

Hai-Jun Rong
Zhao-Xu Yang

Sequential Intelligent Dynamic System Modeling and Control

Sequential Intelligent Dynamic System Modeling and Control

Hai-Jun Rong · Zhao-Xu Yang

Sequential Intelligent Dynamic System Modeling and Control

 Springer

Hai-Jun Rong
Xi'an Jiaotong University
Xi'an, Shaanxi, China

Zhao-Xu Yang
Xi'an Jiaotong University
Xi'an, Shaanxi, China

ISBN 978-981-97-1540-4 ISBN 978-981-97-1541-1 (eBook)
<https://doi.org/10.1007/978-981-97-1541-1>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Paper in this product is recyclable.

Preface

This book consists of research outcomes developed by the authors and their co-authors. Its contents mainly are focused on sequential intelligent dynamic system modeling and control. In reality, many complex nonlinear systems, such as airplane, missile, magnetic bearing systems, mobile robot, and so on, appear dynamic characteristics, which may be caused by their own dynamical properties or the dynamic environments where they operate. Moreover, these dynamic characteristics are uncertain and varied at any time, which leads to the impossibility of modeling the uncertain dynamic systems with exact mathematical models. This further results in the difficulty of the controller design for these dynamic systems under the lack of exact mathematical models. If one hopes to design the satisfied controller for these uncertain dynamic nonlinear systems, the sequential model-free intelligent modeling approaches are expected with their online learning capabilities to perform satisfactorily even if the controlling objects or their environments change real time.

The book offers the novel research results of sequential intelligent dynamic system modeling and control in a unified framework from theory proposals to real applications. It covers an in-depth study on various learning algorithms for permanent adaptation of intelligent model parameters as well as of structural parts of the model. The comprehensive researches on sequential fuzzy and neural controller design schemes for some complex real applications are included, which is particularly suited for readers who are interested to learn practical solutions for controlling the nonlinear systems that are uncertain and varied at any time. Specifically, these contents are presented in four main parts, each of which is comprised of some chapters around a similar subject.

- The first part involves chapters that mainly describe the basic theories about fuzzy inference systems, neural networks, optimization methods, modeling, and controlling of nonlinear dynamic systems. These are the basis of the subsequent three parts about new intelligent models and intelligent controllers.
- The second part consists of chapters where some novel sequential fuzzy system modeling approaches are presented. These approaches aim to model the optimal fuzzy system when the data are sequentially arrived. Apart from the consequent

parameter optimization faced by the fuzzy inference system, its structure identification that is the determination of fuzzy rules is considered according to the coming data streams. These novel intelligent fuzzy systems provide the basis of designing the sequential fuzzy controllers in the subsequent part.

- The third part covers chapters with the theme about the sequential fuzzy controller design for some complex applications. The utilization of the intelligent fuzzy systems described in the second part to design the sequential intelligent controllers will be presented. These fuzzy controllers are capable of learning and fully adapting their structure and parameters simultaneously.
- The fourth part includes chapters with the theme about the sequential neural controller design for some complex applications. The utilization of the feedforward neural networks to develop controller design will be presented. Among these neural controllers, the controller parameters are updated based on the extreme learning machine algorithm where the parameters of hidden nodes of the feedforward neural networks are randomly assigned or determined according to the kernel method without optimizing. This simplifies the controller design process.

The organization of the book from addressing fundamental concepts and presenting the novel intelligent models to solving real applications is one of the major features of the book, which makes it a valuable resource for both beginners and researchers wanting to further their understanding and studying about real time online intelligent modeling and control of nonlinear dynamic systems. The book can also benefit researchers, engineers, and graduate students in the fields of control engineering, artificial intelligence, computational intelligence, intelligent control, nonlinear system modeling and control, etc.

Xi'an, China
December 2023

Hai-Jun Rong
Zhao-Xu Yang

Acknowledgements The authors would like to acknowledge everyone on the academic work who helped them so much. They are also immensely grateful for the unwavering support of their families and friends.

Contents

Part I Basic Theories

1	Fuzzy Inference Systems	3
1.1	Introduction	3
1.2	Fuzzy Sets	4
1.3	Fuzzy Rules	6
	1.3.1 Mamdani Fuzzy System	8
	1.3.2 TS Fuzzy System	8
1.4	Fuzzifier	8
1.5	Fuzzy Inference Engine	9
1.6	Defuzzifier	10
1.7	Structure of Fuzzy Inference Systems	11
1.8	Anya Fuzzy System	13
1.9	Summary	17
	References	17
2	Neural Networks	19
2.1	Introduction	19
2.2	FNNs with Sigmoid Activation Functions	20
2.3	Radial Basis Function Networks	21
2.4	Functional Equivalence Between Fuzzy Inference Systems and Radial Basis Function Networks	23
2.5	FNN Equivalence of Fuzzy Inference Systems	24
2.6	Summary	27
	References	27
3	Optimization Algorithms	29
3.1	Introduction	29
3.2	Gradient-Based Algorithms	30
	3.2.1 Steepest Descent Method	31
	3.2.2 Newton Method	33

- 3.3 Least Squares Algorithm 34
- 3.4 Recursive Least Squares Algorithm 36
- 3.5 Extended Kalman Filter 37
- 3.6 Extreme Learning Machine 40
- 3.7 Summary 42
- References 43
- 4 Modeling and Control of Nonlinear Dynamic Systems 45**
 - 4.1 Introduction 45
 - 4.2 Lyapounv Stability Theory 46
 - 4.3 Modeling of Nonlinear Dynamic Systems 48
 - 4.4 Adaptive Control 50
 - 4.4.1 Indirect Adaptive Control 50
 - 4.4.2 Direct Adaptive Control 51
 - 4.4.3 Model Reference Adaptive Control 52
 - 4.4.4 Feedback Linearization 53
 - 4.4.5 Backstepping Control 56
 - 4.5 Summary 60
 - References 61
- Part II Sequential Fuzzy System Modeling**
- 5 Online Sequential Fuzzy Extreme Learning Machine 65**
 - 5.1 Introduction 65
 - 5.2 Fuzzy-ELM 66
 - 5.3 Online Sequential Fuzzy Extreme Learning Machine 69
 - 5.4 Performance Evaluation of OS-Fuzzy-ELM 74
 - 5.4.1 Nonlinear System Identification 75
 - 5.4.2 Regression Problems 76
 - 5.4.3 Classification Problems 77
 - 5.4.4 Performance Evaluation of OS-Fuzzy-ELM:
Chunk by Chunk Mode 79
 - 5.4.5 Performance Evaluation of OS-Fuzzy-ELM: TS
Versus Mamdani Model 80
 - 5.5 Summary 80
 - References 81
- 6 Sequential Adaptive Fuzzy Inference System 83**
 - 6.1 Introduction 83
 - 6.2 SAFIS Algorithm 84
 - 6.2.1 “Influence” of a Fuzzy Rule 84
 - 6.3 Learning Procedure of SAFIS 87
 - 6.3.1 Adding a Fuzzy Rule 87
 - 6.3.2 Allocation of Antecedent and Consequent
Parameters 88

- 6.3.3 Parameter Adjustment 88
- 6.3.4 Removing a Fuzzy Rule 89
- 6.4 ESAFIS Algorithm 91
 - 6.4.1 Modified Influence of a Fuzzy Rule 91
 - 6.4.2 Adding of a Fuzzy Rule 93
 - 6.4.3 Parameter Adjustment 94
 - 6.4.4 Removing of Fuzzy Rules 95
- 6.5 Performance Evaluation of SAFIS and ESAFIS 97
 - 6.5.1 Study on Memory Factor M 98
 - 6.5.2 Performance Comparisons with Other Learning Approaches 99
- 6.6 Summary 101
- References 102
- 7 Evolving Fuzzy Systems Based on Data Clouds 105**
 - 7.1 Introduction 105
 - 7.2 Learning of EFS Based on Data Clouds 106
 - 7.2.1 Formation of Data Clouds 106
 - 7.2.2 Parameter Learning 109
 - 7.3 Numerical Examples 111
 - 7.3.1 Box-Jenkins Gas Furnace Problem 111
 - 7.3.2 Nonlinear System Identification 112
 - 7.3.3 Mackey-Glass Time Series Prediction 113
 - 7.3.4 Regression Problems 115
 - 7.4 Summary 115
 - References 117
- 8 Stability of a Class of Evolving Fuzzy Systems 119**
 - 8.1 Introduction 119
 - 8.2 Stability and Convergence Analysis 120
 - 8.3 Numerical Examples 125
 - 8.3.1 Box-Jenkins Gas Furnace Problem 125
 - 8.3.2 Nonlinear System Identification 125
 - 8.3.3 Mackey-Glass Time Series Prediction 126
 - 8.4 Summary 129
 - References 129

Part III Sequential Fuzzy Controller Design

- 9 Adaptive Self-Learning Fuzzy Autopilot Design for Uncertain Bank-To-Turn Missiles 133**
 - 9.1 Introduction 133
 - 9.2 Missile Dynamics 136
 - 9.3 Adaptive Self-Learning Fuzzy Autopilot Design 140
 - 9.3.1 Autopilot Design Procedure 141
 - 9.3.2 Stability Analysis 144

9.4	Simulation Results	147
9.5	Summary	152
	References	154
10	Self-evolving Fuzzy Model-Based Controller for Hypersonic Vehicle	157
10.1	Introduction	157
10.2	Hypersonic Vehicle Model	159
10.3	SAFIS Structure for HV Model	166
10.4	Design Procedure of Self-evolving Fuzzy Model-Based Controller	168
10.4.1	System Error Dynamics	168
10.4.2	Controller Stability Analysis	173
10.5	Performance Verification	176
10.6	Summary	179
	References	180
11	Self-Evolving Data Cloud-Based PID-Like Controller for Nonlinear Uncertain Systems	183
11.1	Introduction	183
11.2	Problem Formulation	185
11.3	Self-Evolving Data Cloud-Based PID-Like Controller	187
11.3.1	Controller Structure	187
11.3.2	Self-Evolving Learning Process	190
11.4	Stability and Convergence Analysis	192
11.5	Simulation and Experimental Results	195
11.5.1	Example 1	195
11.5.2	Example 2	197
11.6	Summary	199
	References	200
12	Adaptive Nonparametric Evolving Fuzzy Controller for Nonlinear Uncertain Systems with Dead Zone	203
12.1	Introduction	203
12.2	System Statement and Preliminaries	204
12.2.1	System Description	204
12.3	Adaptive Nonparametric Evolving Fuzzy Controller	205
12.3.1	Controller Design Procedure	205
12.3.2	Parameters Learning	209
12.4	Stability and Convergence Analysis	210
12.5	Simulation Results	213
12.6	Summary	213
	References	214

Part IV Sequential Neural Controller Design

- 13 Adaptive Backstepping Neural Controller for Magnetic Bearing System** 217
 - 13.1 Introduction 217
 - 13.2 Dynamics of Active Magnetic Bearing System 219
 - 13.3 Design Process of Adaptive Backstepping Neural Controller 221
 - 13.4 Summary 227
 - References 227

- 14 Simplified Adaptive Backstepping Neural Controller for Magnetic Bearing System** 229
 - 14.1 Introduction 229
 - 14.2 Design Procedure of Simplified Adaptive Backstepping Neural Controller 230
 - 14.3 Performance Evaluation 234
 - 14.3.1 Performance Comparison Between ABNC and Simpl_ABNC 236
 - 14.3.2 Performance Comparison with Other Control Approaches 239
 - 14.4 Summary 239
 - References 240

- 15 Robust Kernel-Based Model Reference Adaptive Control for Unstable Aircraft** 241
 - 15.1 Introduction 241
 - 15.2 Aircraft Model 243
 - 15.3 Kernel-Based Model Reference Adaptive Control 244
 - 15.3.1 Offline Identification 244
 - 15.3.2 Online Learning Control 249
 - 15.4 Simulation Results 253
 - 15.4.1 Simulation Results of Offline Identification 253
 - 15.4.2 Simulation Results of Online Control 256
 - 15.5 Summary 258
 - References 259

Part I
Basic Theories

Chapter 1

Fuzzy Inference Systems



Abstract This chapter mainly describes the fundamental concepts of the fuzzy inference systems. It starts with fuzzy sets, fuzzifier, fuzzy inference engine and defuzzifier. Then, two commonly used fuzzy systems, viz., Mamdani and Takagi-Sugeno fuzzy systems are introduced. Based on these, the general structure representing the two kinds of fuzzy inference systems with any fuzzy membership functions and fuzzy inference engine is presented in the chapter. Finally, different from Mamdani and Takagi-Sugeno fuzzy systems, a novel fuzzy system called Anya fuzzy system with the data clouds representing the fuzzy antecedence is described in the chapter. These constitute the basis of subsequent researches of the book.

1.1 Introduction

Fuzzy inference systems are developed based on the fuzzy logic and fuzzy set theory introduced by Zadeh in 1965 [1] and have been widely used in many disciplines such as engineering, economics and other areas [2–16]. A fuzzy inference system using fuzzy if-then rules can model the qualitative aspects of human knowledge and reasoning processes for dealing with ill-defined and uncertain systems without employing precise quantitative analysis. Takagi and Sugeno [2] first explored the fuzzy identification systematically by using a fuzzy model which described fuzzy rules by local linear input-output functions. This kind of fuzzy model has been employed in numerous practical applications like control [3, 4], prediction and inference [5]. An adaptive fuzzy controller was proposed by Mamdani [6, 7] by using the proportional and derivative error signals and the control actions were produced based on the plant performance. The proposed controller was successfully applied in controlling a steam engine of a model industrial plant. Fuzzy systems can be utilized as fuzzy controllers for autonomous mobile robots which have complex control architectures. Saffiotti et al. [8] presented a fuzzy controller for an autonomous mobile robot to pursue strategic goals such as a reactive behavior to avoid obstacles on the way and a goal-oriented behavior to reach a given location. In addition, the fuzzy systems have been employed as the fuzzy controllers to solve the aircraft fault-tolerant problem during landing phase for achieving the safe landing under disturbances [9–11]. Many

applications in the pattern classification and prediction field have been explored successfully by the researchers [12–16] using fuzzy systems. Gopal et al. [12] utilized fuzzy logic for classification of Partial Discharge (PD) patterns for the diagnosis of High Voltage insulation system. Wei and Mendel [13] employed fuzzy inference systems to construct a classifier for non-ideal environments where precise probabilistic methods are difficult or impossible to use. In [14] fuzzy inference systems were employed to predict the Gross Domestic Product (GDP) development by designing a prediction model. Konjic et al. [15] utilized fuzzy inference systems to predict load curves at low voltage substations used by different types of consumers such as residents, industry and so on. In [16] fuzzy systems were employed to predict the global solar radiation data. A higher accuracy was achieved by these compared with the conventional methods.

From the above overview, it can be seen that the fuzzy inference systems are very useful to solve many practical problems which involve a high level of uncertainty, complexity, or nonlinearity and are difficult to solve by using conventional modelling methods. In general, a fuzzy inference system consists of four principal components, viz., a fuzzifier, a fuzzy rule base, a fuzzy inference engine, and a defuzzifier. Next, we will give a brief description for different types of fuzzy inference systems, their components and their structures [17] which will help the algorithms described in later chapters.

1.2 Fuzzy Sets

Fuzzy set theory is an extension of the classical set theory assessed in binary terms, which is an element either belongs or does not belong to the set. In the fuzzy set, the membership of the elements in relation to the set is gradually assessed with the aid of a membership function $\mu \rightarrow [0, 1]$. In general, any bounded nonconstant continuous function can be chosen as a candidate for the membership function [18, 19]. The following list gives the types of membership functions which are most commonly used [20].

(1) Triangular Membership Function

The triangular membership function (Trimf) includes two parameters $\{c, a\}$ and is given by

$$\text{Trimf}(x; c, a) = \begin{cases} 0 & x \leq c - a \\ \frac{x-c+a}{a} & c - a < x \leq c \\ \frac{c+a-x}{a} & c < x \leq c + a \\ 0 & c + a < x \end{cases} \quad (1.1)$$

(2) Trapezoid Membership Function

Trapezoid membership function (Trapmf) includes four parameters $\{c_1, a_1, c_2, a_2\}$ and is given as

$$\text{Trapmf}(x; c_1, a_1, c_2, a_2) = \begin{cases} 0 & x \leq c_1 - a_1 \\ \frac{x - c_1 + a_1}{a_1} & c_1 - a_1 < x \leq c_1 \\ 1 & c_1 < x \leq c_2 \\ \frac{c_2 + a_2 - x}{a_2} & c_2 < x \leq c_2 + a_2 \\ 0 & c_2 + a_2 < x \end{cases} \quad (1.2)$$

(3) Gaussian Membership Function

Gaussian membership function (Gaussmf) includes two parameters $\{c, a\}$ and is given by

$$\text{Gaussmf}(x; c, a) = \exp\left(-\left(\frac{x - c}{a}\right)^2\right) \quad (1.3)$$

(4) Two-sided Gaussian Membership Function

Two-sided Gaussian membership function (Gauss2mf) includes four parameters $\{c_1, a_1, c_2, a_2\}$ and is given by

$$\text{Gauss2mf}(x; c_1, a_1, c_2, a_2) = \begin{cases} \exp\left(-\left(\frac{x - c_1}{a_1}\right)^2\right) & x \leq c_1 \\ 1 & c_1 < x < c_2 \\ \exp\left(-\left(\frac{x - c_2}{a_2}\right)^2\right) & x \geq c_2 \end{cases} \quad (1.4)$$

(5) Cauchy Membership Function

The Cauchy membership function (Cauchymf) includes two parameters $\{c, a\}$ and is given by

$$\text{Cauchymf}(x; c, a) = \frac{1}{1 + \left(\frac{x - c}{a}\right)^2} \quad (1.5)$$

(6) π -shaped Membership Function

π -shaped membership function is the product of S membership function and Z membership function. π -shaped membership function (Pimf) includes two parameters $\{c, a\}$ and is given by

$$\text{Pimf}(x; c, a) = \begin{cases} S(x; c - a, c) & x \leq c \\ Z(x; c, c + a) & x > c \end{cases} \quad (1.6)$$

where $S(x; c - a, c)$ is the S membership function and given by

$$S(x; c - a, c) = \begin{cases} 0 & x \leq c - a \\ 2 \left(\frac{x - c + a}{a}\right)^2 & c - a < x \leq \frac{2c - a}{2} \\ 1 - 2 \left(\frac{c - x}{a}\right)^2 & \frac{2c - a}{2} < x \leq c \\ 1 & c < x \end{cases} \quad (1.7)$$

$Z(x; c, c + a)$ is the Z membership function and given by

$$Z(x; c, c + a) = \begin{cases} 1 & x \leq c \\ 1 - 2\left(\frac{x-c}{a}\right)^2 & c < x \leq \frac{2c+a}{2} \\ 2\left(\frac{c+a-x}{a}\right)^2 & \frac{2c+a}{2} < x \leq c + a \\ 0 & c + a < x \end{cases} \quad (1.8)$$

(7) Difference between two Sigmoidally-shaped Membership Functions

The sigmoid function includes two parameters $\{c, a\}$ and is given by

$$f(x; c, a) = \frac{1}{1 + \exp(-(cx + a))} \quad (1.9)$$

The membership function of the difference between two sigmoid functions (Dsigmf) includes four parameters $\{c_1, a_1, c_2, a_2\}$ and is given by

$$\text{Dsigmf}(x; c_1, a_1, c_2, a_2) = f(x; c_1, a_1) - f(x; c_2, a_2) \quad (1.10)$$

(8) Product of two Sigmoidally-shaped Membership Functions

The membership function of the product of two sigmoid functions (Psigmf) includes four parameters $\{c_1, a_1, c_2, a_2\}$ and is given as

$$\text{Psigmf}(x; c_1, a_1, c_2, a_2) = f(x; c_1, a_1) * f(x; c_2, a_2) \quad (1.11)$$

To visualize each membership function described above, the graphs from these membership functions are illustrated in Fig. 1.1.

1.3 Fuzzy Rules

The fuzzy rule base comprises of a series of fuzzy rules in the format of “if-then” form that are consistent with the human languages. The fuzzy rules are generally classified into two types. One type is that the antecedent (if) part and the consequent (then) part are both described by the fuzzy sets. The second type is that only the antecedent part is described by fuzzy sets whereas the consequent part is described by real values. The most common Mamdani fuzzy model uses the first type of fuzzy rules while the Takagi-Sugeno (TS) fuzzy model utilizes the second type of fuzzy rules. Based on the two kinds of fuzzy models, the fuzzy inference systems can be classified into two types, viz., Mamdani fuzzy inference systems and TS fuzzy inference systems.

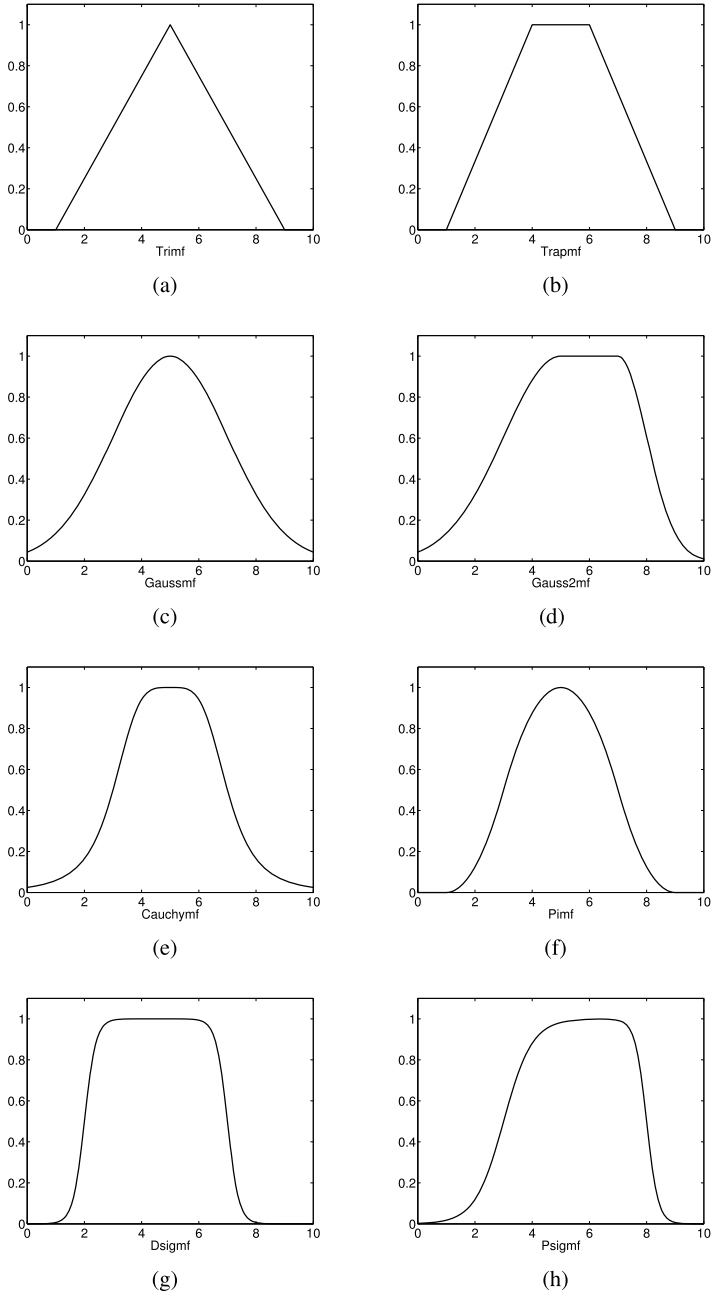


Fig. 1.1 Graph examples of eight membership functions: **a** Trimf, **b** Trapmf, **c** Gaussmf, **d** Gauss2mf, **e** Cauchymf, **f** Pimf, **g** Dsigmf, **h** Psigmf

1.3.1 Mamdani Fuzzy System

The Mamdani fuzzy system uses the following rules [6],

$$\begin{aligned} \text{Rule } i & : \text{if } (x_1 \text{ is } A_{1i}) \text{ AND } (x_2 \text{ is } A_{2i}) \cdots \text{ AND } (x_n \text{ is } A_{ni}), \\ & \text{then } (\hat{y}_1 \text{ is } B_{1i}) \cdots (\hat{y}_m \text{ is } B_{mi}) \end{aligned}$$

where A_{ji} ($j = 1, 2, \dots, n, i = 1, 2, \dots, \tilde{N}$) and B_{li} ($l = 1, 2, \dots, m, i = 1, 2, \dots, \tilde{N}$) are the fuzzy sets of the j th input variable x_j and the l th output variable y_l in rule i , n is the dimension of the input vector \mathbf{x} ($\mathbf{x} = [x_1, \dots, x_n]^T$), m is the dimension of the output vector $\hat{\mathbf{y}}$ ($\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_m]^T$), and \tilde{N} is the number of fuzzy rules.

1.3.2 TS Fuzzy System

The TS fuzzy system is based on the following rules [2],

$$\begin{aligned} \text{Rule } i & : \text{if } (x_1 \text{ is } A_{1i}) \text{ AND } (x_2 \text{ is } A_{2i}) \cdots \text{ AND } (x_n \text{ is } A_{ni}), \\ & \text{then } (\hat{y}_1 \text{ is } \beta_{1i}) \cdots (\hat{y}_m \text{ is } \beta_{mi}) \end{aligned}$$

where β_{li} ($l = 1, 2, \dots, m, i = 1, 2, \dots, \tilde{N}$) is the crisp value and it may be any function of the input variables or a constant value. When β_{li} is the constant, it corresponds to the zero-order TS model. In case of a linear function $\beta_{li} = q_{li0} + q_{li1}x_1 + \cdots + q_{lin}x_n$, it is commonly called as the first-order TS model.

The following gives the details about the learning procedure of fuzzy rules.

1.4 Fuzzifier

The fuzzifier aims to perform a mapping from a crisp input \mathbf{x}' into a fuzzy set A' . One of the most commonly used fuzzifier methods is the singleton fuzzifier, which is $\mu_{A'}(\mathbf{x}') = 1$ for $\mathbf{x}' = \mathbf{x}$ and $\mu_{A'}(\mathbf{x}') = 0$ for $\mathbf{x}' \neq \mathbf{x}$. All the studies in this book are based on the singleton fuzzifier.

1.5 Fuzzy Inference Engine

In fuzzy rule i , a fuzzy implication is employed to define a fuzzy set as given below,

$$\Psi_i : A_{1i} \otimes A_{2i} \otimes \cdots \otimes A_{ni} \longrightarrow B_{1i} + \cdots + B_{mi} \quad (1.12)$$

where ‘ \otimes ’ is the T-norm operator, ‘+’ represents the union of the independent variables.

T-norm includes many types, such as minimum, algebraic product and so on. When the degree to which the given j th input variable x_j and the l th output variable y_l satisfy the quantifier A_{ji} and B_{li} in rule i are specified by their membership function $\mu_{A_{ji}}(x_j)$ and $\mu_{B_{li}}(y_l)$, the minimum T-norm operation is given by

$$\begin{aligned} R_i &= \mu_{A_{1i}}(x_1) \otimes \mu_{A_{2i}}(x_2) \otimes \cdots \otimes \mu_{A_{ni}}(x_n) \\ &= \min \{ \mu_{A_{1i}}(x_1), \mu_{A_{2i}}(x_2), \dots, \mu_{A_{ni}}(x_n) \} \end{aligned} \quad (1.13)$$

For the algebraic product it is given as

$$R_i = \mu_{A_{1i}}(x_1) * \mu_{A_{2i}}(x_2) * \cdots * \mu_{A_{ni}}(x_n) = \prod_{j=1}^n \mu_{A_{ji}}(x_j) \quad (1.14)$$

Equation (1.12) with any membership function as described above is expressed as

$$\begin{aligned} \mu_{\Psi_i}(\mathbf{x}, \hat{\mathbf{y}}) &= \mu_{A_{1i}}(x_1) \otimes \mu_{A_{2i}}(x_2) \otimes \cdots \otimes \mu_{A_{ni}}(x_n) \\ &\quad \otimes (\mu_{B_{1i}}(\hat{y}_1) + \mu_{B_{2i}}(\hat{y}_2) \cdots + \mu_{B_{mi}}(\hat{y}_m)) \end{aligned} \quad (1.15)$$

The fuzzy inference engine aims to determine a mapping from the fuzzy sets in the input space to the fuzzy sets in the output space based on the sup-star composition. Letting A' be an arbitrary fuzzy set in the input space, then the fuzzy set B in the output space is given by

$$\begin{aligned} \mu_B(\hat{\mathbf{y}}) &= \mu_{A' \circ \Psi_i}(\hat{\mathbf{y}}) \\ &= \sup_{\mathbf{x}'} [\mu_{A'}(\mathbf{x}') \otimes \mu_{\Psi_i}(\mathbf{x}', \hat{\mathbf{y}})] \\ &= \sup_{\mathbf{x}'} [\mu_{A'}(\mathbf{x}') \otimes \mu_{A_{1i}}(x'_1) \otimes \mu_{A_{2i}}(x'_2) \otimes \mu_{A_{ni}}(x'_n) \\ &\quad \otimes (\mu_{B_{1i}}(\hat{y}_1) + \mu_{B_{2i}}(\hat{y}_2) \cdots + \mu_{B_{mi}}(\hat{y}_m))] \end{aligned} \quad (1.16)$$

where \circ denotes the sup-star composition where star represents the T-norm operation.

1.6 Defuzzifier

The defuzzifier performs a mapping from the fuzzy sets in the output space to crisp points in the output space. Many schemes including center average, mean of maximum, maximum criterion, etc. [7, 20] have been proposed to realize the defuzzifier. The center average defuzzifier is the most widely adopted defuzzification strategy, which is given as

$$\hat{y} = \frac{\sum_{i=1}^{\tilde{N}} \bar{y}_i \mu_B(\bar{y}_i)}{\sum_{i=1}^{\tilde{N}} \mu_B(\bar{y}_i)} \quad (1.17)$$

where $\bar{y}_i = [\bar{y}_{1i}, \dots, \bar{y}_{mi}]$ and \bar{y}_{li} is the point at which B_{li} achieves its maximum value, which is $\mu_B(\bar{y}_i) = 1$.

For the TS fuzzy model, its consequence is the crisp values and thus the defuzzifier operation is ignored.

For Mamdani fuzzy systems, by using a center average defuzzifier and singleton fuzzifier, the system output \hat{y} for given input \mathbf{x} is given by [21]

$$\hat{y} = \frac{\sum_{i=1}^{\tilde{N}} \beta_i \mu_{A' \circ \Psi_i}(\beta_i)}{\sum_{i=1}^{\tilde{N}} \mu_{A' \circ \Psi_i}(\beta_i)} \quad (1.18)$$

where

$$\begin{aligned} \mu_{A' \circ \Psi_i}(\beta_i) = \sup_{\mathbf{x}'} & \left[\mu_{A'}(\mathbf{x}') \otimes \mu_{A_{1i}}(x'_1) \otimes \mu_{A_{2i}}(x'_2) \otimes \mu_{A_{ni}}(x'_n) \right. \\ & \left. \otimes (\mu_{B_{1i}}(\beta_{1i}) + \mu_{B_{2i}}(\beta_{2i}) \cdots + \mu_{B_{mi}}(\beta_{mi})) \right] \end{aligned} \quad (1.19)$$

Due to the singleton fuzzifier, $\mu_{A'}(\mathbf{x}') = 1$ for $\mathbf{x}' = \mathbf{x}$ and because of center average defuzzifier, $\mu_{B_{1i}}(\beta_{1i}) = \mu_{B_{2i}}(\beta_{2i}) = \cdots = \mu_{B_{mi}}(\beta_{mi}) = 1$.

Thus, Eq. (1.18) becomes as

$$\hat{y} = \frac{\sum_{i=1}^{\tilde{N}} \beta_i R_i}{\sum_{i=1}^{\tilde{N}} R_i} \quad (1.20)$$

where $\beta_i = [\beta_{1i}, \dots, \beta_{mi}]$ and β_{li} is the point at which B_{li} achieves its maximum value, which is $\mu_B(\beta_{li}) = 1$.

For the TS fuzzy systems, since the consequent parts are the crisp values, the defuzzifier is removed. The system crisp output is achieved by the weighted average sum of each rule's output and given by

$$\hat{y} = \frac{\sum_{i=1}^{\tilde{N}} \beta_i R_i}{\sum_{i=1}^{\tilde{N}} R_i} \quad (1.21)$$

where R_i is the weight and computed based on Eq.(1.13). $\beta_i = [\beta_{1i}, \dots, \beta_{mi}]$ ($i = 1, 2, \dots, \tilde{N}$) is the crisp consequence of rule i and its elements may be a constant for the zero-order TS model or a linear function of the input variables for the first-order TS model. In case of linear function, the l th element β_{li} equals to $\beta_{li} = q_{li0} + q_{li1}x_1 + \dots + q_{lin}x_n$.

1.7 Structure of Fuzzy Inference Systems

For the fuzzy models described above, a general five-layer structure can be adopted to represent their learning process, as illustrated in Fig. 1.2.

Layer 1: In Layer 1, each node represents an input variable and directly transmits the input signal to Layer 2.

Layer 2: In this layer each node represents the membership value of each input variable. The membership value $\mu_{A_{ji}}(x_j)$ of the j th input variable x_j in the i th rule can be achieved by any bounded nonconstant piecewise continuous membership function g ,

$$\mu_{A_{ji}}(x_j; c_{ji}, a_i) = g(x_j; c_{ji}, a_i) \quad (1.22)$$

where c_{ji} and a_i are the parameters existing in the membership function g corresponding to the j th input variable x_j and the i th rule.

Layer 3: Each node in this layer represents the if part of if-then rules obtained by fuzzy logic AND operation, which can be any type of T-norm such as the product composition. The firing strength (if part) of the i th rule is given by

$$R_i(\mathbf{x}; \mathbf{c}_i, a_i) = \mu_{A_{1i}}(x_1; c_{1i}, a_i) \otimes \mu_{A_{2i}}(x_2; c_{2i}, a_i) \otimes \dots \otimes \mu_{A_{ni}}(x_n; c_{ni}, a_i) \quad (1.23)$$

where symbol \otimes represents any type of T-norm operation. If the triangular membership function with the algebraic product operation is employed, it will be simply as

$$R_i(\mathbf{x}; \mathbf{c}_i, a_i) = \prod_{j=1}^n \mu_{A_{ji}}(x_j; c_{ji}, a_i) = \prod_{j=1}^n \left(1 - \frac{|x_j - c_{ji}|}{a_i}\right) \quad (1.24)$$

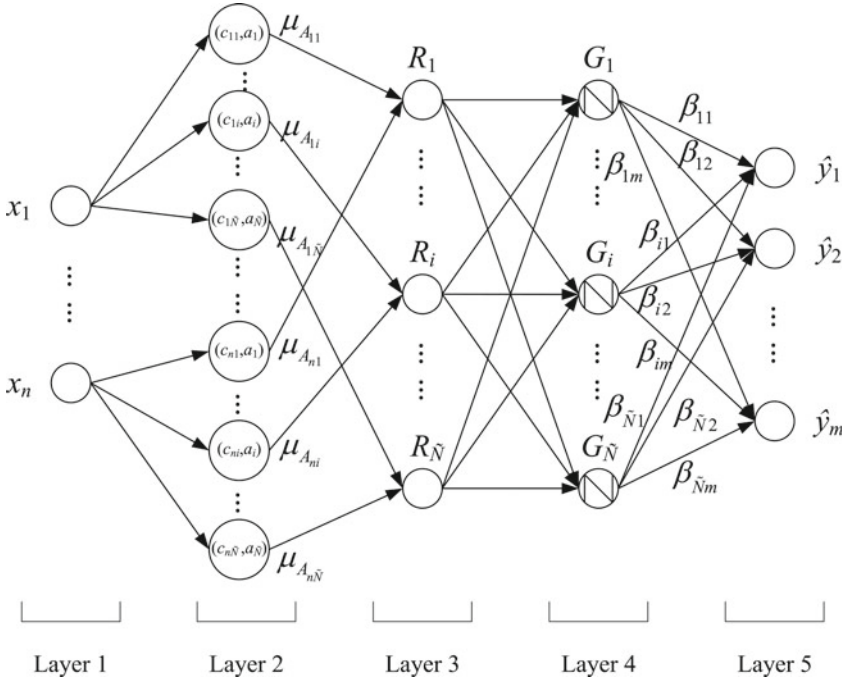


Fig. 1.2 General structure of fuzzy inference systems

Layer 4: The nodes in this layer are named as normalized nodes whose number is equal to the number of the nodes in the third layer. The i th normalized node is equal to the following equation,

$$G(\mathbf{x}; \mathbf{c}_i, a_i) = \frac{R_i(\mathbf{x}; \mathbf{c}_i, a_i)}{\sum_{i=1}^{\tilde{N}} R_i(\mathbf{x}; \mathbf{c}_i, a_i)} \quad (1.25)$$

G can be called Fuzzy Basis Function (FBF). Different from Zeng and Singh [22] where only product operation is used, any T-norm fuzzy logic operation, such as minimum operation, can be used in the fuzzy basis function G defined here.

Layer 5: Each node in this layer corresponds to an output variable.

For the Mamdani fuzzy model, by using center average defuzzifier [21] the system output $\hat{\mathbf{y}}$ for given input \mathbf{x} is calculated by

$$\hat{\mathbf{y}} = \frac{\sum_{i=1}^{\tilde{N}} \beta_i R_i(\mathbf{x}; \mathbf{c}_i, a_i)}{\sum_{i=1}^{\tilde{N}} R_i(\mathbf{x}; \mathbf{c}_i, a_i)} = \sum_{i=1}^{\tilde{N}} \beta_i G(\mathbf{x}; \mathbf{c}_i, a_i) \quad (1.26)$$

where $\beta_i = [\beta_{1i}, \dots, \beta_{mi}]$ and β_{li} ($l = 1, 2, \dots, m$) is the point at which B_{li} achieves its maximum value, which is $\mu_B(\beta_{li}) = 1$.

For the TS fuzzy model, its system output is achieved by the weighted sum of the output of each normalized rule. As such the system output \hat{y} for the given input \mathbf{x} is calculated by

$$\hat{y} = \frac{\sum_{i=1}^{\tilde{N}} \beta_i R_i(\mathbf{x}; \mathbf{c}_i, a_i)}{\sum_{i=1}^{\tilde{N}} R_i(\mathbf{x}; \mathbf{c}_i, a_i)} = \sum_{i=1}^{\tilde{N}} \beta_i G(\mathbf{x}; \mathbf{c}_i, a_i) \quad (1.27)$$

where the weight $\beta_i = [\beta_{1i}, \dots, \beta_{mi}]$ is the crisp consequence of each rule. In case of the zero-order TS model, β_i is the constant consequence. For the first-order TS model, it is a linear function about input variables. For β_{li} , it equals to $\beta_{li} = q_{li0} + q_{li1}x_1 + \dots + q_{lin}x_n$.

Remark 1 In Eqs. (1.26) and (1.27), the antecedent (if) part of fuzzy rules (if-then rules) for the two fuzzy models, $R_i(\cdot)$, is the same in the form and $R_i(\cdot) / \sum_{i=1}^{\tilde{N}} R_i(\cdot)$ represents the normalized firing strength of fuzzy rules while the consequent (then) part β_i ($= [\beta_{1i}, \dots, \beta_{mi}]$) is the same in the form but represents different meaning. For the Mamdani fuzzy model, the β_{li} ($l = 1, \dots, m$) contains the linguistic information since it is related with the linguistic variable B_{li} while the β_{li} in the TS fuzzy model is only the crisp value and has no linguistic information. Note that when the Mamdani fuzzy model applies the center average defuzzifier, the obtained model output (1.26) is functionally equivalent to the output of the zero-order TS model where the consequent parts are constant.

1.8 Anya Fuzzy System

Recently, a simplified type of fuzzy rule based (FRB) system called AnYa [23] is introduced which offers a new way of defining the “if” part of the rules without defining the membership functions per variable in an explicit manner. The antecedent parts of fuzzy rules are formed using so-called “data clouds” that are sets of data samples around focal points. The identified “data clouds” objectively represent the local peaks of the data density distribution and then are used as the antecedent (if) parts of fuzzy rules. In the Anya fuzzy system, “data clouds” are uniquely defined by the data samples associated with the nearest peak of the density which serves as a focal point. Therefore, in Anya fuzzy system we only need to determine the focal points. Then, data samples will be attracted to the nearest focal point and form a number of shape-free “data clouds” around the focal points automatically. In the Anya fuzzy system, the antecedent part of fuzzy rules is reduced to the vector form. The fuzzy rules are changed to

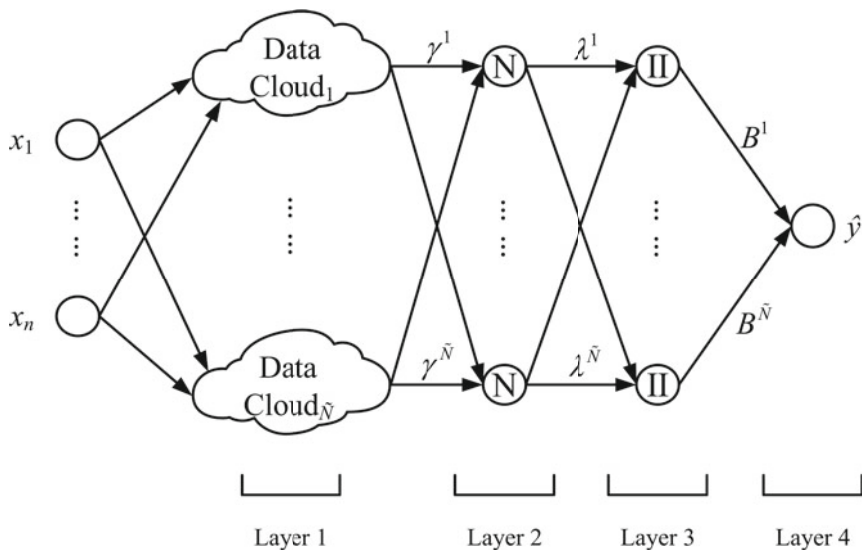


Fig. 1.3 Structure of Anya fuzzy system

$$\text{Rule } i : \text{ if } (\mathbf{x} \sim \text{prototype}_i) \text{ then } (\hat{y} \text{ is } B^i)$$

Here $\mathbf{x} = [x_1, x_2, \dots, x_n]$. prototype_i is the prototype of the i th data cloud and determined according to its local density γ^i that is the prototype of the i th data cloud in the input space. B^i ($i = 1, 2, \dots, \tilde{N}$) represents the crisp consequence of the i th rule that can be a constant or a linear combination of input variables. For a linear consequence, $B^i = q_0^i + q_1^i x_1 + \dots + q_n^i x_n$ is utilized. \tilde{N} represents the number of "data clouds". The structure of the Anya is depicted in Fig. 1.3. Different from those of the Mamdani and TS fuzzy systems, the structure of Anya fuzzy system consists of four layers. Layer 1 represents the local density of each "data cloud". The normalization value of local density for each "data cloud" is obtained in layer 2. Layer 3 is used to implement the weighted average defuzzification. The approximated output is represented in layer 4.

Unlike the Mamdani type and TS type fuzzy systems, the antecedent part of any fuzzy rule is represented by the prototype/focal points of the "data clouds" and derived from the data automatically based on the density of the data that is determined from the empirical data analysis (EDA). EDA is a nonparametric, assumptions-free, entirely data driven methodological framework recently introduced in [24] and empirical fuzzy sets (eFSs) [25]. It is entirely based on the empirical observations of the data samples and their ensemble properties. It is close to statistical learning in its nature but is free from the range of assumptions required by the traditional probability theory and statistical learning methods. Below we introduce some definitions from the empirical data analytics technique [25].

Definition 1 Cumulative proximity $\pi_k(\mathbf{x}_v)$ [25] is a measure indicating the degree of closeness/similarity of the data point \mathbf{x}_k at the current instant k to all other available data points in some data space \mathbf{x}_v , $v = 1, 2, \dots, L$. This is given as

$$\pi_k(\mathbf{x}_v) = \sum_{v=1}^L d_{kv} = \sum_{v=1}^L \|\mathbf{x}_k - \mathbf{x}_v\|^2 \quad (1.28)$$

Definition 2 Eccentricity $\varpi_k(\mathbf{x}_v)$ [25] is defined as the normalized $\pi_k(\mathbf{x}_v)$ of the data point \mathbf{x}_k at the current instant k as a fraction of $\pi_o(\mathbf{x}_v)$, $o = 1, 2, \dots, L$ of all other data samples, which is expressed as

$$\varpi_k(\mathbf{x}_v) = \frac{2\pi_k(\mathbf{x}_v)}{\frac{1}{L} \sum_{o=1}^L \pi_o(\mathbf{x}_v)} = \frac{2 \sum_{v=1}^L d_{kv}}{\frac{1}{L} \sum_{o=1}^L \sum_{v=1}^L d_{ov}} = \frac{2 \sum_{v=1}^L \|\mathbf{x}_k - \mathbf{x}_v\|^2}{\frac{1}{L} \sum_{o=1}^L \sum_{v=1}^L \|\mathbf{x}_o - \mathbf{x}_v\|^2} \quad (1.29)$$

The coefficient 2 is due to the fact that each distance is counted twice and can be seen as a normalization coefficient.

Definition 3 Data density [25] is a measure of similarity of a data point to all available data points and inversely proportional to the eccentricity. This is equal to

$$\gamma_k(\mathbf{x}_v) = \frac{1}{\varpi_k(\mathbf{x}_v)} = \frac{\frac{1}{L} \sum_{o=1}^L \sum_{v=1}^L d_{ov}}{2 \sum_{v=1}^L d_{kv}} \quad (1.30)$$

Density is a measure derived empirically from the observed data directly without any prior knowledge or assumptions about the data. $\pi_k(\mathbf{x}_v)$, $\varpi_k(\mathbf{x}_v)$ and $\gamma_k(\mathbf{x}_v)$ can be defined either locally for a part of the dataset or globally for all data points. It is well-known that a coupled system can be decomposed into a set of loosely connected local simpler systems aggregated in a fuzzy way. In the Anya fuzzy system, each local sub-system is represented by a “data cloud” that describes a certain sub-set of the entire data set. Thus, the approach replaces the scalar membership functions with a non-parametric function that is represented by the local data density of each “data cloud”. The local density of the i th “data cloud” is defined as follows [24, 25],

$$\gamma_k^i = \gamma_k^i(\mathbf{x}_v) = \frac{\frac{1}{M^i} \sum_{o=1}^{M^i} \sum_{v=1}^{M^i} d_{ov}}{2 \sum_{v=1}^{M^i} d_{kv}} \quad (1.31)$$

M^i denotes the number of samples in the i th “data cloud”.

This can be recursively updated as [24, 25]

$$\sum_{v=1}^{M^i} d_{kv} = \sum_{v=1}^{M^i} \|\mathbf{x}_k - \mathbf{x}_v\|^2 = M^i \left(\|\mathbf{x}_k - \Gamma_k^i\|^2 + \Xi_k^i - \|\Gamma_k^i\|^2 \right) \quad (1.32)$$