = \left\{ q_a, \ldots, q_f \,|q_1 \in C \right\}, \tag{10.22}$$

where Q is the set of nodes representing a continuous path with no obstacles; $q_a, \ldots, q_f$ are former nodes of the path and C is the set of possible configurations of the free space. If $x^k(t)$ denotes a Q solution in time t, $f(x^k(t))$ expresses the quality of the solution. The general steps of S-ACO are the following:

- Each link (i, j) is associated with a pheromone concentration denoted as τ_{ij}.
- A number $k = 1, 2, \ldots, n_k$ are placed in the nest.
- On each iteration, all ants build a path to the food source (destiny node). For selecting the next node, a probabilistic equation is used:

$$p_{ij}^k \begin{cases} \dfrac{\tau_{ij}^k}{\sum_{j \in N_{ij}^k} \tau_{ij}^{\alpha}(t)} & if \quad j \in N_i^k \\ 0 & if \quad j \notin N_i^k \end{cases}, \tag{10.23}$$

  where N_i^k is the set of feasible nodes (in a neighborhood) connected to node i with respect to ant k, τ_{ij} is the total pheromone concentration of link ij, and α is a positive constant used again for the pheromone influence.

- Remove cycles and compute each route weight $f(x^k(t))$. A cycle could be generated when there are no feasible candidate nodes, that is, for any i and any k, $N_i^k = \varnothing$; then, the predecessor of that node is included as a former node of the path.
- Pheromone evaporation is calculated with the equation

$$\tau_{ij}(t) \leftarrow (1 - \rho)\tau_{ij}(t) \tag{10.24}$$

  where $\rho \in [0,1]$ is the evaporation rate value of the pheromone trail. The evaporation is added to the algorithm in order to force the exploration of the ants and avoid

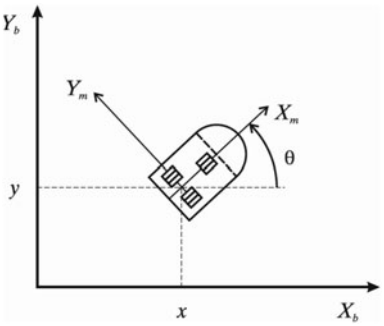


Fig. 10.6 Wheeled mobile robot [10]

premature convergence to suboptimal solutions. For $\rho = 1$, the search becomes completely random.

- The update of the pheromone concentration is realized using the equation

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t) \tag{10.25}$$

where $\Delta\tau_{ij}^{k}$ is the amount of pheromone that an ant k deposits in a link ij in a time t.

- Finally, the algorithm can be ended in three different ways:
  - When a maximum number of epochs has been reached
  - When it has found an acceptable solution, with $f(x_k(t)) < \varepsilon$
  - When all ants follow the same path

10.3.3 Problem Statement

The model of the robot considered in this chapter is a unicycle mobile robot (see Fig. 10.6), that consists of two driving wheels mounted on the same axis and a front free wheel [8, 28, 30].

A unicycle mobile robot is an autonomous, wheeled vehicle capable of performing missions in fixed or uncertain environments. The robot body is symmetrical around the perpendicular axis, and the center of mass is at the geometrical center of the body. It has two driving wheels that are fixed to the axis that passes through C and one passive wheel prevents the robot from tipping over as it moves on a plane. In what follows, it is assumed that the motion of the passive wheel can be ignored in the dynamics of the mobile robot presented by the following set of equations:

$$M(q)\dot{\vartheta} + C(q,\dot{q})\dot{\vartheta} + D\vartheta = \tau + F_{ext}(t) \tag{10.26}$$

$$\dot{q} = \underbrace{\begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_{\vartheta}, \tag{10.27}$$

where $q = (x, y, \theta)^T$ is the vector of the configuration coordinates; $\vartheta = (v, w)^T$ is the vector of linear and angular velocities; $\tau = (\tau_1, \tau_2)$ is the vector of torques applied to the wheels of the robot where τ_1 and τ_2 denote the torques of the right and left wheel, respectively (Fig. 10.6); $F_{ext} \in \mathbb{R}^2$ uniformly bounded disturbance vector; $M(q) \in \mathbb{R}^{2\times 2}$ is the positive-definite inertia matrix; $C(q, \dot{q})\vartheta$ is the vector of centripetal and Coriolis forces; and $D \in \mathbb{R}^{2\times 2}$ is a diagonal positive-definite damping matrix. Equation 10.27 represents the kinematics of the system, where (x, y) is the position of the mobile robot in the X–Y (world) reference frame, θ is the angle between heading direction and the x-axis, and v and w are the angular and angular velocities, respectively.

Furthermore, the system (10.26)–(10.27) has the following nonholonomic constraint:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0, \tag{10.28}$$

which corresponds to a no-slip wheel condition preventing the robot from moving sideways. The system (10.27) fails to meet Brockett's necessary condition for feedback stabilization, which implies that a noncontinuous static state-feedback controller exists that stabilizes the closed-loop system around the equilibrium point [15].

The control objective is to design an FLC of τ that ensures

$$\lim_{t\to\infty} \|q_d(t) - q(t)\| = 0 \tag{10.29}$$

for any continuously, differentiable, bounded desired trajectory $q_d \in \mathbb{R}^3$ while attenuating external disturbances.

10.3.4 Fuzzy Logic Control Design

In order to satisfy the control objective, it is necessary to design an FLC for the real velocities of the mobile robot. To do that, a Takagi–Sugeno FLC was designed using linguistic variables in the input and mathematical functions in the output. The errors of the linear and angular velocities (v_d, w_d respectively) were taken as input variables, while the right (τ_1) and left (τ_2) torques were taken as outputs. The membership functions used in the input are trapezoidal for the negative (N) and positive (P), and a triangular was used for the zero (C) linguistic terms. The interval used for this fuzzy controller is $[-50, 50]$ [10]. Figure 10.7 shows the input and output variables, and Fig. 10.8 shows the general FLC architecture.

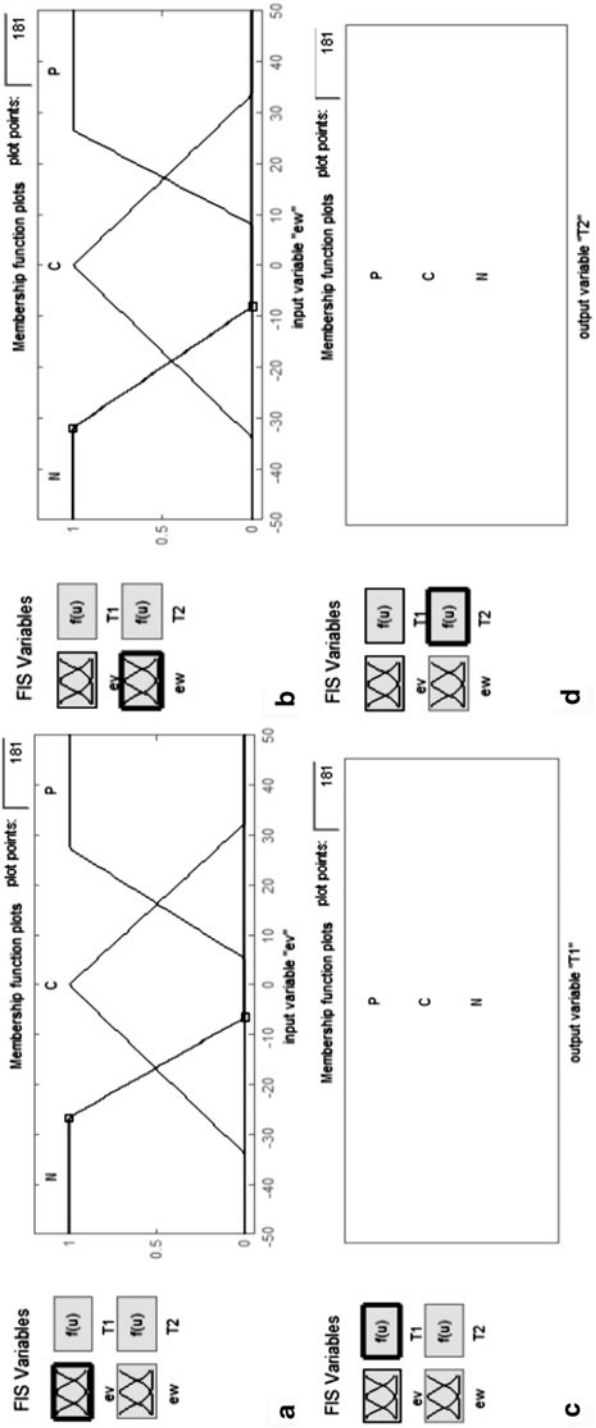


Fig. 10.7 **a** Linear velocity error. **b** Angular velocity error. **c** Right output (τ_1). **d** Left output (τ_2)

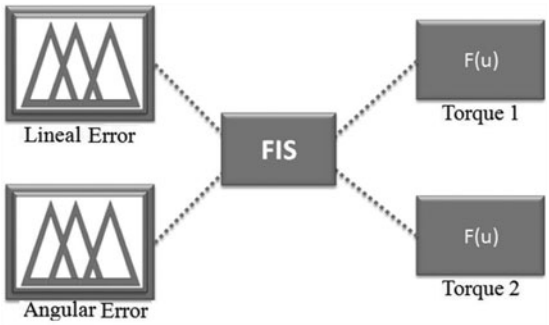


Fig. 10.8 Fuzzy logic controller architecture

Table 10.1 Fuzzy rules set

e_v/e_w	N	Z	P
N	N/N	N/Z	N/P
Z	Z/N	Z/Z	Z/P
P	P/N	P/Z	P/P

The FLC has a rule set that contain 9 rules, which governs the input–output relationship of the FLC and this adopts the Takagi–Sugeno-style inference engine, and it is used with a single point in the outputs, considering that the outputs are constant values, obtained using weighted average defuzzification procedure. In Table 10.1, we present the rule set whose format is established as follows:

$$\text{Rule } i\text{: if } e_v \text{ is } \mathrm{G}_1 \text{ and } e_w \text{ is } G_2 \text{ then } F \text{ is } G_3 \text{ and } N \text{ is } G_4,$$

where $G_1 \ldots G_4$ are the fuzzy set associated to each variable $i = 1, 2, \ldots, 9$.

To find the best FLC, we used an S-ACO to find the parameters of the membership functions. Table 10.2 shows the parameters of the membership functions, the minimal and maximum values in the search range for the S-ACO algorithm to find the best FLC.

It is important to remark that values shown in Table 10.2 are applied to both inputs and both outputs of the FLC.

10.3.5 ACO Architecture

The S-ACO algorithm was applied for the optimization of the membership functions for the FLC. For developing the architecture of the algorithm, it was necessary to follow the next steps:

1. Marking the limits of the problem in order to eliminate unnecessary complexity
2. Representing the architecture of the FLC as a graph that artificial ants could traverse
3. Achieving an adequate handling of the pheromone but permitting the algorithm to evolve by itself

Table 10.2 Parameters of the membership functions

MF type	Point	Minimal value	Maximal value
Trapezoidal	a	− 50	− 50
	b	− 50	− 50
	c	− 15	− 5.1
	d	− 1.5	− 0.5
Triangular	a	− 5	− 1.8
	b	0	0
	c	1.8	5
Trapezoidal	a	0.5	1.5
	b	5.1	15
	c	50	50
	d	50	50
Constant (N)	a	− 50	− 50
Constant (C)	a	0	0
Constant (P)	a	50	50

Table 10.3 Parameters of membership functions included in S-ACO search

MF type	Point	Minimal value	Maximal value
Trapezoidal	c	− 15	− 5.1
	d	− 1.5	− 0.5
Triangular	a	− 5	− 1.8
	c	1.8	5
Trapezoidal	a	0.5	1.5
	b	5.1	15

10.3.5.1 Limiting the Problem and Graph Representation

One of the problems found in the development of the S-ACO algorithm was to make a good representation of FLC. First, we reduced the number of elements that the method needed to find by deleting the elements whose minimal value and maximal values are the same (see Table 10.2), and therefore if they were included they will not change any way. Table 10.3 shows the parameters of the membership functions included in the search.

The next step was to represent those parameters shown in Table 10.3; to that was necessary to discretize the parameters in a range of possible values in order to represent every possible value as a node in the graph of search. The level of discretization between the minimal and maximal value was of 0.1 (by example: − 1.5, − 1.4, − 1.3, . . . , − 0.5).

Table 10.4 shows the number of possible values that each parameter can take.

Figure 10.9 shows the search graph for the proposed S-ACO algorithm; the graph can be viewed as a tree where the root is the nest and the last node is the food source.

Table 10.4 Number of possible values of the parameters of membership functions

MF type	Point	Combinations
Trapezoidal	c d	100 15
Triangular	a c	33 33
Trapezoidal	a b	15 100

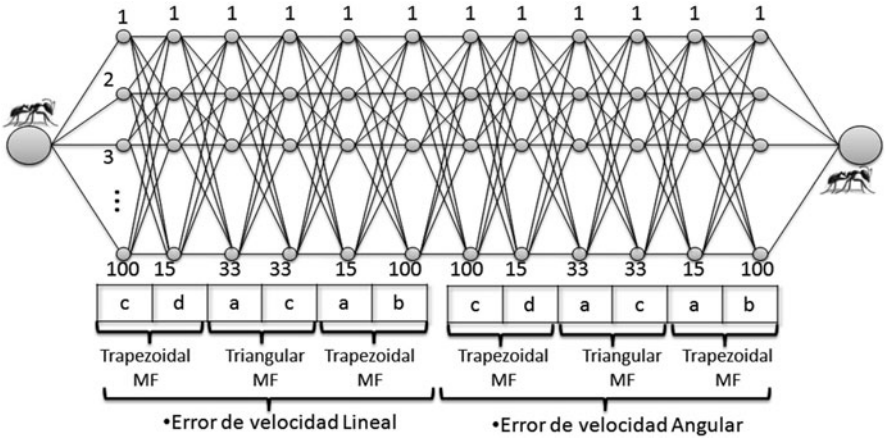


Fig. 10.9 S-ACO search graph

10.3.5.2 Updating Pheromone Trail

An important issue is that the update of pheromone trail be applied in the best way possible. In this sense, we need to handle the evaporation (Eq. 10.24), and increase or deposit of pheromone (Eq. 10.25), where the key parameter in evaporation is denoted by ρ that represents the rate of evaporation and in deposit of pheromone is denoted by $\Delta\tau$ that represents the amount of pheromone that an ant k deposits in a link ij in a time t. For ρ, we assign a random value and Eq. (10.30) shows the way how the increase of pheromone is calculated:

$$\Delta\tau = \frac{(e_{\max} - e_k)}{e_{\max}}, \tag{10.30}$$

where $e_{\max} = 10$ is the maximum error of control permitted and e_k is the error of control generated by a complete path of an ant k. We decided to allocate $e_{\max} = 10$ in order to stand $\Delta\tau \in [0,1]$.

10.3.6 Simulation Results

In this section, we present the results of the proposed controller to stabilize the unicycle mobile robot, defined by Eq. (10.26) and Eq. (10.27), where the matrix values are defined as

$$M(q) = \begin{bmatrix} 0.3749 & -0.0202 \\ -0.0202 & 0.3739 \end{bmatrix},$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & 0.1350\dot{\theta} \\ -0.150\dot{\theta} & 0 \end{bmatrix},$$

and

$$D = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}.$$

The evaluation was made through computer simulation performed in MATLAB® and SIMULINK®.

The desired trajectory is the following one:

$$\vartheta_d(t) = \begin{cases} v_d(t) = 0.2(1 - \exp(-t)) \\ w_d(t) = 0.4 \sin(0.5t) \end{cases} \tag{10.31}$$

and was chosen in terms of its corresponding desired linear v_d and angular w_d velocities, subject to the initial conditions

$$q(0) = (0.1, 0.1, 0,)^T \text{and } \vartheta(0) = 0 \in \mathbb{R}^2.$$

The gains γ_i, $i = 1, 2, 3$, of the kinematic model are $\gamma_1 = 5, \gamma_2 = 24$, and $\gamma_3 = 3$.

10.3.6.1 S-ACO Algorithm Results for the FLC Optimization

Table 10.5 shows the results of the FLC, obtained varying the values of maximum iterations and number of artificial ants, where the italicized row shows the best result obtained with the method. Figure 10.10 shows the evolvement of the method.

Figure 10.11 shows the membership functions of the FLC obtained by the S-ACO algorithm.

Table 10.5 S-ACO results of simulations for FLC optimization

Iterations	Ants	α	ρ	Average error	Time
20	10	0.2	Random	1.5589	00:01:30
20	10	0.2	Random	1.451	00:01:34
25	10	0.2	Random	1.5566	00:01:46
25	10	0.2	Random	1.4767	00:01:51
25	10	0.2	Random	1.4739	00:02:05
25	10	0.2	Random	1.6137	00:02:08
25	10	0.2	Random	1.6642	00:01:54
25	100	0.2	Random	1.3484	00:20:30
25	100	0.2	Random	1.3413	00:18:44
25	100	0.2	Random	1.3360	00:18:31
25	100	0.2	Random	1.2954	00:18:32
25	100	0.2	Random	1.4877	00:18:41
25	100	0.2	Random	1.2391	00:18:31
10	15	0.2	Random	1.6916	00:01:14
10	15	0.2	Random	1.4256	00:01:09
40	65	0.2	Random	1.2783	00:19:17
40	65	0.2	Random	1.4011	00:19:45
40	65	0.2	Random	1.2216	00:19:33
40	65	0.2	Random	1.2487	00:19:49
50	70	0.2	Random	1.3782	00:26:09
50	70	0.2	Random	1.0875	00:27:35
50	70	0.2	Random	1.4218	00:33:45
50	70	0.2	Random	1.475	01:08:48
25	80	0.2	Random	1.4718	00:14:55
25	80	0.2	Random	1.4212	00:15:00
25	80	0.2	Random	1.3221	00:14:52
25	80	0.2	Random	1.1391	00:15:41
50	80	0.2	Random	1.2148	00:28:43
62	50	0.2	Random	1.0322	00:24:49
50	80	0.2	Random	1.1887	00:29:55
50	80	0.2	Random	1.2158	00:29:56
60	90	0.2	Random	1.3493	00:41:56
60	90	0.2	Random	1.3060	00:39:48
60	90	0.2	Random	1.3161	00:40:00

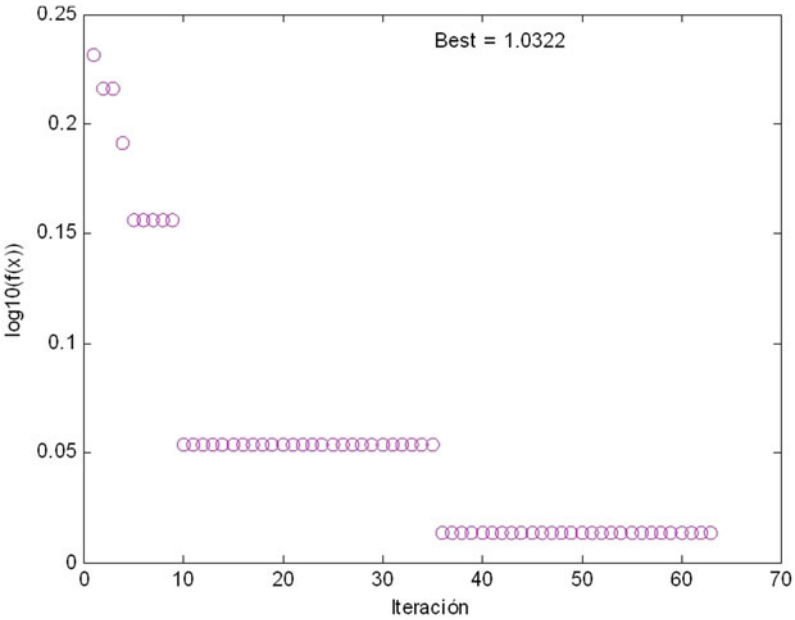


Fig. 10.10 Evolution of the S-ACO for FLC optimization

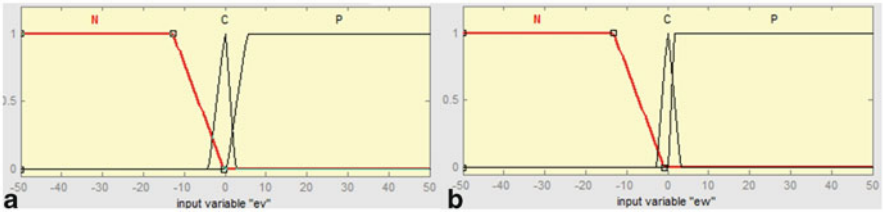


Fig. 10.11 **a** Linear velocity error and **b** angular velocity error optimized by S-ACO algorithm

Figure 10.12 shows the block diagram used for the FLC that obtained the best results. Figure 10.13 shows the results of linear and angular errors, and Fig. 10.14 shows the output results of the fuzzy controller that represents the torque applied to the wheels of the autonomous mobile robot.

The position errors of the autonomous mobile robot can be observed in Fig. 10.15. Figure 10.16 shows the desired trajectory and obtained trajectory.

A trajectory-tracking controller has been designed based on the dynamics and kinematics of the autonomous mobile robot through the application of ACO for the optimization of membership functions for the FLC with good results obtained after simulations.

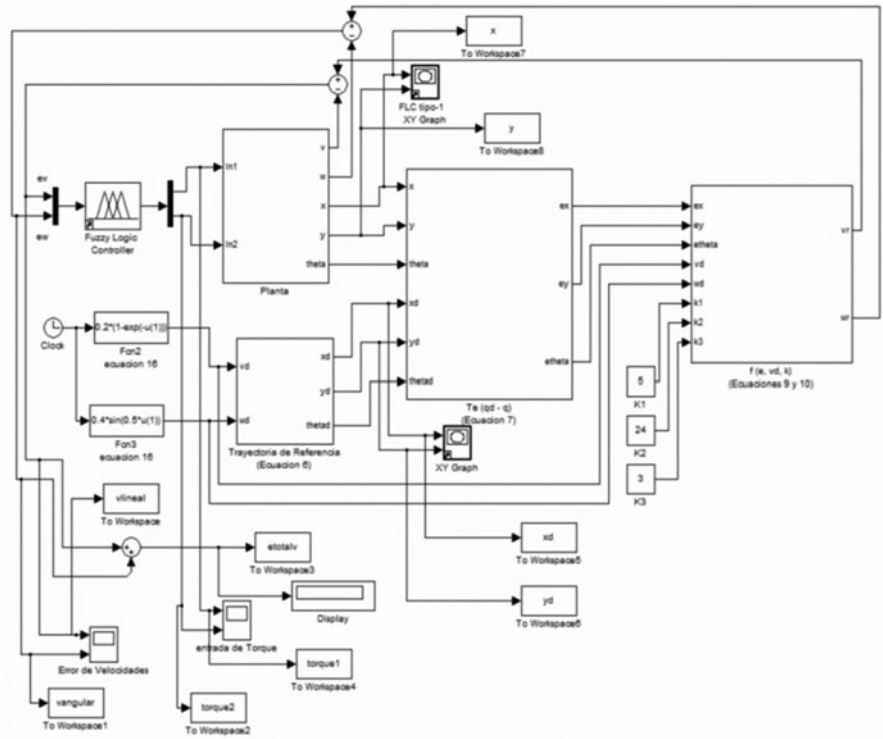


Fig. 10.12 Block diagram for simulation of the FLC

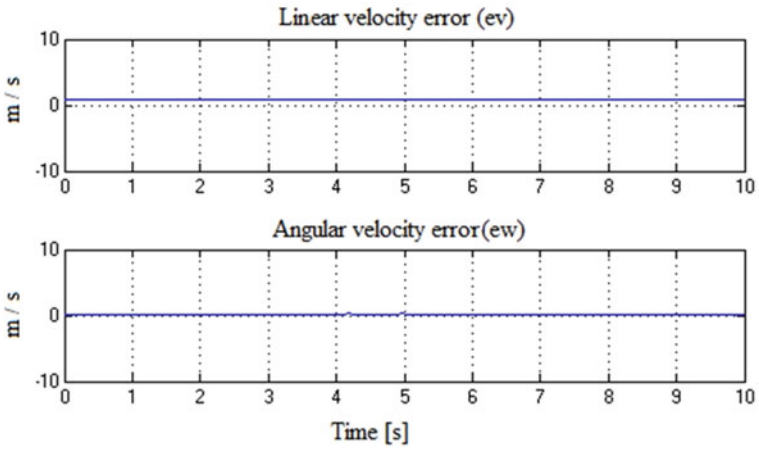


Fig. 10.13 Linear and angular velocity errors

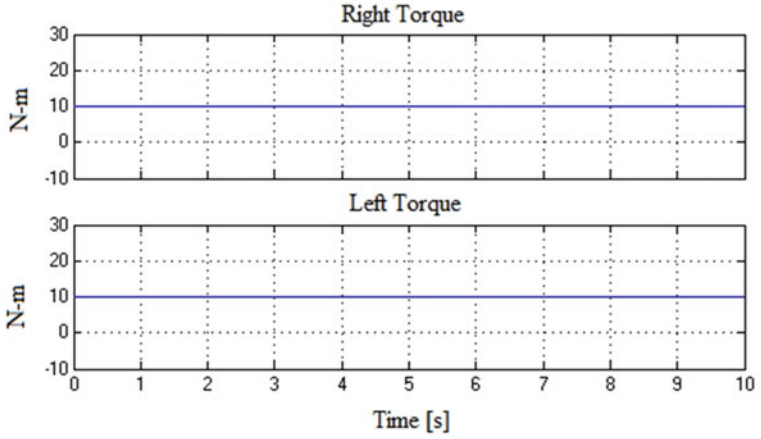


Fig. 10.14 Right and left torques

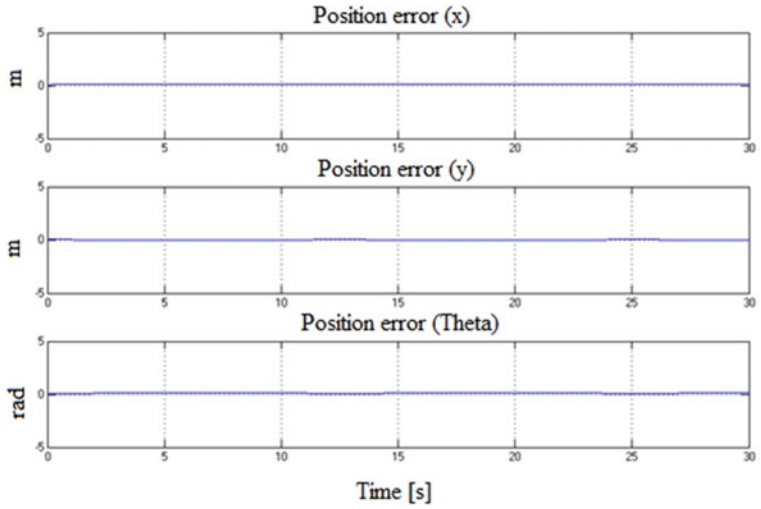


Fig. 10.15 Position errors in x, y, θ

10.4 Optimization of an Interval Type-2 Fuzzy Controller for an Autonomous Mobile Robot Using the PSO Algorithm

This section describes the application of the optimization algorithm for particle swarm known by its acronym as PSO [18], used to adjust the parameters of membership functions of an FLC to find the optimal intelligent control for a wheeled autonomous mobile robot [29]. Results of several simulations show that the PSO is able to optimize the type-1 and type-2 FLC for this specific application [8].

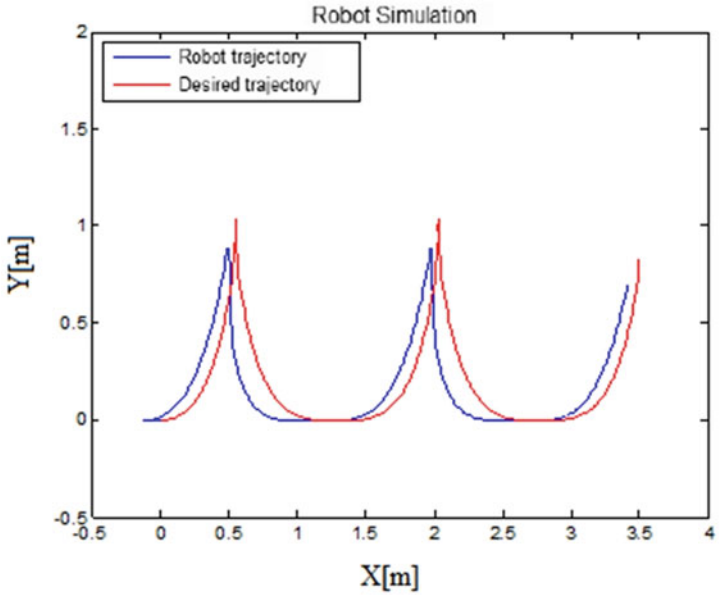


Fig. 10.16 Obtained trajectory

This section describes the application of particle swarm algorithm (PSO) as a method of optimizing the parameters of membership functions of the proposed FLC in order to find the best intelligent controller for an autonomous mobile robot.

10.4.1 Optimization Algorithm Using Particle Swarm

Optimization by swarm of particles (PSO) is a relatively new technique that is slowly emerging and gaining recognition as an effective and efficient algorithm [18]. The PSO algorithm shares similarities with other evolutionary computation techniques while also differing in certain respects and needs no evolution operators such as crossover and mutation [16].

PSO emulates the swarm behavior of insects, a herd of grazing animals, a swarm of birds, and a host of fish in these swarms or clouds that made the search for food in a collaborative manner. Each member of a swarm adapts its search patterns, learning from his own experience and experiences of other members, i.e., taking into account their cognitive beliefs and social beliefs.

These phenomena are discussed in the algorithm and the mathematical models are built on the method for updating the positions of each particle.

In the PSO algorithm, a member in the swarm, called a particle, represents a possible solution, is a point in the search space. The global optimum is considered as the location of food; the problem would be implemented as the optimal solution

found. Each particle has a fitness value and a speed to adjust the flying direction according to the best.

The general formula (Eq. 10.32) for determining the motion of particles, which are presented below, is shown in two parts, the cognitive and social part of which are crucial to identify the type of algorithm that is being implemented; in our case, we used the Full GBEST, i.e., both C1 and C2 must have values greater than 0 but less than 4, respectively:

$$V_{id} = V_{id}\,(t+1) + C_1 r_{id}\,(t)\,[Y_{id}\,(t)\,X_{id}\,(t)] + C_2 r_2\,(t)\,[Y_{id}\,(t)\,X_{id}\,(t)] \quad (10.32)$$

Another formula that is critical (Eq. 10.33) for the update of each particle; this assesses the current position of the particle and the previous position to choose which is the most appropriate to find more quickly the result this position is recalculated at each new iteration that is the algorithm:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (10.33)$$

10.4.2 Design of the Fuzzy Controller

As determined by one of the most used and effective techniques to solve control problems is to use fuzzy systems to meet the control objective, it is necessary to design a fuzzy controller for the actual speed of the mobile robot. In this work the fuzzy inference system if of Takagi-Sugeno type with only input linguistic variables and mathematical functions in the outputs. The errors of linear and angular velocities (respectively) were taken as input variables, while the left and right pairs were taken as outputs. The membership functions used in the entry are trapezoidal for negative (N) and positive (P), and a triangle was used to zero (C) linguistic terms. Figure 10.17 shows the input and output variables used; these are used for both types of fuzzy logic (1 and 2).

The FLC has nine rules, which are adapted to the style of Takagi–Sugeno controller, so the output has a single point, and for that, the results are constants values (P, C, N), which are obtained through a procedure using a weighted average defuzzification by

- R1: If Vangular is C and VLinear is C, then τ 1 is C and τ 2 is C
- R2: If Vangular is C and VLinear is P, then τ 1 is C and τ 2 is P
- R3: If Vangular is C and VLinear is N, then τ 1 is C and τ 2 is N
- R4: If Vangular is P and VLinear is C, then τ 1 is P and τ 2 is C
- R5: If Vangular is P and VLinear is P, then τ 1 is P and τ 2 is P
- R6: If VLinear is P and Vangular is N, then τ 1 is P and τ 2 is N
- R7: If VLinear is N and Vangular is C, then τ 1 is N and τ 2 is C
- R8: If VLinear is N and Vangular is P, then τ 1 is N and τ 2 is P
- R9: If Vangular is N and VLinear is N, then τ 1 is N and τ 2 is N

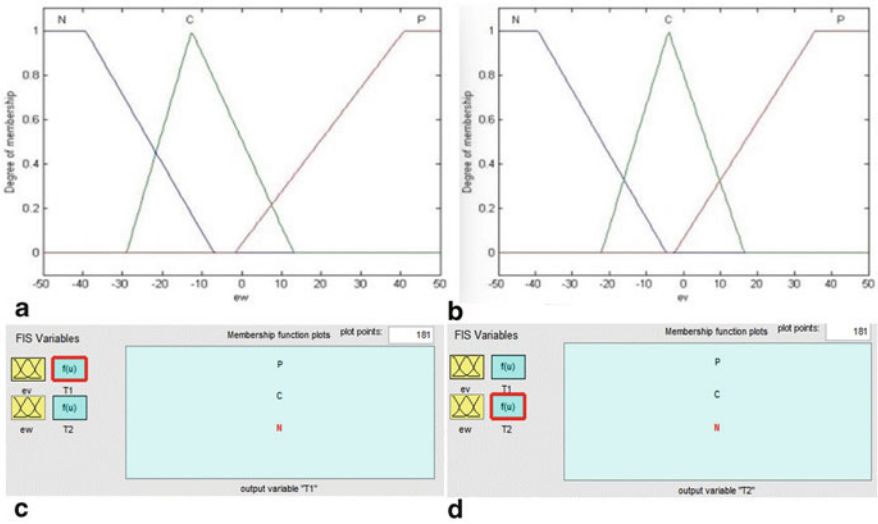


Fig. 10.17 Variable input/output: **a** error of the linear velocity (e_v). **b** Angular velocity error (e_w). **c** Exit fee ($\tau 1$). **d** Exit left ($\tau 2$)

Table 10.6 FLC rules

e_v/e_w	*N*	*C*	*P*
N	N/N	N/C	N/P
C	C/N	C/C	C/P
P	P/N	P/C	P/P

The linguistic terms of input variables are shown in the first row and column of Table 10.6; the rest of the content corresponds to the linguistic terms of output variables.

10.4.3 Results of the Simulations

This section presents the results of the proposed controller to stabilize the autonomous mobile robot. The evaluation is done through computer simulation done in MATLAB ® and SIMULINK ® 2007b.

Table 10.7 shows the results of the FLC, obtained by varying the values of maximum iterations and the number of particles, where the italicized row shows the best result obtained with the method. Figure 10.18 shows the behavior of the optimization method.

Figure 10.19 shows the membership functions of the FLC obtained by the PSO algorithm, and achieved the desired path and the degree of error was obtained.

Table 10.7 Results for PSO using the constriction coefficient for type 1

Experiment no	Iterations	Swarm	Coefficient constriction	Average error	Runtime
1	50	50	1.0	0.0606	00:20:05
2	100	200	1.0	0.2670	02:44:53
3	100	100	1.0	0.0301	01:49:55
4	100	150	1.0	0.0315	01:54:31
5	100	50	1.0	0.0266	03:16:34
6	100	57	1.0	0.0211	00:41:38
7	100	300	1.0	0.0276	04:04:54
8	100	80	1.0	0.0527	01:03:45
9	150	80	1.0	0.0260	01:35:17
10	300	150	1.0	0.0307	04:30:16
11	500	200	1.0	0.0529	39:56:23
12	200	90	1.0	0.0345	01:54:59
13	150	100	1.0	0.0496	01:36:37
14	100	53	1.0	0.0230	00:39:29

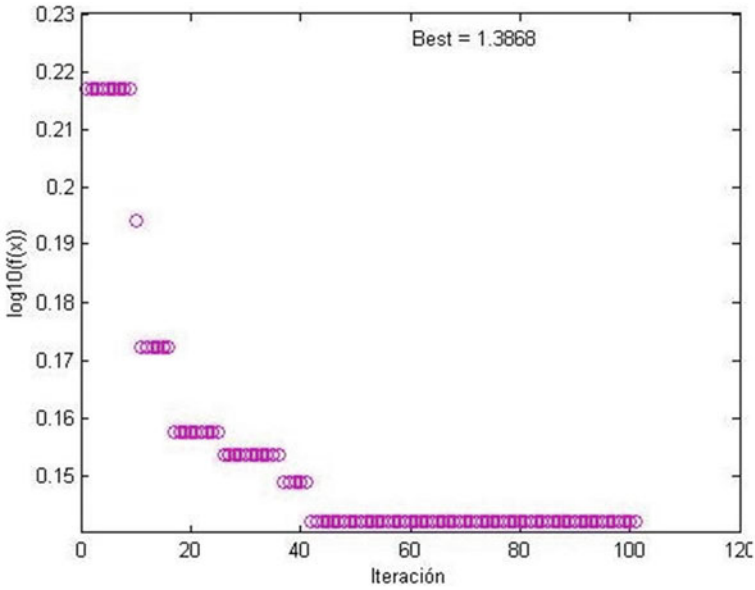


Fig. 10.18 Evolution of PSO for the optimization of FLC

As you can see, the above results are acceptable for type-1 FLC obtaining a final result of 0.0211 using 57 particles and 100 iterations in a time of 47 min and 38 s for

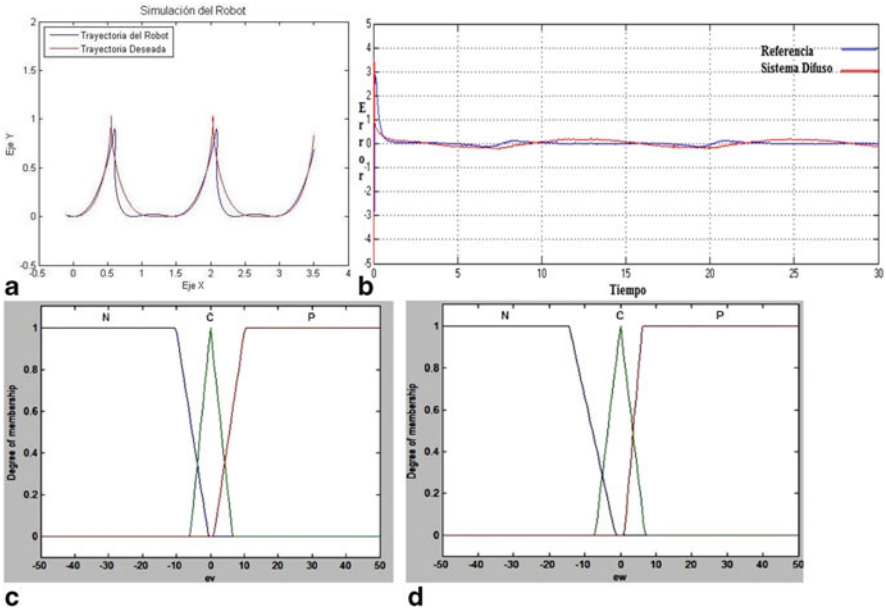


Fig. 10.19 **a** linear velocity (e_v), **b** angular velocity (e_w), **c** shows the desired path and the trajectory obtained, **d** plots representing the degree of error in the simulation

this experiment, but as previously mentioned there were also simulations with type 2, taking into account the same parameters and conditions of Takagi–Sugeno controller; the results of these simulations are presented below in Table 10.8. Figure 10.20 shows the convergence of the PSO algorithm.

Figure 10.21 shows the membership functions of the FLC obtained by the PSO algorithm, and achieved the desired path and the degree of error was obtained.

The above results are acceptable for type-2 fuzzy logic control because we obtain a final result of 0.0500 using 380 particles and 300 iterations in a time of 7 h 55 min and 08 s for this experiment, which cannot be considered high due to the complexity of the problem.

The type-2 fuzzy controllers were implemented with the type-2 fuzzy logic toolbox that was developed previously by our research group [9, 10].

With the results of the experiments shown in Table 10.9, we can determine that for this particular problem the optimized type-2 FLC clearly outperforms its type-1 counterpart.

The trajectory-tracking controller is designed based on the dynamics and kinematics of mobile autonomous robot through the application of PSO for the optimization of membership functions of fuzzy controller both type 1 and type 2 with the good results obtained after simulations.

Table 10.8 Results of PSO using the constriction coefficient for type 2

Experiment no	Iterations	Swarm	Coefficient constriction	Average error	Runtime
1	100	150	1.0	0.0659	02:20:31
2	100	200	1.0	0.0675	02:56:21
3	100	250	1.0	0.0666	03:05:31
4	200	150	1.0	0.0663	03:55:06
5	200	200	1.0	0.0651	04:07:14
6	200	250	1.0	0.0642	04:20:10
7	200	300	1.0	0.0536	04:54:43
8	200	350	1.0	0.0554	05:12:09
9	250	350	1.0	0.0600	05:38:20
10	300	300	1.0	0.0531	06:12:57
11	300	350	1.0	0.0503	07:30:28
12	300	380	1.0	0.0500	07:55:08
13	350	400	1.0	0.0501	08:10:15
14	350	450	1.0	0.0503	08:59:01
15	400	300	1.0	0.0502	12:31:11

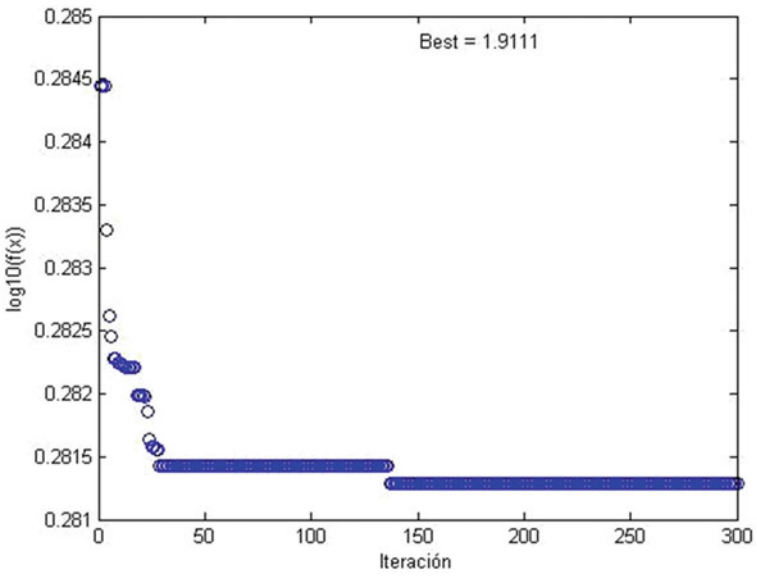


Fig. 10.20 Plot of convergence of PSO algorithm

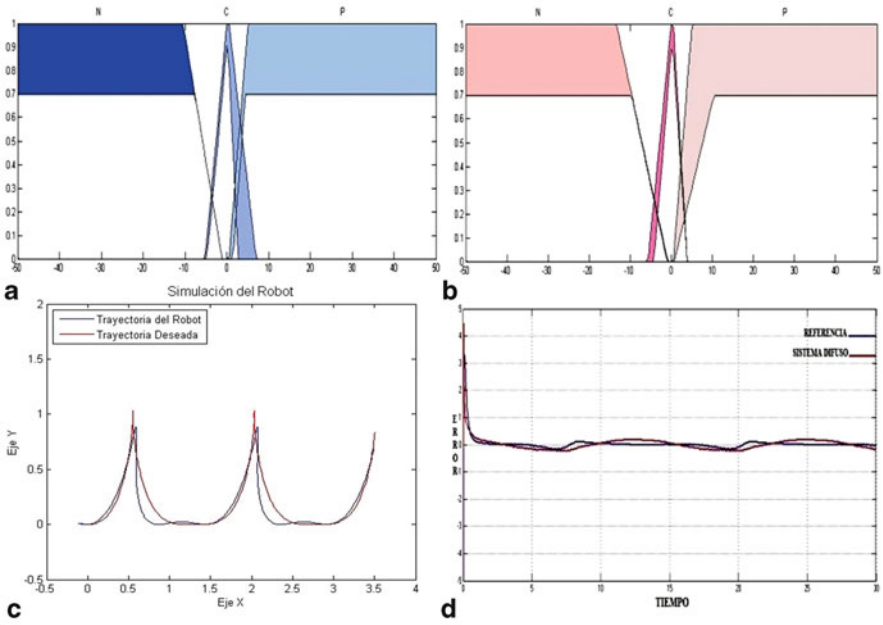


Fig. 10.21 **a** linear velocity (e_v), **b** angular velocity (e_w), **c** shows the desired path and the trajectory obtained, **d** plots representing the degree of error in the simulation

Table 10.9 Comparison of results of PSO algorithm for type 1 and type 2

	Iterations	Swarm	Average error	Runtime
PSO with type-1 fuzzy logic	100	57	0.0211	00:41:38
PSO with type-2 fuzzy logic	300	380	0.0050	07:55:08

10.5 Conclusions

This chapter has presented a general framework for designing interval type-2 fuzzy controllers based on bio-inspired optimization techniques. A trajectory-tracking controller has been designed based on the dynamics and kinematics of the autonomous mobile robot through the application of ACO for the optimization of membership functions for the FLC with good results obtained after simulations. Also, a trajectory-tracking controller is designed based on the dynamics and kinematics of mobile autonomous robot through the application of PSO for the optimization of membership functions of the fuzzy controller for both type 1 and type 2 with the good results obtained after simulations. As future work, we plan to test other bio-inspired optimization techniques, like the bat or firefly algorithms for this particular robotic application and in other different kinds of applications.

References

1. L. Astudillo, P. Melin, O. Castillo, A new optimization method base on a paradigm inspired by nature. Stud. Comput. Intell. **312**, 277–283 (2010)
2. M. Biglarbegian, W.W. Melek, J.M. Mendel, Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments. IEEE Trans. Industrial Electronics **58**, 1371–1384 (2011)
3. Z. Bingül, O. Karahan, A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control. Expert Syst. Appl. **38**, 1017–1031 (2011)
4. O. Castillo, P. Melin, *Type-2 Fuzzy Logic: Theory and Applications* (Springer, Heidelberg, 2008)
5. O. Castillo, A.I. Martinez, A.C. Martinez, Evolutionary computing for topology optimization of type-2 fuzzy systems. Adv. Soft Comput. **41**, 63–75 (2007)
6. O. Castillo, G. Huesca, F. Valdez, Evolutionary computing for topology optimization of type-2 fuzzy controllers. Stud. Fuzziness Soft Comput. **208**, 163–178 (2008)
7. O. Castillo, L.T. Aguilar, N.R. Cazarez-Castro, S. Cardenas, Systematic design of a stable type-2 fuzzy logic controller. Appl. Soft Comput. J. **8**, 1274–1279 (2008)
8. O. Castillo, R. Martinez-Marroquin, P. Melin, F. Valdez, J. Soria, Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inform. Sci. (2011)
9. J.R. Castro, O. Castillo, P. Melin, An interval type-2 fuzzy logic toolbox for control applications, in *Proceedings of FUZZ-IEEE*, London, 2007, pp. 1–6
10. J.R. Castro, O. Castillo, L.G. Martinez, Interval type-2 fuzzy logic toolbox. Eng. Lett. **15**(1), 14 (2007)
11. J.R. Castro, O. Castillo, P. Melin, L.G. Martinez, S. Escobar, I. Camacho, Building fuzzy inference systems with the interval type-2 fuzzy logic toolbox. Adv. Soft Comput. **41**, 53–62 (2007)
12. J.R. Castro, O. Castillo, P. Melin, A. Rodriguez-Diaz, A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. Inform. Sci. **179**, 2175–2193 (2009)
13. N.R. Cazarez-Castro, L.T. Aguilar, O. Castillo, Hybrid genetic-fuzzy optimization of a type-2 fuzzy logic controller, in *Proceedings of the 8th International Conference on Hybrid Intelligent Systems*, HIS 2008, Barcelona, 2008, pp. 216–221
14. L. Cervantes, O. Castillo, Design of a fuzzy system for the longitudinal control of an F-14 airplane. Stud. Comput. Intell. **318**, 213–224 (2010)
15. H. Chaoui, W. Gueaieb, Type-2 fuzzy logic control of a flexible-joint manipulator. J. Intell. Robot. Syst. **51**, 159–186 (2008)
16. O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Set. Syst. **141**, 5–31 (2004)
17. T. Dereli, A. Baykasoglu, K. Altun, A. Durmusoglu, I.B. Turksen, Industrial applications of type-2 fuzzy sets and systems: a concise review. Comput. Ind. **62**, 125–137 (2011)
18. A.P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence* (Wiley, England, 2005)
19. M. Galluzzo, B. Cosenza, Adaptive type-2 fuzzy logic control of a bioreactor. Chem. Eng. Sci. **65**, 4208–4221 (2010)
20. H. Hagras, Hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Trans. Fuzzy Syst. **12**, 524–539 (2004)
21. M. Hsiao, T.H.S. Li, J.Z. Lee, C.H. Chao, S.H. Tsai, Design of interval type-2 fuzzy sliding-mode controller. Inform. Sci. **178**, 1686–1716 (2008)
22. H.C. Huang, C.M. Chu, J.S. Pan, The optimized copyright protection system with genetic watermarking. Soft Comput. **13**, 333–343 (2009)
23. C.-F. Juang, C.-H. Hsu, Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control. IEEE Trans. Industrial Electronics **56**, 3931–3940 (2009)
24. C.-F. Juang, C.-H. Hsu, Reinforcement interval type-2 fuzzy controller design by online rule generation and Q-value-aided ant colony optimization. IEEE Trans. Syst. Man Cybern. B **39**, 1528–1542 (2009)

25. C.-F. Juang, C.-H. Hsu, C.-F. Chuang, Reinforcement self-organizing interval type-2 fuzzy system with ant colony optimization, in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, 2009, pp. 771–776
26. W.-D. Kim, H.-J. Jang, S.-K. Oh, The design of optimized fuzzy cascade controller: focused on type-2 fuzzy controller and HFC-based genetic algorithms. Trans. Korean Inst. Electr. Eng. **59**, 972–980 (2010)
27. G.O. Koca, Z.H. Akpolat, M. Özdemir, Type-2 fuzzy sliding mode control of a four-bar mechanism. Int. J. Model. Simul. **31**, 60–68 (2011)
28. R. Martinez, O. Castillo, L.T. Aguilar, Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Inform. Sci. **179**, 2158–2174 (2009)
29. R. Martinez, A. Rodriguez, O. Castillo, L.T. Aguilar, Type-2 fuzzy logic controllers optimization using genetic algorithms and particle swarm optimization, in *Proceedings of the IEEE International Conference on Granular Computing*, GrC 2010, 2010, pp. 724–727
30. R. Martinez-Marroquin, O. Castillo, J. Soria, Parameter tuning of membership functions of a type-1 and type-2 fuzzy logic controller for an autonomous wheeled mobile robot using ant colony optimization, in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, 2009, pp. 4770–4775
31. P. Melin, O. Castillo, A new method for adaptive control of non-linear plants using Type-2 fuzzy logic and neural networks. Int. J. Gen. Syst. **33**, 289–304 (2004)
32. P. Melin, O. Castillo, An intelligent hybrid approach for industrial quality control combining neural networks, fuzzy logic and fractal theory. Inform. Sci. **177**, 1543–1557 (2007)
33. J.M. Mendel, Uncertainty, fuzzy logic, and signal processing. Signal Process. J. **80**, 913–933 (2000)
34. S.M.A. Mohammadi, A.A. Gharaveisi, M. Mashinchi, An evolutionary tuning technique for type-2 fuzzy logic controller in a non-linear system under uncertainty, in *Proceedings of the 18th Iranian Conference on Electrical Engineering*, ICEE, 2010, pp. 610–616
35. K.J. Poornaselvan, T. Gireesh Kumar, V.P. Vijayan, Agent based ground flight control using type-2 fuzzy logic and hybrid ant colony optimization to a dynamic environment, in *Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology*, ICETET 2008, 2008, pp. 343–348
36. J.T. Starczewski, Efficient triangular type-2 fuzzy logic systems. Int. J. Approx. Reason. **50**, 799–811 (2009)
37. K.R. Sudha, R. Vijaya Santhi, Robust decentralized load frequency control of interconnected power system with generation rate constraint using type-2 fuzzy approach. Int. J. Elec. Power **33**, 699–707 (2011)
38. C. Wagner, H. Hagras, A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots, in *Proceedings of the IEEE Conference on Fuzzy Systems*, London, 2007
39. C. Wagner, H. Hagras, Evolving type-2 fuzzy logic controllers for autonomous mobile robots. Adv. Soft Comput. **41**, 16–25 (2007)
40. C.-H. Wang, C.-S. Cheng, T.-T. Lee, Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN). IEEE Trans. Syst. Man Cybern. B **34**(3), 1462–1477 (2004)
41. D. Wu, W.-W. Tan, A type-2 fuzzy logic controller for the liquid level process, in *Proceedings of the IEEE Conference on Fuzzy Systems*, Budapest, 2004, pp. 953–958
42. D. Wu, W.-W. Tan, Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers. Eng. Appl. Artificial Intell. **19**, 829–841 (2006)
43. R.R. Yager, Fuzzy subsets of type II in decisions. J. Cybern. **10**, 137–159 (1980)
44. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning. Inform. Sci. **8**, 43–80 (1975)

Chapter 11
Image Processing and Pattern Recognition with Interval Type-2 Fuzzy Inference Systems

Patricia Melin

Abstract Interval type-2 fuzzy systems can be of great help in achieving efficient image processing and pattern recognition applications. In particular, edge detection is an operation 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 have been proposed, but it is very difficult to demonstrate which one is better before the recognition results are obtained. In this chapter, we present experimental results where several edge detectors were used to preprocess the same image sets. Each resultant 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 image recognition.

11.1 Introduction

In previous works, we have proposed some extensions to traditional edge detectors to improve their performance by using fuzzy systems [14, 16, 19]. In all the experiments, we show the resulting images, demonstrating that the images obtained with fuzzy systems were visually better than the ones obtained with the traditional methods.

There is still work to be done on developing formal validations for fuzzy edge detectors using different methods. In the literature, we can find comparisons of edge detectors based on human observations [5, 8, 9, 11, 12], and some others that found the optimal values for parametric edge detectors [12].

Edge detectors can be used in recognition systems for different purposes, but in our 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 clearly show that fuzzy edge detectors are a good choice to improve the performance of neural or other types of recognition systems, and for this reason, we propose

P. Melin (✉)
Division of Graduate Studies and Research, Tijuana Institute of Technology, Tijuana, Mexico
e-mail: pmelin@tectijuana.mx

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_11

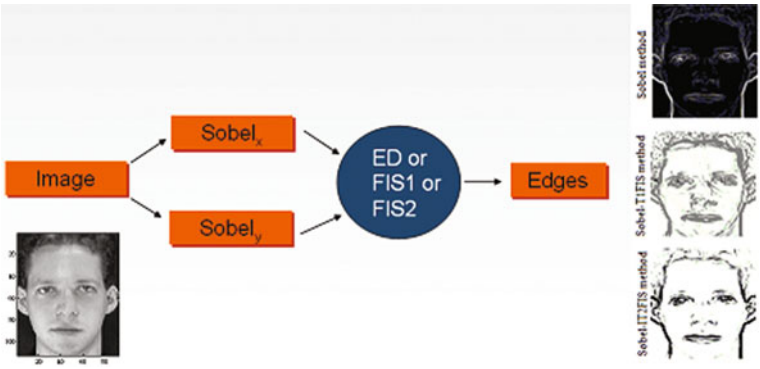


Fig. 11.1 Sobel edge detector enhanced with fuzzy systems

that the recognition rate of the neural networks can be used as an edge detection performance index.

11.2 Overview of Existing Fuzzy Edge Detectors

11.2.1 Sobel Edge Detector Improved with Fuzzy Logic

In the Sobel fuzzy edge detector, the Sobel operators $Sobel_x$ and $Sobel_y$ are used as in the traditional method, and then we substitute the Euclidean distance of Eq. (11.1) by a fuzzy system that uses these operators as inputs, as can be appreciated in Fig. 11.1 [23]:

$$\text{Sobel\_edges} = \sqrt{\text{Sobel}_x^2 + \text{Sobel}_y^2}. \tag{11.1}$$

Sobel operators are the main inputs for 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. 11.2 and 11.3, respectively.

The use of the FIS2 [6, 7] 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 (the knowledge is 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).
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).

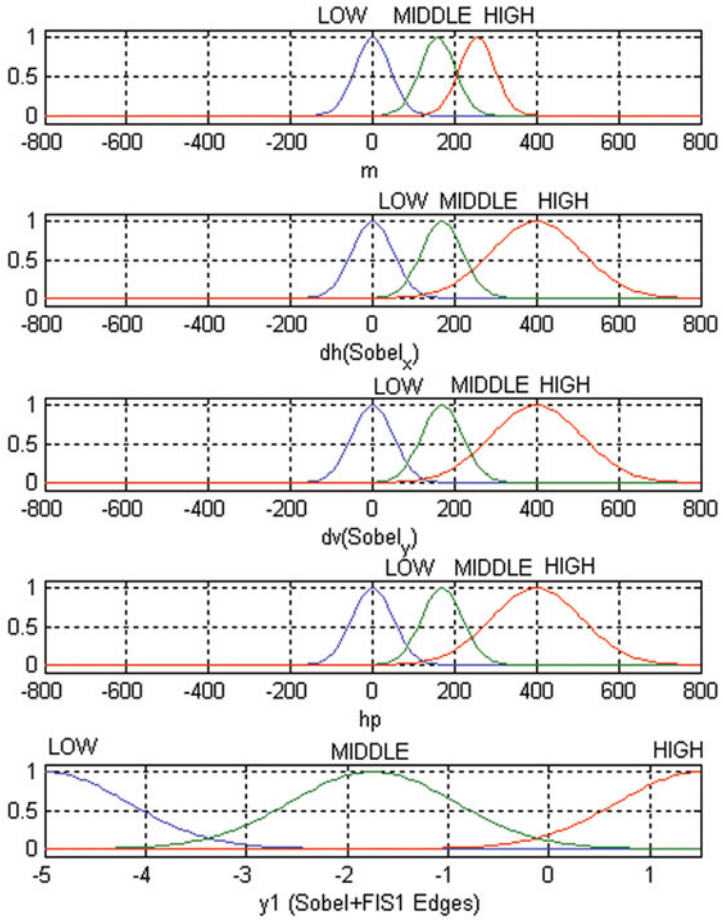


Fig. 11.2 Membership functions of the variables for the Sobel + FIS1 edge detector

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 neighboring pixels, then 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.

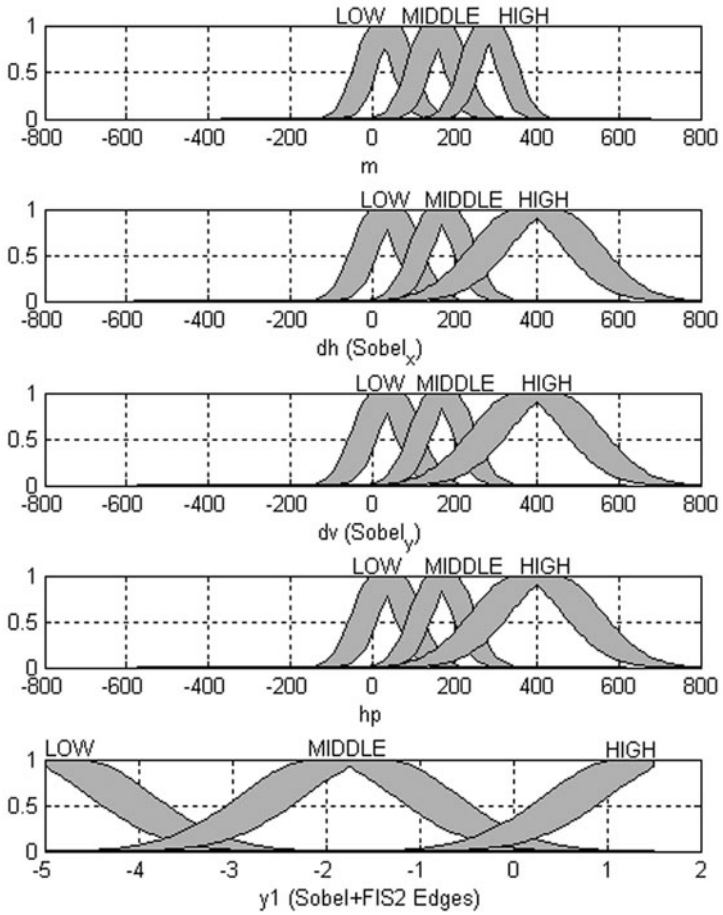


Fig. 11.3 Membership functions of the variables for the Sobel + FIS2 edge detector

11.3 Morphological Gradient Edge Detectors Improved with Fuzzy Logic

In the morphological gradient (MG), the four gradients are calculated as in the traditional method [1, 4], and substitute the sum of gradients Eq. (11.2), using instead a fuzzy inference system that uses as inputs these gradients, as can be appreciated in Fig. 11.4:

$$\text{MG\_edges} = D_1 + D_2 + D_3 + D_4. \tag{11.2}$$

The linguistic variables used in the MG + FIS1 and MG + FIS2 edge detectors are shown in Figs. 11.5 and 11.6, respectively.

The rules for both the FIS1 and FIS2 are the same and are shown below:

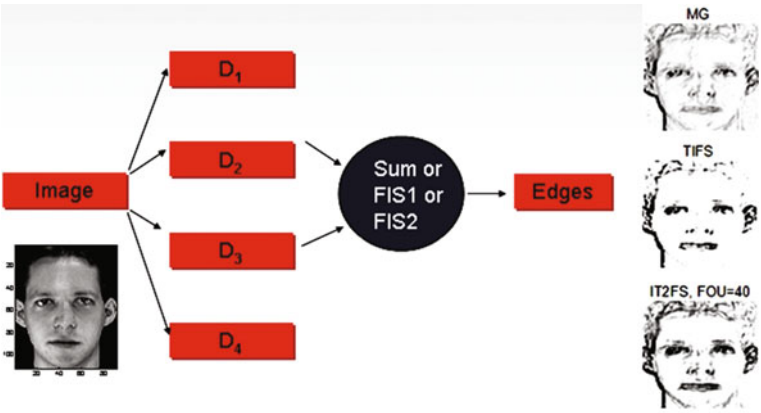


Fig. 11.4 Morphological gradient edge detector enhanced with fuzzy systems

1. If (D1 is HIGH) or (D2 is HIGH) or (D3 is HIGH) or (D4 is HIGH) then (E is BLACK).
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 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 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.

11.4 Experimental Setup To Test the Proposed Approach

The experiment consists of applying a neural recognition system using each of the following edge detectors: Sobel, Sobel + FIS1, Sobel + FIS2, MG, MG + FIS1, and MG + FIS2.

11.4.1 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.

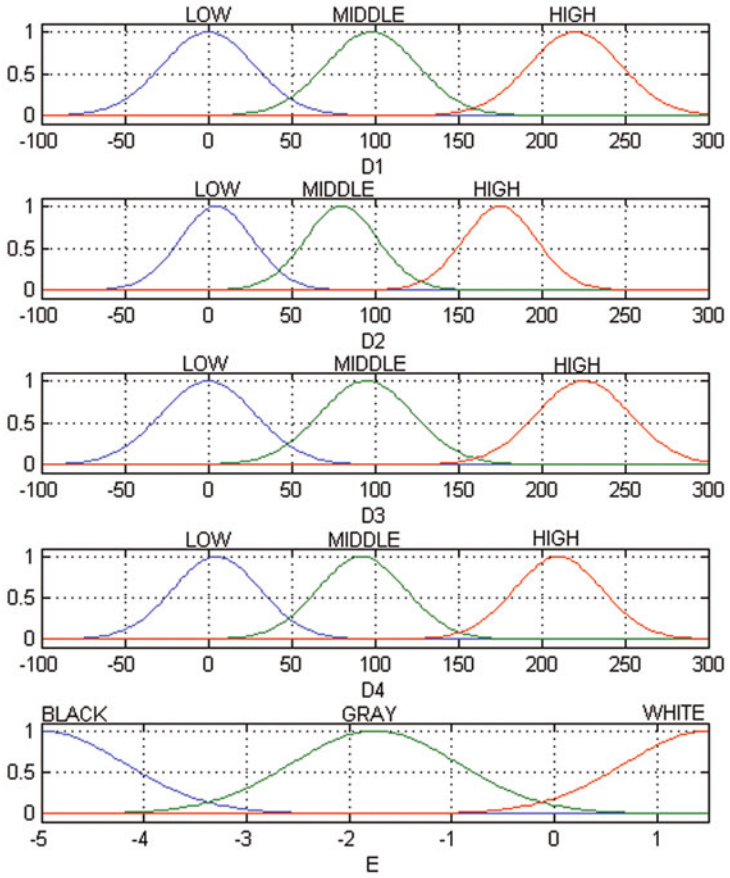


Fig. 11.5 Membership functions of the variables for the MG + FIS1 edge detector

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.

11.4.2 Parameters for the Images Databases

The experiments can be performed with images databases used for identification purposes. That is the case of the face recognition applications, then we use three of the most popular sets of images, the ORL face database [3], the Cropped Yale face database [2, 10], and the FERET face database [22].

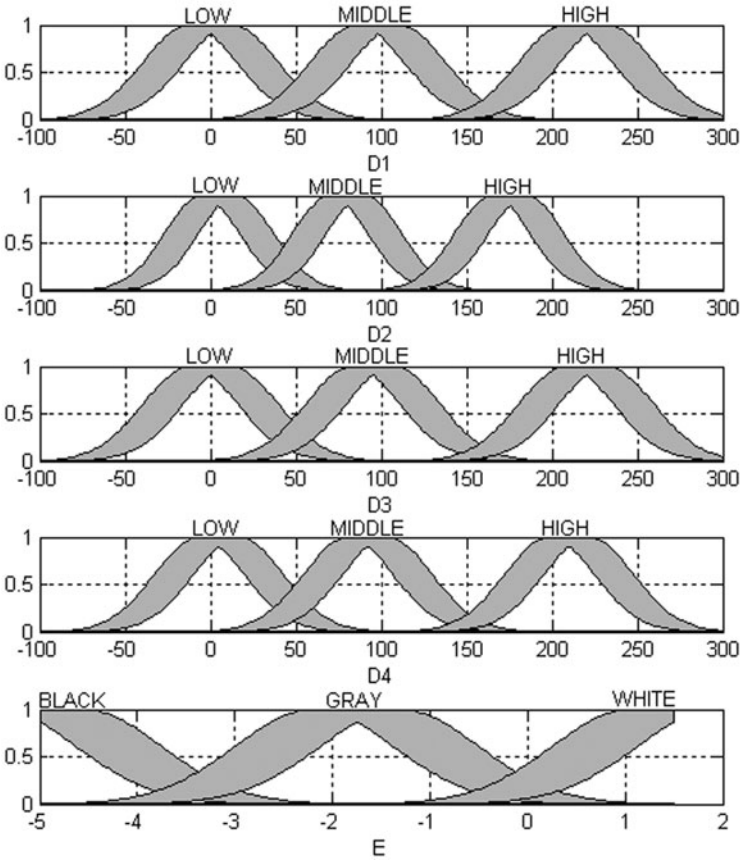


Fig. 11.6 Membership functions of the variables for the MG + FIS2 edge detector

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 the k-fold cross-validation method, with $k = 5$ for both databases. We can generalize the calculation of folds size m or number of samples in each fold, dividing the total number of samples for each person s by the folds number, and then multiplying the result by the person number p (11.3), then the train dataset size i (11.4) can be calculated as the number of samples in $k-1$ folds m, and test dataset size t (11.5) are the number of samples in only one fold:

$$m = (s/k) * p \tag{11.3}$$

$$i = m(k-1) \tag{11.4}$$

$$t = m \tag{11.5}$$

The total number of samples used for each person were 10 for the ORL and YALE databases; then if the size m of each fivefold is 2, the number of samples for training

Table 11.1 Particular information for the tested database of faces

Database	Pearson 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

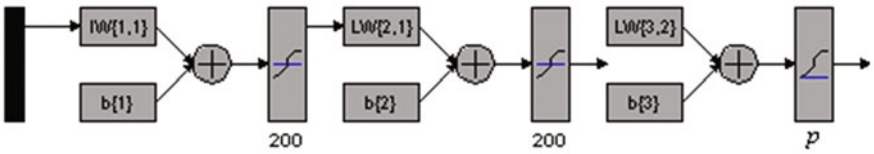


Fig. 11.7 General structure for the monolithic neural network

for each person is 8 and for testing is 2. For the experiments with the FERET face database, we use only the samples of 74 persons who have four frontal sample images. The particular information for each database is shown in Table 11.1.

11.4.3 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 optimal. We used the same structure for multi-net modular neural networks, in order to establish a standard for comparison for all the experiments [13, 15, 17, 21]. The general structure for the monolithic neural network is shown in Fig. 11.7:

- Two hidden layers with 200 neurons
- Learning Algorithm: Gradient descent with momentum and adaptive learning rate back-propagation
- Error goal 0.0001

11.5 Simulation Results

In this section, we show the numerical results of the experiments. Table 11.2 contains the results for the ORL face database, Table 11.3 contains the results for the Cropped Yale database, and Table 11.4 contains the results for the FERET face database.

For a better appreciation of the results, we made plots for the values presented in Tables 11.2, 11.3 and 11.4. Even if this work does not pretend to make a comparison

Table 11.2 Recognition rates for the ORL database of faces

Training set preprocessing 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 11.3 Recognition rates for the Cropped Yale database of faces

Training set preprocessing 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 11.4 Recognition rates for the FERET database of faces

Training set preprocessing 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

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. 11.8, the lowest training times are for the MG + 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. 11.9, where we can notice that the best mean performance of the neural network was achieved when it was trained with the datasets obtained with the MG + FIS2 and Sobel + FIS2 edge detectors.

Figure 11.10 shows that the recognition rates are also better for the edge detectors improved with interval type-2 fuzzy systems. The maximum recognition rates

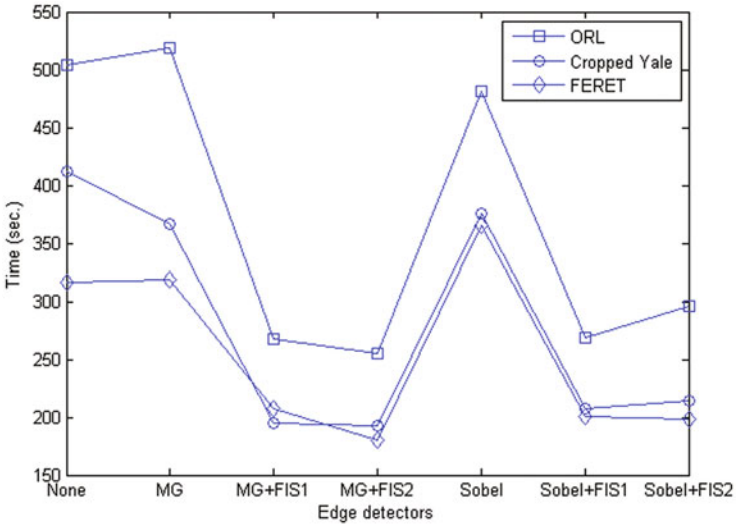


Fig. 11.8 Training time for the compared edge detectors tested with the ORL, Cropped Yale, and FERET face databases

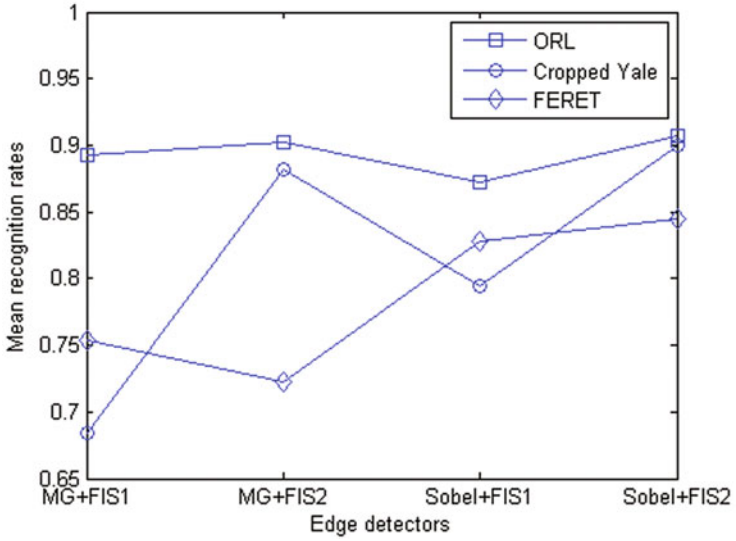


Fig. 11.9 Mean recognition rates for the compared edge detectors with ORL, Cropped Yale, and FERET face databases

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 the ORL

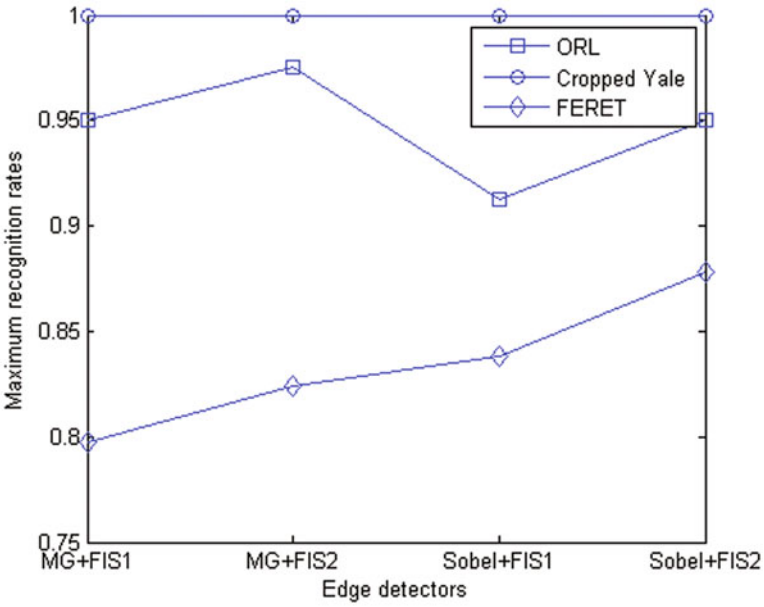


Fig. 11.10 Maximum recognition rates for the compared edge detectors with ORL, Cropped Yale, and FERET face database

dataset preprocessed with the MG + FIS2 edge detector. 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.

11.6 Conclusions

This work is the first effort for developing a formal comparison method for edge detectors as a function of their performance in different types of recognition systems. In this work, we demonstrate that Sobel and MG edge detectors improved with type-2 fuzzy logic have a better performance than the traditional methods in an image recognition system based on neural networks.

References

1. A.N. Evans, X.U. Liu, Morphological gradient approach for color edges detection. IEEE Trans. Image Process. **15**(6), 1454–1463 (2006)
2. A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, 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)
3. AT & T Laboratories Cambridge, The ORL database of faces, http://www.cl.cam.ac.uk /research/dtg/attarchive/facedatabase.html. Accessed 9 Dec 2013

4. F. Russo, G. Ramponi, Edge extraction by FIRE operators fuzzy systems. IEEE World Congr. Comput. Intell., **1**, 249–253 (1994)
5. H. Bustince, E. Berrenechea, M. Pagola, J. Fernandez, *Interval-Valued Fuzzy Sets Constructed from Matrices: Application to Edge Detection, Fuzzy Sets and Systems* (Elsevier), http://www.sciencedirect.com. Accessed 13 Dec 2013
6. J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems : Introduction and New Directions* (Prentice-Hall, Upper Saddle River, 2001)
7. J.R. Castro, O. Castillo, P. Melin, A. Rodriguez-Diaz, Building fuzzy inference systems with a new interval type-2 fuzzy logic toolbox. *Transactions on Computational Science*, vol. 4750 (Springer, Heidelberg, 2008), pp. 104–114
8. K. Revathy, S. Lekshmi, S.R. Prabhakaran Nayar, Fractal-based fuzzy technique for detection of active regions from solar. J. Solar Phys. **228**, 43–53 (2005)
9. K. Suzuki, I. Horiba, N. Sugie, M. Nanki, Contour extraction of left ventricular cavity from digital subtraction angiograms using a neural edge detector. Syst. Comput. Jpn., 55–69 (2003)
10. K.C. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell., **27**(5), 684–698 (2005)
11. L. Hua, H.D. Cheng, Ming Zhanga, A high performance edge detector based on fuzzy inference rules. Int. J. Inf. Sci. **177**(21), 4768–4784 (2007) (Elsevier, New York)
12. M. Heath, S. Sarkar, T. Sanocki, K.W. Bowyer, A robust visual method for assessing the relative performance of edge-detection algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **19**(12), 1338–1359 (1997)
13. O. Mendoza, P. Melin, The fuzzy Sugeno integral as a decision operator in the recognition of images with modular neural networks. *Hybrid Intelligent Systems* (Springer, Germany, 2007), pp. 299–310
14. O. Mendoza, P. Melin, G. Licea, 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, 2007)
15. O. Mendoza, P. Melin, G. Licea, A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks and the sugeno integral. Inf. Sci. **179**(13), 2078–2101 (2007) (Elsevier, New York)
16. O. Mendoza, P. Melin, G. Licea, Fuzzy inference systems type-1 and type-2 for digital images edges detection. Eng. Lett., Int. Assoc. Eng., E.U.A., **15**(1) (2007) http://www.engineeringletters.com/issues_v15/issue_1/EL_15_1_7.pdf
17. O. Mendoza, P. Melin, G. Licea, Interval type-2 fuzzy logic for module relevance estimation in Sugeno integration of modular neural networks. *Soft Computing for Hybrid Intelligent Systems* (Springer, Germany, 2008), pp. 115–127.
18. O. Mendoza, P. Melin, G. Licea, A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks and the Sugeno integral. Inf. Sci. **179**(3), 2078–2101 (2008) (Elsevier)
19. O. Mendoza, P. Melin, G. Licea, Interval type-2 fuzzy logic for edges detection in digital images. Int. J. Intell. Syst. **24**(11), 1115–1134 (2009) (Wiley, New York)
20. O. Mendoza, P. Melin, G. Licea, Interval type-2 fuzzy logic and modular neural networks for face recognition applications. Appl. Soft Comput. J. **9**(4), 1377–1387 (2009)
21. O. Mendoza, P. Melin, O. Castillo, G. Licea, 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)* (Springer, Germany, 2007), pp. 604–612
22. P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10), 1090–1104 (2000)
23. Y. Yitzhaky, E. Peli, A method for objective edge detection evaluation and detector parameter selection. IEEE Trans. Pattern Anal. Mach. Intell. **25**(8), 1027–1033 (2003)

Chapter 12
Big Data Analytic via Soft Computing Paradigms

Mo Jamshidi, Barney Tannahill, Yunus Yetis and Halid Kaplan

Abstract Large sets of data have been accumulating in all aspects of our lives for a long time. Advances in sensor technology, the Internet, social networks, wireless communication, and inexpensive memory have all contributed to an explosion of "Big Data." System of Systems (SoS) integrate independently operating, nonhomogeneous systems to achieve a higher goal than the sum of the parts. Today's SoS are also contributing to the existence of unmanageable "Big Data." Recent efforts have developed a promising approach, called "data analytic," which uses statistical and computational intelligence (CI) tools such as principal component analysis (PCA), clustering, fuzzy logic, neuro-computing, evolutionary computation, Bayesian networks, etc. to reduce the size of "Big Data" to a manageable size and apply these tools to (a) extract information, (b) build a knowledge base using the derived data, and (c) eventually develop a nonparametric model for the "Big Data." This chapter attempts to construct a bridge between SoS and data analytic to develop reliable models for such systems. The first application prediction of the stock market close is presented using a neural network paradigm. In the second application, a photovoltaic energy forecasting problem of a micro-grid SoS will be offered here for a case study of this modeling relation. The given input data represent the market price of the years which are between 2012 and 2013. The real exchange rate value of the NASDAQ stock market index is used. The results in both applications are quite promising.

The early versions of the techniques in this chapter were presented at the 8th SoSE Conference, Maui, HI, June 4-6, 2013, and at the 3rd World Conference on Soft Computing, San Antonio, TX, December 16-18, 2013.

M. Jamshidi (✉) · Y. Yetis · H. Kaplan
ACE Laboratory, The University of Texas, 1 UTSA Circle, San Antonio, TX, USA
e-mail: moj@wacong.org

B. Tannahill
Southwest Research Institute, San Antonio, TX, USA

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9_12

12.1 Introduction

System of Systems (SoS) are integrated, independently operating systems working in a cooperative mode to achieve a higher performance. A detailed literature survey on definitions to applications of SoS and many applications can be found in recent texts by Jamshidi [1, 2]. The application areas of SoS are vast indeed. They are software systems like the Internet, cloud computing, health care, and cyber-physical systems all the way to such hardware-dominated cases like military, energy, transportation, etc. Data analytic and its statistical and intelligent tools including clustering, fuzzy logic, neuro-computing, data mining, pattern recognition, principal component analysis (PCA), regression analysis, and postprocessing, such as evolutionary computations have their own applications in forecasting, marketing, politics, and all domains of SoS. SoSs are generating "Big Data" which makes modeling of such complex systems a challenge indeed.

A typical example of SoS is the future smart grid, destined to replace conventional electric grids and their small-scale version known as a micro-grid designed to provide electric power to a home, an office complex, or a small local community. A micro-grid is an aggregation of multiple distributed generators (DGs) such as renewable energy sources and conventional generators, in association with energy storage units which work together as a power supply networked in order to provide both electric power and thermal energy for small communities which may vary from one common building to a smart house or even a set of complicated loads consisting of a mixture of different structures such as buildings, factories, etc. [2]. Typically, a micro-grid operates synchronously parallel to the main grid. However, there are cases in which a micro-grid operates in an islanded mode, or in a disconnected state [3]. Accurate predictions of received solar power can reduce operating costs by influencing decisions regarding buying or selling power from the main grid or utilizing nonrenewable energy generation sources.

Another typical example is stock market prediction. Prediction of stock market price is one of the most important issues in finance. Many researchers have presented their ideas about how to forecast the market price in order to make gain using different techniques, such as technical analysis and statistical analysis, with different methods [4].

Energy systems is one example of Big Data and SoS. Another important example is the financial markets in any region, continent, or a nation. Nowadays, artificial neural networks (ANNs), as a data analytic tool, have been applied in order to predict exchange stock market index prediction. ANNs are one of the data mining techniques that are learning the capability of the human brain. Data patterns may perform dynamics and are unpredictable because of complex financial data used. Several research efforts have been made to improve the computational efficiency of share values [5, 6].

ANNs have been used in stock market prediction during the decade. One of the first projects was by Kimoto et al. [7] who had used ANN for the prediction of the Tokyo stock exchange index. Mizuno et al. [8] applied ANN again to the Tokyo

stock exchange to predict buying and selling signals with an overall prediction rate of 63 %. Sexton and friends [9] concluded in 1998 that use of momentum and start of learning at random points may solve the problems that may occur in the training process. Phua et al. [10] applied neural network with genetic algorithm to the stock exchange market of Singapore and predicted the market direction with an accuracy of 81 %.

The object of this chapter is to use big data analytic approaches to predict or forecast the behavior of two important aspects of our times—stock market prediction and renewable energy availability. In each case, a massive amount of data is used to achieve these goals.

The remainder of this chapter is as follows. Section 12.2 briefly describes two such tools: PCA and ANN. Stock market prediction using ANN will be presented in Sect. 12.3, while Sect. 12.4 describes irradiance components forecast of a PB-based micro-grid. Section 12.5 then discusses the application and effectiveness of different data analytic tools in the generation of models and relations that could be leveraged to better optimize the operation of the micro-grid. Finally, Sect. 12.6 concludes the chapter.

12.2 Brief Description of Soft Computing Tools for Data Analytic

12.2.1 Principal Component Analysis

PCA is a scheme to identify patterns in data sets so that its similarities and differences are highlighted. Since patterns in data can be hard to find in data of high dimensions, PCA can help reduce the dimension of the data while bringing up the principal meaning of the information in the data. In other words, PCA can first find the pattern and then compress the data. PCA can work with both numerical and image data. The following steps can summarize simple steps to perform PCA [11]:

Step 1: Get a data set
Step 2: Subtract the mean from each data value.
Step 3: Calculate the covariance matrix.
Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix.
Step 5: Choose components and form a feature vector.

The principal component of the data will be near to the eigenvector of the covariance matrix with the largest eigenvalue.

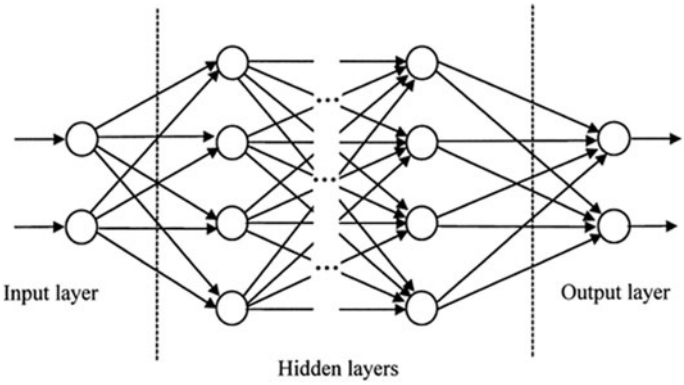


Fig. 12.1 Architecture of a feed-forward multilayer perceptron

12.2.2 Artificial Neural Networks

ANNs are an information processing system that was first inspired by generalizations of mathematical of human neuron (Fig. 12.1).

Each neuron receives signals from other neurons or from outside (input layer). The multilayer perceptron (MLP), shown in Fig. 12.1, has three layers of neurons, where one input layer is present. Every neuron employs an activation function that fires when total input is more than a given threshold. In this chapter, we focus on MLP networks that are layered feed-forward networks typically trained with static backpropagation. These networks are used for application static pattern classification [12, 13].

One of the learning methods in MLP neural networks selects an example of training, making a forward and a backward pass. The first advantage of MLP networks is their ease of use and approximation of any input or output map. The first disadvantage is that they train very slowly and require a lot of training data. It should be said that the learning speed will dramatically decrease according to the increase of the number of neurons and layers of the networks.

12.3 Stock Market Data Analytic

The stock prediction data analytic was divided into the next two sections.

12.3.1 Training Process

Training is the process by which the free parameters of the networks, such as optimal weight values, are determined. Learning models, which are used for MLP, train

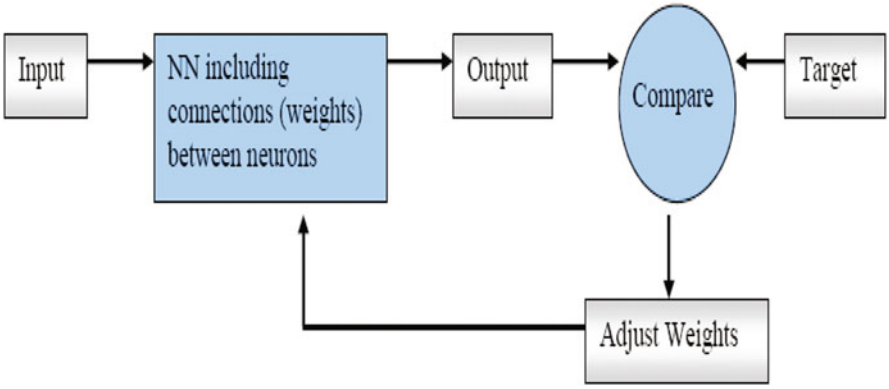


Fig. 12.2 Network based on a comparison of the output and the target. (Source: Ball and Tissot [14])

certain output nodes to respond to certain input patterns and the changes in connection weights, due to learning, cause those same nodes to respond to more general classes of patterns. In these models, input layer units distribute input signals to the network. Connection weights modify the signals that pass through it. Hidden layers and output layer contain a vector of processing elements with an activation function. Usually, the sigmoid is used as the activation function.

Once the network weight and biases have been initialized, the network is ready for training. During the training, the network is adjusted based on a comparison of the output and the target (Fig. 12.2)

The training process requires a set of examples of proper network behavior and target outputs. During training, the weights and biases of the network are iteratively adjusted to minimize the network performance function. The most common performance function for feed-forward networks is mean square error (MSE), which is the average squared error between the networks outputs and the target outputs [15].

Training of the network takes place through the backpropagation algorithm (similar to a conjugate gradient optimization approach) from output layer to the input layer. Backpropagation is one of the important ways for the training process because hidden units do not have training target value that can be used, so they must be trained based on errors from previous layers. Output is the only layer which has a target value in order to compare. Training occurs until the errors in the weights. Network training continues until the norm $\|.\|$ of the error between current and previous set of weights of the network is less than a threshold value. MLP is the most common type of feed-forward networks (Fig. 12.3). MLP has three types of layers: an input, an output, and a hidden layer.

Table 12.1 shows the stock data from the 2013 NASDAQ [16]. The data include opening, high, low, and close prices of NASDAQ and the volume of stocks for given dates.

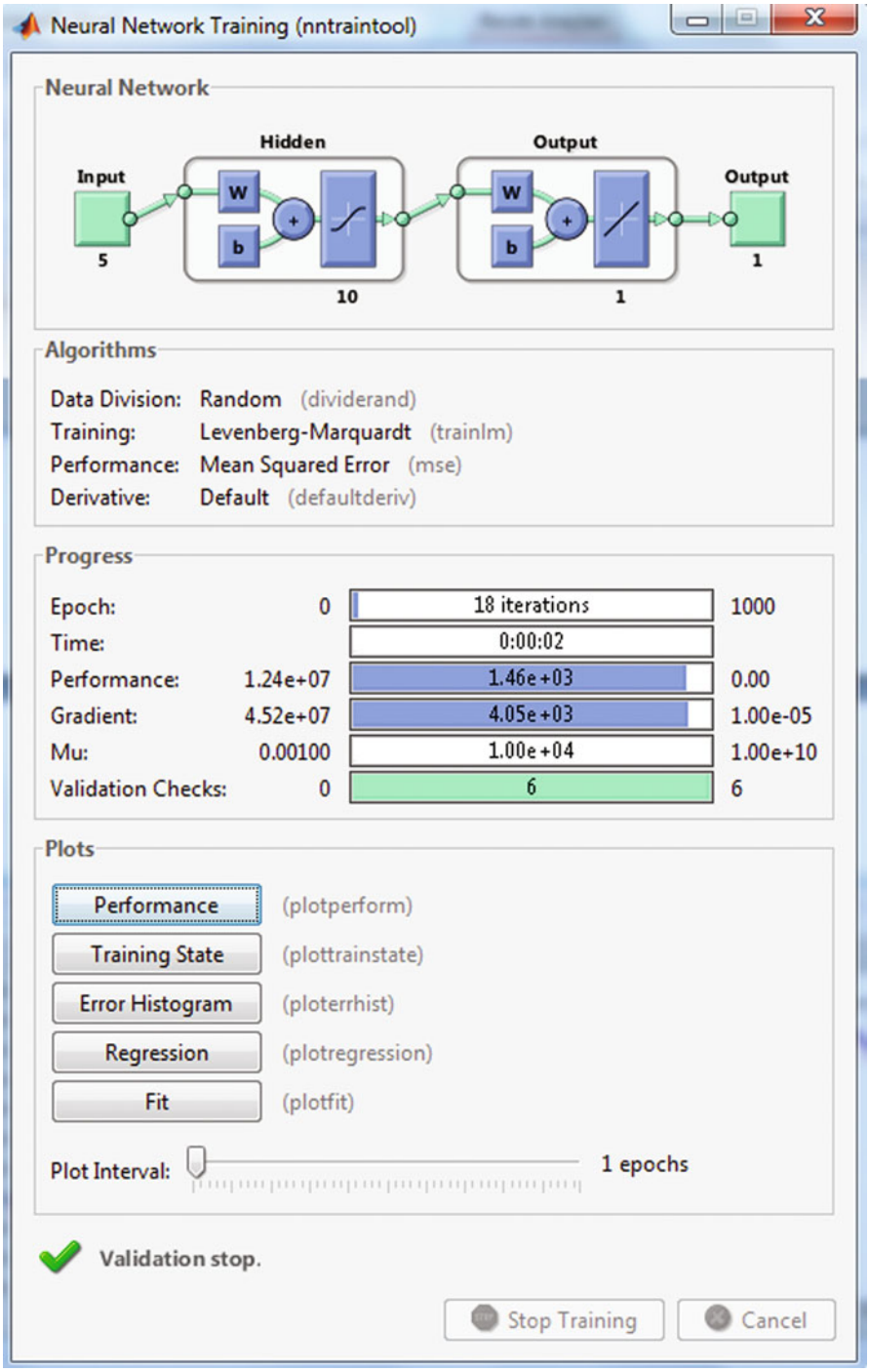


Fig. 12.3 Neural network training

Table 12.1 Stock data from the 2013 NASDAQ. (Source from NASDAQ, April 2013)

Date	Open	High	Low	Close	Volume	Adjusted close
4/15/2013	3277.58	3283.4	3213.46	3216.49	1,779,320,000	3216.49
4/12/2013	3292.39	3296.5	3271.02	3294.95	1,471,180,000	3294.95
4/11/2013	3289.59	3306.95	3287.74	3300.16	1,829,170,000	3300.16
4/10/2013	3246.06	3299.16	3245.8	3297.25	1,769,870,000	3297.25
4/9/2013	3229.81	3249.95	3215.02	3237.86	1,498,130,000	3237.86
4/8/2013	3207.15	3222.26	3195.57	3222.25	1,323,520,000	3222.25
4/5/2013	3174	3206.21	3168.88	3203.86	1,594,090,000	3203.86
4/4/2013	3219.11	3226.24	3206.02	3224.98	1,475,720,000	3224.98
4/3/2013	3257.38	3260.15	3210.39	3218.6	1,813,910,000	3218.6
4/2/2013	3252.55	3267.93	3245.41	3254.86	1,580,800,000	3254.86
4/1/2013	3268.63	3270.23	3230.57	3239.17	1,481,360,000	3239.17
3/28/2013	3257.32	3270.3	3253.21	3267.52	1,636,800,000	3267.52
3/27/2013	3230.76	3258.26	3227.02	3256.52	1,420,130,000	3256.52
3/26/2013	3249.95	3252.93	3239.92	3252.48	1,444,500,000	3252.48
3/25/2013	3255.85	3263.63	3222.48	3235.3	1,666,010,000	3235.3
3/22/2013	3235.3	3247.94	3230.86	3245	1,681,360,000	3245
3/21/2013	3228.17	3237.57	3215.69	3222.6	1,692,260,000	3222.6

Table 12.1 (continued)

Date	Open	High	Low	Close	Volume	Adjusted close
3/20/2013	3251.91	3257.99	3240.9	3254.19	1,599,120,000	3254.19
3/19/2013	3246.7	3252.6	3205.42	3229.1	1,690,680,000	3229.1
3/18/2013	3215.71	3249.37	3211.1	3237.59	1,550,510,000	3237.59
3/15/2013	3260.46	3260.62	3242.65	3249.07	2,305,230,000	3249.07
3/14/2013	3253	3258.93	3250.24	3258.93	1,651,650,000	3258.93
3/13/2013	3243.04	3251.45	3230.62	3245.12	1,577,280,000	3245.12
3/12/2013	3244.85	3249.78	3229.92	3242.32	1,673,740,000	3242.32
3/11/2013	3237.74	3252.87	3233.67	3252.87	1,628,500,000	3252.87
3/8/2013	3245.85	3248.7	3227.89	3244.37	1,611,700,000	3244.37
3/7/2013	3224.5	3235.1	3221.47	3232.09	1,675,640,000	3232.09
3/6/2013	3233.31	3233.44	3217.67	3222.36	1,764,020,000	3222.36
3/5/2013	3200.38	3227.31	3200.27	3224.13	1,891,510,000	3224.13
3/4/2013	3159.46	3182.27	3154.79	3182.03	1,718,290,000	3182.03
3/1/2013	3143.54	3171.5	3129.4	3169.74	1,870,250,000	3169.74
2/28/2013	3161.43	3182.6	3159.72	3160.19	2,022,530,000	3160.19
2/27/2013	3129.72	3177.8	3127.27	3162.26	1,727,260,000	3162.26
2/26/2013	3126.23	3135.57	3105.36	3129.65	1,847,750,000	3129.65

Table 12.1 (continued)

Date	Open	High	Low	Close	Volume	Adjusted close
2/25/2013	3180.59	3186.25	3116.25	3116.25	1,930,990,000	3116.25
2/22/2013	3149.09	3161.82	3139.55	3161.82	1,581,500,000	3161.82
2/21/2013	3154.88	3155.19	3118.62	3131.49	2,052,630,000	3131.49
2/20/2013	3211.99	3213.25	3163.95	3164.41	2,001,800,000	3164.41
2/19/2013	3197.46	3213.6	3194.92	3213.59	1,843,840,000	3213.59
2/15/2013	3202.84	3206.21	3184.03	3192.03	1,858,670,000	3192.03
2/14/2013	3182.74	3202.33	3182.39	3198.66	1,924,900,000	3198.66
2/13/2013	3195.34	3205.52	3187.06	3196.88	1,822,450,000	3196.88
2/12/2013	3190.73	3196.92	3184.84	3186.49	1,786,800,000	3186.49
2/11/2013	3192.53	3194.01	3182.19	3192	1,551,370,000	3192
2/8/2013	3178.06	3196.89	3177.18	3193.87	1,816,480,000	3193.87
2/7/2013	3167.44	3170.42	3135.98	3165.13	1,955,960,000	3165.13
2/6/2013	3159.38	3174.82	3157.35	3168.48	2,002,740,000	3168.48
2/5/2013	3140.9	3178.52	3136.82	3171.58	2,150,080,000	3171.58
2/4/2013	3161.72	3169.63	3130.57	3131.17	1,874,750,000	3131.17
2/1/2013	3162.94	3183.14	3154.91	3179.1	2,012,930,000	3179.1
1/31/2013	3140.67	3154.18	3136.82	3142.13	2,190,840,000	3142.13
1/30/2013	3157.43	3164.06	3135.83	3142.31	2,014,350,000	3142.31

Table 12.1 (continued)

Date	Open	High	Low	Close	Volume	Adjusted close
1/29/2013	3149.62	3156.94	3133.11	3153.66	2,050,670,000	3153.66
1/28/2013	3152.17	3161.83	3144.89	3154.3	1,935,590,000	3154.3
1/25/2013	3140.65	3156.2	3135.86	3149.71	1,920,250,000	3149.71
1/24/2013	3125.67	3153.56	3124.45	3130.38	2,046,990,000	3130.38
1/23/2013	3155.82	3161.06	3149.74	3153.67	1,698,190,000	3153.67
1/22/2013	3135.63	3143.18	3121.54	3143.18	1,790,730,000	3143.18
1/18/2013	3127.91	3134.73	3119.2	3134.71	1,860,070,000	3134.71
1/17/2013	3130.49	3144.05	3125.79	3136	1,766,510,000	3136
1/16/2013	3110.72	3124.65	3106.79	3117.54	1,692,380,000	3117.54
1/15/2013	3101.06	3112.29	3093.32	3110.78	1,852,870,000	3110.78
1/14/2013	3113.65	3123.48	3104.08	3117.5	1,876,050,000	3117.5
1/11/2013	3122.12	3126.59	3114.1	3125.64	1,772,600,000	3125.64
1/10/2013	3125.64	3127.72	3098.47	3121.76	1,754,240,000	3121.76
1/9/2013	3099.65	3111.22	3096.34	3105.81	1,732,510,000	3105.81
1/8/2013	3098.46	3103.39	3076.6	3091.81	1,744,380,000	3091.81
1/7/2013	3089.17	3102.35	3083.88	3098.81	1,702,540,000	3098.81
1/4/2013	3100.88	3108.44	3090.81	3101.66	1,745,140,000	3101.66
1/3/2013	3108.49	3118.18	3092.28	3100.57	1,769,420,000	3100.57
1/2/2013	3091.33	3112.65	3083.49	3112.26	2,111,300,000	3112.26

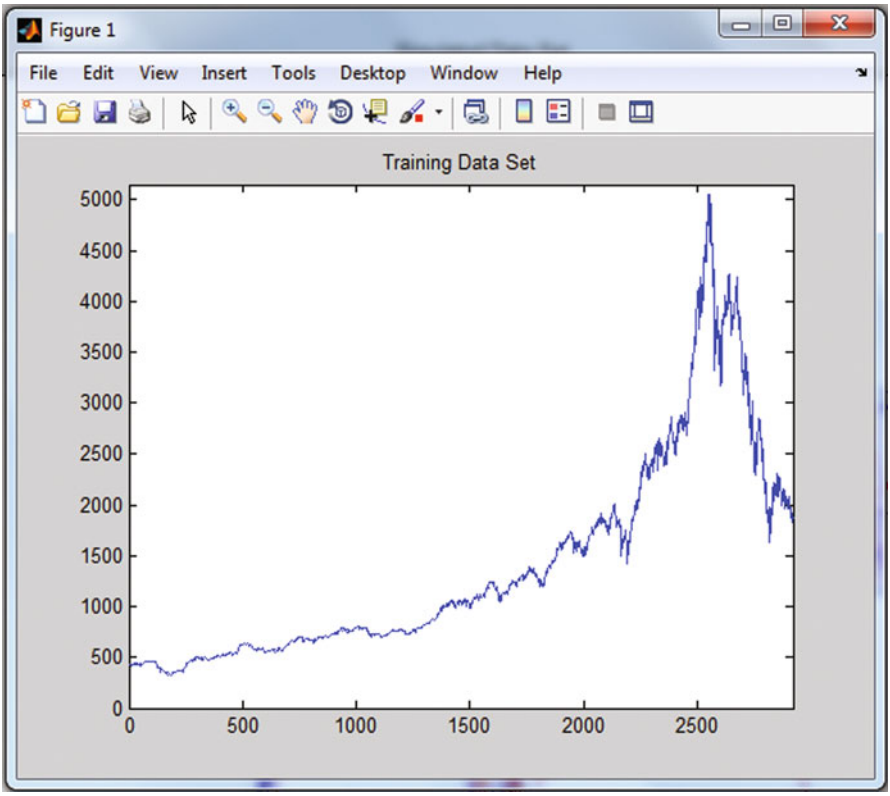


Fig. 12.4 Training data set

Fig. 12.5 Randomly divided three sets of input and target vectors

With these settings, the input vectors and target vectors will be randomly divided into three sets as follows (Fig. 12.4): 70 % will be used for training and 15 % will be used to validate the network and to stop training before overfitting.

The last 15 % will be used as a completely independent test of network generalization (Fig. 12.5).

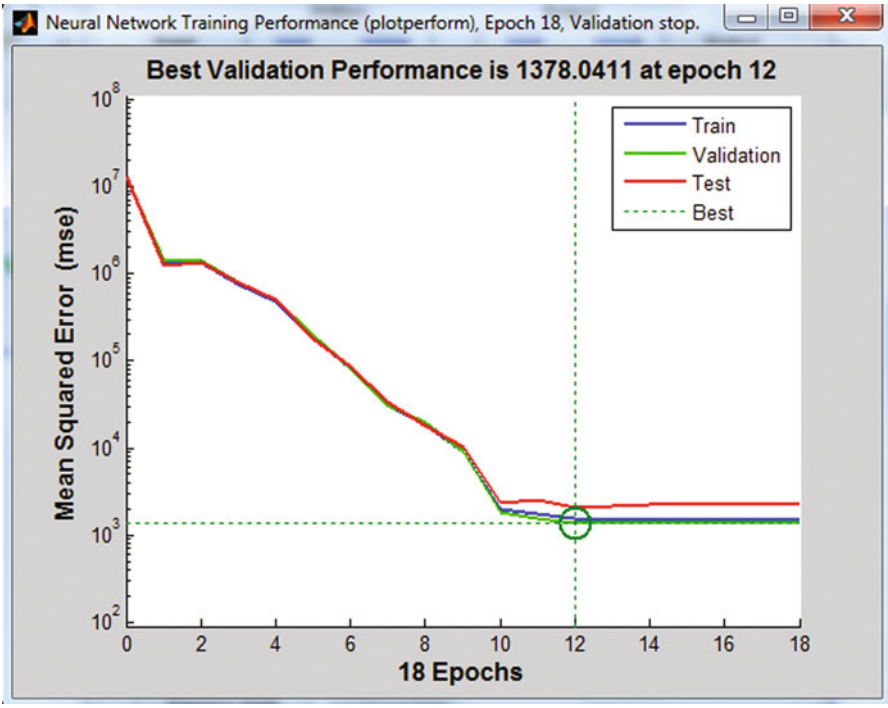


Fig. 12.6 Performance curve for training

12.3.2 Experimental Results

The data set of NASDAQ daily stock price has been used for experiments [16]. There are six parameters of opening price, high, low, volume, adjusted close, and closing price. The inputs are used opening price, low and high during the day, volume, and adj. (adjusted) closing price for predicting stock price (Fig. 12.10). The output is the closing price of the day. Data are shown above in Table 12.1 and Fig. 12.11.

This observation is taken when other parameters are:

- Hidden neurons: 10
- Learning rate (alpha): 0.4
- Momentum constant (mom): 0.75
- Max epochs (epochs): 1000

Mean squared error I (MSE) is the average squared difference between outputs and targets. Lower values are better. If the test curve had increased significantly before the validation curve increased, it means it is possible that some overfitting might have occurred [17]. According to the above information which explains the diagram's principle of operation, the result is reasonable because of the following considerations which are the train set error, the test curve had increased before the validation curve increased (Fig. 12.6).

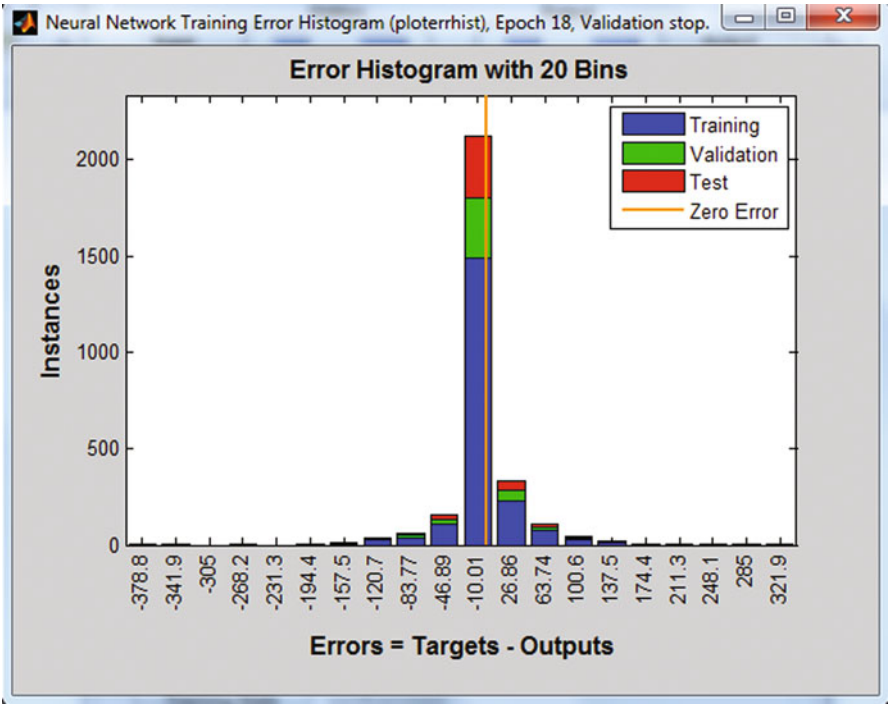


Fig. 12.7 Error histogram

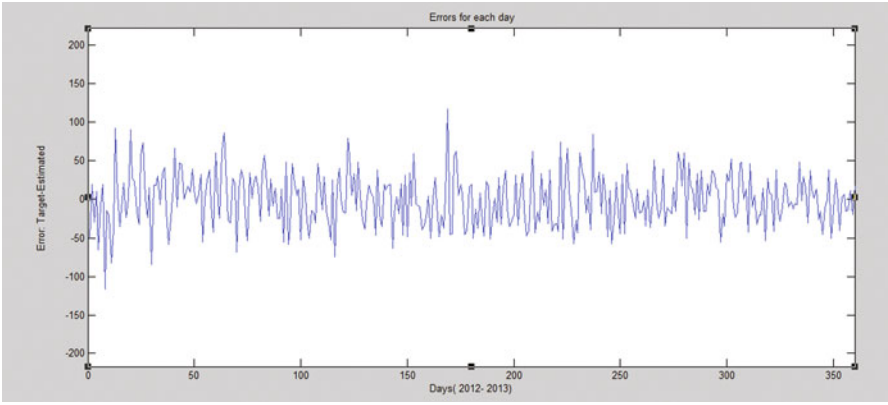


Fig. 12.8 Errors for each day

Error histogram obtains additional verification of network performance. It can be clearly seen that errors are between -120 and $+100$. Data sets are represented in hundreds of thousands so these errors are negligible considering the error is smaller than approximately 0.02 % of targets (Fig. 12.7 Fig. 12.8).

One of the important issues is that making the ANN implement better relates to input normalization. Each input variable should be preprocessed. The mean value and average of training set are small compared to its standard deviation. The index range is between − 1 and + 1 [18]. We are able to use a simple formula which is *Index(x) = (Index(x) − Min (Index))/ (Max (Index) − Min (Index))* [18]. The regression plot of the training set can be seen clearly. Each of the figures corresponds to the target from the output array. *R* parameters are very close to 1. This means that correlation between the outputs and the target is very high.

Regression is used to validate the network performance. The following regression plots display the network outputs with respect to targets for training, validation, and test sets. For a perfect fit, the data should fall along a 45 degree line, where the network outputs are equal to the targets. For this problem, the fit is reasonably good for all data sets, with *R* values in each case of 0.99 or above (Fig. 12.9).

12.4 Photovoltaic Data Analytic

The proposed micro-grid SoS is shown in Fig. 12.12. Solar array, battery storage, direct current (DC)–alternating current (AC) inverter, load, and a controller to manage the entire system are shown in the figure. Ultimately, we want to forecast received solar power as a model based on real-time environmental measurements to be used in an energy management system [3, 19] to minimize operating costs.

This micro-grid represents a facility-scale cyber-physical system (CPS) or an SoS consisting of a building with:

- A fixed (or with tracking system) solar photovoltaic (PV) system
- A load demand in the form of overall energy consumption; heating, ventilation, and air conditioning (HVAC); and lighting, with bidirectional communications (e.g., bidirectional inverter)
- A reconfigurable control and acquisition system (i.e., with open I/O modules, embedded controller for communication, processing, and a user-programmable field-programmable gate array (FPGA))
- A local, off-site, or cloud-based computing infrastructure for simulation/computational analysis.

12.4.1 PV Data Description

To ensure that the PV input data for the different data analysis tools are comprehensive, data from different sources were combined to form the full data set. This was possible because of the solar research projects occurring in Golden, CO, where the National Renewable Energy Laboratory (NREL) is conducting long-term research and data recording to support the growing renewable energy industry.

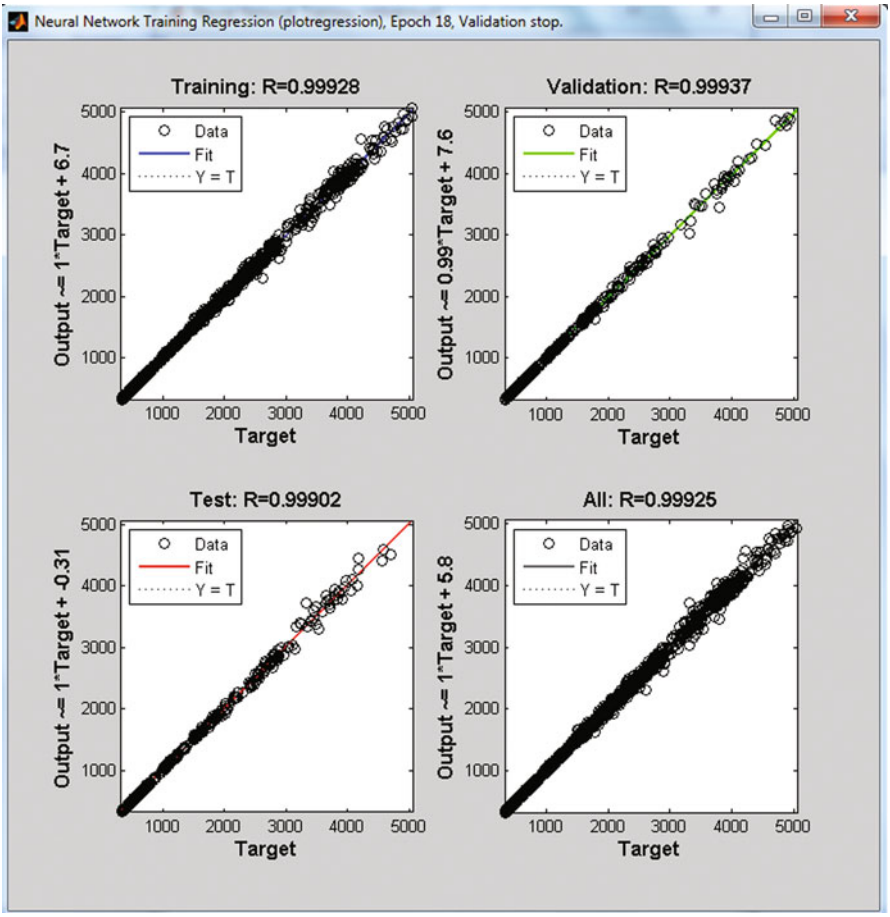


Fig. 12.9 Regression plot for training

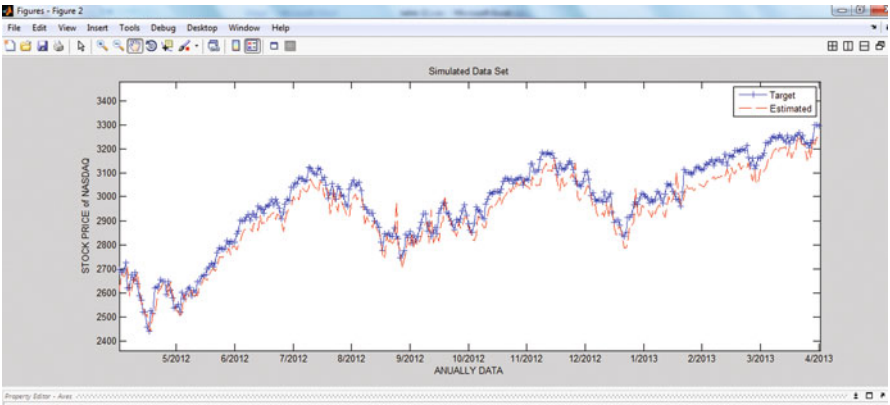


Fig. 12.10 Target and estimated data of annual NASDAQ data

Fig. 12.11 Real data of NASDAQ stock price. (Source: finance.yahoo.com)

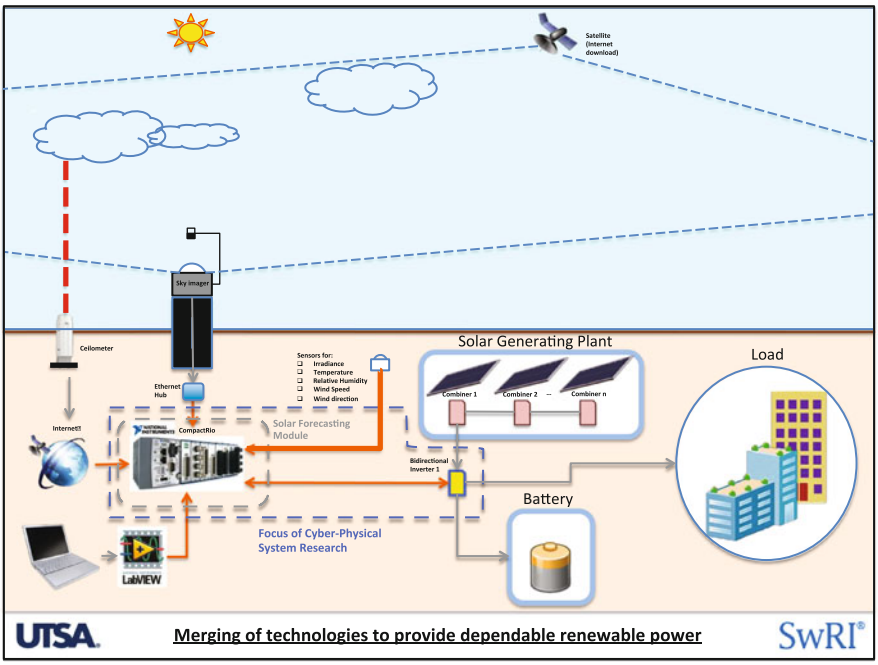


Fig. 12.12 A PV forecasting system as a constituent of a micro-grid SoS [19]

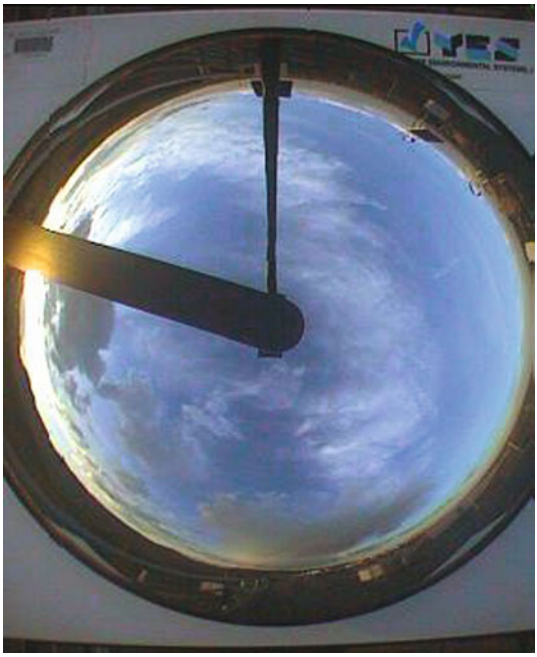


Fig. 12.13 Sample sky image

The first source was the data recorded by the Solar Radiation Research Laboratory (SRRL), which employs over 70 instruments to measure solar conditions and environmental parameters [20]. Also, this data set includes 180° images of the sky that are used to determine current cloud conditions directly. An example of this is shown in Fig. 12.13.

The second source of data was the SOLPOS data, made available by the Measurement and Instrumentation Data Center (MIDC), which has stations throughout North America to capture information on solar position and available solar energy [21]. Luckily, the MIDC has a station near NREL, so their data can be used in conjunction with the SRRL data.

The final set of data originates from the Iowa Environmental Mesonet (IEM) [22]. Their Automated Surface Observing System (ASOS) station near the Golden, CO, site was also included to have current weather data in the set.

Data from the month of October 2012 were combined from the different sources of data. This final set includes one sample for each minute of the month and incorporates measured values for approximately 250 different variables at each data point. The data set was sanitized to only include data points containing valid sensor data prior to the analysis.

12.4.2 Data Analytic of PV Data and Objective

In this section, the analysis steps are described, and the results from the different techniques are compared. The goal is to use data analytic tools to generate a useful model from the data set without needing to resort to parametric analysis and the use of subject matter experts.

Since the micro-grid would benefit from predicted values of solar irradiance, it was decided that the output of the data analytic should be 60-minute predicted values of three key irradiance parameters (Global Horizontal Irradiance, GHI; Direct Horizontal Irradiance, DHI; and Direct Normal Irradiance, DNI).

12.4.3 Input Variable Down-Selection and Data Cleanup

The input variables were down-selected from the full data set to only include cloud levels, humidity, temperature, wind speed, and current irradiance levels. If this exercise was conducted using "Cloud" computing, the number of variables might not need to be down-selected; however, since this effort took place on a single PC, the number of variables was reduced.

Next, the data set was further reduced by removing data points in which GHI, DHI, and DNI levels were very low. The primary reason for this second step was to reduce the amount of time and memory necessary for analysis. Figure 12.14 is a graph containing the measurements of GHI, DHI, and DNI over 1 day in the cleaned data set.

12.5 Nonparametric Model General Tools

After cleaning took place, the data could be fed into either of the two nonparametric model-generating tools, the fuzzy inference system generator and backpropagation neural network training tools included in the MATLAB Fuzzy Logic Toolbox and the Neural Network Toolbox.

12.5.1 Nonparametric Model Generation Tools

The Fuzzy Logic Toolbox function used in this exercise, *genfis3,* uses fuzzy C-means clustering to cluster values for each variable which produces fuzzy membership functions for each of the variables in the input matrix and output matrix. It then determines the rules necessary to map each of the fuzzy inputs to the outputs to best match the training data set. These membership functions and rules can be viewed using the MATLAB FIS GUI tools such as *ruleview*. When run with default parameters, the *genfis3* function ran significantly slower and performed worse than MATLAB's neural network fitting function.

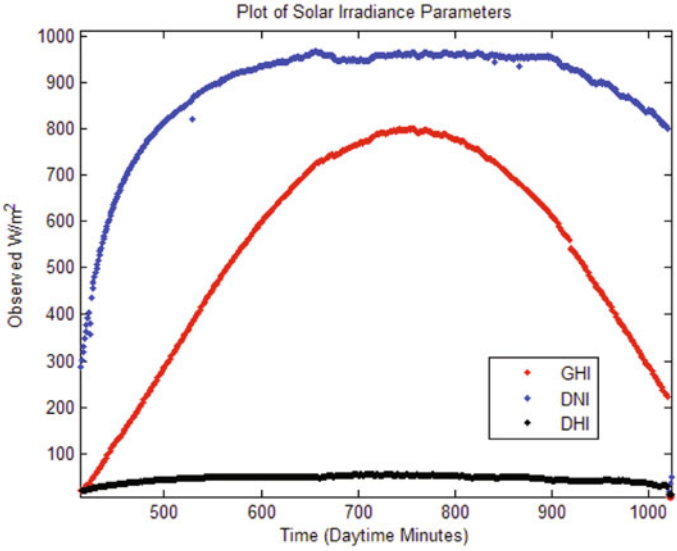


Fig. 12.14 Three key irradiance parameter plots for a clear day

Note that in Fig. 12.15, the differences in the observed and predicted data points generally correspond to the presence of clouds or other anomalies that could not be predicted an hour in advance using the variables input to the function.

12.5.2 Neural Network Fitting Tool

The second model-generating method was the MATLAB Neural Network training tool. By default, this tool uses the Levenberg–Marquardt backpropagation method to train the network to minimize its mean squared error performance. Results from training one sample set are shown in Figs. 12.16–12.18.

12.5.3 Additional Preprocessing Discussion

Once the initial performance of these two tools was evaluated, it was decided that further effort should go into including a greater number of original input variables and including additional preprocessed parameters in the training data in an effort to enhance the performance of the derived model. This effort took three paths, the calculation of nonlinear input parameters, the inclusion of a greater number of input parameters, and the reduction of input data dimension when necessary in order to support the execution requirements of the two model generation tools.

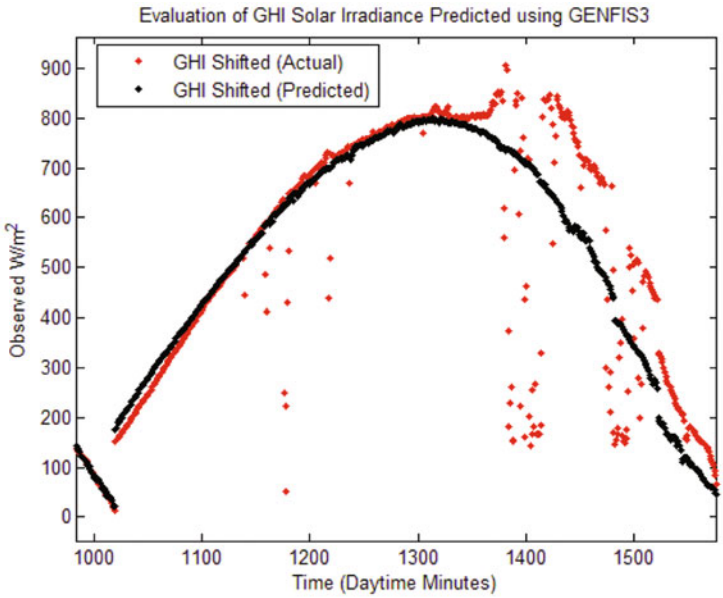


Fig. 12.15 Data generated using GENFIS3 based on 13 input variables

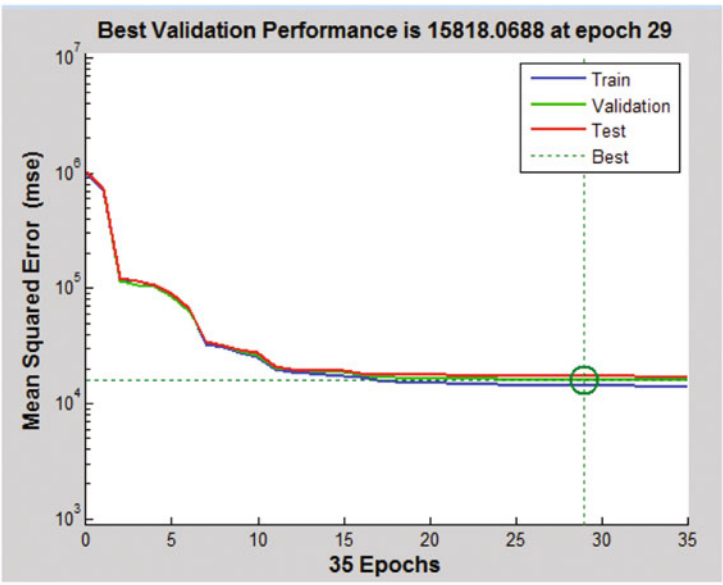


Fig. 12.16 Backpropagation performance curve

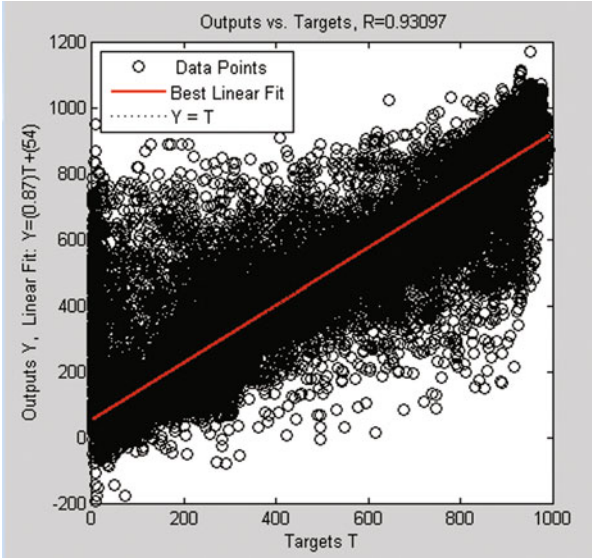


Fig. 12.17 Post-training network regression performance

12.5.3.1 Nonlinear Data-Set Expansion

In an effort to derive additional useful input parameters from the existing data set, each variable included in the data set generated several additional variables based on nonlinear functions and past values of the variable itself. Inclusion of these parameters in the training data set greatly improved the performance of the training tools. A subject matter expert would be useful in this step to identify useful derived parameters such as these to add to the training data set.

12.5.3.2 Large Data Sets and PCA

Models were generated using different sets of input variables to try to assess the impact of incorporating increasing numbers of variables in the training data set. In general, the trained model performed better when more variables were included in the training data set; however, as the number of variables increased, the training execution time became excessive and out-of-memory errors occurred when the data sets became too large.

In order to combat this issue, the dimension of the training data set was reduced to a manageable size using PCA. PCA can be used to compress the information from a large number of variables to a smaller data set while minimizing the information lost during this process [10]. This can be performed directly on a data set using the *princomp* function in MATLAB.

The columns of the SCORE matrix returned by the *princomp* function represent the columns of the input data transformed to place the majority of the information in the data set in the first few principal components. The information distribution among the principal components is illustrated in Fig. 12.19. The higher eigenvalues

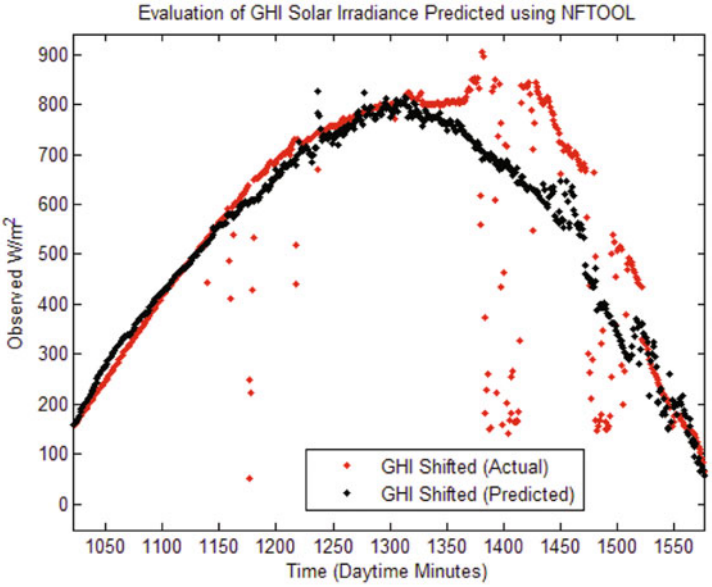


Fig. 12.18 Data generated using *NFTOOL* based on 13 input variables and ten hidden neurons

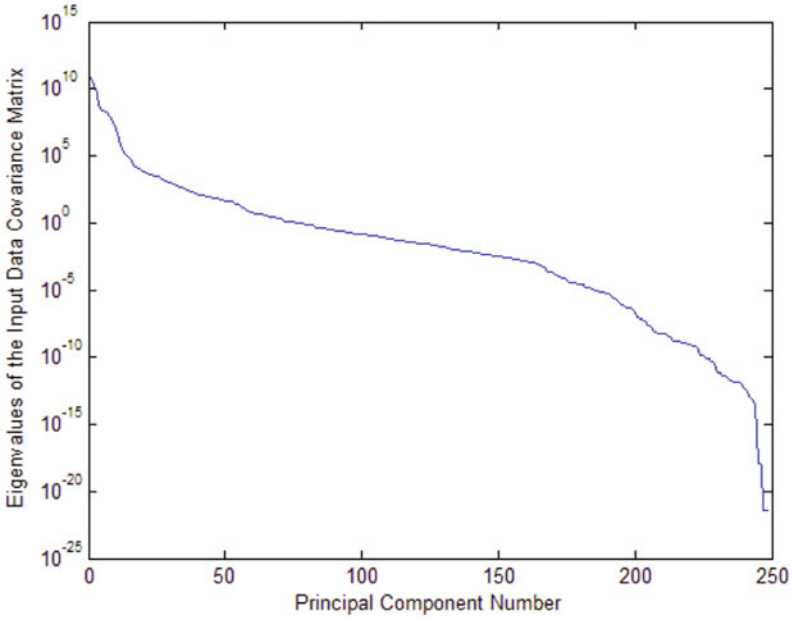


Fig. 12.19 Principal component information graph

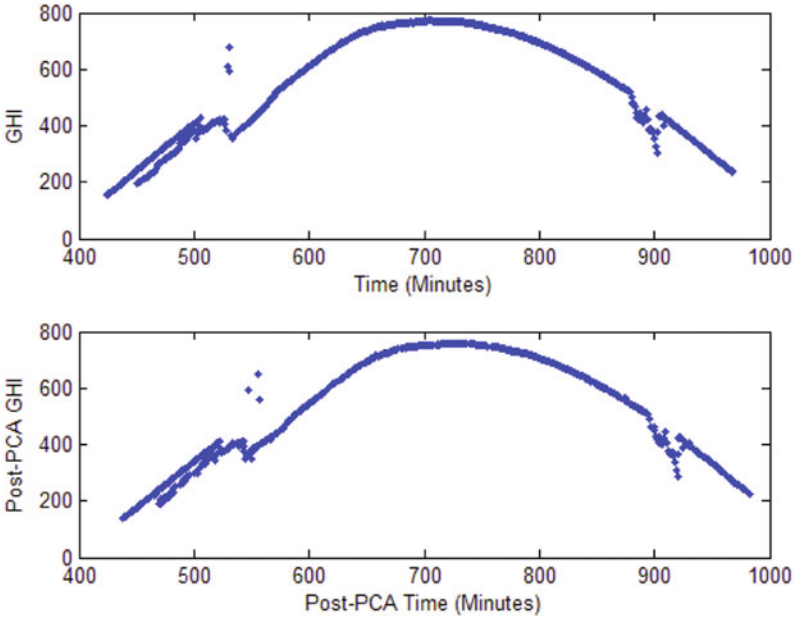


Fig. 12.20 Data recovery demonstration using first 50 principal components

represent the principal components with the most information. Incorporating principal components past 10 provides minimal additional information.

Figure 12.20 shows the quality of information recovery if transforming back to the original basis using only information from the first 50 principal components.

In this application, PCA was primarily useful because it allowed the reduction of very high-dimension data sets to smaller, more manageable data sets that could be used as training inputs to the model generation tools.

12.5.4 PV Forecasting Results

In order to generate the best nonparametric model possible, different combinations of data inputs to the *GENFIS3* and *NFTOOL* were considered. Different implementations of the options discussed above were evaluated during this analysis.

The best-performing NFTOOL-generated model utilized data from 244 original variables, which were then expanded to 1945 variables using derivatives calculations such as sin(.), d(.)/dt, cos(.), etc. Next, the dimension of the data was shrunk to 150 so that the training function had sufficient memory to train the network. The resulting network was the best of all the generated models.

The best-performing *GENFIS3*-generated model (Figs. 12.21–12.24) evaluated during this effort used the same input data set as mentioned in the paragraph above with the exception that the dimension was shrunk down to 50 using PCA. It was

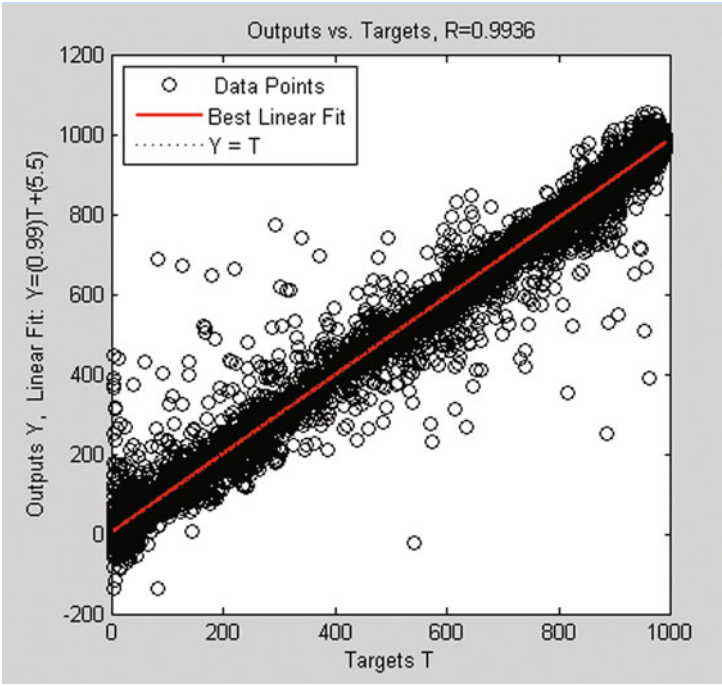


Fig. 12.21 Best neural network linear regression performance

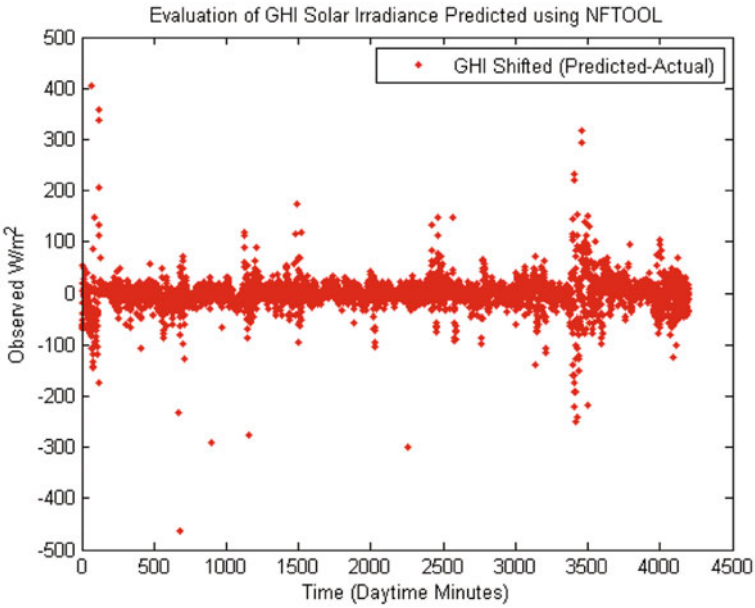


Fig. 12.22 Best neural network GHI error

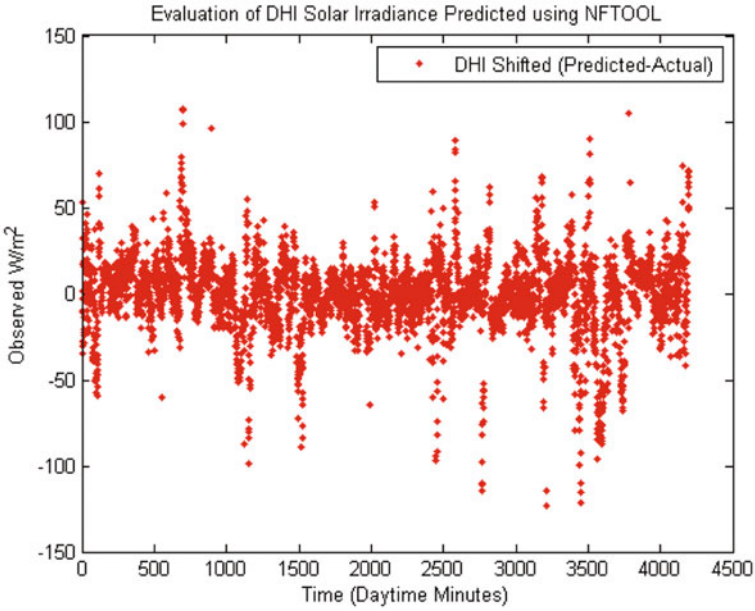


Fig. 12.23 Best neural network DHI error

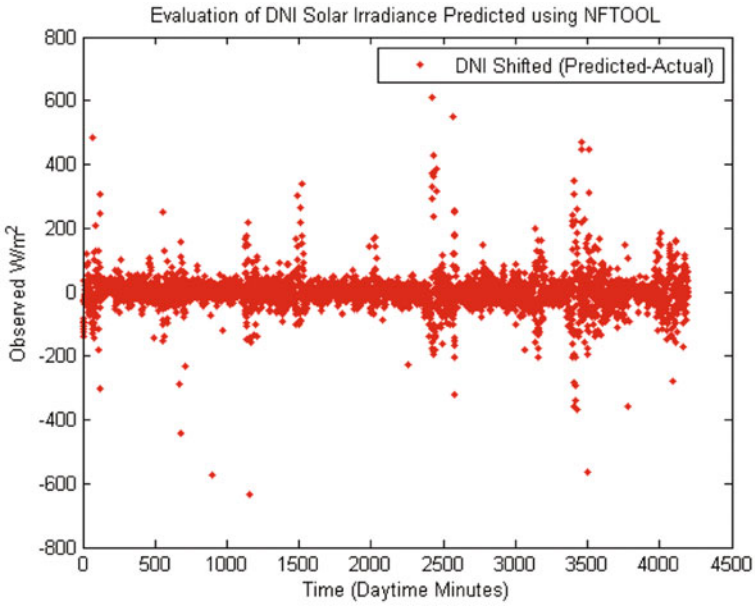


Fig. 12.24 Best neural network DNI error

Table 12.2 Performance comparison of the generated nonparametric models (GENFIS3)

Model Type	Input Params	Add Nonlin. Params	PCA?	Final Dim	MSE GHI	MSE DNI	MSE GHI	R
FIS	13	N	N	13	8.81E+03	4.98E+04	2.03E+03	0.906
FIS	32	N	N	32	8.68E+03	4.90E+04	1.99E+03	0.907
FIS	32	Y	Y	10	6.64E+03	3.58E+04	1.70E+03	0.938
FIS	32	Y	Y	50	6.35E+03	3.12E+04	1.85E+03	0.945
FIS	13	Y	N	97	4.13E+03	2.31E+04	1.14E+03	0.961
FIS	244	N	Y	50	1.06E+04	5.00E+04	2.40E+03	0.903
FIS	244	Y	Y	50	2.96E+03	2.08E+04	1.05E+03	0.967
FIS	244	Y	Y	150	1.82E+04	5.50E+04	3.35E+03	0.901

Table 12.3 Performance comparison of the generated nonparametric models (NN10)

Model Type	Input Params	Add Nonlin. Params	PCA?	Final Dim	MSE GHI	MSE DNI	MSE GHI	R
NN10	13	N	N	13	7.01E+03	3.41E+04	1.65E+03	0.934
NN10	32	N	N	32	6.94E+03	3.40E+04	1.96E+03	0.934
NN10	32	Y	Y	249	1.22E+03	3.83E+03	5.28E+02	0.992
NN10	32	Y	Y	10	4.17E+03	1.74E+04	1.14E+03	0.969
NN10	32	Y	Y	50	2.21E+03	7.14E+03	9.74E+02	0.986
NN10	13	Y	N	97	1.72E+03	4.57E+03	6.58E+02	0.991
NN10	32	Y	N	249	1.20E+03	3.52E+03	4.20E+02	0.993
NN10	244	N	Y	50	5.26E+03	1.63E+04	1.57E+03	0.966
NN10	244	Y	Y	50	1.46E+03	4.91E+03	5.44E+02	0.991
NN10	244	Y	Y	150	9.97E+02	2.95E+03	4.57E+02	0.994

observed during this effort that the effectiveness of the *GENFIS3* tool appears to be less tolerant of high-dimension training data sets than the *NFTOOL*.

Tables 12.2 and 12.3 describe the performance of the models generated using these tools. Note that these performance numbers should be compared qualitatively since the different input parameter configurations can yield different numbers of training data points.

A suboptimal predictor was constructed in order to show its performance relative to that of the nonparametric models. This predictor was based on the average GHI, DHI, and DNI values for each time bin in the data set. Table 12.4 shows the improvement of the nonparametric models when compared to this suboptimal predictor, named "Time Bin Mean" in Table 12.4.

During this analysis, the aspect of the scalability of the *GENFIS3* and *NFTOOL* tools was evaluated. The model generation time for *NFTOOL* was always shorter

Table 12.4 Performance of best nonparametric to mean time bin suboptimal predictor

	Model Type		
Parameter	Time Bin Mean	Best FIS	Best NN10
MSE GHI	1.17E+04	2.96E+03	9.97E+02
% MSE GHI Improvement	0%	294%	1070%
MSE DNI	8.44E+04	2.08E+04	2.95E+03
% MSE DNI Improvement	0%	306%	2765%
MSE GHI	4.51E+03	1.05E+03	4.57E+02
% MSE GHI Improvement	0%	331%	886%
R	8.39E-01	9.67E-01	9.94E-01
% R Improvement	100%	115%	119%

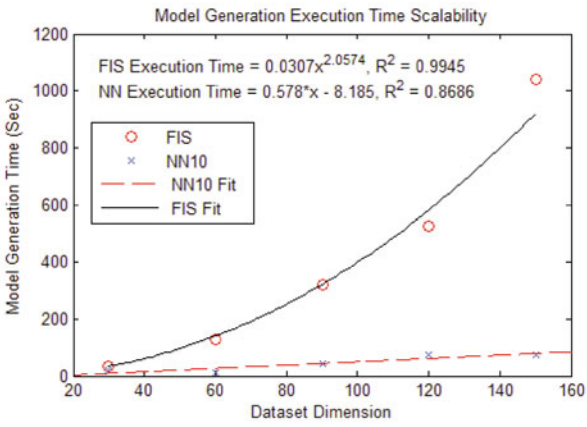


Fig. 12.25 Model generation execution time relationship with data set dimension

than *GENFIS3* for the same data sets. The relationship of *NFTOOL* execution time to data set length and dimension was generally linear for the test cases evaluated. The relationship of *GENFIS3* execution time to data set length was also linear; however, its relationship between data set dimension and execution time was exponential. This is shown in Figs. 12.25 and 12.26.

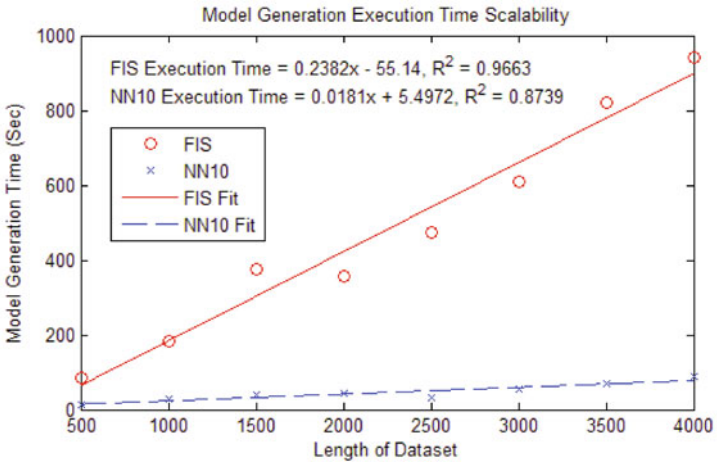


Fig. 12.26 Model generation execution time relationship with data set length

12.6 Conclusions

Applications of ANN in stock market forecast have been documented fro quite some time. In this section ANN is used to predict future behavior of stock market indices such as NASDAQ in the United States. This paper presents that ANN does indeed offer an opportunity for them to improve price in selecting stocks.

On the basis of the above applications, it is clear ,that the ANN is a very effective tool in soft computing for determining nonparametric models for Big Data. In order to create the prediction model, the implementation process should include different steps like data collection, data preprocessing, classification, and model evaluation. The experiment conducted in this paper uses a simple and efficient approach to stock prediction using backpropagation with a feed-forward network. The accuracy of the network recorded was 99 % in case of training with five layers, ten neurons in input layer, and one in output layer, and the best validation performance was 1378,0411.

This model can be very beneficial for individual and corporate investors, financial analysts, and users of financial news. They can foresee the future behavior and movement of stock prices, take corrective actions immediately, and act properly in their trading to gain more profit and prevent loss. In conclusion, we can say that if we train our system with more input data sets, it generates more error-free prediction price.

The second application was solar forecasting for a micro-grid system. That application presented a high-level look at some of the tools available in the MATLAB toolset that enable the user to extract information from “Big Data” sources in order to draw useful conclusions. As described in Sect. 12.4, the specific application discussed in this chapter is the prediction of the amount of solar power generated by a micro-grid. Section 12.4 then discusses the data that were gathered to support this exercise. Section 12.5 discussed the steps and techniques considered while trying to

generate the best solar irradiance prediction model. Techniques discussed included data-set sanitation, training input parameter selection, model generation via fuzzy C-means clustering and rule inference (*GENFIS3*), Neural Network training using backpropagation (*NFTOOL*), preprocessing nonlinear variables to add to the training data set, and the use of PCA to reduce the dimension of the training data while maximizing the information retained in the data set.

It was observed in the results presented in Sect. 5 that the best model predicting solar irradiance was one utilizing the maximum number of original and preprocessed variables, which was then reduced to a manageable dimension using PCA prior to use in training the model. The results in this section also showed that the nonparametric model generation methods discussed in this chapter performed significantly better than a suboptimal predictor. Finally, the results describing the model generation times for the two techniques showed that *NFTOOL* provides significantly better training times, especially when the dimension of the data set is high.

References

1. M. Jamshidi (ed.), *Systems of Systems Engineering—Principles and Applications* (CRC/Taylor & Francis, London, 2008) (also in Mandarin language, China Machine Press, ISBN 978-7-111-38955-2, Beijing, 2013)
2. M. Jamshidi (ed.), *System of Systems Engineering—Innovations for the 21st Century* (Wiley, New York, 2009)
3. Y.S. Manjili, A. Rajaee, M. Jamshidi, B. Kelley, *Fuzzy Control of Electricity Storage Unit for Energy Management of Micro-Grids1* (World Automation Congress (WAC), Puerto Vallarta, 2012)
4. M.T. Philip, K. Poul, S.O. Choy, K. Reggie, S.C. Ng, J. Mark, T. Jonathan, K. Kai, W. Tak-Lam, Design and implementation of nn5 for hong stock price forecasting. J. Eng. Appl. Artif. Intel. **20**, 453–461 (2007)
5. D. Enke, N. Mehdiyev, Stock market prediction using a combination of stepwise regression analysis, differential evolution-based fuzzy clustering, and a fuzzy inference neural network. Intel. Autom. Soft Co. **19**(4), 636–648 (2013)
6. X. Wu, M. Fund, A. Flitman, *Forecasting Stock Performance using Intelligent Hybrid Systems* (Springer, Heidelberg, 2001), pp. 447–456
7. T. Kimoto, K. Asawaka, M. Yoda, M. Takeoka, *Stock Market Prediction System with Modular Neural Network*. Processing of international joint conference on neural networks, 1990, pp. 1–6
8. H. Mizono, M. Kosaka, H. Yajma, N. Komoda, Application of neural network to technical analysis of stock market prediction. Stud. Inform. Control **7**(3), 111–120 (1998)
9. S.R. Sexton, R.E. Dorsey J.D. Johnson, Toward global optimization of neural network: A comparison of the genetic algorithm and back propagation. Decis. Support Syst. **22**, 117–185 (1998)
10. P.K. Phua, K.H. Ming, W. Lin, *Neural Network with Genetic Algorithms for Stocks Prediction*. Fifth conferences of the association of Asian-Pacific operations research societies, Singapore, 5–7 July 2000
11. L. I. Smith, A tutorial on principal components analysis, (2002), http://www.cs.otago.ac.nz/cosc453/studenttutorials/principalcomponents.pdfor–http://scholar.google.com/scholar?hl=en&q=PCA+Tutorial+Smith&btnG=&assdt=1%2C44&assdtp=. Accessed 26 Feb 2002
12. D.E. Rumelhart, G.E. Hinton, R.J. Wiliams, in *Parallel Distributed Processing Explorations in the Microstructures of Cognition*, ed. by J.L. McClelland. Learning internal representation by error propagation, vol 1 (MIT Press, Cambridge, 1986) pp. 318–362

13. A. Zilouchian, M. Jamshidi (eds.), *Intelligent Control Systems with Soft Computing Methodologies,* (CRC Publishers, Boca Raton, 2001) (Chap. 2)
14. R. Ball, P. Tissot, Demonstration of Artificial Neural Network in Matlab, Division of Nearhsore Research, Texas A&M University, 2006.http://www.learningace.com/doc/625993/22d317fb06b0c2eded47a44f2b4203cc/nnet-demo
15. S. Neenwi, P.O. Asagba, L.G. Kabari, Predicting the Nigerian stock market using artificial neural network. Eur. J. Comp. Sci. Inf. **1**(1) 30–39 (June 2013)
16. Yahoo Historical Prices of NASDAQ, (2013), http://finance.yahoo.com/q/hp?s=%5EIXIC&a=00&b=2&c=2013&d=03&e=15&f=2013&g=d. Accessed 5 May 2013
17. Mathworks, (2013), http://www.mathworks.com/help/nnet/ug/analyze-neural-network-performance-after-training.html. Accessed Nov 2013
18. C. Sergiu, Financial predictor via neural networ, (2011), http://www.codeproject.com. Accessed 25 May 2011
19. M. Jamshidi PI, Texas Sustainable Energy Research Institute, Proposal to National Science Foundation on PV Forecasting and Energy Management, San Antonio, Feb 2013
20. National Renewable Energy Laboratory, Current irradiance and meteorological conditions data set, (2012), http://www.nrel.gov/midc/srrl_bms/. Accessed 11 Feb 2013
21. National Renewable Energy Laboratory, SOLPOS Data Set, (2012), http://www.nrel.gov/midc/solpos/solpos.html. Accessed 10 Nov 2012
22. Iowa Environmental Mesonet, Automated surface observing system data set, (2012), http://mesonet.agron.iastate.edu/ASOS/. Accessed 21 Sept 2013

Index

A. Sadeghian, H. Tahayori (eds.), *Frontiers of Higher Order Fuzzy Sets,*
DOI 10.1007/978-1-4614-3442-9