INTERNATIONAL EDITION

A COURSE IN FUZZY SYSTEMS AND CONTROL

Li-Xin Wang

WANG A COURSE IN FUZZY SYSTEMS AND CONTROL

A Course in Fuzzy Systems and Control

Li-Xin Wang

Prentice-Hall International, Inc.

Contents

Preface

The field of fuzzy systems and control has been making rapid progress in recent years. Motivated by the practical success of fuzzy control in consumer products and industrial process control, there has been an increasing amount of work on the rigorous theoretical studies of fuzzy systems and fuzzy control. Researchers are trying to explain why the practical results are good, systematize the existing approaches, and develop more powerful ones. As a result of these efforts, the whole picture of fuzzy systems and fuzzy control theory is becoming clearer. Although there are many books on fuzzy theory, most of them are either research monographs that concentrate on special topics, or collections of papers, or books on fuzzy mathematics. We desperately need a real textbook on fuzzy systems and control that provides the skeleton of the field and summarizes the fundamentals.

This book, which is based on a course developed at the Hong Kong University of Science and Technology, is intended as a textbook for graduate and senior students, and as a self-study book for practicing engineers. When writing this book, we required that it be:

- **Well-Structured:** This book is not intended as a collection of existing results on fuzzy systems and fuzzy control; rather, we first establish the structure that a reasonable theory of fuzzy systems and fuzzy control should follow, and then fill in the details. For example, when studying fuzzy control systems, we should consider the stability, optimality, and robustness of the systems, and classify the approaches according to whether the plant is linear, nonlinear, or modeled by fuzzy systems. Fortunately, the major existing results fit very well into this structure and therefore are covered in detail in this book. Because the field is not mature, as compared with other mainstream fields, there are holes in the structure for which no results exist. For these topics, we either provide our preliminary approaches, or point out that the problems are open.
- **Clear and Precise:** Clear and logical presentation is crucial for any book, especially for a book associated with the word "fuzzy." Fuzzy theory itself is precise; the "fuzziness" appears in the phenomena that fuzzy theory tries

to study. Once a fuzzy description (for example, "hot day") is formulated in terms of fuzzy theory, nothing will be fuzzy anymore. We pay special attention to the use of precise language to introduce the concepts, to develop the approaches, and to justify the conclusions.

- **Practical:** We recall that the driving force for fuzzy systems and control is practical applications. Most approaches in this book are tested for problems that have practical significance. In fact, a main objective of the book is to teach students and practicing engineers how to use the fuzzy systems approach to solving engineering problems in control, signal processing, and communications.

- **Rich and Rigorous:** This book should be intelligently challenging for students. In addition to the emphasis on practicality, many theoretical results are given (which, of course, have practical relevance and importance). All the theorems and lemmas are proven in a mathematically rigorous fashion, and some effort may have to be taken for an average student to comprehend the details.

- **Easy to Use as Textbook:** To facilitate its use as a textbook, this book is written in such a style that each chapter is designed for a one and one-half hour lecture. Sometimes, three chapters may be covered by two lectures, or vice versa, depending upon the emphasis of the instructor and the background of the students. Each chapter contains some exercises and mini-projects that form an integrated part of the text.

The book is divided into six parts. Part I (Chapters 2-6) introduces the fundamental concepts and principles in the general field of fuzzy theory that are particularly useful in fuzzy systems and fuzzy control. Part II (Chapters 7-11) studies the fuzzy systems in detail. The operations inside the fuzzy systems are carefully analyzed and certain properties of the fuzzy systems (for example, approximation capability and accuracy) are studied. Part III (Chapters 12-15) introduces four methods for designing fuzzy systems from sensory measurements, and all these methods are tested for a number of control, signal processing, or communication problems. Part IV (Chapters 16-22) and Part V (Chapters 23-26) parts concentrate on fuzzy control, where Part IV studies nonadaptive fuzzy control and Part V studies adaptive fuzzy control. Finally, Part VI (Chapters 27-31) reviews a number of topics that are not included in the main structure of the book, but are important and strongly relevant to fuzzy systems and fuzzy control.

The book can be studied in many ways, according to the particular interests of the instructor or the reader. Chapters 1-15 cover the general materials that can be applied to a variety of engineering problems. Chapters 16-26 are more specialized in control problems. If the course is not intended as a control course, then some materials in Chapters 16-26 may be omitted, and the time saved may be used for a more detailed coverage of Chapters 1-15 and 27-31. On the other hand, if it

is a control course, then Chapters 16-26 should be studied in detail. The book also can be used, together with a book on neural networks, for a course on neural networks and fuzzy systems. In this case, Chapters 1-15 and selected topics from Chapters 16-31 may be used for the fuzzy system half of the course. If a practicing engineer wants to learn fuzzy systems and fuzzy control quickly, then the proofs of the theorems and lemmas may be skipped.

This book has benefited from the review of many colleagues, students, and friends. First of all, I would like thank my advisors, Lotfi Zadeh and Jerry Mendel, for their continued encouragement. I would like to thank Karl Åström for sending his student, Mikael Johansson, to help me prepare the manuscript during the summer of 1995. Discussions with Kevin Passino, Frank Lewis, Jyh-Shing Jang, Hua Wang, Hideyuki Takagi, and other researchers in fuzzy theory have helped the organization of the materials. The book also benefited from the input of the students who took the course at HKUST.

Support for the author from the Hong Kong Research Grants Council was greatly appreciated.

Finally, I would like to express my gratitude to my department at HKUST for providing the excellent research and teaching environment. Especially, I would like to thank my colleagues Xiren Cao, Zexiang Li, Li Qiu, Erwei Bai, Justin Chuang, Philip Chan, and Kwan-Fai Cheung for their collaboration and critical remarks on various topics in fuzzy theory.

Li-Xin Wang
The Hong Kong University of Science and Technology

Chapter 1

Introduction

1.1 Why Fuzzy Systems?

According to the Oxford English Dictionary, the word "fuzzy" is defined as "blurred, indistinct; imprecisely defined; confused, vague." We ask the reader to disregard this definition and view the word "fuzzy" as a technical adjective. Specifically, fuzzy systems are systems to be precisely defined, and fuzzy control is a special kind of nonlinear control that also will be precisely defined. This is analogous to linear systems and control where the word "linear" is a technical adjective used to specify "systems and control;" the same is true for the word "fuzzy." Essentially, what we want to emphasize is that although the phenomena that the fuzzy systems theory characterizes may be fuzzy, the theory itself is precise.

In the literature, there are two kinds of justification for fuzzy systems theory:

- The real world is too complicated for precise descriptions to be obtained, therefore approximation (or fuzziness) must be introduced in order to obtain a reasonable, yet trackable, model.
- As we move into the information era, human knowledge becomes increasingly important. We need a theory to formulate human knowledge in a systematic manner and put it into engineering systems, together with other information like mathematical models and sensory measurements.

The first justification is correct, but does not characterize the unique nature of fuzzy systems theory. In fact, almost all theories in engineering characterize the real world in an approximate manner. For example, most real systems are nonlinear, but we put a great deal of effort in the study of linear systems. A good engineering theory should be precise to the extent that it characterizes the key features of the real world and, at the same time, is trackable for mathematical analysis. In this aspect, fuzzy systems theory does not differ from other engineering theories.

The second justification characterizes the unique feature of fuzzy systems theory and justifies the existence of fuzzy systems theory as an independent branch in

engineering. As a general principle, a good engineering theory should be capable of making use of all available information effectively. For many practical systems, important information comes from two sources: one source is human experts who describe their knowledge about the system in natural languages; the other is sensory measurements and mathematical models that are derived according to physical laws. An important task, therefore, is to combine these two types of information into system designs. To achieve this combination, a key question is how to formulate human knowledge into a similar framework used to formulate sensory measurements and mathematical models. In other words, the key question is how to transform a human knowledge base into a mathematical formula. Essentially, what a fuzzy system does is to perform this transformation. In order to understand how this transformation is done, we must first know what fuzzy systems are.

1.2 What Are Fuzzy Systems?

Fuzzy systems are knowledge-based or rule-based systems. The heart of a fuzzy system is a knowledge base consisting of the so-called fuzzy IF-THEN rules. A fuzzy IF-THEN rule is an IF-THEN statement in which some words are characterized by continuous membership functions. For example, the following is a fuzzy IF-THEN rule:

$$IF\ the\ speed\ of\ a\ car\ is\ high,\ THEN\ apply\ less\ force\ to\ the\ accelerator \quad (1.1)$$

where the words "high" and "less" are characterized by the membership functions shown in Figs.1.1 and 1.2, respectively.[1] A fuzzy system is constructed from a collection of fuzzy IF-THEN rules. Let us consider two examples.

Example 1.1. Suppose we want to design a controller to automatically control the speed of a car. Conceptually, there are two approaches to designing such a controller: the first approach is to use conventional control theory, for example, designing a PID controller; the second approach is to emulate human drivers, that is, converting the rules used by human drivers into an automatic controller. We now consider the second approach. Roughly speaking, human drivers use the following three types of rules to drive a car in normal situations:

$$IF\ speed\ is\ low,\ THEN\ apply\ more\ force\ to\ the\ accelerator \quad (1.2)$$
$$IF\ speed\ is\ medium,\ THEN\ apply\ normal\ force\ to\ the\ accelerator \quad (1.3)$$
$$IF\ speed\ is\ high,\ THEN\ apply\ less\ force\ to\ the\ accelerator \quad (1.4)$$

where the words "low," "more," "medium," "normal," "high," and "less" are characterized by membership functions similar to those in Figs.1.1-1.2. Of course, more rules are needed in real situations. We can construct a fuzzy system based on these

[1] A detailed definition and analysis of membership functions will be given in Chapter 2. At this point, an intuitive understanding of the membership functions in Figs. 1.1 and 1.2 is sufficient.

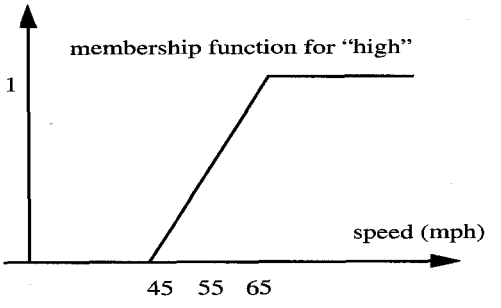


Figure 1.1. Membership function for "high," where the horizontal axis represents the speed of the car and the vertical axis represents the membership value for "high."

rules. Because the fuzzy system is used as a controller, it also is called a fuzzy controller. □

Example 1.2. In Example 1.1, the rules are control instructions, that is, they represent what a human driver does in typical situations. Another type of human knowledge is descriptions about the system. Suppose a person pumping up a balloon wished to know how much air he could add before it burst, then the relationship among some key variables would be very useful. With the balloon there are three key variables: the air inside the balloon, the amount it increases, and the surface tension. We can describe the relationship among these variables in the following fuzzy IF-THEN rules:

$$\textit{IF the amount of air is small and it is increased slightly, THEN the surface tension will increase slightly} \tag{1.5}$$

$$\textit{IF the amount of air is small and it is increased substantially, THEN the surface tension will increase substantially} \tag{1.6}$$

$$\textit{IF the amount of air is large and it is increased slightly, THEN the surface tension will increase moderately} \tag{1.7}$$

$$\textit{IF the amount of air is large and it is increased substantially, THEN the surface tension will increase very substantially} \tag{1.8}$$

where the words "small," "slightly," "substantially," etc., are characterized by membership functions similar to those in Figs.1.1 and 1.2. Combining these rules into a fuzzy system, we obtain a model for the balloon. □

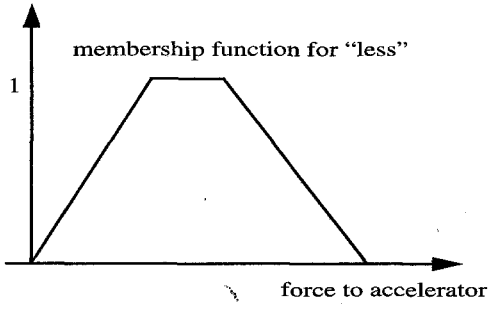


Figure 1.2. *Membership function for "less," where the horizontal axis represents the force applied to the accelerator and the vertical axis represents the membership value for "less."*

In summary, the starting point of constructing a fuzzy system is to obtain a collection of fuzzy IF-THEN rules from human experts or based on domain knowledge. The next step is to combine these rules into a single system. Different fuzzy systems use different principles for this combination. So the question is: what are the commonly used fuzzy systems?

There are three types of fuzzy systems that are commonly used in the literature: (i) pure fuzzy systems, (ii) Takagi-Sugeno-Kang (TSK) fuzzy systems, and (iii) fuzzy systems with fuzzifier and defuzzifier. We now briefly describe these three types of fuzzy systems.

The basic configuration of a pure fuzzy system is shown in Fig. 1.3. The *fuzzy rule base* represents the collection of fuzzy IF-THEN rules. For examples, for the car controller in Example 1.1, the fuzzy rule base consists of the three rules (1.2)-(1.4), and for the balloon model of Example 1.2, the fuzzy rule base consists of the four rules (1.5)-(1.8). The *fuzzy inference engine* combines these fuzzy IF-THEN rules into a mapping from fuzzy sets[2] in the input space $U \subset R^n$ to fuzzy sets in the output space $V \subset R$ based on fuzzy logic principles. If the dashed feedback line in Fig. 1.3 exists, the system becomes the so-called fuzzy dynamic system.

The main problem with the pure fuzzy system is that its inputs and outputs are

[2]The precise definition of fuzzy set is given in Chapter 2. At this point, it is sufficient to view a fuzzy set as a word like, for example, "high," which is characterized by the membership function shown in Fig.1.1.

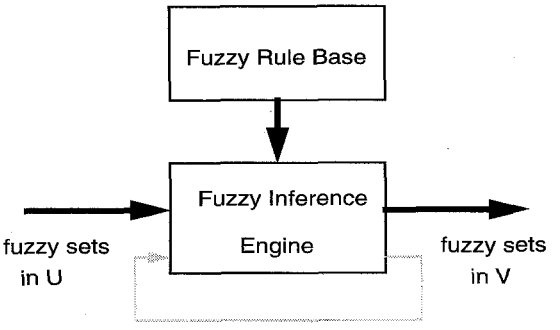


Figure 1.3. Basic configuration of pure fuzzy systems.

fuzzy sets (that is, words in natural languages), whereas in engineering systems the inputs and outputs are real-valued variables. To solve this problem, Takagi, Sugeno, and Kang (Takagi and Sugeno [1985] and Sugeno and Kang [1988]) proposed another fuzzy system whose inputs and outputs are real-valued variables.

Instead of considering the fuzzy IF-THEN rules in the form of (1.1), the Takagi-Sugeno-Kang (TSK) system uses rules in the following form:

$$\begin{aligned} &IF \text{ the speed } x \text{ of a car is high,} \\ &THEN \text{ the force to the accelerator is } y = cx \end{aligned} \tag{1.9}$$

where the word "high" has the same meaning as in (1.1), and c is a constant. Comparing (1.9) and (1.1) we see that the THEN part of the rule changes from a description using words in natural languages into a simple mathematical formula. This change makes it easier to combine the rules. In fact, the Takagi-Sugeno-Kang fuzzy system is a weighted average of the values in the THEN parts of the rules. The basic configuration of the Takagi-Sugeno-Kang fuzzy system is shown in Fig. 1.4.

The main problems with the Takagi-Sugeno-Kang fuzzy system are: (i) its THEN part is a mathematical formula and therefore may not provide a natural framework to represent human knowledge, and (ii) there is not much freedom left to apply different principles in fuzzy logic, so that the versatility of fuzzy systems is not well-represented in this framework. To solve these problems, we use the third type of fuzzy systems—fuzzy systems with fuzzifier and defuzzifier.

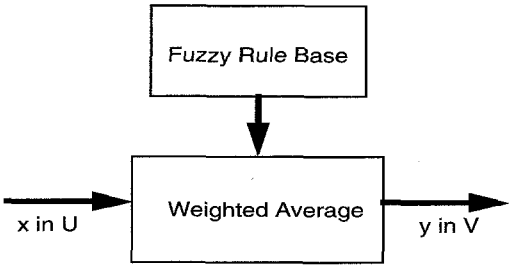


Figure 1.4. Basic configuration of Takagi-Sugeno-Kang fuzzy system.

In order to use pure fuzzy systems in engineering systems, a simple method is to add a fuzzifier, which transforms a real-valued variable into a fuzzy set, to the input, and a defuzzifier, which transforms a fuzzy set into a real-valued variable, to the output. The result is the fuzzy system with fuzzifier and defuzzifier, shown in Fig. 1.5. This fuzzy system overcomes the disadvantages of the pure fuzzy systems and the Takagi-Sugeno-Kang fuzzy systems. Unless otherwise specified, from now on when we refer fuzzy systems we mean fuzzy systems with fuzzifier and defuzzifier.

To conclude this section, we would like to emphasize a distinguished feature of fuzzy systems: on one hand, fuzzy systems are multi-input-single-output mappings from a real-valued vector to a real-valued scalar (a multi-output mapping can be decomposed into a collection of single-output mappings), and the precise mathematical formulas of these mappings can be obtained (see Chapter 9 for details); on the other hand, fuzzy systems are knowledge-based systems constructed from human knowledge in the form of fuzzy IF-THEN rules. *An important contribution of fuzzy systems theory is that it provides a systematic procedure for transforming a knowledge base into a nonlinear mapping.* Because of this transformation, we are able to use knowledge-based systems (fuzzy systems) in engineering applications (control, signal processing, or communications systems, etc.) in the same manner as we use mathematical models and sensory measurements. Consequently, the analysis and design of the resulting combined systems can be performed in a mathematically rigorous fashion. The goal of this text is to show how this transformation is done, and how the analysis and design are performed.

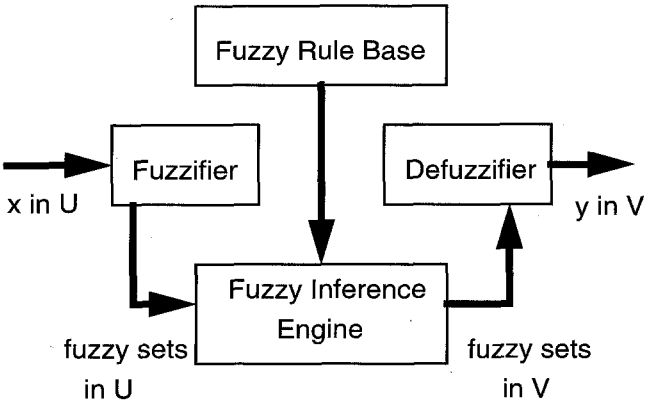


Figure 1.5. Basic configuration of fuzzy systems with fuzzifier and defuzzifier.

1.3 Where Are Fuzzy Systems Used and How?

Fuzzy systems have been applied to a wide variety of fields ranging from control, signal processing, communications, integrated circuit manufacturing, and expert systems to business, medicine, psychology, etc. However, the most significant applications have concentrated on control problems. Therefore, instead of listing the applications of fuzzy systems in the different fields, we concentrate on a number of control problems where fuzzy systems play a major role.

Fuzzy systems, as shown in Fig. 1.5, can be used either as open-loop controllers or closed-loop controllers, as shown in Figs. 1.6 and 1.7, respectively. When used as an open-loop controller, the fuzzy system usually sets up some control parameters and then the system operates according to these control parameters. Many applications of fuzzy systems in consumer electronics belong to this category. When used as a closed-loop controller, the fuzzy system measures the outputs of the process and takes control actions on the process continuously. Applications of fuzzy systems in industrial processes belong to this category. We now briefly describe how fuzzy systems are used in a number of consumer products and industrial systems.

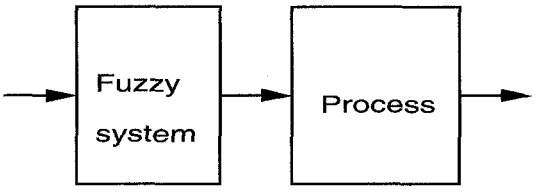


Figure 1.6. Fuzzy system as open-loop controller.

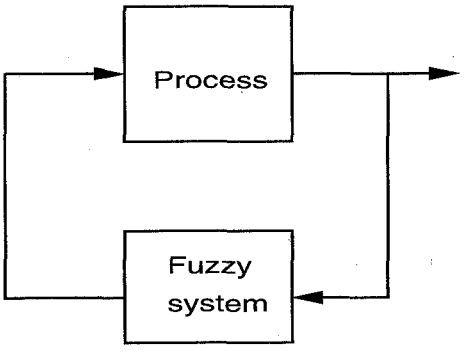


Figure 1.7. Fuzzy system as closed-loop controller.

1.3.1 Fuzzy Washing Machines

The fuzzy washing machines were the first major consumer products to use fuzzy systems. They were produced by Matsushita Electric Industrial Company in Japan around 1990. They use a fuzzy system to automatically set the proper cycle according to the kind and amount of dirt and the size of the load. More specifically, the fuzzy system used is a three-input-one-output system, where the three inputs

are measurements of dirtiness, type of dirt, and load size, and the output is the correct cycle. Sensors supply the fuzzy system with the inputs. The optical sensor sends a beam of light through the water and measures how much of it reaches the other side. The dirtier the water, the less light crosses. The optical sensor also can tell whether the dirt is muddy or oily. Muddy dirt dissolves faster. So, if the light readings reach minimum quickly, the dirt is muddy. If the downswing is slower, it is oily. And if the curve slopes somewhere in between, the dirt is mixed. The machine also has a load sensor that registers the volume of clothes. Clearly, the more volume of the clothes, the more washing time is needed. The heuristics above were summarized in a number of fuzzy IF-THEN rules that were then used to construct the fuzzy system.

1.3.2 Digital Image Stabilizer

Anyone who has ever used a camcorder realizes that it is very difficult for a human hand to hold the camcorder without shaking it slightly and imparting an irksome quiver to the tape. Smoothing out this jitter would produce a new generation of camcorders and would have tremendous commercial value. Matsushita introduced what it calls a digital image stabilizer, based on fuzzy systems, which stabilizes the picture when the hand is shaking. The digital image stabilizer is a fuzzy system that is constructed based on the following heuristics:

IF all the points in the picture are moving in the same direction,
THEN the hand is shaking (1.10)

IF only some points in the picture are moving,
THEN the hand is not shaking (1.11)

More specifically, the stabilizer compares each current frame with the previous images in memory. If the whole appears to have shifted, then according to (1.10) the hand is shaking and the fuzzy system adjusts the frame to compensate. Otherwise, it leaves it alone. Thus, if a car crosses the field, only a portion of the image will change, so the camcorder does not try to compensate. In this way the picture remains steady, although the hand is shaking.

1.3.3 Fuzzy Systems in Cars

An automobile is a collection of many systems—engine, transmission, brake, suspension, steering, and more—and fuzzy systems have been applied to almost all of them. For example, Nissan has patented a fuzzy automatic transmission that saves fuel by 12 to 17 percent. It is based on the following observation. A normal transmission shifts whenever the car passes a certain speed, it therefore changes quite often and each shift consumes gas. However, human drivers not only shift less frequently, but also consider nonspeed factors. For example, if accelerating up

a hill, they may delay the shift. Nissan's fuzzy automatic transmission device summarized these heuristics into a collection of fuzzy IF-THEN rules that were then used to construct a fuzzy system to guide the changes of gears.

Nissan also developed a fuzzy antilock braking system. The challenge here is to apply the greatest amount of pressure to the brake without causing it to lock. The Nissan system considers a number of heuristics, for example,

IF the car slows down very rapidly,
THEN the system assumes brake – lock and eases up on pressure (1.12)

In April 1992, Mitsubishi announced a fuzzy omnibus system that controls a car's automatic transmission, suspension, traction, four-wheel steering, four-wheel drive, and air conditioner. The fuzzy transmission downshifts on curves and also keeps the car from upshifting inappropriately on bends or when the driver releases the accelerator. The fuzzy suspension contains sensors in the front of the car that register vibration and height changes in the road and adjusts the suspension for a smoother ride. Fuzzy traction prevents excess speed on corners and improves the grip on slick roads by deciding whether they are level or sloped. Finally, fuzzy steering adjusts the response angle of the rear wheels according to road conditions and the car's speed, and fuzzy air conditioning monitors sunlight, temperature, and humidity to enhance the environment inside the car.

1.3.4 Fuzzy Control of a Cement Kiln

Cement is manufactured by finegrinding of cement clinker. The clinkers are produced in the cement kiln by heating a mixture of linestone, clay, and sand components. Because cement kilns exhibit time-varying nonlinear behavior and relatively few measurements are available, they are difficult to control using conventional control theory.

In the late 1970s, Holmblad and Østergaard of Denmark developed a fuzzy system to control the cement kiln. The fuzzy system (fuzzy controller) had four inputs and two outputs (which can be viewed as two fuzzy systems in the form of Fig. 1.5, which share the same inputs). The four inputs are: (i) oxygen percentage in exhausted gases, (ii) temperature of exhaust gases, (iii) kiln drive torque, and (iv) litre weight of clinker (indicating temperature level in the burning zone and quality of clinker). The two outputs are: (i) coal feed rate and (ii) air flow. A collection of fuzzy IF-THEN rules were constructed that describe how the outputs should be related to the inputs. For example, the following two rules were used:

IF the oxygen percentage is high and the temperature is low,
THEN increase air flow (1.13)

IF the oxygen percentage is high and the temperature is high,
THEN reduce the coal feed rate slightly (1.14)

The fuzzy controller was constructed by combining these rules into fuzzy systems. In June 1978, the fuzzy controller ran for six days in the cement kiln of F.L. Smidth & Company in Denmark—the first successful test of fuzzy control on a full-scale industrial process. The fuzzy controller showed a slight improvement over the results of the human operator and also cut fuel consumption. We will show more details about this system in Chapter 16.

1.3.5 Fuzzy Control of Subway Train

The most significant application of fuzzy systems to date may be the fuzzy control system for the Sendai subway in Japan. On a single north-south route of 13.6 kilometers and 16 stations, the train runs along very smoothly. The fuzzy control system considers four performance criteria simutaneously: safety, riding comfort, traceability to target speed, and accuracy of stopping gap. The fuzzy control system consists of two parts: the constant speed controller (it starts the train and keeps the speed below the safety limit), and the automatic stopping controller (it regulates the train speed in order to stop at the target position). The constant speed controller was constructed from rules such as:

For safety; IF the speed of train is approaching the limit speed, THEN select the maximum brake notch (1.15)

For riding comfort; IF the speed is in the allowed range, THEN do not change the control notch (1.16)

More rules were used in the real system for traceability and other factors. The automatic stopping controller was constructed from the rules like:

For riding comfort; IF the train will stop in the allowed zone, THEN do not change the control notch (1.17)

For riding comfort and safety; IF the train is in the allowed zone, THEN change the control notch from acceleration to slight braking (1.18)

Again, more rules were used in the real system to take care of the accuracy of stopping gap and other factors. By 1991, the Sendai subway had carried passengers for four years and was still one of the most advanced subway systems.

1.4 What Are the Major Research Fields in Fuzzy Theory?

By fuzzy theory we mean all the theories that use the basic concept of fuzzy set or continuous membership function. Fuzzy theory can be roughly classified according to Fig.1.8. There are five major branches: (i) fuzzy mathematics, where classical mathematical concepts are extended by replacing classical sets with fuzzy sets; (ii) fuzzy logic and artificial intelligence, where approximations to classical logic

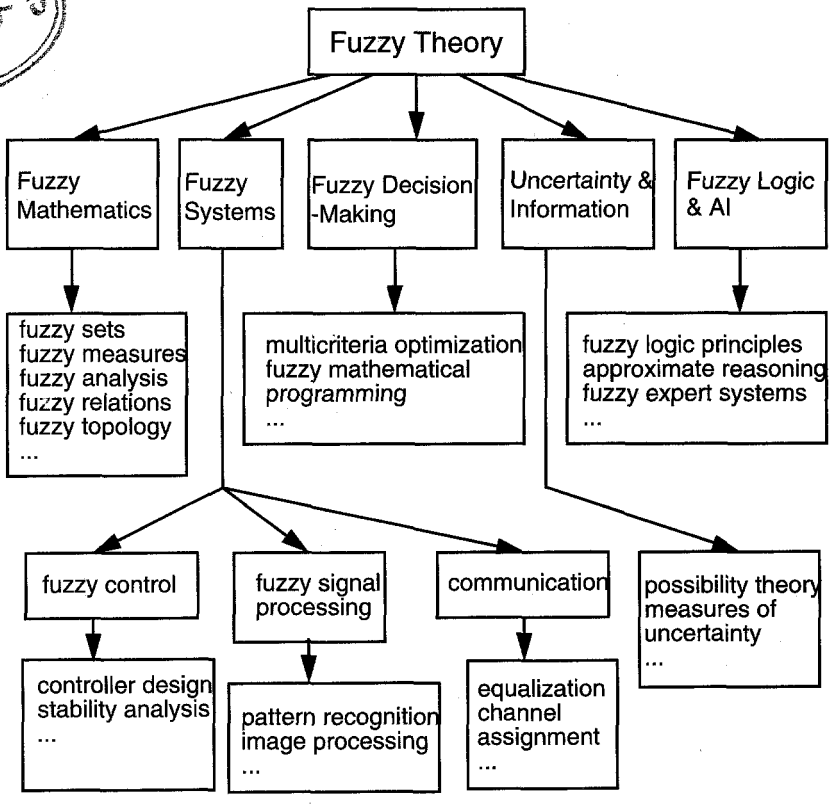


Figure 1.8. Classification of fuzzy theory.

are introduced and expert systems are developed based on fuzzy information and approximate reasoning; (iii) fuzzy systems, which include fuzzy control and fuzzy approaches in signal processing and communications; (iv) uncertainty and information, where different kinds of uncertainties are analyzed; and (v) fuzzy decision making, which considers optimalization problems with soft constraints.

Of course, these five branches are not independent and there are strong interconnections among them. For example, fuzzy control uses concepts from fuzzy mathematics and fuzzy logic.

From a practical point of view, the majority of applications of fuzzy theory has concentrated on fuzzy systems, especially fuzzy control, as we could see from the examples in Section 1.3. There also are some fuzzy expert systems that perform

medical diagnoses and decision support (Terano, Asai and Sugeno [1994]). Because fuzzy theory is still in its infancy from both theoretical and practical points of view, we expect that more solid practical applications will appear as the field matures.

From Fig. 1.8 we see that fuzzy theory is a huge field that comprises a variety of research topics. In this text, we concentrate on fuzzy systems and fuzzy control. We first will study the basic concepts in fuzzy mathematics and fuzzy logic that are useful in fuzzy systems and control (Chapters 2-6), then we will study fuzzy systems and control in great detail (Chapters 7-26), and finally we will briefly review some topics in other fields of fuzzy theory (Chapters 27-31).

1.5 A Brief History of Fuzzy Theory and Applications

1.5.1 The 1960s: The Beginning of Fuzzy Theory

Fuzzy theory was initiated by Lotfi A. Zadeh in 1965 with his seminal paper "Fuzzy Sets" (Zadeh [1965]). Before working on fuzzy theory, Zadeh was a well-respected scholar in control theory. He developed the concept of "state," which forms the basis for modern control theory. In the early '60s, he thought that classical control theory had put too much emphasis on precision and therefore could not handle the complex systems. As early as 1962, he wrote that to handle biological systems "we need a radically different kind of mathematics, the mathematics of fuzzy or cloudy quantities which are not describable in terms of probability distributions" (Zadeh [1962]). Later, he formalized the ideas into the paper "Fuzzy Sets."

Since its birth, fuzzy theory has been sparking controversy. Some scholars, like Richard Bellman, endorsed the idea and began to work in this new field. Other scholars objected to the idea and viewed "fuzzification" as against basic scientific principles. The biggest challenge, however, came from mathematicians in statistics and probability who claimed that probability is sufficient to characterize uncertainty and any problems that fuzzy theory can solve can be solved equally well or better by probability theory (see Chapter 31). Because there were no real practical applications of fuzzy theory in the beginning, it was difficult to defend the field from a purely philosophical point of view. Almost all major research institutes in the world failed to view fuzzy theory as a serious research field.

Although fuzzy theory did not fall into the mainstream, there were still many researchers around the world dedicating themselves to this new field. In the late 1960s, many new fuzzy methods like fuzzy algorithms, fuzzy decision making, etc., were proposed.

1.5.2 The 1970s: Theory Continued to Grow and Real Applications Appeared

It is fair to say that the establishment of fuzzy theory as an independent field is largely due to the dedication and outstanding work of Zadeh. Most of the funda-

mental concepts in fuzzy theory were proposed by Zadeh in the late '60s and early '70s. After the introduction of fuzzy sets in 1965, he proposed the concepts of fuzzy algorithms in 1968 (Zadeh [1968]), fuzzy decision making in 1970 (Bellman and Zadeh [1970]), and fuzzy ordering in 1971 (Zadeh [1971b]). In 1973, he published another seminal paper, "Outline of a new approach to the analysis of complex systems and decision processes" (Zadeh [1973]), which established the foundation for fuzzy control. In this paper, he introduced the concept of linguistic variables and proposed to use fuzzy IF-THEN rules to formulate human knowledge.

A big event in the '70s was the birth of fuzzy controllers for real systems. In 1975, Mamdani and Assilian established the basic framework of fuzzy controller (which is essentially the fuzzy system in Fig.1.5) and applied the fuzzy controller to control a steam engine. Their results were published in another seminal paper in fuzzy theory "An experiment in linguistic synthesis with a fuzzy logic controller" (Mamdani and Assilian [1975]). They found that the fuzzy controller was very easy to construct and worked remarkably well. Later in 1978, Holmblad and Østergaard developed the first fuzzy controller for a full-scale industrial process—the fuzzy cement kiln controller (see Section 1.3).

Generally speaking, the foundations of fuzzy theory were established in the 1970s. With the introduction of many new concepts, the picture of fuzzy theory as a new field was becoming clear. Initial applications like the fuzzy steam engine controller and the fuzzy cement kiln controller also showed that the field was promising. Usually, the field should be founded by major resources and major research institutes should put some manpower on the topic. Unfortunately, this never happened. On the contrary, in the late '70s and early '80s, many researchers in fuzzy theory had to change their field because they could not find support to continue their work. This was especially true in the United States.

1.5.3 The 1980s: Massive Applications Made a Difference

In the early '80s, this field, from a theoretical point of view, progressed very slowly. Few new concepts and approaches were proposed during this period, simply because very few people were still working in the field. It was the application of fuzzy control that saved the field.

Japanese engineers, with their sensitivity to new technology, quickly found that fuzzy controllers were very easy to design and worked very well for many problems. Because fuzzy control does not require a mathematical model of the process, it could be applied to many systems where conventional control theory could not be used due to a lack of mathematical models. In 1980, Sugeno began to create Japan's first fuzzy application—control of a Fuji Electric water purification plant. In 1983, he began the pioneer work on a fuzzy robot, a self-parking car that was controlled by calling out commands (Sugeno and Nishida [1985]). In the early 1980s, Yasunobu and Miyamoto from Hitachi began to develop a fuzzy control system for the Sandai

subway. They finished the project in 1987 and created the most advanced subway system on earth. This very impressive application of fuzzy control made a very big difference.

In July 1987, the Second Annual International Fuzzy Systems Association Conference was held in Tokyo. The conference began three days after the Sendai subway began operation, and attendees were amused with its dreamy ride. Also, in the conference Hirota displayed a fuzzy robot arm that played two-dimensional Ping-Pong in real time (Hirota, Arai and Hachisu [1989]), and Yamakawa demonstrated a fuzzy system that balanced an inverted pendulum (Yamakawa [1989]). Prior to this event, fuzzy theory was not well-known in Japan. After it, a wave of pro-fuzzy sentiment swept through the engineering, government, and business communities. By the early 1990s, a large number of fuzzy consumer products appeared in the market (see Section 1.3 for examples).

1.5.4 The 1990s: More Challenges Remain

The success of fuzzy systems in Japan surprised the mainstream researchers in the United States and in Europe. Some still criticize fuzzy theory, but many others have been changing their minds and giving fuzzy theory a chance to be taken seriously. In February 1992, the first IEEE International Conference on Fuzzy Systems was held in San Diego. This event symbolized the acceptance of fuzzy theory by the largest engineering organization—IEEE. In 1993, the IEEE Transactions on Fuzzy Systems was inaugurated.

From a theoretical point of view, fuzzy systems and control has advanced rapidly in the late 1980s and early 1990s. Although it is hard to say there is any breakthrough, solid progress has been made on some fundamental problems in fuzzy systems and control. For examples, neural network techniques have been used to determine membership functions in a systematic manner, and rigorous stability analysis of fuzzy control systems has appeared. Although the whole picture of fuzzy systems and control theory is becoming clearer, much work remains to be done. Most approaches and analyses are preliminary in nature. We believe that only when the top research institutes begin to put some serious man power on the research of fuzzy theory can the field make major progress.

1.6 Summary and Further Readings

In this chapter we have demonstrated the following:

- The goal of using fuzzy systems is to put human knowledge into engineering systems in a systematic, efficient, and analyzable order.

- The basic architectures of the commonly used fuzzy systems.

- The fuzzy IF-THEN rules used in certain industrial processes and consumer products.
- Classification and brief history of fuzzy theory and applications.

A very good non-technical introduction to fuzzy theory and applications is McNeill and Freiberger [1993]. It contains many interviews and describes the major events. Some historical remarks were made in Kruse, Gebhardt, and Klawonn [1994]. Klir and Yuan [1995] is perhaps the most comprehensive book on fuzzy sets and fuzzy logic. Earlier applications of fuzzy control were collected in Sugeno [1985] and more recent applications (mainly in Japan) were summarized in Terano, Asai, and Sugeno [1994].

1.7 Exercises

Exercise 1.1. Is the fuzzy washing machine an open-loop control system or a closed-loop control system? What about the fuzzy cement kiln control system? Explain your answer.

Exercise 1.2. List four to six applications of fuzzy theory to practical problems other than those in Section 1.3. Point out the references where you find these applications.

Exercise 1.3. Suppose we want to design a fuzzy system to balance the inverted pendulum shown in Fig. 1.9. Let the angle θ and its derivation $\dot{\theta}$ be the inputs to the fuzzy system and the force u applied to the cart be its output.

(a) Determine three to five fuzzy IF-THEN rules based on the common sense of how to balance the inverted pendulum.

(b) Suppose that the rules in (a) can successfully control a particular inverted pendulum system. Now if we want to use the rules to control another inverted pendulum system with different values of m_c, m, and l, what parts of the rules should change and what parts may remain the same.

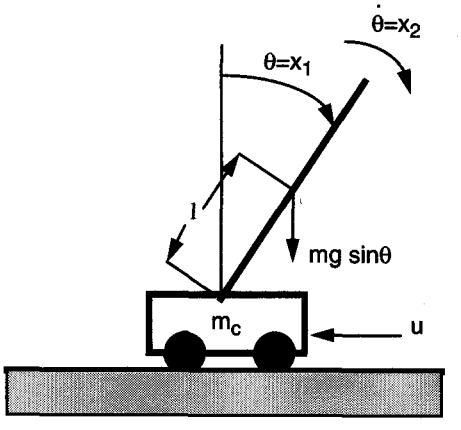


Figure 1.9. The inverted pendulum system.

Part I

The Mathematics of Fuzzy Systems and Control

Fuzzy mathematics provide the starting point and basic language for fuzzy systems and fuzzy control. Fuzzy mathematics by itself is a huge field, where fuzzy mathematical principles are developed by replacing the sets in classical mathematical theory with fuzzy sets. In this way, all the classical mathematical branches may be "fuzzified." We have seen the birth of fuzzy measure theory, fuzzy topology, fuzzy algebra, fuzzy analysis, etc. Understandably, only a small portion of fuzzy mathematics has found applications in engineering. In the next five chapters, we will study those concepts and principles in fuzzy mathematics that are useful in fuzzy systems and fuzzy control.

In Chapter 2, we will introduce the most fundamental concept in fuzzy theory —the concept of fuzzy set. In Chapter 3, set-theoretical operations on fuzzy sets such as complement, union, and intersection will be studied in detail. Chapter 4 will study fuzzy relations and introduce an important principle in fuzzy theory— the extension principle. Linguistic variables and fuzzy IF-THEN rules, which are essential to fuzzy systems and fuzzy control, will be precisely defined and studied in Chapter 5. Finally, Chapter 6 will focus on three basic principles in fuzzy logic that are useful in the fuzzy inference engine of fuzzy systems.

Chapter 2

Fuzzy Sets and Basic Operations on Fuzzy Sets

2.1 From Classical Sets to Fuzzy Sets

Let U be the *universe of discourse*, or *universal set*, which contains all the possible elements of concern in each particular context or application. Recall that a *classical (crisp) set* A, or simply a set A, in the universe of discourse U can be defined by listing all of its members (*the list method*) or by specifying the properties that must be satisfied by the members of the set (*the rule method*). The list method can be used only for finite sets and is therefore of limited use. The rule method is more general. In the rule method, a set A is represented as

$$A = \{x \in U | x \text{ meets some conditions}\} \tag{2.1}$$

There is yet a third method to define a set A—*the membership method*, which introduces a zero-one membership function (also called characteristic function, discrimination function, or indicator function) for A, denoted by $\mu_A(x)$, such that

$$\mu_A(x) = \begin{cases} 1 & if \quad x \in A \\ 0 & if \quad x \notin A \end{cases} \tag{2.2}$$

The set A is mathematically equivalent to its membership function $\mu_A(x)$ in the sense that knowing $\mu_A(x)$ is the same as knowing A itself.

Example 2.1. Consider the set of all cars in Berkeley; this is the universe of discourse U. We can define different sets in U according to the properties of cars. Fig. 2.1 shows two types of properties that can be used to define sets in U: (a) US cars or non-US cars, and (b) number of cylinders. For example, we can define a set A as all cars in U that have 4 cylinders, that is,

$$A = \{x \in U | x \text{ has 4 cylinders}\} \tag{2.3}$$

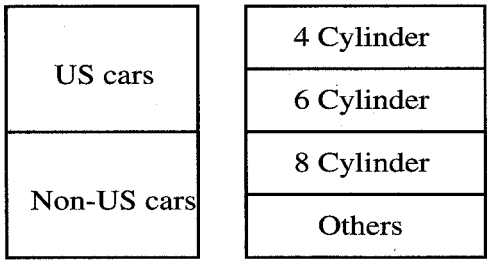


Figure 2.1. Partitioning of the set of all cars in Berkeley into subsets by: (a) US cars or non-US cars, and (b) number of cylinders.

or

$$\mu_A(x) = \begin{cases} 1 & if \quad x \in U \text{ and } x \text{ has 4 cylinders} \\ 0 & if \quad x \in U \text{ and } x \text{ does not have 4 cylinders} \end{cases} \tag{2.4}$$

If we want to define a set in U according to whether the car is a US car or a non-US car, we face a difficulty. One perspective is that a car is a US car if it carries the name of a USA auto manufacturer; otherwise it is a non-US car. However, many people feel that the distinction between a US car and a non-US car is not as crisp as it once was, because many of the components for what we consider to be US cars (for examples, Fords, GM's, Chryslers) are produced outside of the United States. Additionally, some "non-US" cars are manufactured in the USA. How to deal with this kind of problems? □

Essentially, the difficulty in Example 2.1 shows that *some sets do not have clear boundaries.* Classical set theory requires that a set must have a well-defined property, therefore it is unable to define the set like "all US cars in Berkeley." To overcome this limitation of classical set theory, the concept of fuzzy set was introduced. It turns out that this limitation is fundamental and a new theory is needed—this is the fuzzy set theory.

Definition 2.1. A *fuzzy set* in a universe of discourse U is characterized by a membership function $\mu_A(x)$ that takes values in the interval $[0, 1]$.

Therefore, a fuzzy set is a generalization of a classical set by allowing the mem-

bership function to take any values in the interval $[0, 1]$. In other words, the membership function of a classical set can only take two values—zero and one, whereas the membership function of a fuzzy set is a continuous function with range $[0, 1]$. We see from the definition that there is nothing "fuzzy" about a fuzzy set; it is simply a set with a continuous membership function.

A fuzzy set A in U may be represented as a set of ordered pairs of a generic element x and its membership value, that is,

$$A = \{(x, \mu_A(x)) | x \in U\} \tag{2.5}$$

When U is continuous (for example, $U = R$), A is commonly written as

$$A = \int_U \mu_A(x)/x \tag{2.6}$$

where the integral sign does not denote integration; it denotes the collection of all points $x \in U$ with the associated membership function $\mu_A(x)$. When U is discrete, A is commonly written as

$$A = \sum_U \mu_A(x)/x \tag{2.7}$$

where the summation sign does not represent arithmetic addition; it denotes the collection of all points $x \in U$ with the associated membership function $\mu_A(x)$.

We now return to Example 2.1 and see how to use the concept of fuzzy set to define US and non-US cars.

Example 2.1 (Cont'd). We can define the set "US cars in Berkeley," denoted by D, as a fuzzy set according to the percentage of the car's parts made in the USA. Specifically, D is defined by the membership function

$$\mu_D(x) = p(x) \tag{2.8}$$

where $p(x)$ is the percentage of the parts of car x made in the USA and it takes values from 0% to 100%. For example, if a particular car x_0 has 60% of its parts made in the USA, then we say that the car x_0 belongs to the fuzzy set D to the degree of 0.6.

Similarly, we can define the set "non-US cars in Berkeley," denoted by F, as a fuzzy set with the membership function

$$\mu_F(x) = 1 - p(x) \tag{2.9}$$

where $p(x)$ is the same as in (2.8). Thus, if a particular car x_0 has 60% of its parts made in the USA, then we say the car x_0 belongs to the fuzzy set F to the degree of 1-0.6=0.4. Fig. 2.2 shows (2.8) and (2.9). Clearly, an element can belong to different fuzzy sets to the same or different degrees. □

We now consider another example of fuzzy sets and from it draw some remarks.

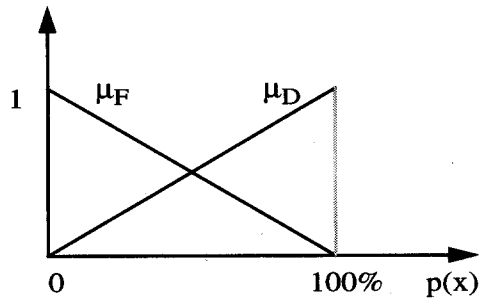


Figure 2.2. Membership functions for US (μ_D) and non-US (μ_F) cars based on the percentage of parts of the car made in the USA ($p(x)$).

Example 2.2. Let Z be a fuzzy set named "numbers close to zero." Then a possible membership function for Z is

$$\mu_Z(x) = e^{-x^2} \tag{2.10}$$

where $x \in R$. This is a Gaussian function with mean equal to zero and standard derivation equal to one. According to this membership function, the numbers 0 and 2 belong to the fuzzy set Z to the degrees of $e^0 = 1$ and e^{-4}, respectively.

We also may define the membership function for Z as

$$\mu_Z(x) = \begin{cases} 0 & if \quad x < -1 \\ x+1 & if \quad -1 \leq x < 0 \\ 1-x & if \quad 0 \leq x < 1 \\ 0 & if \quad 1 \leq x \end{cases} \tag{2.11}$$

According to this membership function, the numbers 0 and 2 belong to the fuzzy set Z to the degrees of 1 and 0, respectively. (2.10) and (2.11) are plotted graphically in Figs. 2.3 and 2.4, respectively. We can choose many other membership functions to characterize "numbers close to zero." □

From Example 2.2 we can draw three important remarks on fuzzy sets:

- The properties that a fuzzy set is used to characterize are usually fuzzy, for example, "numbers close to zero" is not a precise description. Therefore, we

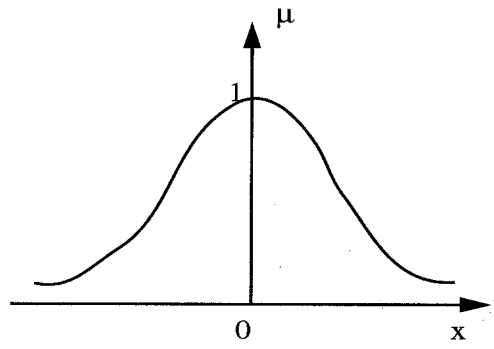


Figure 2.3. A possible membership function to characterize "numbers close to zero."

may use different membership functions to characterize the same description. However, the membership functions themselves are not fuzzy—they are precise mathematical functions. Once a fuzzy property is represented by a membership function, for example, once "numbers close to zero" is represented by the membership function (2.10) or (2.11), nothing will be fuzzy anymore. Thus, by characterizing a fuzzy description with a membership function, we essentially *defuzzify* the fuzzy description. A common misunderstanding of fuzzy set theory is that fuzzy set theory tries to fuzzify the world. We see, on the contrary, that fuzzy sets are used to defuzzify the world.

- Following the previous remark is an important question: how to determine the membership functions? Because there are a variety of choices of membership functions, how to choose one from these alternatives? Conceptually, there are two approaches to determining a membership function. The first approach is to use the knowledge of human experts, that is, ask the domain experts to specify the membership functions. Because fuzzy sets are often used to formulate human knowledge, the membership functions represent a part of human knowledge. Usually, this approach can only give a rough formula of the membership function; fine-tuning is required. In the second approach, we use data collected from various sensors to determine the membership functions. Specifically, we first specify the structures of the membership functions and then fine-tune the parameters of the membership functions based on the data. Both approaches, especially the second approach, will be studied in detail in

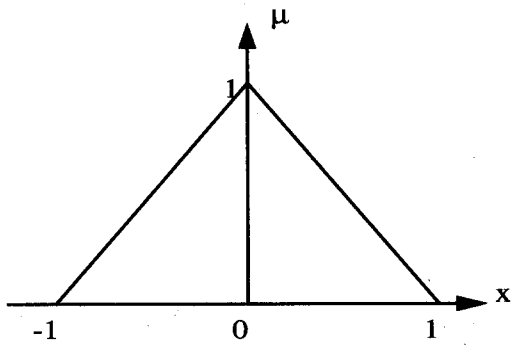


Figure 2.4. Another possible membership function to characterize "numbers close to zero."

later chapters.

- Finally, it should be emphasized that although (2.10) and (2.11) are used to characterize the same description "numbers close to zero," they are different fuzzy sets. Hence, rigorously speaking, we should use different labels to represent the fuzzy sets (2.10) and (2.11); for example, we should use $\mu_{Z_1}(x)$ in (2.10) and $\mu_{Z_2}(x)$ in (2.11). A fuzzy set has a one-to-one correspondence with its membership function. That is, when we say a fuzzy set, there must be a unique membership function associated with it; conversely, when we give a membership function, it represents a fuzzy set. Fuzzy sets and their membership functions are equivalent in this sense.

Let us consider two more examples of fuzzy sets, one in continuous domain and the other in discrete domain; they are classical examples from Zadeh's seminal paper (Zadeh [1965]).

Example 2.3. Let U be the interval $[0, 100]$ representing the age of ordinary humans. Then we may define fuzzy sets "young" and "old" as (using the integral notation (2.6))

$$young = \int_0^{25} 1/x + \int_{25}^{100} (1 + (\frac{x-25}{5})^2)^{-1}/x \tag{2.12}$$

$$old = \int_{50}^{100} (1 + (\frac{x-50}{5})^{-2})^{-1}/x \tag{2.13}$$

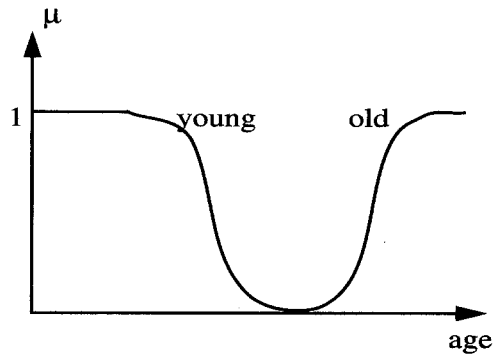


Figure 2.5. Diagrammatic representation of "young" and "old."

See Fig. 2.5. □

Example 2.4. Let U be the integers from 1 to 10, that is, $U = \{1, 2, ..., 10\}$. Then the fuzzy set "several" may be defined as (using the summation notation (2.7))

$$several = 0.5/3 + 0.8/4 + 1/5 + 1/6 + 0.8/7 + 0.5/8 \tag{2.14}$$

That is, 5 and 6 belong to the fuzzy set "several" with degree 1, 4 and 7 with degree 0.8, 3 and 8 with degree 0.5, and 1,2,9 and 10 with degree 0. See Fig. 2.6. □

2.2 Basic Concepts Associated with Fuzzy Set

We now introduce some basic concepts and terminology associated with a fuzzy set. Many of them are extensions of the basic concepts of a classical (crisp) set, but some are unique to the fuzzy set framework.

Definition 2.2. The concepts of support, fuzzy singleton, center, crossover point, height, normal fuzzy set, α-cut, convex fuzzy set, and projections are defined as follows.

The *support* of a fuzzy set A in the universe of discourse U is a crisp set that contains all the elements of U that have nonzero membership values in A, that is,

$$supp(A) = \{x \in U | \mu_A(x) > 0\} \tag{2.15}$$

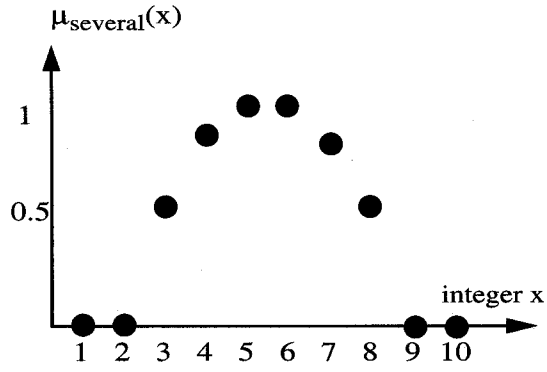


Figure 2.6. Membership function for fuzzy set "several."

where $supp(A)$ denotes the support of fuzzy set A. For example, the support of fuzzy set "several" in Fig. 2.6 is the set of integers $\{3,4,5,6,7,8\}$. If the support of a fuzzy set is empty, it is called an *empty fuzzy set.* A *fuzzy singleton* is a fuzzy set whose support is a single point in U.

The *center* of a fuzzy set is defined as follows: if the mean value of all points at which the membership function of the fuzzy set achieves its maximum value is finite, then define this mean value as the center of the fuzzy set; if the mean value equals positive (negative) infinite, then the center is defined as the smallest (largest) among all points that achieve the maximum membership value. Fig. 2.7 shows the centers of some typical fuzzy sets. The *crossover point* of a fuzzy set is the point in U whose membership value in A equals 0.5.

The *height* of a fuzzy set is the largest membership value attained by any point. For example, the heights of all the fuzzy sets in Figs.2.2-2.4 equal one. If the height of a fuzzy set equals one, it is called a *normal fuzzy set.* All the fuzzy sets in Figs. 2.2-2.4 are therefore normal fuzzy sets.

An *α-cut* of a fuzzy set A is a crisp set A_α that contains all the elements in U that have membership values in A greater than or equal to α, that is,

$$A_\alpha = \{x \in U | \mu_A(x) \geq \alpha\} \tag{2.16}$$

For example, for $\alpha = 0.3$, the α-cut of the fuzzy set (2.11) (Fig. 2.4) is the crisp set $[-0.7, 0.7]$, and for $\alpha = 0.9$, it is $[-0.1, 0.1]$.

When the universe of discourse U is the n-dimensional Euclidean space R^n, the

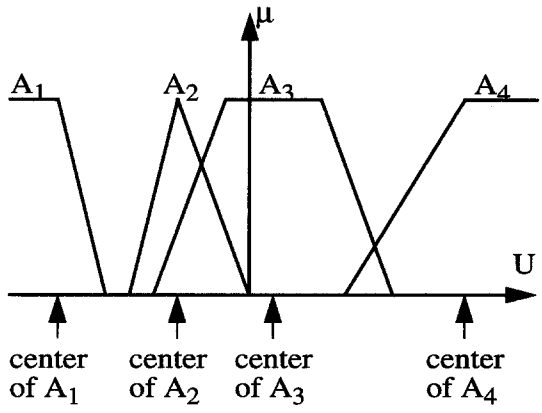


Figure 2.7. Centers of some typical fuzzy sets.

concept of set convexity can be generalized to fuzzy set. A fuzzy set A is said to be *convex* if and only if its α-cut A_α is a convex set for any α in the interval $(0, 1]$. The following lemma gives an equivalent definition of a convex fuzzy set.

Lemma 2.1. A fuzzy set A in R^n is convex if and only if

$$\mu_A[\lambda x_1 + (1-\lambda)x_2] \geq min[\mu_A(x_1), \mu_A(x_2)] \tag{2.17}$$

for all $x_1, x_2 \in R^n$ and all $\lambda \in [0, 1]$.

Proof: First, suppose that A is convex and we prove the truth of (2.17). Let x_1 and x_2 be arbitrary points in R^n and without loss of generality we assume $\mu_A(x_1) \leq \mu_A(x_2)$. If $\mu_A(x_1) = 0$, then (2.17) is trivially true, so we let $\mu_A(x_1) = \alpha > 0$. Since by assumption the α-cut A_α is convex and $x_1, x_2 \in A_\alpha$ (since $\mu_A(x_2) \geq \mu_A(x_1) = \alpha$), we have $\lambda x_1 + (1-\lambda)x_2 \in A_\alpha$ for all $\lambda \in [0, 1]$. Hence, $\mu_A[\lambda x_1 + (1-\lambda)x_2] \geq \alpha = \mu_A(x_1) = min[\mu_A(x_1), \mu_A(x_2)]$.

Conversely, suppose (2.17) is true and we prove that A is convex. Let α be an arbitrary point in $(0, 1]$. If A_α is empty, then it is convex (empty sets are convex by definition). If A_α is nonempty, then there exists $x_1 \in R^n$ such that $\mu_A(x_1) = \alpha$ (by the definition of A_α). Let x_2 be an arbitrary element in A_α, then $\mu_A(x_2) \geq \alpha = \mu_A(x_1)$. Since (2.17) is true by assumption, we have $\mu_A[\lambda x_1 + (1-\lambda)x_2] \geq min[\mu_A(x_1), \mu_A(x_2)] = \mu_A(x_1) = \alpha$ for all $\lambda \in [0, 1]$, which means that $\lambda x_1 + (1-\lambda)x_2 \in A_\alpha$. So A_α is a convex set. Since α is an arbitrary point in $(0, 1]$, the convexity of A_α implies the convexity of A. □

Let A be a fuzzy set in R^n with membership function $\mu_A(x) = \mu_A(x_1, ..., x_n)$ and H be a hyperplane in R^n defined by $H = \{x \in R^n | x_1 = 0\}$ (for notational simplicity, we consider this special case of hyperplane; generalization to general hyperplanes is straightforward). The *projection* of A on H is a fuzzy set A_H in R^{n-1} defined by

$$\mu_{A_H}(x_2, ..., x_n) = \sup_{x_1 \in R} \mu_A(x_1, ..., x_n) \tag{2.18}$$

where $\sup_{x_1 \in R} \mu_A(x_1, ..., x_n)$ denotes the maximum value of the function $\mu_A(x_1, ..., x_n)$ when x_1 takes values in R.

2.3 Operations on Fuzzy Sets

The basic concepts introduced in Sections 2.1 and 2.2 concern only a single fuzzy set. In this section, we study the basic operations on fuzzy sets. In the sequel, we assume that A and B are fuzzy sets defined in the same universe of discourse U.

Definition 2.3. The equality, containment, complement, union, and intersection of two fuzzy sets A and B are defined as follows.

We say A and B are *equal* if and only if $\mu_A(x) = \mu_B(x)$ for all $x \in U$. We say B *contains* A, denoted by $A \subset B$, if and only if $\mu_A(x) \leq \mu_B(x)$ for all $x \in U$. The *complement* of A is a fuzzy set $\bar{A}$ in U whose membership function is defined as

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{2.19}$$

The *union* of A and B is a fuzzy set in U, denoted by $A \cup B$, whose membership function is defined as

$$\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)] \tag{2.20}$$

The *intersection* of A and B is a fuzzy set $A \cap B$ in U with membership function

$$\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)] \tag{2.21}$$

The reader may wonder why we use "max" for union and "min" for intersection; we now give an intuitive explanation. An intuitively appealing way of defining the union is the following: the union of A and B is the smallest fuzzy set containing both A and B. More precisely, if C is any fuzzy set that contains both A and B, then it also contains the union of A and B. To show that this intuitively appealing definition is equivalent to (2.20), we note, first, that $A \cup B$ as defined by (2.20) contains both A and B because $max[\mu_A, \mu_B] \geq \mu_A$ and $max[\mu_A, \mu_B] \geq \mu_B$. Furthermore, if C is any fuzzy set containing both A and B, then $\mu_C \geq \mu_A$ and $\mu_C \geq \mu_B$. Therefore, $\mu_C \geq max[\mu_A, \mu_B] = \mu_{A \cup B}$, which means that $A \cup B$ as defined by (2.20) is the smallest fuzzy set containing both A and B. The intersection as defined by (2.21) can be justified in the same manner.

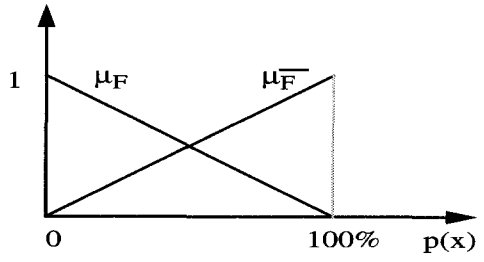


Figure 2.8. The membership functions for $\bar{F}$ and F.

Example 2.5. Consider the two fuzzy sets D and F defined by (2.8) and (2.9) (see also Fig. 2.2). The complement of F, $\bar{F}$, is the fuzzy set defined by

$$\mu_{\bar{F}}(x) = 1 - \mu_F(x) = 1 - p(x) \tag{2.22}$$

which is shown in Fig. 2.8. Comparing (2.22) with (2.9) we see that $\bar{F} = D$. This makes sense because if a car is not a non-US car (which is what the complement of F means intuitively), then it should be a US car; or more accurately, the less a car is a non-US car, the more the car is a US car. The union of F and D is the fuzzy set $F \cup D$ defined by

$$\mu_{F \cup D}(x) = max[\mu_F, \mu_D] = \begin{cases} \mu_F(x) & if \quad 0 \leq p(x) \leq 0.5 \\ \mu_D(x) & if \quad 0.5 \leq p(x) \leq 1 \end{cases} \tag{2.23}$$

which is plotted in Fig. 2.9. The intersection of F and D is the fuzzy set $F \cap D$ defined by

$$\mu_{F \cap D}(x) = min[\mu_F, \mu_D] = \begin{cases} \mu_D(x) & if \quad 0 \leq p(x) \leq 0.5 \\ \mu_F(x) & if \quad 0.5 \leq p(x) \leq 1 \end{cases} \tag{2.24}$$

which is plotted in Fig. 2.10. □

With the operations of complement, union and intersection defined as in (2.19), (2.20) and (2.21), many of the basic identities (not all!) which hold for classical sets can be extended to fuzzy sets. As an example, let us consider the following lemma.

Lemma 2.2. The De Morgan's Laws are true for fuzzy sets. That is, suppose A and B are fuzzy sets, then

$$\overline{A \cup B} = \bar{A} \cap \bar{B} \tag{2.25}$$

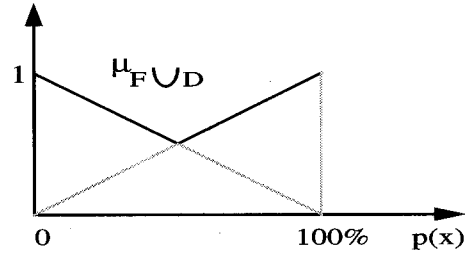


Figure 2.9. The membership function for $F \cup D$, where F and D are defined in Fig. 2.2.

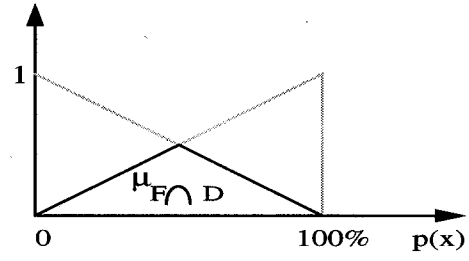


Figure 2.10. The membership function for $F \cap D$, where F and D are defined in Fig. 2.2.

and

$$\overline{A \cap B} = \bar{A} \cup \bar{B} \tag{2.26}$$

Proof: We only prove (2.25); (2.26) can be proven in the same way and is left as an exercise. First, we show that the following identity is true:

$$1 - max[\mu_A, \mu_B] = min[1 - \mu_A, 1 - \mu_B] \tag{2.27}$$

To show this we consider the two possible cases: $\mu_A \geq \mu_B$ and $\mu_A < \mu_B$. If $\mu_A \geq \mu_B$, then $1-\mu_A \leq 1-\mu_B$ and $1-max[\mu_A, \mu_B] = 1-\mu_A = min[1-\mu_A, 1-\mu_B]$, which is (2.27). If $\mu_A < \mu_B$, then $1-\mu_A > 1-\mu_B$ and $1-max[\mu_A, \mu_B] = 1-\mu_B = min[1-\mu_A, 1-\mu_B]$, which is again (2.27). Hence, (2.27) is true. From the definitions (2.19)-(2.21) and the definition of the equality of two fuzzy sets, we see that (2.27) implies (2.25). □

2.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- The definitions of fuzzy set, basic concepts associated with a fuzzy set (support, α-cut, convexity, etc.) and basic operations (complement, union, intersection, etc.) of fuzzy sets.
- The intuitive meaning of membership functions and how to determine intuitively appealing membership functions for specific fuzzy descriptions.
- Performing operations on specific examples of fuzzy sets and proving basic properties concerning fuzzy sets and their operations.

Zadeh's original paper (Zadeh [1965]) is still the best source to learn fuzzy set and related concepts. The paper was extremely well-written and the reader is encouraged to read it. The basic operations and concepts associated with a fuzzy set were also introduced in Zadeh [1965].

2.5 Exercises

Exercise 2.1. Determine reasonable membership functions for "short persons," "tall persons," and "heavy persons."

Exercise 2.2. Model the following expressions as fuzzy sets: (a) hard-working students, (b) top students, and (c) smart students.

Exercise 2.3. Consider the fuzzy sets F, G and H defined in the interval $U = [0, 10]$ by the membership functions

$$\mu_F(x) = \frac{x}{x+2}, \ \mu_G(x) = 2^{-x}, \ \mu_H(x) = \frac{1}{1+10(x-2)^2} \tag{2.28}$$

Determine the mathematical formulas and graphs of membership functions of each of the following fuzzy sets:

(a) $\bar{F}, \bar{G}, \bar{H}$

(b) $F \cup G, F \cup H, G \cup H$

(c) $F \cap G, F \cap H, G \cap H$

(d) $F \cup G \cup H, F \cap G \cap H$

(e) $F \cap \bar{H}, \overline{\bar{G} \cap H}, \overline{F \cup H}$

Exercise 2.4. Determine the α-cuts of the fuzzy sets F, G and H in Exercise 2.3 for: (a) $\alpha = 0.2$, (b) $\alpha = 0.5$, (c) $\alpha = 0.9$, and (d) $\alpha = 1$.

Exercise 2.5. Let fuzzy set A be defined in the closed plane $U = [-1, 1] \times [-3, 3]$ with membership function

$$\mu_A(x_1, x_2) = e^{-(x_1^2 + x_2^2)} \tag{2.29}$$

Determine the projections of A on the hyperplanes $H_1 = \{x \in U | x_1 = 0\}$ and $H_2 = \{x \in U | x_2 = 0\}$, respectively.

Exercise 2.6. Show that the law of the excluded middle, $F \cup \bar{F} = U$, is not true if F is a fuzzy set.

Exercise 2.7. Prove the identity (2.26) in Lemma 2.2.

Exercise 2.8. Show that the intersection of two convex fuzzy sets is also a convex fuzzy set. What about the union?

Chapter 3

Further Operations on Fuzzy Sets

In Chapter 2 we introduced the following basic operators for complement, union, and intersection of fuzzy sets:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{3.1}$$

$$\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)] \tag{3.2}$$

$$\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)] \tag{3.3}$$

We explained that the fuzzy set $A \cup B$ defined by (3.2) is the smallest fuzzy set containing both A and B, and the fuzzy set $A \cap B$ defined by (3.3) is the largest fuzzy set contained by both A and B. Therefore, (3.1)-(3.3) define only one type of operations on fuzzy sets. Other possibilities exist. For example, we may define $A \cup B$ as any fuzzy set containing both A and B (not necessarily the smallest fuzzy set). In this chapter, we study other types of operators for complement, union, and intersection of fuzzy sets.

Why do we need other types of operators? The main reason is that the operators (3.1)-(3.3) may not be satisfactory in some situations. For example, when we take the intersection of two fuzzy sets, we may want the larger fuzzy set to have an impact on the result. But if we use the *min* operator of (3.3), the larger fuzzy set will have no impact. Another reason is that from a theoretical point of view it is interesting to explore what types of operators are possible for fuzzy sets. We know that for nonfuzzy sets only one type of operation is possible for complement, union, or intersection. For fuzzy sets there are other possibilities. But what are they? What are the properties of these new operators? These are the questions we will try to answer in this chapter.

The new operators will be proposed on axiomatic bases. That is, we will start with a few axioms that complement, union, or intersection should satisfy in order to be qualified as these operations. Then, we will list some particular formulas that satisfy these axioms.

3.1 Fuzzy Complement

Let $c : [0,1] \rightarrow [0,1]$ be a mapping that transforms the membership function of fuzzy set A into the membership function of the complement of A, that is,

$$c[\mu_A(x)] = \mu_{\bar{A}}(x) \tag{3.4}$$

In the case of (3.1), $c[\mu_A(x)] = 1 - \mu_A(x)$. In order for the function c to be qualified as a complement, it should satisfy at least the following two requirements:

Axiom c1. $c(0) = 1$ and $c(1) = 0$ (boundary condition).

Axiom c2. For all $a, b \in [0,1]$, if $a < b$, then $c(a) \geq c(b)$ (nonincreasing condition), where (and throughout this chapter) a and b denote membership functions of some fuzzy sets, say, $a = \mu_A(x)$ and $b = \mu_B(x)$.

Axiom c1 shows that if an element belongs to a fuzzy set to degree zero (one), then it should belong to the complement of this fuzzy set to degree one (zero). Axiom c2 requires that an increase in membership value must result in a decrease or no change in membership value for the complement. Clearly, any violation of these two requirements will result in an operator that is unacceptable as complement.

Definition 3.1. Any function $c : [0,1] \rightarrow [0,1]$ that satisfies Axioms c1 and c2 is called a *fuzzy complement.*

One class of fuzzy complements is the *Sugeno class* (Sugeno [1977]) defined by

$$c_\lambda(a) = \frac{1-a}{1+\lambda a} \tag{3.5}$$

where $\lambda \in (-1, \infty)$. For each value of the parameter λ, we obtain a particular fuzzy complement. It is a simple matter to check that the complement defined by (3.5) satisfies Axioms c1 and c2. Fig. 3.1 illustrates this class of fuzzy complements for different values of λ. Note that when $\lambda = 0$ it becomes the basic fuzzy complement (3.1).

Another type of fuzzy complement is the *Yager class* (Yager [1980]) defined by

$$c_w(a) = (1 - a^w)^{1/w} \tag{3.6}$$

where $w \in (0, \infty)$. For each value of w, we obtain a particular fuzzy complement. It is easy to verify that (3.6) satisfies Axioms c1 and c2. Fig. 3.2 illustrates the Yager class of fuzzy complements for different values of w. When $w = 1$, (3.6) becomes (3.1).

3.2 Fuzzy Union—The S-Norms

Let $s : [0,1] \times [0,1] \rightarrow [0,1]$ be a mapping that transforms the membership functions of fuzzy sets A and B into the membership function of the union of A and B, that

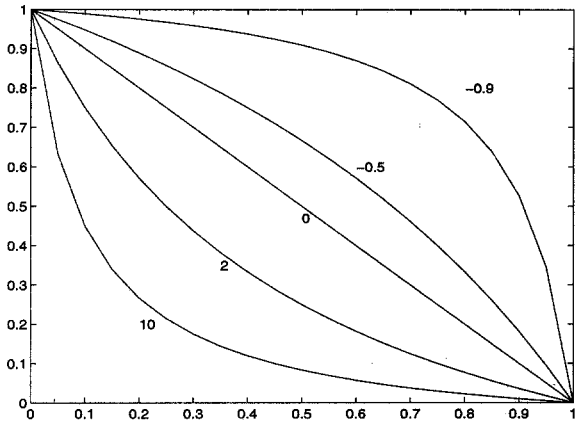


Figure 3.1. Sugeno class of fuzzy complements $c_\lambda(a)$ for different values of λ.

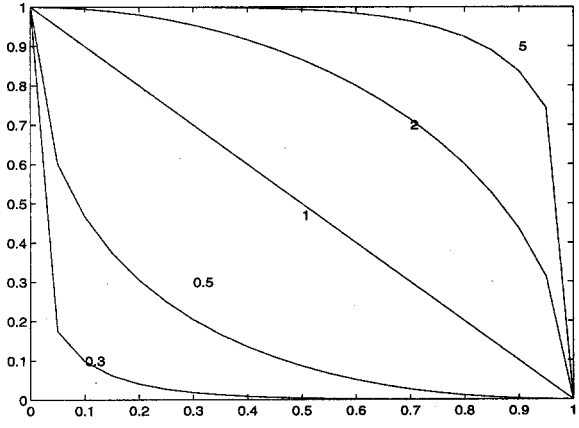


Figure 3.2. Yager class of fuzzy complements $c_w(a)$ for different values of w.

is,

$$s[\mu_A(x), \mu_B(x)] = \mu_{A \cup B}(x) \tag{3.7}$$

In the case of (3.2), $s[\mu_A(x), \mu_B(x)] = max[\mu_A(x), \mu_B(x)]$. In order for the function s to be qualified as an union, it must satisfied at least the following four requirements:

Axiom s1. $s(1,1) = 1, s(0,a) = s(a,0) = a$ (boundary condition).

Axiom s2. $s(a,b) = s(b,a)$ (commutative condition).

Axiom s3. If $a \leq a'$ and $b \leq b'$, then $s(a,b) \leq s(a',b')$ (nondecreasing condition).

Axiom s4. $s(s(a,b),c) = s(a,s(b,c))$ (associative condition).

Axiom s1 indicates what an union function should be in extreme cases. Axiom s2 insures that the order in which the fuzzy sets are combined has no influence on the result. Axiom s3 shows a natural requirement for union: an increase in membership values in the two fuzzy sets should result in an increase in membership value in the union of the two fuzzy sets. Axiom s4 allows us to extend the union operations to more than two fuzzy sets.

Definition 3.2. Any function $s : [0,1] \times [0,1] \rightarrow [0,1]$ that satisfies Axioms s1-s4 is called an *s-norm.*

It is a simple matter to prove that the basic fuzzy union *max* of (3.2) is a s-norm. We now list three particular classes of s-norms:

- Dombi class (Dombi [1982]):

$$s_\lambda(a,b) = \frac{1}{1 + [(\frac{1}{a} - 1)^{-\lambda} + (\frac{1}{b} - 1)^{-\lambda}]^{-1/\lambda}} \tag{3.8}$$

  where the parameter $\lambda \in (0, \infty)$.

- Dubois-Prade class (Dubois and Prade [1980]):

$$s_\alpha(a,b) = \frac{a + b - ab - min(a, b, 1-\alpha)}{max(1-a, 1-b, \alpha)} \tag{3.9}$$

  where the parameter $\alpha \in [0,1]$.

- Yager class (Yager [1980]):

$$s_w(a,b) = min[1, (a^w + b^w)^{1/w}] \tag{3.10}$$

  where the parameter $w \in (0, \infty)$.

With a particular choice of the parameters, (3.8)-(3.10) each defines a particular s-norm. It is straightforward to verify that (3.8)-(3.10) satisfy Axioms s1-s4. These s-norms were obtained by generalizing the union operation for classical sets from different perspectives.

Many other s-norms were proposed in the literature. We now list some of them below:

- Drastic sum:

$$s_{ds}(a,b) = \begin{cases} a \; if \; b = 0 \\ b \; if \; a = 0 \\ 1 \; otherwise \end{cases} \tag{3.11}$$

- Einstein sum:

$$s_{es}(a,b) = \frac{a+b}{1+ab} \tag{3.12}$$

- Algebraic sum:

$$s_{as}(a,b) = a + b - ab \tag{3.13}$$

- Maximum: (3.2)

Why were so many s-norms proposed in the literature? The theoretical reason is that they become identical when the membership values are restricted to zero or one; that is, they are all extensions of nonfuzzy set union. The practical reason is that some s-norms may be more meaningful than others in some applications.

Example 3.1: Consider the fuzzy sets D and F defined in Example 2.1 of Chapter 2 ((2.8) and (2.9)). If we use the Yager s-norm (3.10) for fuzzy union, then the fuzzy set $D \cup F$ is computed as

$$\begin{aligned} \mu_{D \cup F}(x) &= s_w[\mu_D(x), \mu_F(x)] \\ &= min[1, ((p(x))^w + (1 - p(x))^w)^{1/w}] \end{aligned} \tag{3.14}$$

Fig. 3.3 illustrates this $\mu_{D \cup F}(x)$ for $w = 3$. If we use the algebraic sum (3.13) for the fuzzy union, the fuzzy set $D \cup F$ becomes

$$\begin{aligned} \mu_{D \cup F}(x) &= s_{as}[\mu_D(x), \mu_F(x)] \\ &= p(x) + (1 - p(x)) - p(x)(1 - p(x)) \\ &= 1 - p(x) + (p(x))^2 \end{aligned} \tag{3.15}$$

which is plotted in Fig. 3.4. □

Comparing Figs. 3.3 and 3.4 with Fig. 2.9, we see that the Yager s-norm and algebraic sum are larger than the maximum operator. In general, we can show that *maximum* (3.2) is the smallest s-norm and *drastic sum* (3.11) is the largest s-norm.

Theorem 3.1: For any s-norm s, that is, for any function $s : [0,1] \times [0,1] \rightarrow [0,1]$ that satisfies Axioms s1-s4, the following inequality holds:

$$max(a,b) \leq s(a,b) \leq s_{ds}(a,b) \tag{3.16}$$

for any $a, b \in [0,1]$.

Proof: We first prove $max(a,b) \leq s(a,b)$. From the nondecreasing condition Axiom s3 and the boundary condition Axium s1, we obtain

$$s(a,b) \geq s(a,0) = a \tag{3.17}$$

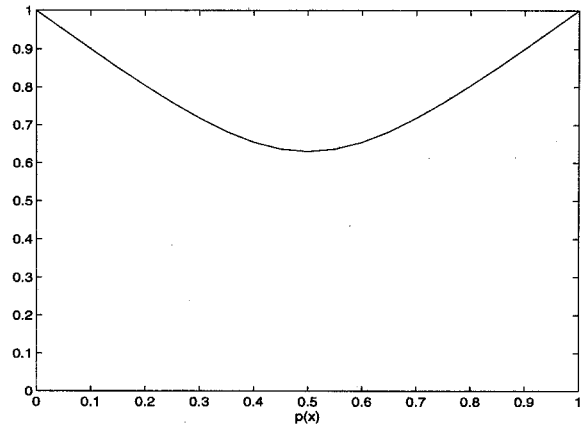


Figure 3.3. Membership function of $D \cup F$ using the Yager s-norm (3.10) with $w = 3$.

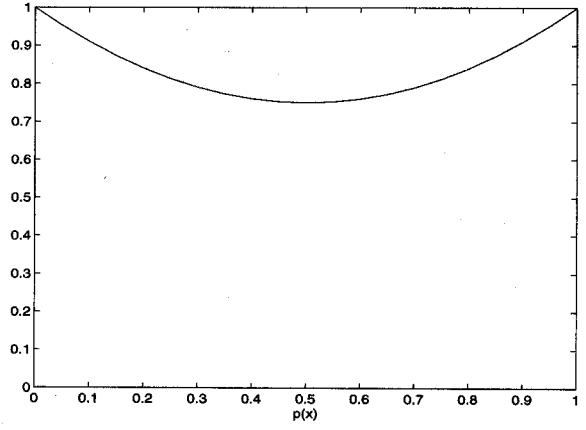


Figure 3.4. Membership function of $D \cup F$ using the algebraic sum (3.13).

Furthermore, the commutative condition Axiom s2 gives

$$s(a, b) = s(b, a) \geq s(b, 0) = b \tag{3.18}$$

Combining (3.17) and (3.18) we have $s(a, b) \geq max(a, b)$.

Next we prove $s(a, b) \leq s_{ds}(a, b)$. If $b = 0$, then from Axiom s1 we have $s(a, b) =$

$s(a,0) = a$, thus $s(a,b) = s_{ds}(a,b)$. By the commutative condition Axiom s2 we have $s(a,b) = s_{ds}(a,b)$ if $a = 0$. If $a \neq 0$ and $b \neq 0$, we have

$$s_{ds}(a,b) = 1 \geq s(a,b) \tag{3.19}$$

Thus $s(a,b) \leq s_{ds}(a,b)$ for all $a, b \in [0,1]$. □

Finally, we prove an interesting property of the Dombi s-norm $s_\lambda(a,b)$ (3.8): $s_\lambda(a,b)$ converges to the basic fuzzy union $max(a,b)$ as the parameter λ goes to infinity and converges to the drastic sum $s_{ds}(a,b)$ as λ goes to zero. Therefore, the Dombi s-norm covers the whole spectrum of s-norms.

Lemma 3.1: Let $s_\lambda(a,b)$ be defined as in (3.8) and $s_{ds}(a,b)$ be defined as in (3.11), then

$$\lim_{\lambda\to\infty} s_\lambda(a,b) = max(a,b) \tag{3.20}$$

$$\lim_{\lambda\to 0} s_\lambda(a,b) = s_{ds}(a,b) \tag{3.21}$$

Proof: We first prove (3.20). If $a = b \neq 0$, then from (3.8) we have $\lim_{\lambda\to\infty} s_\lambda(a,b) = \lim_{\lambda\to\infty}[1/(1+2^{-1/\lambda}(\frac{1}{a}-1))] = a = max(a,b)$. If $a = b = 0$, then $\lim_{\lambda\to\infty} s_\lambda(a,b) = \lim_{\lambda\to\infty} 1/(1+0^{-1/\lambda}) = 0 = max(a,b)$. If $a \neq b$, then without loss of generality (due to Axiom s2) we assume $a < b$. Let $z = [(\frac{1}{a}-1)^{-\lambda} + (\frac{1}{b}-1)^{-\lambda}]^{-1/\lambda}$, then using 1'Hospital's rule, we have

$$\begin{aligned}
\lim_{\lambda\to\infty} ln(z) &= \lim_{\lambda\to\infty} -\frac{ln[(\frac{1}{a}-1)^{-\lambda} + (\frac{1}{b}-1)^{-\lambda}]}{\lambda} \\
&= \lim_{\lambda\to\infty} \frac{(\frac{1}{a}-1)^{-\lambda} ln(\frac{1}{a}-1) + (\frac{1}{b}-1)^{-\lambda} ln(\frac{1}{b}-1)}{(\frac{1}{a}-1)^{-\lambda} + (\frac{1}{b}-1)^{-\lambda}} \\
&= \lim_{\lambda\to\infty} \frac{[(\frac{1}{a}-1)/(\frac{1}{b}-1)]^{-\lambda} ln(\frac{1}{a}-1) + ln(\frac{1}{b}-1)}{[(\frac{1}{a}-1)/(\frac{1}{b}-1)]^{-\lambda} + 1} \\
&= ln(\frac{1}{b}-1)
\end{aligned} \tag{3.22}$$

Hence, $\lim_{\lambda\to\infty} z = \frac{1}{b} - 1$, and

$$\lim_{\lambda\to\infty} s_\lambda(a,b) = \lim_{\lambda\to\infty} \frac{1}{1+z} = b = max(a,b) \tag{3.23}$$

Next we prove (3.21). If $a = 0$ and $b \neq 0$, we have $s_\lambda(a,b) = 1/[1+(\frac{1}{b}-1)^{-\lambda\frac{-1}{\lambda}}] = b = s_{ds}(a,b)$. By commutativity, we have $s_\lambda(a,b) = a = s_{ds}(a,b)$ if $b = 0$ and $a \neq 0$. If $a \neq 0$ and $b \neq 0$, we have $lim_{\lambda\to 0} s_\lambda(a,b) = \lim_{\lambda\to 0} 1/[1+2^{-1/\lambda}] = 1 = s_{ds}(a,b)$. Finally, if $a = b = 0$, we have $\lim_{\lambda\to 0} s_\lambda(a,b) = \lim_{\lambda\to 0} 1/[1+0^{-1/\lambda}] = 0 = s_{ds}(a,b)$. □

Similarly, it can be shown that the Yager s-norm (3.10) converges to the basic fuzzy union $max(a,b)$ as w goes to infinity and converges to the drastic sum $s_{ds}(a,b)$ as w goes to zero; the proof is left as an exercise.

3.3 Fuzzy Intersection—The T-Norms

Let $t : [0,1] \times [0,1] \rightarrow [0,1]$ be a function that transforms the membership functions of fuzzy sets A and B into the membership function of the intersection of A and B, that is,

$$t[\mu_A(x), \mu_B(x)] = \mu_{A \cap B}(x) \tag{3.24}$$

In the case of (3.3), $t[\mu_A(x), \mu_B(x)] = min[\mu_A(x), \mu_B(x)]$. In order for the function t to be qualified as an intersection, it must satisfy at least the following four requirements:

Axiom t1: $t(0,0) = 0; t(a,1) = t(1,a) = a$ (boundary condition).

Axiom t2: $t(a,b) = t(b,a)$ (commutativity).

Axiom t3: If $a \leq a'$ and $b \leq b'$, then $t(a,b) \leq t(a',b')$ (nondecreasing).

Axiom t4: $t[t(a,b),c] = t[a,t(b,c)]$ (associativity).

These axioms can be justified in the same way as for Axioms s1-s4.

Definition 3.3. Any function $t : [0,1] \times [0,1] \rightarrow [0,1]$ that satisfies Axioms t1-t4 is called a *t-norm*.

We can verify that the basic fuzzy intersection *min* of (3.3) is a t-norm. For any t-norm, there is an s-norm associated with it and vice versa. Hence, associated with the s-norms of Dombi, Dubois-Prade and Yager classes ((3.8)-(3.10)), there are t-norms of Dombi, Dubois-Prade and Yager classes, which are defined as follows:

- Dombi class (Dombi [1982]):

$$t_\lambda(a,b) = \frac{1}{1 + [(\frac{1}{a} - 1)^\lambda + (\frac{1}{b} - 1)^\lambda]^{1/\lambda}} \tag{3.25}$$

  where $\lambda \in (0, \infty)$.

- Dubois-Prade class (Dubois and Prade [1980]):

$$t_\alpha(a,b) = \frac{ab}{max(a,b,\alpha)} \tag{3.26}$$

  where $\alpha \in [0,1]$.

- Yager class (Yager [1980]):

$$t_w(a,b) = 1 - min[1, ((1-a)^w + (1-b)^w)^{1/w}] \tag{3.27}$$

  where $w \in (0, \infty)$.

With a particular choice of the parameters, (3.25)-(3.27) each defines a particular t-norm. We can verify that (3.25)-(3.27) satisfy Axiom t1-t4. Associated with the particular s-norms (3.11)-(3.13) and (3.2), there are t-norms that are listed below:

- Drastic product:

$$t_{dp}(a,b) = \begin{cases} a \; if \; b = 1 \\ b \; if \; a = 1 \\ 0 \; otherwise \end{cases} \tag{3.28}$$

- Einstein product:

$$t_{ep}(a,b) = \frac{ab}{2-(a+b-ab)} \tag{3.29}$$

- Algebraic product:

$$t_{ap}(a,b) = ab \tag{3.30}$$

- Minimum: (3.3)

Example 3.2: Consider the fuzzy sets D and F defined in Example 2.1 of Chapter 2. If we use the Yager t-norm (3.27) for fuzzy intersection, then $D \cap F$ is obtained as

$$\begin{aligned} \mu_{D\cap F}(x) &= t_w[\mu_D(x), \mu_F(x)] \\ &= 1 - min[1, ((1-p(x))^w + (p(x))^w)^{1/w}] \end{aligned} \tag{3.31}$$

Fig. 3.5 shows this $\mu_{D\cap F}(x)$ for $w = 3$. If we use the algebraic product (3.30) for fuzzy intersection, the fuzzy set $D \cap F$ becomes

$$\begin{aligned} \mu_{D\cap F}(x) &= t_{ap}[\mu_D(x), \mu_F(x)] \\ &= p(x)(1-p(x)) \end{aligned} \tag{3.32}$$

which is plotted in Fig. 3.6. □

Comparing Figs. 3.5 and 3.6 with Fig. 2.10, we see that the Yager t-norm and algebraic product are smaller than the minimum operator. In general, we can show that minimum is the largest t-norm and drastic product is the smallest t-norm.

Theorem 3.2: For any t-norm t, that is, for any function $t : [0,1] \times [0,1] \rightarrow [0,1]$ that satisfies Axioms t1-t4, the following inequality holds:

$$t_{dp}(a,b) \leq t(a,b) \leq min(a,b) \tag{3.33}$$

for any $a, b \in [0,1]$.

The proof of this theorem is very similar to that of Theorem 3.1 and is left as an exercise. Similar to Lemma 3.1, we can show that the Dombi t-norm $t_\lambda(a,b)$ of (3.25) converges to the basic fuzzy intersection $min(a,b)$ as λ goes to infinity and converges to the drastic product $t_{dp}(a,b)$ as λ goes to zero. Hence, the Dombi t-norm covers the whole range of t-norms.

Lemma 3.2: Let $t_\lambda(a,b)$ be defined as in (3.25), then

$$\lim_{\lambda \rightarrow \infty} t_\lambda(a,b) = min(a,b) \tag{3.34}$$

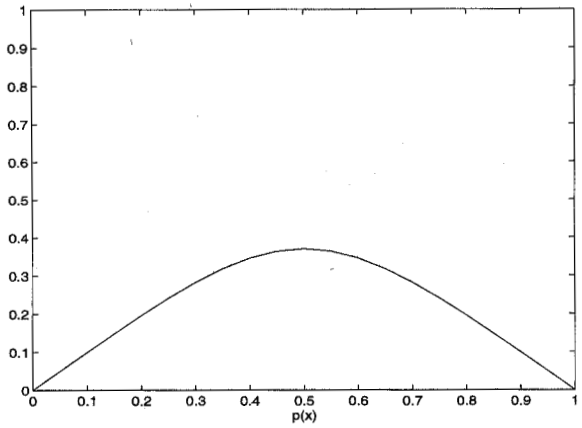


Figure 3.5. Membership function of $D \cap F$ using the Yager t-norm (3.27) with $w = 3$.

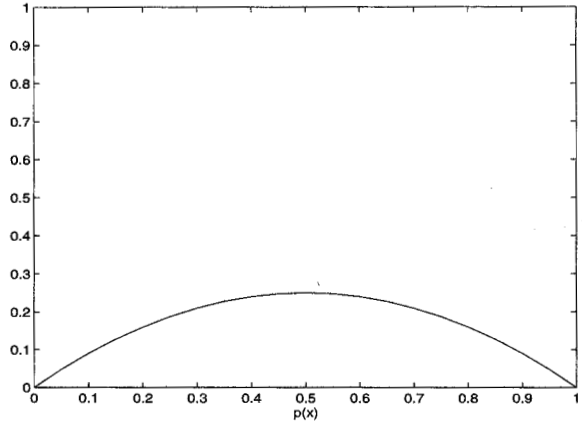


Figure 3.6. Membership function of $D \cap F$ using the algebraic product (3.30).

and

$$\lim_{\lambda \to 0} t_\lambda(a, b) = t_{dp}(a, b) \tag{3.35}$$

This lemma can be proven in a similar way as for Lemma 3.1.

Comparing (3.8)-(3.13) with (3.25)-(3.30), respectively, we see that for each s-

norm there is a t-norm associated with it. But what does this "associated" mean? It means that there exists a fuzzy complement such that the three together satisfy the DeMorgan's Law. Specifically, we say the s-norm $s(a,b)$, t-norm $t(a,b)$ and fuzzy complement $c(a)$ form *an associated class* if

$$c[s(a,b)] = t[c(a),c(b)] \tag{3.36}$$

Example 3.3: The Yager s-norm $s_w(a,b)$ of (3.10), Yager t-norm $t_w(a,b)$ of (3.27), and the basic fuzzy complement (3.1) form an associated class. To show this, we have from (3.1) and (3.10) that

$$c[s_w(a,b)] = 1 - min[1,(a^w + b^w)^{1/w}] \tag{3.37}$$

where $c(a)$ denotes the basic fuzzy complement (3.1). On the other hand, we have from (3.1) and (3.27) that

$$t_w[c(a),c(b)] = 1 - min[1,((1-1+a)^w + (1-1+b)^w)^{1/w}] \tag{3.38}$$

From (3.37) and (3.38) we obtain (3.36). □

Example 3.4: The algebraic sum (3.13), algebraic product (3.30), and the basic fuzzy complement (3.1) form an associated class. To show this, we have from (3.1) and (3.13) that

$$c[s_{as}(a,b)] = 1 - a - b + ab \tag{3.39}$$

On the other hand, from (3.1) and (3.30) we have

$$t_{ap}[c(a),c(b)] = (1-a)(1-b) = 1 - a - b + ab \tag{3.40}$$

Hence, they satisfy the DeMorgan's Law (3.36). □

3.4 Averaging Operators

From Theorem 3.1 we see that for any membership values $a = \mu_A(x)$ and $b = \mu_B(x)$ of arbitrary fuzzy sets A and B, the membership value of their union $A \cup B$ (defined by any s-norm) lies in the interval $[max(a,b), s_{ds}(a,b)]$. Similarly, from Theorem 3.2 we have that the membership value of the intersection $A \cap B$ (defined by any t-norm) lies in the interval $[t_{dp}(a,b), min(a,b)]$. See Fig.3.7. Therefore, the union and intersection operators cannot cover the interval between $min(a,b)$ and $max(a,b)$. The operators that cover the interval $[min(a,b), max(a,b)]$ are called *averaging operators*. Similar to the s-norms and t-norms, an averaging operator, denoted by v, is a function from $[0,1] \times [0,1]$ to $[0,1]$.

Many averaging operators were proposed in the literature. Here we list four of them:

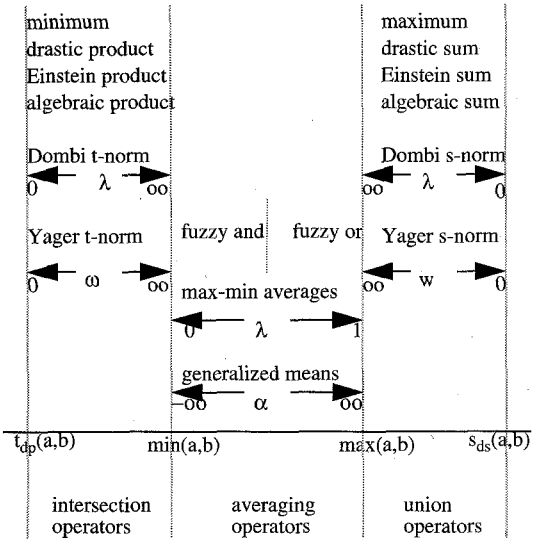


Figure 3.7. The full scope of fuzzy aggregation operators.

- Max-min averages:

$$v_\lambda(a,b) = \lambda max(a,b) + (1-\lambda)min(a,b) \tag{3.41}$$

where $\lambda \in [0,1]$.

- Generalized means:

$$v_\alpha(a,b) = (\frac{a^\alpha + b^\alpha}{2})^{1/\alpha} \tag{3.42}$$

where $\alpha \in R$ $(\alpha \neq 0)$.

- "Fuzzy and":

$$v_p(a,b) = pmin(a,b) + \frac{(1-p)(a+b)}{2} \tag{3.43}$$

where $p \in [0,1]$.

- "Fuzzy or":

$$v_\gamma(a,b) = \gamma max(a,b) + \frac{(1-\gamma)(a+b)}{2} \tag{3.44}$$

where $\gamma \in [0,1]$.

Clearly, the max-min averages cover the whole interval $[min(a,b), max(a,b)]$ as the parameter λ changes from 0 to 1. The "fuzzy and" covers the range from $min(a,b)$ to $(a+b)/2$, and the "fuzzy or" covers the range from $(a+b)/2$ to $max(a,b)$. It also can be shown that the generalized means cover the whole range from $min(a,b)$ to $max(a,b)$ as α changes from $-\infty$ to ∞.

3.5 Summary and Further Readings

In this chapter we have demonstrated the following:

- The axiomatic definitions of fuzzy complements, s-norms (fuzzy unions), and t-norms (fuzzy intersections).
- Some specific classes of fuzzy complements, s-norms, t-norms, and averaging operators, and their properties.
- How to prove various properties of some particular fuzzy complements, s-norms, t-norms, and averaging operators.

The materials in this chapter were extracted from Klir and Yuan [1995] where more details on the operators can be found. Dubois and Prade [1985] provided a very good review of fuzzy union, fuzzy intersection, and averaging operators.

3.6 Exercises

Exercise 3.1. The *equilibrium* of a fuzzy complement c is defined as $a \in [0,1]$ such that $c(a) = a$.

(a) Determine the equilibrium of the Yager fuzzy complement (3.6).

(b) Prove that every fuzzy complement has at most one equilibrium.

(c) Prove that a continuous fuzzy complement has a unique equilibrium.

Exercise 3.2. Show that the Yager s-norm (3.10) converges to the basic fuzzy union (3.2) as w goes to infinity and converges to the drastic sum (3.11) as w goes to zero.

Exercise 3.3. Let the fuzzy sets F and G be defined as in Exercise 2.3.

(a) Determine the membership functions for $F \cup G$ and $F \cap G$ using the Yager s-norm (3.10) and t-norm (3.27) with $w = 2$.

(b) Using (3.1) as fuzzy complement, algebraic sum (3.13) as fuzzy union, and algebraic product (3.30) as fuzzy intersection, compute the membership functions for $F \cap \bar{G}$, $\overline{\bar{F} \cap G}$, and $\overline{F \cup G}$.

Exercise 3.4. Prove Theorem 3.2.

Exercise 3.5. A fuzzy complement c is said to be *involutive* if $c[c(a)] = a$ for all $a \in [0,1]$.

(a)Show that the Sugeno fuzzy complement (3.5) and the Yager fuzzy complement (3.6) are involutive.

(b) Let c be an involutive fuzzy complement and t be any t-norm. Show that the operator $u : [0,1] \times [0,1] \to [0,1]$ defined by

$$u(a,b) = c[t(c(a), c(b))] \tag{3.45}$$

is an s-norm.

(c) Prove that the c, t, and u in (b) form an associated class.

Exercise 3.6. Determine s-norm $s_x(a,b)$ such that $s_x(a,b)$, the minimum t-norm (3.3), and the Yager complement (3.6) with $w = 2$ form an associated class.

Exercise 3.7. Prove that the following triples form an associated class with respect to any fuzzy complement c: (a) (min, max, c), and (b) (t_{dp}, s_{ds}, c).

Exercise 3.8. Prove that the generalized means (3.42) become *min* and *max* operators as $\alpha \to -\infty$ and $\alpha \to \infty$, respectively.

Chapter 4

Fuzzy Relations and the Extension Principle

4.1 From Classical Relations to Fuzzy Relations

4.1.1 Relations

Let U and V be two arbitrary classical (nonfuzzy, crisp) sets. The *Cartesian product* of U and V, denoted by $U \times V$, is the nonfuzzy set of all ordered pairs (u, v) such that $u \in U$ and $v \in V$; that is,

$$U \times V = \{(u, v) | u \in U \text{ and } v \in V\} \tag{4.1}$$

Note that the order in which U and V appears is important; that is, if $U \neq V$, then $U \times V \neq V \times U$. In general, the Cartesian product of arbitrary n nonfuzzy sets $U_1, U_2, ..., U_n$, denoted by $U_1 \times U_2 \times \cdots \times U_n$, is the nonfuzzy set of all n-tuples $(u_1, u_2, ..., u_n)$ such that $u_i \in U_i$ for $i \in \{1, 2, ..., \}$; that is,

$$U_1 \times U_2 \times \cdots \times U_n = \{(u_1, u_2, ..., u_n) | u_1 \in U_1,\ u_2 \in U_2, \cdots,\ u_n \in U_n\} \tag{4.2}$$

A (nonfuzzy) *relation* among (nonfuzzy) sets $U_1, U_2, ..., U_n$ is a subset of the Cartesian product $U_1 \times U_2 \times \cdots \times U_n$; that is, if we use $Q(U_1, U_2, ..., U_n)$ to denote a relation among $U_1, U_2, ..., U_n$, then

$$Q(U_1, U_2, ..., U_n) \subset U_1 \times U_2 \times \cdots \times U_n \tag{4.3}$$

As a special case, a *binary relation* between (nonfuzzy) sets U and V is a subset of the Cartesian product $U \times V$.

Example 4.1. Let $U = \{1, 2, 3\}$ and $V = \{2, 3, 4\}$. Then the cartesian product of U and V is the set $U \times V = \{(1,2), (1,3), (1,4), (2,2), (2,3), (2,4), (3,2), (3,3), (3,4)\}$. A relation between U and V is a subset of $U \times V$. For example, let $Q(U, V)$ be a relation named "the first element is no smaller than the second element," then

$$Q(U, V) = \{(2,2), (3,2), (3,3)\} \tag{4.4}$$

□

Because a relation is itself a set, all of the basic set operations can be applied to it without modification. Also, we can use the following membership function to represent a relation:

$$\mu_Q(u_1, u_2, ..., u_n) = \begin{cases} 1 \; if \; (u_1, u_2, ..., u_n) \in Q(U_1, U_2, ..., U_n) \\ 0 \; otherwise \end{cases} \tag{4.5}$$

For binary relation $Q(U, V)$ defined over $U \times V$ which contains finite elements, we often collect the values of the membership function μ_Q into a *relational matrix*; see the following example.

Example 4.1 (Cont'd). The relation $Q(U, V)$ of (4.4) can be represented by the following relational matrix:

$$\begin{array}{cc|ccc} & & & V & \\ & & 2 & 3 & 4 \\ & 1 & 0 & 0 & 0 \\ U & 2 & 1 & 0 & 0 \\ & 3 & 1 & 1 & 0 \end{array} \tag{4.6}$$

□

A classical relation represents a crisp relationship among sets, that is, either there is such a relationship or not. For certain relationships, however, it is difficult to give a zero-one assessment; see the following example.

Example 4.2. Let $U = \{SanFrancisco, HongKong, Tokyo\}$ and $V = \{Boston, HongKong\}$. We want to define the relational concept "very far" between these two sets of cities. Clearly, classical relations are not useful because the concept "very far" is not well-defined in the framework of classical sets and relations. However, "very far" does mean something and we should find a numerical system to characterize it. If we use a number in the interval $[0, 1]$ to represent the degree of "very far," then the concept "very far" may be represented by the following (fuzzy) relational matrix:

$$\begin{array}{cc|cc} & & & V & \\ & & Boston & HK \\ & SF & 0.3 & 0.9 \\ U & HK & 1 & 0 \\ & Tokyo & 0.95 & 0.1 \end{array} \tag{4.7}$$

□

Example 4.2 shows that we need to generalize the concept of classical relation in order to formulate more relationships in the real world. The concept of fuzzy relation was thus introduced.

Definition 4.1. A *fuzzy relation* is a fuzzy set defined in the Cartesian product of crisp sets $U_1, U_2, ..., U_n$. With the representation scheme (2.5), a fuzzy relation

Q in $U_1 \times U_2 \times \cdots \times U_n$ is defined as the fuzzy set

$$Q = \{((u_1, u_2, ..., u_n), \mu_Q(u_1, u_2, ..., u_n)) | (u_1, u_2, ..., u_n) \in U_1 \times U_2 \times \cdots \times U_n\} \quad (4.8)$$

where $\mu_Q : U_1 \times U_2 \times \cdots \times U_n \to [0, 1]$.

As a special case, a *binary fuzzy relation* is a fuzzy set defined in the Cartesian product of two crisp sets. A binary relation on a finite Cartesian product is usually represented by a *fuzzy relational matrix*, that is, a matrix whose elements are the membership values of the corresponding pairs belonging to the fuzzy relation. For example, (4.7) is a fuzzy relational matrix representing the fuzzy relation named "very far" between the two groups of cities.

Example 4.3. Let U and V be the set of real numbers, that is, $U = V = R$. A fuzzy relation "x is approximately equal to y," denoted by AE, may be defined by the membership function

$$\mu_{AE}(x, y) = e^{-(x-y)^2} \quad (4.9)$$

Similarly, a fuzzy relation "x is much larger than y," denoted by ML, may be defined by the membership function

$$\mu_{ML}(x, y) = \frac{1}{1 + e^{-(x-y)}} \quad (4.10)$$

Of course, other membership functions may be used to represent these fuzzy relations. □

4.1.2 Projections and Cylindric Extensions

Because a crisp relation is defined in the product space of two or more sets, the concepts of projection and cylindric extension were proposed. For example, consider the set $A = \{(x, y) \in R^2 | (x-1)^2 + (y-1)^2 \leq 1\}$ which is a relation in $U \times V = R^2$. Then the projection of A on U is $A_1 = [0, 1] \subset U$, and the projection of A on V is $A_2 = [0, 1] \subset V$; see Fig. 4.1. The cylindric extension of A_1 to $U \times V = R^2$ is $A_{1E} = [0, 1] \times (-\infty, \infty) \subset R^2$. These concepts can be extended to fuzzy relations.

Definition 4.2. Let Q be a fuzzy relation in $U_1 \times \cdots \times U_n$ and $\{i_1, ..., i_k\}$ be a subsequence of $\{1, 2, ..., n\}$, then the *projection* of Q on $U_{i_1} \times \cdots \times U_{i_k}$ is a fuzzy relation Q_P in $U_{i_1} \times \cdots \times U_{i_k}$ defined by the membership function

$$\mu_{Q_P}(u_{i_1}, ..., u_{i_k}) = \max_{u_{j1} \in U_{j1}, \cdots, u_{j(n-k)} \in U_{j(n-k)}} \mu_Q(u_1, ..., u_n) \quad (4.11)$$

where $\{u_{j1}, ..., u_{j(n-k)}\}$ is the complement of $\{u_{i_1}, ..., u_{i_k}\}$ with respect to $\{u_1, ..., u_n\}$.

As a special case, if Q is a binary fuzzy relation in $U \times V$, then the projection of Q on U, denoted by Q_1, is a fuzzy set in U defined by

$$\mu_{Q_1}(x) = \max_{y \in V} \mu_Q(x, y) \quad (4.12)$$

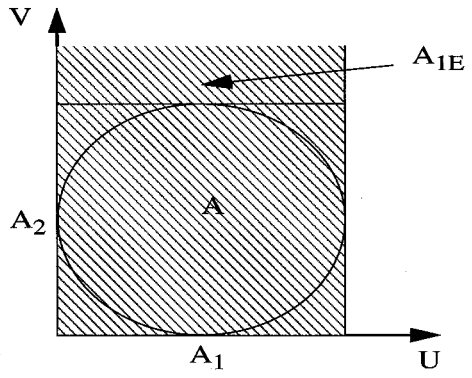


Figure 4.1. Projections and cylindric extensions of a relation.

Note that (4.12) is still valid if Q is a crisp relation. For example, if Q is the crisp relation A in Fig. 4.1, then its projection Q_1 defined by (4.12) is equal to the A_1 in Fig. 4.1. Hence, the projection of fuzzy relation defined by (4.11) is a natural extension of the projection of crisp relation.

Example 4.4. According to (4.12), the projection of fuzzy relation (4.7) on U and V are the fuzzy sets

$$Q_1 = 0.9/SF + 1/HK + 0.95/Tokyo \tag{4.13}$$

and

$$Q_2 = 1/Boston + 0.9/HK \tag{4.14}$$

respectively. Similarly, the projections of AE defined by (4.9) on U and V are the fuzzy sets

$$AE_1 = \int_U \max_{y \in V} e^{-(x-y)^2}/x = \int_U 1/x \tag{4.15}$$

and

$$AE_2 = \int_V \max_{x \in U} e^{-(x-y)^2}/y = \int_V 1/y \tag{4.16}$$

respectively. Note that AE_1 equals the crisp set U and AE_2 equals the crisp set V. □

The projection constrains a fuzzy relation to a subspace; conversely, the cylindric extension extends a fuzzy relation (or fuzzy set) from a subspace to the whole space. Formally, we have the following definition.

Definition 4.3. Let Q_P be a fuzzy relation in $U_{i_1} \times \cdots \times U_{i_k}$ and $\{i_1, ..., i_k\}$ is a subsequence of $\{1, 2, ..., n\}$, then the *cylindric extension* of Q_P to $U_1 \times \cdots \times U_n$ is a fuzzy relation Q_{PE} in $U_1 \times \cdots \times U_n$ defined by

$$\mu_{Q_{PE}}(u_1, ..., u_n) = \mu_{Q_P}(u_{i_1}, ..., u_{i_k}) \tag{4.17}$$

As a special case, if Q_1 is a fuzzy set in U, then the cylindric extension of Q_1 to $U \times V$ is a fuzzy relation Q_{1E} in $U \times V$ defined by

$$\mu_{Q_{1E}}(x, y) = \mu_{Q_1}(x) \tag{4.18}$$

The definition (4.17) is also valid for crisp relations; check Fig. 4.1 for an example.

Example 4.5. Consider the projections Q_1 and Q_2 in Example 4.4 ((4.13) and (4.14)). According to (4.18), their cylindric extensions to $U \times V$ are

$$\begin{aligned} Q_{1E} = {} & 0.9/(SF, Boston) + 0.9/(SF, HK) + 1/(HK, Boston) \\ & + 1/(HK, HK) + 0.95/(Tokyo, Boston) \\ & + 0.95/(Tokyo, HK) \end{aligned} \tag{4.19}$$

and

$$\begin{aligned} Q_{2E} = {} & 1/(SF, Boston) + 1/(HK, Boston) + 1/(Tokyo, Boston) \\ & + 0.9/(SF, HK) + 0.9/(HK, HK) + 0.9/(Tokyo, HK) \end{aligned} \tag{4.20}$$

Similarly, the cylindric extensions of AE_1 and AE_2 in (4.15) and (4.16) to $U \times V$ are

$$AE_{1E} = \int_{U \times V} 1/(x, y) = U \times V \tag{4.21}$$

and

$$AE_{2E} = \int_{U \times V} 1/(x, y) = U \times V \tag{4.22}$$

□

From Examples 4.4 and 4.5 we see that when we take the projection of a fuzzy relation and then cylindrically extend it, we obtain a fuzzy relation that is larger than the original one. To characterize this property formally, we first introduce the concept of Cartesian product of fuzzy sets. Let $A_1, ..., A_n$ be fuzzy sets in $U_1, ..., U_n$, respectively. The *Cartesian product* of $A_1, ..., A_n$, denoted by $A_1 \times \cdots \times A_n$, is a fuzzy relation in $U_1 \times \cdots \times U_n$ whose membership function is defined as

$$\mu_{A_1 \times \cdots \times A_n}(u_1, ..., u_n) = \mu_{A_1}(u_1) \star \cdots \star \mu_{A_n}(u_n) \tag{4.23}$$

where $\star$ represents any t-norm operator.

Lemma 4.1. If Q is a fuzzy relation in $U_1 \times \cdots \times U_n$ and $Q_1, ..., Q_n$ are its projections on $U_1, ..., U_n$, respectively, then (see Fig. 4.2 for illustration)

$$Q \subset Q_1 \times \cdots \times Q_n \tag{4.24}$$

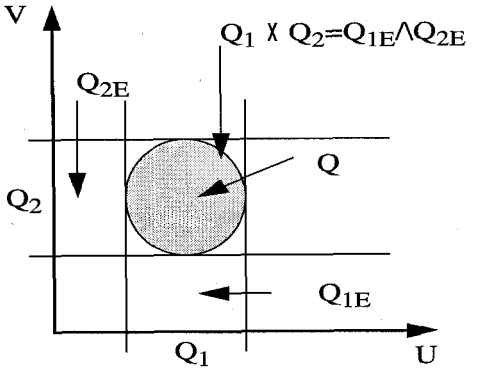


Figure 4.2. Relation between the Cartesian product and intersection of cylindric sets.

where we use *min* for the t-norm in the definition (4.23) of $Q_1 \times \cdots \times Q_n$.

Proof: Substituting (4.11) into (4.17), we have

$$\mu_{Q_{PE}}(u_1, ..., u_n) = \max_{u_{j1} \in U_{j1}, ..., u_{j(n-k)} \in U_{j(n-k)}} \mu_Q(u_1, ..., u_n) \tag{4.25}$$

Hence,

$$Q \subset Q_{iE} \tag{4.26}$$

for all $i = 1, 2, ..., n$, where Q_{iE} is the cylindric extension of Q_i to $U_1 \times \cdots \times U_n$. Therefore, if we use *min* for intersection, we have

$$Q \subset Q_{1E} \cap \cdots \cap Q_{nE} = Q_1 \times \cdots \times Q_n \tag{4.27}$$

□

4.2 Compositions of Fuzzy Relations

Let $P(U, V)$ and $Q(V, W)$ be two crisp binary relations that share a common set V. The *composition* of P and Q, denoted by $P \circ Q$, is defined as a relation in $U \times W$ such that $(x, z) \in P \circ Q$ if and only if there exists at least one $y \in V$ such that $(x, y) \in P$ and $(y, z) \in Q$. Using the membership function representation of relations (see (4.5)), we have an equivalent definition for composition that is given in the following lemma.

Lemma 4.2. $P \circ Q$ is the composition of $P(U,V)$ and $Q(V,W)$ if and only if

$$\mu_{P\circ Q}(x,z) = \max_{y\in V} t[\mu_P(x,y), \mu_Q(y,z)] \tag{4.28}$$

for any $(x,z) \in U \times W$, where t is any t-norm.

Proof: We first show that if $P \circ Q$ is the composition according to the definition, then (4.28) is true. If $P \circ Q$ is the composition, then $(x,z) \in P \circ Q$ implies that there exists $y \in V$ such that $\mu_P(x,y) = 1$ and $\mu_Q(y,z) = 1$. Hence, $\mu_{P\circ Q}(x,z) = 1 = \max_{y\in V} t[\mu_P(x,y), \mu_Q(y,z)]$, that is, (4.28) is true. If $(x,z) \notin P \circ Q$, then for any $y \in V$ either $\mu_P(x,y) = 0$ or $\mu_Q(y,z) = 0$. Hence, $\mu_{P\circ Q}(x,z) = 0 = \max_{y\in V} t[\mu_P(x,y), \mu_Q(y,z)]$. Therefore, (4.28) is true for any $(x,z) \in U \times W$.

Conversely, if (4.28) is true, then $(x,z) \in P \circ Q$ implies $\max_{y\in V} t[\mu_P(x,y), \mu_Q(y,z)] = 1$, which means that there exists at least one $y \in V$ such that $\mu_P(x,y) = \mu_Q(y,z) = 1$ (see Axiom t1 in Section 3.3); this is the definition. For $(x,z) \notin P \circ Q$, we have from (4.28) that $\max_{y\in V} t[\mu_P(x,y), \mu_Q(y,z)] = 0$, which means that there is no $y \in V$ such that $\mu_P(x,y) = \mu_Q(y,z) = 1$. Therefore, (4.28) implies that $P \circ Q$ is the composition according to the definition. □

Now we generalize the concept of composition to fuzzy relations. From Lemma 4.2 we see that if we use (4.28) to define composition of fuzzy relations (suppose P and Q are fuzzy relatioins), then the definition is valid for the special case where P and Q are crisp relations. Therefore, we give the following definition.

Definition 4.4. The *composition of fuzzy relations* $P(U,V)$ and $Q(V,W)$, denoted by $P \circ Q$, is defined as a fuzzy relation in $U \times W$ whose membership function is given by (4.28).

Because the t-norm in (4.28) can take a variety of formulas, for each t-norm we obtain a particular composition. The two most commonly used compositions in the literature are the so-called *max-min* composition and *max-product* composition, which are defined as follows:

- The *max-min composition* of fuzzy relations $P(U,V)$ and $Q(V,W)$ is a fuzzy relation $P \circ Q$ in $U \times W$ defined by the membership function

$$\mu_{P\circ Q}(x,z) = \max_{y\in V} min[\mu_P(x,y), \mu_Q(y,z)] \tag{4.29}$$

  where $(x,z) \in U \times W$.

- The *max-product composition* of fuzzy relations $P(U,V)$ and $Q(V,W)$ is a fuzzy relation $P \circ Q$ in $U \times W$ defined by the membership function

$$\mu_{P\circ Q}(x,z) = \max_{y\in V} [\mu_P(x,y)\mu_Q(y,z)] \tag{4.30}$$

  where $(x,z) \in U \times W$.

We see that the max-min and max-product compositions use minimum and algebraic product for the t-norm in the definition (4.28), respectively. We now consider two examples for how to compute the compositions.

Example 4.6. Let U and V be defined as in Example 4.2 and $W = \{New\ York\ City, Beijing\}$. Let $P(U,V)$ denote the fuzzy relation "very far" defined by (4.7). Define the fuzzy relation "very near" in $V \times W$, denoted by $Q(V,W)$, by the relational matrix

$$\begin{array}{cccc} & & W & \\ & & NYC & Beijing \\ V & Boston & 0.95 & 0.1 \\ & HK & 0.1 & 0.9 \end{array} \tag{4.31}$$

Using the notation (2.7), we can write P and Q as

$$\begin{aligned} P = {} & 0.3/(SF, Boston) + 0.9/(SF, HK) + 1/(HK, Boston) \\ & +0/(HK, HK) + 0.95/(Tokyo, Boston) + 0.1/(Tokyo, HK) \end{aligned} \tag{4.32}$$

$$\begin{aligned} Q = {} & 0.95/(Boston, NYC) + 0.1/(Boston, Beijing) + 0.1/(HK, NYC) \\ & +0.9/(HK, Beijing) \end{aligned} \tag{4.33}$$

We now compute the max-min and max-product compositions of P and Q. First, we note that $U \times W$ contains six elements: (SF,NYC), (SF,Beijing), (HK,NYC), (HK,Beijing), (Tokyo,NYC) and (Tokyo,Beijing). Thus, our task is to determine the membership values of $\mu_{P \circ Q}$ at these six elements. Using the max-min composition (4.29), we have

$$\begin{aligned} \mu_{P \circ Q}(SF, NYC) &= max\{min[\mu_P(SF, Boston), \mu_Q(Boston, NYC)], \\ &\quad min[\mu_P(SF, HK), \mu_Q(HK, NYC)]\} \\ &= max[min(0.3, 0.95), min(0.9, 0.1)] \\ &= 0.3 \end{aligned} \tag{4.34}$$

Similarly, we have

$$\begin{aligned} \mu_{P \circ Q}(SF, Beijing) &= max\{min[\mu_P(SF, Boston), \mu_Q(Boston, Beijing)], \\ &\quad min[\mu_P(SF, HK), \mu_Q(HK, Beijing)]\} \\ &= max[min(0.3, 0.1), min(0.9, 0.9)] \\ &= 0.9 \end{aligned} \tag{4.35}$$

The final $P \circ Q$ is

$$\begin{aligned} P \circ Q = {} & 0.3/(SF, NYC) + 0.9/(SF, Beijing) + 0.95/(HK, NYC) \\ & +0.1/(HK, Beijing) + 0.95/(Tokyo, NYC) \\ & +0.1/(Tokyo, Beijing) \end{aligned} \tag{4.36}$$

If we use the max-product composition (4.30), then following the same procedure as above (replacing *min* by *product*), we obtain

$$\begin{aligned} P \circ Q = & 0.285/(SF, NYC) + 0.81/(SF, Beijing) + 0.95/(HK, NYC) \\ & +0.1/(HK, Beijing) + 0.9025/(Tokyo, NYC) \\ & +0.095/(Tokyo, Beijing) \end{aligned} \tag{4.37}$$

□

From (4.36), (4.37) and the relational matrices (4.7) and (4.31), we see that a simpler way to compute $P \circ Q$ is to use relational matrices and matrix product. Specifically, let P and Q be the relational matrices for the fuzzy relations $P(U, V)$ and $Q(V, W)$, respectively. Then, the relational matrix for the fuzzy composition $P \circ Q$ can be computed according to the following method:

- For max-min composition, write out each element in the matrix product PQ, but treat each multiplication as a *min* operation and each addition as a *max* operation.
- For max-product composition, write out each element in the matrix product PQ, but treat each addition as a *max* operation.

We now check that (4.36) and (4.37) can be obtained by this method. Specifically, we have

$$\begin{pmatrix} 0.3 & 0.9 \\ 1 & 0 \\ 0.95 & 0.1 \end{pmatrix} \circ \begin{pmatrix} 0.95 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} = \begin{pmatrix} 0.3 & 0.9 \\ 0.95 & 0.1 \\ 0.95 & 0.1 \end{pmatrix} \tag{4.38}$$

for max-min composition, and

$$\begin{pmatrix} 0.3 & 0.9 \\ 1 & 0 \\ 0.95 & 0.1 \end{pmatrix} \circ \begin{pmatrix} 0.95 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} = \begin{pmatrix} 0.285 & 0.81 \\ 0.95 & 0.1 \\ 0.9025 & 0.095 \end{pmatrix} \tag{4.39}$$

for max-product composition.

In Example 4.6, the universal sets U, V and W contain a finite number of elements. In most engineering applications, however, the U, V and W are real-valued spaces that contain an infinite number of elements. We now consider an example for computing the composition of fuzzy relations in continuous domains.

Example 4.7: Let $U = V = W = R$. Consider the fuzzy relation AE (approximately equal) and ML (much larger) defined by (4.9) and (4.10) in Example 4.3. We now want to determine the composition $AE \circ ML$. Using the max-product composition, we have

$$\mu_{AE \circ ML}(x, z) = \max_{y \in R} [\frac{e^{-(x-y)^2}}{1 + e^{-(y-z)}}] \tag{4.40}$$

To compute the right hand side of (4.40), we must determine the $y \in R$ at which $\frac{e^{-(x-y)^2}}{1+e^{-(y-z)}}$ achieves its maximum value, where x and z are considered to be fixed values in R. The necessary condition for such y is

$$\frac{\partial}{\partial y}[\frac{e^{-(x-y)^2}}{1+e^{-(y-z)}}] = 0 \tag{4.41}$$

Because it is impossible to obtain a closed form solution for (4.41), we cannot further simplify (4.40). In practice, for given values of x and z we can first determine the numerical solution of (4.41) and then substitute it into (4.40). Comparing this example with Example 4.6, we see that fuzzy compositions in continuous domains are much more difficult to compute than those in discrete domains. □

4.3 The Extension Principle

The extension principle is a basic identity that allows the domain of a function to be extended from crisp points in U to fuzzy sets in U. More specifically, let $f : U \to V$ be a function from crisp set U to crisp set V. Suppose that a fuzzy set A in U is given and we want to determine a fuzzy set $B = f(A)$ in V that is induced by f. If f is an one-to-one mapping, then we can define

$$\mu_B(y) = \mu_A[f^{-1}(y)], \ y \in V \tag{4.42}$$

where $f^{-1}(y)$ is the inverse of f, that is, $f[f^{-1}(y)] = y$. If f is not one-to-one, then an ambiguity arises when two or more distinct points in U with different membership values in A are mapped into the same point in V. For example, we may have $f(x_1) = f(x_2) = y$ but $x_1 \neq x_2$ and $\mu_A(x_1) \neq \mu_A(x_2)$, so the right hand side of (4.42) may take two different values $\mu_A(x_1 = f^{-1}(y))$ or $\mu_A(x_2 = f^{-1}(y))$. To resolve this ambiguity, we assign the larger one of the two membership values to $\mu_B(y)$. More generally, the membership function for B is defined as

$$\mu_B(y) = \max_{x \in f^{-1}(y)} \mu_A(x), \ y \in V \tag{4.43}$$

where $f^{-1}(y)$ denotes the set of all points $x \in U$ such that $f(x) = y$. The identity (4.43) is called the *extension principle.*

Example 4.8. Let $U = \{1, 2, ..., 10\}$ and $f(x) = x^2$. Let *small* be a fuzzy set in U defined by

$$small = 1/1 + 1/2 + 0.8/3 + 0.6/4 + 0.4/5 \tag{4.44}$$

Then, in consequence of (4.43), we have

$$small^2 = 1/1 + 1/4 + 0.8/9 + 0.6/16 + 0.4/25 \tag{4.45}$$

□

4.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- The concepts of fuzzy relations, projections, and cylindric extensions.
- The max-min and max-product compositions of fuzzy relations.
- The extension principle and its applications.

The basic ideas of fuzzy relations, projections, cylindric extensions, compositions of fuzzy relations, and the extension principle were proposed in Zadeh [1971b] and Zadeh [1975]. These original papers of Zadeh were very clearly written and are still the best sources to learn these fundamental concepts.

4.5 Exercises

Exercise 4.1. Given an n-ary relation, how many different projections of the relation can be taken?

Exercise 4.2. Consider the fuzzy relation Q defined in $U_1 \times \cdots \times U_4$ where $U_1 = \{a, b, c\}, U_2 = \{s, t\}, U_3 = \{x, y\}$ and $U_4 = \{i, j\}$:

$$Q = 0.4/(b,t,y,i) + 0.6/(a,s,x,i) + 0.9/(b,s,y,i) + 1/(b,s,y,j) \\ +0.6/(a,t,y,j) + 0.2/(c,s,y,i)$$

(a) Compute the projections of Q on $U_1 \times U_2 \times U_4$, $U_1 \times U_3$ and U_4.

(b) Compute the cylindric extensions of the projections in (a) to $U_1 \times U_2 \times U_3 \times U_4$.

Exercise 4.3. Consider the three binary fuzzy relations defined by the relational matrices:

$$Q_1 = \begin{pmatrix} 1 & 0 & 0.7 \\ 0.3 & 0.2 & 0 \\ 0 & 0.5 & 1 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 0.6 & 0.6 & 0 \\ 0 & 0.6 & 0.1 \\ 0 & 0.1 & 0 \end{pmatrix}, \quad Q_3 = \begin{pmatrix} 1 & 0 & 0.7 \\ 0 & 1 & 0 \\ 0.7 & 0 & 1 \end{pmatrix} \tag{4.46}$$

Compute the max-min and max-product compositions $Q_1 \circ Q_2, Q_1 \circ Q_3$ and $Q_1 \circ Q_2 \circ Q_3$.

Exercise 4.4. Consider fuzzy set $A = 0.5/-1 + 0.8/0 + 1/1 + 0.4/2$ and function $f(x) = x^2$. Determine the fuzzy set $f(A)$ using the extension principle.

Exercise 4.5. Compute the $\mu_{AE \circ ML}(x, z)$ in Example 4.7 for $(x, z) = (0, 0), (0, 1), (1, 0), (1, 1)$.

Chapter 5

Linguistic Variables and Fuzzy IF-THEN Rules

5.1 From Numerical Variables to Linguistic Variables

In our daily life, words are often used to describe variables. For example, when we say "today is hot," or equivalently, "today's temperature is high," we use the word "high" to describe the variable "today's temperature." That is, the variable "today's temperature" takes the word "high" as its value. Clearly, the variable "today's temperature" also can take numbers like $25^oc, 19^oc$, etc., as its values. When a variable takes numbers as its values, we have a well-established mathematical framework to formulate it. But when a variable takes words as its values, we do not have a formal framework to formulate it in classical mathematical theory. In order to provide such a formal framework, the concept of linguistic variables was introduced. Roughly speaking, if a variable can take words in natural languages as its values, it is called a *linguistic variable*. Now the question is how to formulate the words in mathematical terms? Here we use fuzzy sets to characterize the words. Thus, we have the following definition.

Definition 5.1. If a variable can take words in natural languages as its values, it is called a *linguistic variable*, where the words are characterized by fuzzy sets defined in the universe of discourse in which the variable is defined.

Example 5.1. The speed of a car is a variable x that takes values in the interval $[0, V_{max}]$, where V_{max} is the maximum speed of the car. We now define three fuzzy sets "slow," "medium," and "fast" in $[0, V_{max}]$ as shown in Fig. 5.1. If we view x as a linguistic variable, then it can take "slow," "medium" and "fast" as its values. That is, we can say "x is slow," "x is medium," and "x is fast." Of course, x also can take numbers in the interval $[0, V_{max}]$ as its values, for example, $x = 50$mph, 35mph, etc. □

Definition 5.1 gives a simple and intuitive definition for linguistic variables. In the fuzzy theory literature, a more formal definition of linguistic variables was usu-

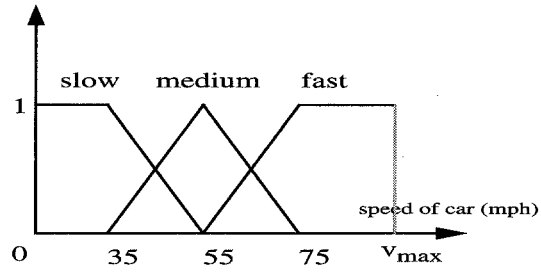


Figure 5.1. The speed of a car as a linguistic variable that can take fuzzy sets "slow," "medium" and "fast" as its values.

ally employed (Zadeh [1973] and [1975]). This definition is given below.

Definition 5.2. A *linguistic variable* is characterized by (X, T, U, M), where:

- X is the name of the linguistic variable; in Example 5.1, X is the speed of the car.
- T is the set of linguistic values that X can take; in Example 5.1, $T = \{\text{slow, medium, fast}\}$.
- U is the actual physical domain in which the linguistic variable X takes its quantitative (crisp) values; in Example 5.1, $U = [0, V_{max}]$.
- M is a semantic rule that relates each linguistic value in T with a fuzzy set in U; in Example 5.1, M relates "slow," "medium," and "fast" with the membership functions shown in Fig. 5.1.

Comparing Definitions 5.1 with 5.2, we see that they are essentially equivalent. Definition 5.1 is more intuitive, whereas Definition 5.2 looks more formal. From these definitions we see that linguistic variables are extensions of numerical variables in the sense that they are allowed to take fuzzy sets as their values; see Fig. 5.2.

Why is the concept of linguistic variable important? Because linguistic variables are the most fundamental elements in human knowledge representation. When we use sensors to measure a variable, they give us numbers; when we ask human experts to evaluate a variable, they give us words. For example, when we use a radar gun to measure the speed of a car, it gives us numbers like $39mph, 42mph$, etc.; when

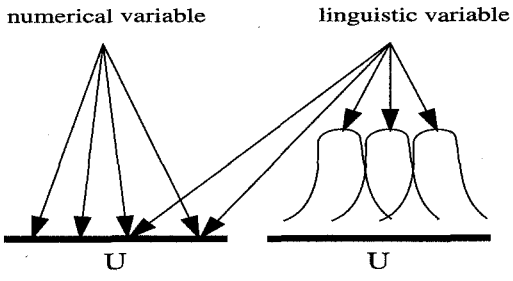


Figure 5.2. From numerical variable to linguistic variable.

we ask a human to tell us about the speed of the car, he/she often tells us in words like "it's slow," "it's fast," etc. Hence, by introducing the concept of linguistic variables, we are able to formulate vague descriptions in natural languages in precise mathematical terms. This is the first step to incorporate human knowledge into engineering systems in a systematic and efficient manner.

5.2 Linguistic Hedges

With the concept of linguistic variables, we are able to take words as values of (linguistic) variables. In our daily life, we often use more than one word to describe a variable. For example, if we view the speed of a car as a linguistic variable, then its values might be "not slow," "very slow," "slightly fast," "more or less medium," etc. In general, the value of a linguistic variable is a composite term $x = x_1 x_2 \cdots x_n$ that is a concatenation of atomic terms $x_1, x_2, ..., x_n$. These *atomic terms* may be classified into three groups:

- *Primary terms*, which are labels of fuzzy sets; in Example 5.1, they are "slow," "medium," and "fast."
- Complement "not" and connections "and" and "or."
- *Hedges*, such as "very," "slightly," "more or less," etc.

The terms "not," "and," and "or" were studied in Chapters 2 and 3. Our task now is to characterize hedges.

Although in its everyday use the hedge *very* does not have a well-defined meaning, in essence it acts as an intensifier. In this spirit, we have the following definition for the two most commonly used hedges: *very* and *more or less.*

Definition 5.3. Let A be a fuzzy set in U, then *very* A is defined as a fuzzy set in U with the membership function

$$\mu_{very\ A}(x) = [\mu_A(x)]^2 \tag{5.1}$$

and *more or less* A is a fuzzy set in U with the membership function

$$\mu_{more\ or\ less\ A}(x) = [\mu_A(x)]^{1/2} \tag{5.2}$$

Example 5.2. Let $U = \{1, 2, ..., 5\}$ and the fuzzy set *small* be defined as

$$small = 1/1 + 0.8/2 + 0.6/3 + 0.4/4 + 0.2/5 \tag{5.3}$$

Then, according to (5.1) and (5.2), we have

$$very\ small = 1/1 + 0.64/2 + 0.36/3 + 0.16/4 + 0.04/5 \tag{5.4}$$

$$\begin{aligned} very\ very\ small &= very\ (very\ small) \\ &= 1/1 + 0.4096/2 + 0.1296/3 + 0.0256/4 \\ &\quad +0.0016/5 \end{aligned} \tag{5.5}$$

$$\begin{aligned} more\ or\ less\ small &= 1/1 + 0.8944/2 + 0.7746/3 + 0.6325/4 \\ &\quad +0.4472/5 \end{aligned} \tag{5.6}$$

□

5.3 Fuzzy IF-THEN Rules

In Chapter 1 we mentioned that in fuzzy systems and control, human knowledge is represented in terms of fuzzy IF-THEN rules. A *fuzzy IF-THEN rule* is a conditional statement expressed as

$$IF\ <fuzzy\ proposition>,\ THEN\ <fuzzy\ proposition> \tag{5.7}$$

Therefore, in order to understand fuzzy IF-THEN rules, we first must know what are fuzzy propositions.

5.3.1 Fuzzy Propositions

There are two types of fuzzy propositions: atomic fuzzy propositions, and compound fuzzy propositions. An *atomic fuzzy proposition* is a single statement

$$x\ is\ A \tag{5.8}$$

where x is a linguistic variable, and A is a linguistic value of x (that is, A is a fuzzy set defined in the physical domain of x). A *compound fuzzy proposition* is a composition of atomic fuzzy propositions using the connectives "and," "or," and "not" which represent fuzzy intersection, fuzzy union, and fuzzy complement, respectively. For example, if x represents the speed of the car in Example 5.1, then the following are fuzzy propositions (the first three are atomic fuzzy propositions and the last three are compound fuzzy propositions):

$$x \text{ is } S \tag{5.9}$$

$$x \text{ is } M \tag{5.10}$$

$$x \text{ is } F \tag{5.11}$$

$$x \text{ is } S \text{ or } x \text{ is not } M \tag{5.12}$$

$$x \text{ is not } S \text{ and } x \text{ is not } F \tag{5.13}$$

$$(x \text{ is } S \text{ and } x \text{ is not } F) \text{ or } x \text{ is } M \tag{5.14}$$

where S, M and F denote the fuzzy sets "slow," "medium," and "fast," respectively.

Note that in a compound fuzzy proposition, the atomic fuzzy propositions are independent, that is, the x's in the same proposition of (5.12)-(5.14) can be different variables. Actually, the linguistic variables in a compound fuzzy proposition are in general not the same. For example, let x be the speed of a car and $y = \dot{x}$ be the acceleration of the car, then if we define fuzzy set $large(L)$ for the acceleration, the following is a compound fuzzy proposition

$$x \text{ is } F \text{ and } y \text{ is } L$$

Therefore, *compound fuzzy propositions should be understood as fuzzy relations.* How to determine the membership functions of these fuzzy relations?

- *For connective "and" use fuzzy intersections.* Specifically, let x and y be linguistic variables in the physical domains U and V, and A and B be fuzzy sets in U and V, respectively, then the compound fuzzy proposition

  $$x \text{ is } A \text{ and } y \text{ is } B \tag{5.15}$$

  is interpreted as the fuzzy relation $A \cap B$ [1] in $U \times V$ with membership function

  $$\mu_{A \cap B}(x, y) = t[\mu_A(x), \mu_B(y)] \tag{5.16}$$

  where $t : [0,1] \times [0,1] \to [0,1]$ is any t-norm.

- *For connective "or" use fuzzy unions.* Specifically, the compound fuzzy proposition

  $$x \text{ is } A \text{ or } y \text{ is } B \tag{5.17}$$

[1] Note that in Chapters 2 and 3, A and B are fuzzy sets defined in the same universal set U and $A \cup B$ and $A \cap B$ are fuzzy sets in U; here, $A \cup B$ and $A \cap B$ are fuzzy relations in $U \times V$, where U may or may not equal V.

is interpreted as the fuzzy relation $A \cup B$ in $U \times V$ with membership function

$$\mu_{A \cup B}(x, y) = s[\mu_A(x), \mu_B(y)] \tag{5.18}$$

where $s : [0,1] \times [0,1] \to [0,1]$ is any s-norm.

- *For connective "not" use fuzzy complements.* That is, replace *not* A by $\bar{A}$, which is defined according to the complement operators in Chapter 3.

Example 5.3. The fuzzy proposition (5.14), that is,

$$FP = (x \text{ is } S \text{ and } x \text{ is not } F) \text{ or } x \text{ is } M \tag{5.19}$$

is a fuzzy relation in the product space $[0, V_{max}]^3$ with the membership function

$$\mu_{FP}(x_1, x_2, x_3) = s\{t[\mu_S(x_1), c(\mu_F(x_2))], \mu_M(x_3)\} \tag{5.20}$$

where s, t and c are s-norm, t-norm and fuzzy complement operators, respectively, the fuzzy sets $S = small, M = medium$, and $F = fast$ are defined in Fig. 5.1, and $x_1 = x_2 = x_3 = x$. □

We are now ready to interpret the fuzzy IF-THEN rules in the form of (5.7).

5.3.2 Interpretations of Fuzzy IF-THEN Rules

Because the fuzzy propositions are interpreted as fuzzy relations, the key question remaining is how to interpret the IF-THEN operation. In classical propositional calculus, the expression *IF* p *THEN* q is written as $p \to q$ with the implication $\to$ regarded as a connective defined by Table 5.1, where p and q are propositional variables whose values are either truth (T) or false (F). From Table 5.1 we see that if both p and q are true or false, then $p \to q$ is true; if p is true and q is false, then $p \to q$ is false; and, if p is false and q is true, then $p \to q$ is true. Hence, $p \to q$ is equivalent to

$$\bar{p} \vee q \tag{5.21}$$

and

$$(p \wedge q) \vee \bar{p} \tag{5.22}$$

in the sense that they share the same truth table (Table 5.1) as $p \to q$, where $\bar{}$, $\vee$ and $\wedge$ represent (classical) logic operations "not," "or," and "and," respectively.

Because fuzzy IF-THEN rules can be viewed as replacing the p and q with fuzzy propositions, we can interpret the fuzzy IF-THEN rules by replacing the $\bar{}, \vee$ and $\wedge$ operators in (5.21) and (5.22) with fuzzy complement, fuzzy union, and fuzzy intersection, respectively. Since there are a wide variety of fuzzy complement, fuzzy union, and fuzzy intersection operators, a number of different interpretations of fuzzy IF-THEN rules were proposed in the literature. We list some of them below.

Table 5.1. Truth table for $p \rightarrow q$

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

In the following, we rewrite (5.7) as $IF \ < FP_1 > \ THEN \ < FP_2 >$ and replace the p and q in (5.21) and (5.22) by FP_1 and FP_2, respectively, where FP_1 and FP_2 are fuzzy propositions. We assume that FP_1 is a fuzzy relation defined in $U = U_1 \times \cdots \times U_n$, FP_2 is a fuzzy relation defined in $V = V_1 \times \cdots \times V_m$, and x and y are linguistic variables (vectors) in U and V, respectively.

- **Dienes-Rescher Implication**: If we replace the logic operators $^-$ and $\vee$ in (5.21) by the basic fuzzy complement (3.1) and the basic fuzzy union (3.2), respectively, then we obtain the so-called *Dienes-Rescher implication.* Specifically, the fuzzy IF-THEN rule $IF \ < FP_1 > \ THEN \ < FP_2 >$ is interpreted as a fuzzy relation Q_D in $U \times V$ with the membership function

$$\mu_{Q_D}(x, y) = max[1 - \mu_{FP_1}(x), \mu_{FP_2}(y)] \tag{5.23}$$

- **Lukasiewicz Implication**: If we use the Yager s-norm (3.10) with $w = 1$ for the $\vee$ and basic fuzzy complement (3.1) for the $^-$ in (5.21), we obtain the *Lukasiewicz implication.* Specifically, the fuzzy IF-THEN rule $IF \ < FP_1 > \ THEN \ < FP_2 >$ is interpreted as a fuzzy relation Q_L in $U \times V$ with the membership function

$$\mu_{Q_L}(x, y) = min[1, 1 - \mu_{FP_1}(x) + \mu_{FP_2}(y)] \tag{5.24}$$

- **Zadeh Implication**: Here the fuzzy IF-THEN rule $IF \ < FP_1 > \ THEN \ < FP_2 >$ is interpreted as a fuzzy relation Q_Z in $U \times V$ with the membership function

$$\mu_{Q_Z}(x, y) = max[min(\mu_{FP_1}(x), \mu_{FP_2}(y)), 1 - \mu_{FP_1}(x)] \tag{5.25}$$

  Clearly, (5.25) is obtained from (5.22) by using basic fuzzy complement (3.1), basic fuzzy union (3.2), and basic fuzzy intersection (3.3) for $^-\vee$ and $\wedge$, respectively.

- **Gödel Implication**: The Gödel implication is a well-known implication formula in classical logic. By generalizing it to fuzzy propositions, we obtain

the following: the fuzzy IF-THEN rule $IF \ < FP_1 > \ THEN \ < FP_2 >$ is interpreted as a fuzzy relation Q_G in $U \times V$ with the membership function

$$\mu_{Q_G}(x,y) = \begin{cases} 1 & if \ \mu_{FP_1}(x) \leq \mu_{FP_2}(y) \\ \mu_{FP_2}(y) & otherwise \end{cases} \tag{5.26}$$

It is interesting to explore the relationship among these implications. The following lemma shows that the Zadeh implication is smaller than the Dienes-Rescher implication, which is smaller than the Lukasiewicz implication.

Lemma 5.1. For all $(x,y) \in U \times V$, the following is true

$$\mu_{Q_Z}(x,y) \leq \mu_{Q_D}(x,y) \leq \mu_{Q_L}(x,y) \tag{5.27}$$

Proof: Since $0 \leq 1 - \mu_{FP_1}(x) \leq 1$ and $0 \leq \mu_{FP_2}(y) \leq 1$, we have $max[1 - \mu_{FP_1}(x), \mu_{FP_2}(y)] \leq 1 - \mu_{FP_1}(x) + \mu_{FP_2}(y)$ and $max[1 - \mu_{FP_1}(x), \mu_{FP_2}(y)] \leq 1$. Hence, $\mu_{Q_D}(x,y) = max[1 - \mu_{FP_1}(x), \mu_{FP_2}(y)] \leq min[1, 1 - \mu_{FP_1}(x) + \mu_{FP_2}(y)] = \mu_{Q_L}(x,y)$. Since $min[\mu_{FP_1}(x), \mu_{FP_2}(y)] \leq \mu_{FP_2}(y)$, we have $max[min(\mu_{FP_1}(x), \mu_{FP_2}(y)), 1 - \mu_{FP_1}(x)] \leq max[\mu_{FP_2}(y), 1 - \mu_{FP_1}(x)]$, which is $\mu_{Q_Z}(x,y) \leq \mu_{Q_D}(x,y)$. □

Conceptually, we can replace the $\bar{}$, $\vee$ and $\wedge$ in (5.21) and (5.22) by any fuzzy complement, s-norm and t-norm, respectively, to obtain a particular interpretation. So a question arises: Based on what criteria do we choose the combination of fuzzy complements, s-norms, and t-norms? This is an important question and we will discuss it in Chapters 7-10. Another question is: Are (5.21) and (5.22) still "equivalent" to $p \rightarrow q$ when p and q are fuzzy propositions and what does this "equivalent" mean? We now try to answer this question. When p and q are crisp propositions (that is, p and q are either true or false), $p \rightarrow q$ is a *global* implication in the sense that Table 5.1 covers all the possible cases. However, when p and q are fuzzy propositions, $p \rightarrow q$ may only be a *local* implication in the sense that $p \rightarrow q$ has large truth value only when both p and q have large truth values. For example, when we say "IF speed is high, THEN resistance is high," we are concerned only with a local situation in the sense that this rule tells us nothing about the situations when "speed is slow," "speed is medium," etc. Therefore, the fuzzy IF-THEN rule

$$IF \ < FP_1 > \ THEN \ < FP_2 > \tag{5.28}$$

should be interpreted as

$$IF \ < FP_1 > \ THEN \ < FP_2 > \ ELSE \ < NOTHING > \tag{5.29}$$

where $NOTHING$ means that this rule does not exist. In logic terms, it becomes

$$p \rightarrow q = p \wedge q \tag{5.30}$$

Using *min* or *algebraic product* for the $\wedge$ in (5.30), we obtain the *Mamdani implications.*

- **Mamdani Implications**: The fuzzy IF-THEN rule (5.28) is interpreted as a fuzzy relation Q_{MM} or Q_{MP} in $U \times V$ with the membership function

$$\mu_{Q_{MM}}(x,y) = min[\mu_{FP_1}(x), \mu_{FP_2}(y)] \tag{5.31}$$

or

$$\mu_{Q_{MP}}(x,y) = \mu_{FP_1}(x)\mu_{FP_2}(y) \tag{5.32}$$

Mamdani implications are the most widely used implications in fuzzy systems and fuzzy control. They are supported by the argument that fuzzy IF-THEN rules are local. However, one may not agree with this argument. For example, one may argue that when we say "IF speed is high, THEN resistance is high," we *implicitly* indicate that "IF speed is slow, THEN resistance is low." In this sense, fuzzy IF-THEN rules are nonlocal. This kind of debate indicates that when we represent human knowledge in terms of fuzzy IF-THEN rules, different people have different interpretations. Consequently, different implications are needed to cope with the diversity of interpretations. For example, if the human experts think that their rules are local, then the Mamdani implications should be used; otherwise, the global implications (5.23)-(5.26) should be considered.

We now consider some examples for the computation of Q_D, Q_L, Q_Z, Q_{MM} and Q_{MP}.

Example 5.4. Let x_1 be the speed of a car, x_2 be the acceleration, and y be the force applied to the accelerator. Consider the following fuzzy IF-THEN rule:

$$IF\ x_1\ is\ slow\ and\ x_2\ is\ small,\ THEN\ y\ is\ large \tag{5.33}$$

where "slow" is the fuzzy set defined in Fig. 5.1, that is,

$$\mu_{slow}(x_1) = \begin{cases} 1 & if \quad x_1 \leq 35 \\ \frac{55-x_1}{20} & if \quad 35 < x_1 \leq 55 \\ 0 & if \quad x_1 > 55 \end{cases} \tag{5.34}$$

"small" is a fuzzy set in the domain of acceleration with the membership function

$$\mu_{small}(x_2) = \begin{cases} \frac{10-x_2}{10} & if \quad 0 \leq x_2 \leq 10 \\ 0 & if \quad x_2 > 10 \end{cases} \tag{5.35}$$

and "large" is a fuzzy set in the domain of force applied to the accelerator with the membership function

$$\mu_{large}(y) = \begin{cases} 0 & if \quad y \leq 1 \\ y-1 & if \quad 1 \leq y \leq 2 \\ 1 & if \quad y > 2 \end{cases} \tag{5.36}$$

Let the domains of x_1, x_2 and y be $U_1 = [0,100], U_2 = [0,30]$, and $V = [0,3]$, respectively. If we use algebraic product for the t-norm in (5.16), then the fuzzy proposition

$$FP_1 = x_1\ is\ slow\ and\ x_2\ is\ small \tag{5.37}$$

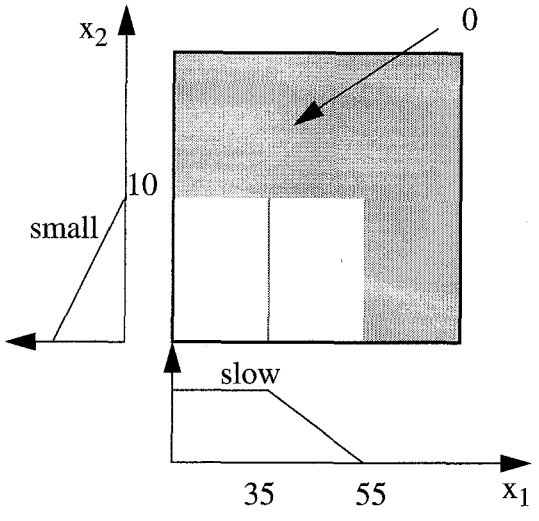


Figure 5.3. Illustration for how to compute $\mu_{slow}(x_1)\mu_{small}(x_2)$ in Example 5.4.

is a fuzzy relation in $U_1 \times U_2$ with the membership function

$$\mu_{FP_1}(x_1, x_2) = \mu_{slow}(x_1)\mu_{small}(x_2)$$
$$= \begin{cases} 0 & if\ x_1 \geq 55\ or\ x_2 > 10 \\ \frac{10-x_2}{10} & if\ x_1 \leq 35\ and\ x_2 \leq 10 \\ \frac{(55-x_1)(10-x_2)}{200} & if\ 35 < x_1 \leq 55\ and\ x_2 \leq 10 \end{cases} \tag{5.38}$$

Fig. 5.3 illustrates how to compute $\mu_{FP_1}(x_1, x_2)$.

If we use the Dienes-Rescher implication (5.23), then the fuzzy IF-THEN rule (5.33) is interpreted as a fuzzy relation $Q_D(x_1, x_2, y)$ in $U_1 \times U_2 \times V$ with the membership function

$$\mu_{Q_D}(x_1, x_2, y) = max[1 - \mu_{FP_1}(x_1, x_2), \mu_{large}(y)] \tag{5.39}$$

From (5.38) we have

$$1 - \mu_{FP_1}(x_1, x_2) = \begin{cases} 1 & if\ \ x_1 \geq 55\ or\ x_2 > 10 \\ x_2/10 & if\ \ x_1 \leq 35\ and\ x_2 \leq 10 \\ 1 - \frac{(55-x_1)(10-x_2)}{200} & if\ \ 35 < x_1 \leq 55\ and\ x_2 \leq 10 \end{cases} \tag{5.40}$$

To help us combining $1 - \mu_{FP_1}(x_1, x_2)$ of (5.40) with $\mu_{large}(y)$ of (5.36) using the *max* operator, we illustrate in Fig. 5.4 the division of the domains of $1 - \mu_{FP_1}(x_1, x_2)$

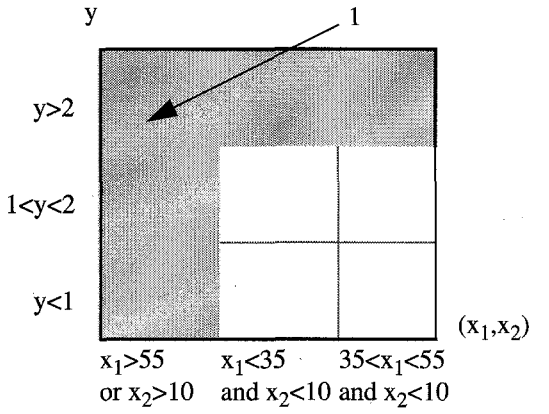


Figure 5.4. Division of the domains of $1 - \mu_{FP_1}(x_1, x_2)$ and $\mu_{large}(y)$ and their combinations for Example 5.4.

and $\mu_{large}(y)$ and their combinations. From Fig. 5.4, we obtain

$$\mu_{Q_D}(x_1, x_2, y) = \begin{cases} 1 & if\ x_1 \geq 55\ or\ x_2 > 10\ or\ y > 2 \\ x_2/10 & if\ x_1 \leq 35\ and\ x_2 \leq 10\ and\ y \leq 1 \\ 1 - \frac{(55-x_1)(10-x_2)}{200} & if\ 35 < x_1 \leq 55\ and\ x_2 \leq 10 \\ & \quad and\ y \leq 1 \\ max[y-1, x_2/10] & if\ x_1 \leq 35\ and\ x_2 \leq 10 \\ & \quad and\ 1 < y \leq 2 \\ max[y-1, 1 - \frac{(55-x_1)(10-x_2)}{200}] & if\ 35 < x_1 \leq 55\ and\ x_2 \leq 10 \\ & \quad and\ 1 < y \leq 2 \end{cases} \tag{5.41}$$

For Lukasiewicz, Zadeh and Mamdani implications, we can use the same procedure to determine the membership functions. □

From Example 5.4 we see that if the membership functions in the atomic fuzzy propositions are not smooth functions (for example, (5.34)-(5.36)), the computation of the final membership functions μ_{Q_D}, μ_{Q_Z}, etc., is cumbersome, although it is straightforward. A way to resolve this complexity is to use a single smooth function to approximate the nonsmooth functions; see the following example.

Example 5.4 (Cont'd). Suppose we use

$$\mu_{slow}(x_1) = \frac{1}{1 + e^{\frac{x_1 - 45}{5}}} \tag{5.42}$$

to approximate the $\mu_{slow}(x_1)$ of (5.34),

$$\mu_{small}(x_2) = \frac{1}{1 + e^{\frac{x_2 - 5}{2}}} \tag{5.43}$$

to approximate the $\mu_{small}(x_2)$ of (5.35), and

$$\mu_{large}(y) = \frac{1}{1 + e^{2(-y+1.25)}} \tag{5.44}$$

to approximate the $\mu_{large}(y)$ of (5.36). Now if we use Mamdani product implication (5.32) and algebraic product for the t-norm in (5.16), then the membership function $\mu_{Q_{MP}}(x_1, x_2, y)$ can be easly computed as

$$\begin{aligned} \mu_{Q_{MP}}(x_1, x_2, y) &= \mu_{slow}(x_1)\mu_{small}(x_2)\mu_{large}(y) \\ &= \frac{1}{(1 + e^{\frac{x_1 - 45}{5}})(1 + e^{\frac{x_2 - 5}{2}})(1 + e^{2(-y+1.5)})} \end{aligned} \tag{5.45}$$

□

Example 5.5. Let $U = \{1, 2, 3, 4\}$ and $V = \{1, 2, 3\}$. Suppose we know that $x \in U$ is somewhat inversely propositional to $y \in V$. To formulate this knowledge, we may use the following fuzzy IF-THEN rule:

$$IF\ x\ is\ large,\ THEN\ y\ is\ small \tag{5.46}$$

where the fuzzy sets "large" and "small" are defined as

$$large = 0/1 + 0.1/2 + 0.5/3 + 1/4 \tag{5.47}$$

$$small = 1/1 + 0.5/2 + 0.1/3 \tag{5.48}$$

If we use the Dienes-Rescher implication (5.23), then the fuzzy IF-THEN rule (5.46) is interpreted as the following fuzzy relation Q_D in $U \times V$:

$$\begin{aligned} Q_D = {} & 1/(1,1) + 1/(1,2) + 1/(1,3) + 1/(2,1) + 0.9/(2,2) + 0.9/(2,3) \\ & + 1/(3,1) + 0.5/(3,2) + 0.5/(3,3) + 1/(4,1) + 0.5/(4,2) \\ & + 0.1/(4,3) \end{aligned} \tag{5.49}$$

If we use the Lukasiewicz implication (5.24), the rule (5.46) becomes

$$\begin{aligned} Q_L = {} & 1/(1,1) + 1/(1,2) + 1/(1,3) + 1/(2,1) + 1/(2,2) + 1/(2,3) + 1/(3,1) \\ & + 1/(3,2) + 0.6/(3,3) + 1/(4,1) + 0.5/(4,2) + 0.1/(4,3) \end{aligned} \tag{5.50}$$

For the Zadeh implication (5.25) and the Gödel implication (5.26), we have

$$\begin{aligned} Q_Z = {} & 1/(1,1) + 1/(1,2) + 1/(1,3) + 0.9/(2,1) + 0.9/(2,2) + 0.9/(2,3) \\ & +0.5/(3,1) + 0.5/(3,2) + 0.5/(3,3) + 1/(4,1) \\ & +0.5/(4,2) + 0.1/(4,3) \end{aligned} \tag{5.51}$$

and

$$\begin{aligned} Q_G = {} & 1/(1,1) + 1/(1,2) + 1/(1,3) + 1/(2,1) + 1/(2,2) + 1/(2,3) + 1/(3,1) \\ & +1/(3,2) + 0.1/(3,3) + 1/(4,1) + 0.5/(4,2) + 0.1/(4,3) \end{aligned} \tag{5.52}$$

Finally, if we use the Mamdani implication (5.31) and (5.32), then the fuzzy IF-THEN rule (5.46) becomes

$$\begin{aligned} Q_{MM} = {} & 0/(1,1) + 0/(1,2) + 0/(1,3) + 0.1/(2,1) + 0.1/(2,2) \\ & +0.1/(2,3) + 0.5/(3,1) + 0.5/(3,2) + 0.1/(3,3) \\ & +1/(4,1) + 0.5/(4,2) + 0.1/(4,3) \end{aligned} \tag{5.53}$$

and

$$\begin{aligned} Q_{MP} = {} & 0/(1,1) + 0/(1,2) + 0/(1,3) + 0.1/(2,1) + 0.05/(2,2) \\ & +0.01/(2,3) + 0.5/(3,1) + 0.25/(3,2) + 0.05/(3,3) \\ & +1/(4,1) + 0.5/(4,2) + 0.1/(4,3) \end{aligned} \tag{5.54}$$

From (5.49)-(5.52) we see that for the combinations not covered by the rule (5.46), that is, the pairs $(1,1), (1,2)$ and $(1,3)$ (because $\mu_{large}(1) = 0$), Q_D, Q_L, Q_Z and Q_G give full membership value to them, but Q_{MM} and Q_{MP} give zero membership value. This is consistent with our earlier discussion that Dienes-Rescher, Lukasiewicz, Zadeh and Gödel implications are global, whereas Mamdani implications are local. □

5.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- The concept of linguistic variables and the characterization of hedges.
- The concept of fuzzy propositions and fuzzy IF-THEN rules.
- Different interpretations of fuzzy IF-THEN rules, including Dienes-Rescher, Lukasiewicz, Zadeh, Gödel and Mamdani implications.
- Properties and computation of these implications.

Linguistic variables were introduced in Zadeh's seminal paper Zadeh [1973]. This paper is another piece of art and the reader is highly recommended to study it. The comprehensive three-part paper Zadeh [1975] summarized many concepts and principles associated with linguistic variables.

5.5 Exercises

Exercise 5.1. Give three examples of linguistic variables. Combine these linguistic variables into a compound fuzzy proposition and determine its membership function.

Exercise 5.2. Consider some other linguistic hedges than those in Section 5.2 and propose reasonable operations that represent them.

Exercise 5.3. Let Q_L, Q_G, Q_{MM} and Q_{MP} be the fuzzy relations defined in (5.24), (5.26), (5.31), and (5.32), respectively. Show that

$$Q_{MP} \subset Q_{MM} \subset Q_G \subset Q_L \tag{5.55}$$

Exercise 5.4. Use basic fuzzy operators (3.1)-(3.3) for "not," "or," and "and," respectively, and determine the membership functions for the fuzzy propositions (5.12) and (5.13). Plot these membership functions.

Exercise 5.5. Consider the fuzzy IF-THEN rule (5.33) with the fuzzy sets "slow," "small" and "large" defined by (5.42), (5.43) and (5.44), respectively. Use *min* for the t-norm in (5.16) and compute the fuzzy relations $Q_D, Q_L, Q_Z, Q_G, Q_{MM}$ and Q_{MP}.

Exercise 5.6. Let Q be a fuzzy relation in $U \times U$. Q is called *reflexive* if $\mu_Q(u,u) = 1$ for all $u \in U$. Show that if Q is reflexive, then: (a) $Q \circ Q$ is also reflexive, and (b) $Q \subseteq Q \circ Q$, where $\circ$ denotes $max - min$ composition.

Chapter 6

Fuzzy Logic and Approximate Reasoning

6.1 From Classical Logic to Fuzzy Logic

Logic is the study of methods and principles of reasoning, where *reasoning* means obtaining new propositions from existing propositions. In classical logic, the propositions are required to be either true or false, that is, the truth value of a proposition is either 0 or 1. Fuzzy logic generalizes classical two-value logic by allowing the truth values of a proposition to be any number in the interval $[0, 1]$. This generalization allows us to perform *approximate reasoning*, that is, deducing imprecise conclusions (fuzzy propositions) from a collection of imprecise premises (fuzzy propositions). In this chapter, we first review some basic concepts and principles in classical logic and then study their generalization to fuzzy logic.

6.1.1 Short Primer on Classical Logic

In classical logic, the relationships between propositions are usually represented by a truth table. The fundamental truth table for conjunction $\vee$, disjunction $\wedge$, implication $\rightarrow$, equivalence $\leftrightarrow$ and negation $^-$ are collected together in Table 6.1, where the symbols T and F denote truth and false, respectively.

Given n basic propositions $p_1, ..., p_n$, a new proposition can be defined by a function that assigns a particular truth value to the new proposition for each combination of truth values of the given propositions. The new proposition is usually called a *logic function*. Since n propositions can assume 2^n possible combinations of truth values, there are 2^{2^n} possible logic functions defining n propositions. Because 2^{2^n} is a huge number for large n, a key issue in classical logic is to express all the logic functions with only a few basic logic operations; such basic logic operations are called *a complete set of primitives*. The most commonly used complete set of primitives is negation $^-$, conjunction $\vee$, and disjunction $\wedge$. By combining $^-$, $\vee$ and $\wedge$ in appropriate algebraic expressions, referred to as *logic formulas*, we can form any

Table 6.1. Truth table for five operations that are frequently applied to propositions.

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$	$\bar{p}$
T	T	T	T	T	T	F
T	F	F	T	F	F	F
F	T	F	T	T	F	T
F	F	F	F	T	T	T

other logic function. Logic formulas are defined recursively as follows:

- The truth values 0 and 1 are logic formulas.
- If p is a proposition, then p and $\bar{p}$ are logic formulas.
- If p and q are logic formulas, then $p \vee q$ and $p \wedge q$ are also logic formulas.
- The only logic formulas are those defined by (a)-(c).

When the proposition represented by a logic formula is always true regardless of the truth values of the basic propositions participating in the formula, it is called a *tautology*; when it is always false, it is called a *contradiction*.

Example 6.1. The following logic formulas are tautologies:

$$(p \rightarrow q) \leftrightarrow (\bar{p} \vee q) \tag{6.1}$$

$$(p \rightarrow q) \leftrightarrow ((p \wedge q) \vee \bar{p}) \tag{6.2}$$

To prove (6.1) and (6.2), we use the truth table method, that is, we list all the possible values of (6.1) and (6.2) and see whether they are all true. Table 6.2 shows the results, which indicates that (6.1) and (6.2) are tautologies. □

Table 6.2. Proof of $(p \rightarrow q) \leftrightarrow (\bar{p} \vee q)$ and $(p \rightarrow q) \leftrightarrow ((p \wedge q) \vee \bar{p})$.

p	q	$p \rightarrow q$	$\bar{p} \vee q$	$(p \wedge q) \vee \bar{p}$	$(p \rightarrow q) \leftrightarrow (\bar{p} \vee q)$	$(p \rightarrow q) \leftrightarrow ((p \wedge q) \vee \bar{p})$
T	T	T	T	T	T	T
T	F	F	F	F	T	T
F	T	T	T	T	T	T
F	F	T	T	T	T	T

Various forms of tautologies can be used for making deductive inferences. They are referred to as *inference rules*. The three most commonly used inference rules are:

- **Modus Ponens**: This inference rule states that given two propositions p and $p \rightarrow q$ (called the *premises*), the truth of the proposition q (called the *conclusion*) should be inferred. Symbolically, it is represented as

$$(p \wedge (p \rightarrow q)) \rightarrow q \tag{6.3}$$

A more intuitive representation of modus ponens is

$$\begin{aligned} &\textit{Premise } 1: && x \textit{ is } A \\ &\textit{Premise } 2: && \textit{IF } x \textit{ is } A \textit{ THEN } y \textit{ is } B \\ &\textit{Conclusion}: && y \textit{ is } B \end{aligned}$$

- **Modus Tollens**: This inference rule states that given two propositions $\bar{q}$ and $p \rightarrow q$, the truth of the proposition $\bar{p}$ should be inferred. Symbolically, it becomes

$$(\bar{q} \wedge (p \rightarrow q)) \rightarrow \bar{p} \tag{6.4}$$

A more intuitive representation of modus tollens is

$$\begin{aligned} &\textit{Premise } 1: && y \textit{ is not } B \\ &\textit{Premise } 2: && \textit{IF } x \textit{ is } A \textit{ THEN } y \textit{ is } B \\ &\textit{Conclusion}: && x \textit{ is not } A \end{aligned}$$

- **Hypothetical Syllogism**: This inference rule states that given two propostions $p \rightarrow q$ and $q \rightarrow r$, the truth of the proposition $p \rightarrow r$ should be inferred. Symbolically, we have

$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r) \tag{6.5}$$

A more intuitive representation of it is

$$\begin{aligned} &\textit{Premise } 1: && \textit{IF } x \textit{ is } A \textit{ THEN } y \textit{ is } B \\ &\textit{Premise } 2: && \textit{IF } y \textit{ is } B \textit{ THEN } z \textit{ is } C \\ &\textit{Conclusion}: && \textit{IF } x \textit{ is } A \textit{ THEN } z \textit{ is } C \end{aligned}$$

6.1.2 Basic Principles in Fuzzy Logic

In fuzzy logic, the propositions are fuzzy propositions that, as defined in Chapter 5, are represented by fuzzy sets. The ultimate goal of fuzzy logic is to provide foundations for approximate reasoning with imprecise propositions using fuzzy set theory as the principal tool. To achieve this goal, the so-called *generalized modus ponens, generalized modus tollens,* and *generalized hypothetical syllogism* were proposed. They are the fundamental principles in fuzzy logic.

- **Generalized Modus Ponens**: This inference rule states that given two fuzzy propositions x *is* A' and *IF* x *is* A *THEN* y *is* B, we should infer a new fuzzy proposition y *is* B' such that the closer the A' to A, the closer the B' to B, where A, A', B and B' are fuzzy sets; that is,

$$\begin{aligned} \textit{Premise } 1: &\quad x \textit{ is } A' \\ \textit{Premise } 2: &\quad \textit{IF } x \textit{ is } A \textit{ THEN } y \textit{ is } B \\ \textit{Conclusion}: &\quad y \textit{ is } B' \end{aligned}$$

Table 6.3 shows the intuitive criteria relating Premise 1 and the Conclusion in generalized modus ponens. We note that if a causal relation between "x is A" and "y is B" is not strong in Premise 2, the satisfaction of criterion p3 and criterion p5 is allowed. Criterion p7 is interpreted as: "IF x is A THEN y is B, ELSE y is not B." Although this relation is not valid in classical logic, we often make such an interpretation in everyday reasoning.

Table 6.3. Intuitive criteria relating Premise 1 and the Conclusion for given Premise 2 in generalized modus ponens.

	x is A' (Premise 1)	y is B' (Conclusion)
criterion p1	x is A	y is B
criterion p2	x is very A	y is very B
criterion p3	x is very A	y is B
criterion p4	x is more or less A	y is more or less B
criterion p5	x is more or less A	y is B
criterion p6	x is not A	y is unknown
criterion p7	x is not A	y is not B

- **Generalized Modus Tollens**: This inference rule states that given two fuzzy propositions y *is* B' and *IF* x *is* A *THEN* y *is* B, we should infer a new fuzzy proposition x *is* A' such that the more difference between B' and B, the more difference between A' and A, where A', A, B' and B are fuzzy sets; that is,

$$\begin{aligned} \textit{Premise } 1: &\quad y \textit{ is } B' \\ \textit{Premise } 2: &\quad \textit{IF } x \textit{ is } A \textit{ THEN } y \textit{ is } B \\ \textit{Conclusion}: &\quad x \textit{ is } A' \end{aligned}$$

Table 6.4 shows some intuitive criteria relating Premise 1 and the Conclusion in generalized modus tollens. Similar to the criteria in Table 6.3, some criteria in Table 6.4 are not true in classical logic, but we use them approximately in our daily life.

Table 6.4. Intuitive criteria relating Premise 1 and the Conclusion for given Premise 2 in generalized modus tollens.

	y is B' (Premise 1)	x is A' (Conclusion)
criterion t1	y is not B	x is not A
criterion t2	y is not very B	x is not very A
criterion t3	y is not more or less B	x is not more or less A
criterion t4	y is B	x is unknown
criterion t5	y is B	x is A

- **Generalized Hypothetical Syllogism**: This inference rule states that given two fuzzy propositions *IF* x *is* A *THEN* y *is* B and *IF* y *is* B' *THEN* z *is* C, we could infer a new fuzzy proposition *IF* x *is* A *THEN* z *is* C' such that the closer the B to B', the closer the C' to C, where A, B, B', C and C' are fuzzy sets; that is,

$$\begin{aligned} \textit{Premise } 1: &\quad \textit{IF } x \textit{ is } A \textit{ THEN } y \textit{ is } B \\ \textit{Premise } 2: &\quad \textit{IF } y \textit{ is } B' \textit{ THEN } z \textit{ is } C \\ \textit{Conclusion}: &\quad \textit{IF } x \textit{ is } A \textit{ THEN } z \textit{ is } C' \end{aligned}$$

Table 6.5 shows some intuitive criteria relating y *is* B' with z *is* C' in the generalized hypothetical syllogism. Criteria s2 is obtained from the following intuition: To match the B in Premise 1 with the $B' = \textit{very } B$ in Premise 2, we may change Premise 1 to *IF* x *is very* A *THEN* y *is very* B, so we have *IF* x *is very* A *THEN* z *is* C. By applying the hedge *more or less* to cancel the *very*, we have *IF* x *is* A *THEN* z *is more or less* C, which is criterion s2. Other criteria can be justified in a similar manner.

Table 6.5. Intuitive criteria relating y *is* B' in Premise 2 and z *is* C' in the Conclusion in generalized hypothetical syllogism.

	y is B' (Premise 2)	z is C' (Conclusion)
criterion s1	y is B	z is C
criterion s2	y is very B	z is more or less C
criterion s3	y is very B	z is C
criterion s4	y is more or less B	z is very C
criterion s5	y is more or less B	z is C
criterion s6	y is not B	z is unknown
criterion s7	y is not B	z is not C

We call the criteria in Tables 6.3-6.5 *intuitive* criteria because they are not necessarily true for a particular choice of fuzzy sets; this is what *approximate reasoning* means. Although these criteria are not absolutely correct, they do make some sense. They should be viewed as guidelines (or soft constraints) in designing specific inferences.

We have now shown the basic ideas of three fundamental principles in fuzzy logic: generalized modus ponens, generalized modus tollens, and generalized hypothetical syllogism. The next question is how to determine the membership functions of the fuzzy propositions in the conclusions given those in the premises. The *compositional rule of inference* was proposed to answer this question.

6.2 The Compositional Rule of Inference

The compositional rule of inference is a generalization of the following procedure (referring to Fig. 6.1): suppose we have a curve $y = f(x)$ from $x \in U$ to $y \in V$ and are given $x = a$, then from $x = a$ and $y = f(x)$ we can infer $y = b = f(a)$.

Let us generalize the above procedure by assuming that a is an interval and $f(x)$ is an interval-valued function as shown in Fig. 6.2. To find the interval b which is inferred from a and $f(x)$, we first construct a cylindrical set a_E with base a and find its intersection I with the interval-valued curve. Then we project I on V yielding the interval b.

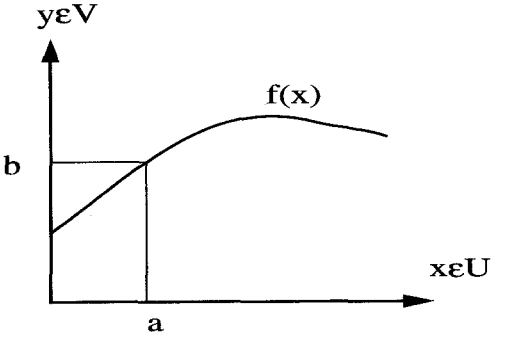


Figure 6.1. Inferring $y = b$ from $x = a$ and $y = f(x)$.

Going one step further in our chain of generalization, assume the A' is a fuzzy set in U and Q is a fuzzy relation in $U \times V$. Again, forming a cylindrical extension

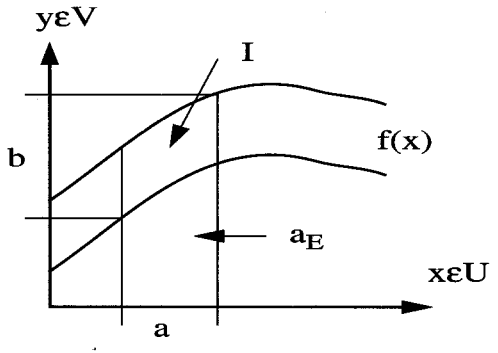


Figure 6.2. Inferring interval b from interval a and interval-valued function $f(x)$.

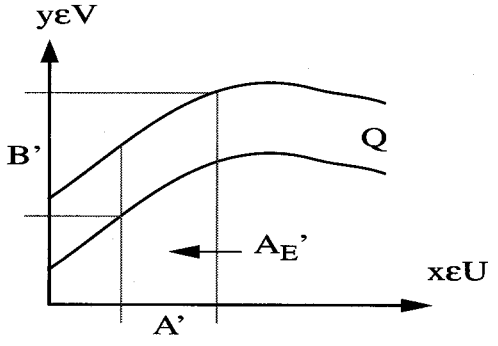


Figure 6.3. Inferring fuzzy set B' from fuzzy set A' and fuzzy relation Q.

A'_E of A' and intersecting it with the fuzzy relation Q (see Fig. 6.3), we obtain a fuzzy set $A'_E \cap Q$ which is analog of the intersection I in Fig. 6.2. Then, projecting $A'_E \cap Q$ on the y-axis, we obtain the fuzzy set B'.

More specifically, given $\mu_{A'}(x)$ and $\mu_Q(x,y)$, we have

$$\mu_{A'_E}(x,y) = \mu_{A'}(x) \tag{6.6}$$

(see (4.18)) and, consequently,

$$\begin{aligned}\mu_{A'_E \cap Q}(x,y) &= t[\mu_{A'_E}(x,y), \mu_Q(x,y)] \\ &= t[\mu_{A'}(x), \mu_Q(x,y)] \end{aligned} \tag{6.7}$$

Finally, from (4.12) we obtain B', the projection of $A'_E \cap Q$ on V, as

$$\mu_{B'}(y) = \sup_{x \in U} t[\mu_{A'}(x), \mu_Q(x,y)] \tag{6.8}$$

(6.8) is called the *compositional rule of inference.* In the literature, the symbol "$\star$" is often used for the t-norm operator, so (6.8) is also written as

$$\mu_{B'}(y) = \sup_{x \in U} [\mu_{A'}(x) \star \mu_Q(x,y)] \tag{6.9}$$

The compositional rule of inference is also called the *sup-star composition.*

In Chapter 5, we learned that a fuzzy IF-THEN rule, for example, IF x is A THEN y is B, is interpreted as a fuzzy relation in the Cartesian product of the domains of x and y. Different implication principles give different fuzzy relations; see (5.23)-(5.26), (5.31), and (5.32). Therefore, the Premise 2s in the generalized modus ponens and generalized modus tollens can be viewed as the fuzzy relation Q in (6.9). For generalized hypothetical syllogism, we see that it is simply the composition of two fuzzy relations, so we can use the composition (4.28) to determine the conclusion. In summary, we obtain the detailed formulas for computing the conclusions in generalized modus ponens, generalized modus tollens, and generalized hypothetical syllogism, as follows:

- **Generalized Modus Ponens**: Given fuzzy set A' (which represents the premise x *is* A') and fuzzy relation $A \to B$ in $U \times V$ (which represents the premise *IF* x *is* A *THEN* y *is* B), a fuzzy set B' in V (which represents the conclusion y *is* B') is inferred as

$$\mu_{B'}(y) = \sup_{x \in U} t[\mu_{A'}(x), \mu_{A \to B}(x,y)] \tag{6.10}$$

- **Generalized Modus Tollens**: Given fuzzy set B' (which represents the premise y *is* B') and fuzzy relation $A \to B$ in $U \times V$ (which represents the premise *IF* x *is* A *THEN* y *is* B), a fuzzy set A' in U (which represents the conclusion x *is* A') is inferred as

$$\mu_{A'}(x) = \sup_{y \in V} t[\mu_{B'}(y), \mu_{A \to B}(x,y)] \tag{6.11}$$

- **Generalized Hypothetical Syllogism**: Given fuzzy relation $A \to B$ in $U \times V$ (which represents the premise *IF* x *is* A *THEN* y *is* B) and fuzzy relation $B' \to C$ in $V \times W$ (which represents the premise *IF* y *is* B' *THEN* z

is C), a fuzzy relation $A \to C'$ in $U \times W$ (which represents the conclusion *IF x is A THEN z is C'*) is inferred as

$$\mu_{A\to C'}(x,z) = \sup_{y\in V} t[\mu_{A\to B}(x,y), \mu_{B'\to C}(y,z)] \tag{6.12}$$

Using different t-norms in (6.10)-(6.12) and different implication rules (5.23)-(5.26), (5.31) and (5.32), we obtain a diversity of results. These results show the properties of the implication rules. We now study some of these properties.

6.3 Properties of the Implication Rules

In this section, we apply specific implication rules and t-norms to (6.10)-(6.12) and see what the $\mu_{B'}(y), \mu_{A'}(x)$ and $\mu_{A\to C'}(x,z)$ look like for some typical cases of A' and B'. We consider the generalized modus ponens, generalized modus tollens, and generalized hypothetical syllogism in sequal.

6.3.1 Generalized Modus Ponens

Example 6.2. Suppose we use *min* for the t-norm and Mamdani's product implication (5.32) for the $\mu_{A\to B}(x,y)$ in the generalized modus ponens (6.10). Consider four cases of A': (a) $A' = A$, (b) $A' = very\ A$, (c) $A' = more\ or\ less\ A$, and (d) $A = \bar{A}$. Our task is to determine the corresponding B'. We assume that $\sup_{x\in U}[\mu_A(x)] = 1$ (the fuzzy set A is normal). If $A' = A$, we have

$$\begin{aligned}\mu_{B'}(y) &= \sup_{x\in U}\{min[\mu_A(x), \mu_A(x)\mu_B(y)]\} \\ &= \sup_{x\in U}[\mu_A(x)\mu_B(y)] \\ &= \mu_B(y)\end{aligned} \tag{6.13}$$

If $A' = very\ A$, we have

$$\mu_{B'}(y) = \sup_{x\in U}\{min[\mu_A^2(x), \mu_A(x)\mu_B(y)]\} \tag{6.14}$$

Since $\sup_{x\in U}[\mu_A(x)] = 1$ and x can take any values in U, for any $y \in V$ there exists $x \in U$ such that $\mu_A(x) \geq \mu_B(y)$. Thus (6.14) can be simplified to

$$\begin{aligned}\mu_{B'}(y) &= \sup_{x\in U}[\mu_A(x)\mu_B(y)] \\ &= \mu_B(y)\end{aligned} \tag{6.15}$$

If $A' = more\ or\ less\ A$, then from $\mu_A^{1/2}(x) \geq \mu_A(x) \geq \mu_A(x)\mu_B(x)$, we have

$$\begin{aligned}\mu_{B'}(y) &= \sup_{x\in U}\{min[\mu_A^{1/2}(x), \mu_A(x)\mu_B(y)]\} \\ &= \mu_B(y)\end{aligned} \tag{6.16}$$

Finally, if $A' = \bar{A}$, we obtain

$$\mu_{B'}(y) = \sup_{x \in U}\{min[1 - \mu_A(x), \mu_A(x)\mu_B(y)]\} \tag{6.17}$$

Since for fixed $y \in V$, $\mu_A(x)\mu_B(y)$ is an increasing function with $\mu_A(x)$ while $1 - \mu_A(x)$ is a decreasing function with $\mu_A(x)$, the $\sup_{x \in U} min$ in (6.17) is achieved when $1 - \mu_A(x) = \mu_A(x)\mu_B(y)$, that is, when $\mu_A(x) = \frac{1}{1+\mu_B(y)}$. Hence,

$$\mu_{B'}(y) = \frac{\mu_B(y)}{1 + \mu_B(y)} \tag{6.18}$$

From (6.13), (6.15), (6.16), (6.18), and Table 6.3 we see that the particular generalized modus ponens considered in this example satisfies critera p1, p3 and p5, but does not satisfy criteria p2, p4, p6, and p7. □

Example 6.3. In this example, we still use *min* for the t-norm but use Zadeh implication (5.25) for the $\mu_{A \to B}(x, y)$ in the generalized modus ponens (6.10). Again, we consider the four typical cases of A' in Example 6.2 and assume that $\sup_{x \in U}[\mu_A(x)] = 1$.

(a) For $A' = A$, we have

$$\mu_{B'}(y) = \sup_{x \in U} min\{\mu_A(x), max[min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x)]\} \tag{6.19}$$

Since $\sup_{x \in U} \mu_A(x) = 1$, the $\sup_{x \in U} min$ in (6.19) is achieved at the particular $x_0 \in U$ when

$$\mu_A(x_0) = max[min(\mu_A(x_0), \mu_B(y)), 1 - \mu_A(x_0)] \tag{6.20}$$

If $\mu_A(x_0) < \mu_B(y)$, then (6.20) becomes

$$\mu_A(x_0) = max[\mu_A(x_0), 1 - \mu_A(x_0)] \tag{6.21}$$

which is true when $\mu_A(x_0) \geq 0.5$; thus from (6.19) and (6.20) we have $\mu_{B'}(y) = \mu_A(x_0)$. Since $\sup_{x \in U}[\mu_A(x)] = 1$, it must be true that $\mu_A(x_0) = 1$, but this leads to $\mu_B(y) > \mu_A(x_0) = 1$, which is impossible. Thus, we cannot have $\mu_A(x_0) < \mu_B(y)$. Now consider the only possible case $\mu_A(x_0) \geq \mu_B(y)$. In this case, (6.20) becomes

$$\mu_A(x_0) = max[\mu_B(y), 1 - \mu_A(x_0)] \tag{6.22}$$

If $\mu_B(y) < 1 - \mu_A(x_0)$, then $\mu_A(x_0) = 1 - \mu_A(x_0)$, which is true when $\mu_A(x_0) = 0.5$. If $\mu_B(y) \geq 1 - \mu_A(x_0)$, then from (6.22) we have $\mu_A(x_0) = \mu_B(y) \geq 0.5$. Hence, $\mu_A(x_0) = max[0.5, \mu_B(y)]$ and we obtain

$$\mu_{B'}(y) = \mu_A(x_0) = max[0.5, \mu_B(y)] \tag{6.23}$$

(b) For $A' = very\ A$, we have

$$\mu_{B'}(y) = \sup_{x \in U} min\{\mu_A^2(x), max[min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x)]\} \tag{6.24}$$

Similar to the $A' = A$ case, the $\sup_{x \in U} min$ is achieved at $x_0 \in U$ when

$$\mu_A^2(x_0) = max[min(\mu_A(x_0), \mu_B(y)), 1 - \mu_A(x_0)] \tag{6.25}$$

If $\mu_A(x_0) < \mu_B(y)$, then

$$\mu_A^2(x_0) = max[\mu_A(x_0), 1 - \mu_A(x_0)] \tag{6.26}$$

which is true only when $\mu_A(x_0) = 1$, but this leads to the contradiction $\mu_B(y) > 1$. Thus $\mu_A(x_0) \geq \mu_B(y)$ is the only possible case. If $\mu_A(x_0) \geq \mu_B(y)$, then (6.25) becomes

$$\mu_A^2(x_0) = max[\mu_B(y), 1 - \mu_A(x_0)] \tag{6.27}$$

If $\mu_B(y) < 1 - \mu_A(x_0)$, then $\mu_A^2(x_0) = 1 - \mu_A(x_0)$, which is true when $\mu_A(x_0) = \frac{\sqrt{5}-1}{2}$. Hence, if $\mu_B(y) < 1 - \mu_A(x_0) = \frac{3-\sqrt{5}}{2}$, we have $\mu_{B'}(y) = \mu_A^2(x_0) = \frac{3-\sqrt{5}}{2}$. If $\mu_B(y) \geq 1 - \mu_A(x_0)$, we have $\mu_B(y) = \mu_A^2(x_0) = \mu_B(y) \geq \frac{3-\sqrt{5}}{2}$. In summary, we obtain

$$\mu_{B'}(y) = \mu_A^2(x_0) = max[\frac{3-\sqrt{5}}{2}, \mu_B(y)] \tag{6.28}$$

(c) If $A' = more\ or\ less\ A$, we have

$$\mu_{B'}(y) = \sup_{x \in U} min\{\mu_A^{1/2}(x), max[min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x)]\} \tag{6.29}$$

where the $\sup_{x \in U} min$ is achieved at $x_0 \in U$ when

$$\mu_A^{1/2}(x_0) = max[min(\mu_A(x_0), \mu_B(y)), 1 - \mu_A(x_0)] \tag{6.30}$$

Similar to the $A' = very\ A$ case, we can show that $\mu_A(x_0) < \mu_B(y)$ is impossible. For $\mu_A(x_0) \geq \mu_B(y)$, we have

$$\mu_A^{1/2}(x_0) = max[\mu_B(y), 1 - \mu_A(x_0)] \tag{6.31}$$

If $\mu_B(y) < 1 - \mu_A(x_0)$, then $\mu_A^{1/2}(x_0) = 1 - \mu_A(x_0)$, which is true when $\mu_A(x_0) = \frac{3-\sqrt{5}}{2}$. Thus, if $\mu_B(y) < 1 - \mu_A(x_0) = \frac{\sqrt{5}-1}{2}$, we have $\mu_{B'}(y) = \mu_A^{1/2}(x_0) = \frac{\sqrt{5}-1}{2}$. If $\mu_B(y) \geq 1 - \mu_A(x_0)$, we have $\mu_{B'}(y) = \mu_A^{1/2}(x_0) = \mu_B(y) \geq \frac{\sqrt{5}-1}{2}$. To summarize, we obtain

$$\mu_{B'}(y) = \mu_A^{1/2}(x_0) = max[\frac{\sqrt{5}-1}{2}, \mu_B(y)] \tag{6.32}$$

(d) Finally, when $A' = \bar{A}$, we have

$$\mu_{B'}(y) = \sup_{x \in U} min\{1 - \mu_A(x), max[min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x)]\} \tag{6.33}$$

By inspecting (6.33) we see that if we choose $x_0 \in U$ such that $\mu_A(x_0) = 0$, then $1 - \mu_A(x_0) = 1$ and $max[min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x)] = 1$, thus the $\sup_{x \in U} min$ is achieved at $x = x_0$. Hence, in this case we have

$$\mu_{B'}(y) = 1 \tag{6.34}$$

From (6.23), (6.28), (6.32), and (6.34), we see that for all the criteria in Table 6.3, only criterion p6 is satisfied. (This approximate reasoning is truely *approximate*!) □

6.3.2 Generalized Modus Tollens

Example 6.4. Similar to Example 6.2, we use *min* for the t-norm and Mamdani's product implication (5.32) for the $\mu_{A \to B}(x, y)$ in the generalized modus tollens (6.11). Consider four cases of B': (a) $B' = \bar{B}$, (b) $B' = not\ very\ B$, (c) $B' = not\ more\ or\ less\ B$, and (d) $B' = B$. We assume that $\sup_{y \in V}[\mu_B(y)] = 1$. If $B' = \bar{B}$, we have from (6.11) that

$$\mu_{A'}(x) = \sup_{y \in V} min[1 - \mu_B(y), \mu_A(x)\mu_B(y)] \tag{6.35}$$

The $\sup_{y \in V} min$ is achieved at $y_0 \in V$ when $1 - \mu_B(y_0) = \mu_A(x)\mu_B(y_0)$, which implies $\mu_B(y_0) = \frac{1}{1+\mu_A(x)}$, hence,

$$\mu_{A'}(x) = 1 - \mu_B(y_0) = \frac{\mu_A(x)}{1 + \mu_A(x)} \tag{6.36}$$

If $B' = not\ very\ B$, then

$$\mu_{A'}(x) = \sup_{y \in V} min[1 - \mu_B^2(y), \mu_A(x)\mu_B(y)] \tag{6.37}$$

where the $\sup_{y \in V} min$ is achieved at $y_0 \in V$ when $1 - \mu_B^2(y_0) = \mu_A(x)\mu_B(y_0)$, which gives $\mu_B(y_0) = \frac{\sqrt{\mu_A^2(x)+4} - \mu_A(x)}{2}$. Hence,

$$\mu_{A'}(x) = \mu_A(x)\mu_B(y_0) = \frac{\mu_A(x)\sqrt{\mu_A^2(x) + 4} - \mu_A^2(x)}{2} \tag{6.38}$$

If $B' = not\ more\ or\ less\ B$, we have

$$\mu_{A'}(x) = \sup_{y \in V} min[1 - \mu_B^{1/2}(y), \mu_A(x)\mu_B(y)] \tag{6.39}$$

Again, the $\sup_{y \in V} min$ is achieved at $y_0 \in V$ when $1 - \mu_B^{1/2}(y_0) = \mu_A(x)\mu_B(y_0)$, which gives $\mu_B(y_0) = \frac{1+2\mu_A(x) - \sqrt{\mu_A^2(x)+1}}{2\mu_A^2(x)}$. Hence,

$$\mu_{A'}(x) = \mu_A(x)\mu_B(y_0) = \frac{1 + 2\mu_A(x) - \sqrt{\mu_A^2(x) + 1}}{2\mu_A(x)} \tag{6.40}$$

Finally, when $B' = B$, we have

$$\begin{aligned}\mu_{A'}(x) &= \sup_{y \in V} min[\mu_B(y), \mu_A(x)\mu_B(y)] \\ &= \sup_{y \in V} \mu_A(x)\mu_B(y) \\ &= \mu_A(x) \end{aligned} \tag{6.41}$$

From (6.36), (6.38), (6.40) and (6.41) we see that among the seven intuitive criteria in Table 6.4 only criterion t5 is satisfied. □

6.3.3 Generalized Hypothetical Syllogism

Example 6.5. Similar to Examples 6.2 and 6.4, we use *min* for the t-norm and Mamdani product implication for the $\mu_{A \to B}(x, y)$ and $\mu_{B' \to C}(y, z)$ in the generalized hypothetical syllogism (6.12). We assume $\sup_{y \in V}[\mu_B(y)] = 1$ and consider four typical cases of B': (a) $B' = B$, (b) $B' = very\ B$, (c) $B' = more\ or\ less\ B$, and (d) $B' = \bar{B}$. If $B' = B$, we have from (6.12) that

$$\begin{aligned}\mu_{A \to C'}(x, z) &= \sup_{y \in V} min[\mu_A(x)\mu_B(y), \mu_B(y)\mu_C(z)] \\ &= (\sup_{y \in V} \mu_B(y)) min[\mu_A(x), \mu_C(z)] \\ &= min[\mu_A(x), \mu_C(z)] \end{aligned} \tag{6.42}$$

If $B' = very\ B$, we have

$$\mu_{A \to C'}(x, z) = \sup_{y \in V} min[\mu_A(x)\mu_B(y), \mu_B^2(y)\mu_C(z)] \tag{6.43}$$

If $\mu_A(x) > \mu_C(z)$, then it is always true that $\mu_A(x)\mu_B(y) > \mu_B^2(y)\mu_C(z)$, thus, $\mu_{A \to C'}(x, z) = \sup_{y \in V} \mu_B^2(y)\mu_C(z) = \mu_C(z)$. If $\mu_A(x) \leq \mu_C(z)$, then the $\sup_{y \in V} min$ is achieved at $y_0 \in V$, when $\mu_A(x)\mu_B(y_0) = \mu_B^2(y_0)\mu_C(z)$, which gives $\mu_B(y_0) = \frac{\mu_A(x)}{\mu_C(z)}$; hence, in this case $\mu_{A \to C'}(x, z) = \mu_A(x)\mu_B(y_0) = \frac{\mu_A^2(x)}{\mu_C(z)}$. In summary, we obtain

$$\mu_{A \to C'}(x, z) = \begin{cases} \mu_C(z) & if \quad \mu_C(z) < \mu_A(x) \\ \frac{\mu_A^2(x)}{\mu_C(z)} & if \quad \mu_C(z) \geq \mu_A(x) \end{cases} \tag{6.44}$$

If $B' = more\ or\ less\ B$, then using the same method as for the $B' = very\ B$ case, we have

$$\mu_{A \to C'}(x, z) = \begin{cases} \mu_A(x) & if \quad \mu_A(x) < \mu_C(z) \\ \frac{\mu_C^2(z)}{\mu_A(x)} & if \quad \mu_A(x) \geq \mu_C(z) \end{cases} \tag{6.45}$$

Finally, when $B' = \bar{B}$, we have

$$\mu_{A \to C'}(x, z) = \sup_{y \in V} min[\mu_A(x)\mu_B(y), (1 - \mu_B(y))\mu_C(z)] \tag{6.46}$$

where the $\sup_{y \in V} min$ is achieved at $y_0 \in V$ when $\mu_A(x)\mu_B(y_0) = (1-\mu_B(y_0))\mu_C(z)$, that is, when $\mu_B(y_0) = \frac{\mu_C(z)}{\mu_A(x)+\mu_C(z)}$. Hence,

$$\mu_{A \to C'}(x, z) = \frac{\mu_A(x)\mu_C(z)}{\mu_A(x) + \mu_C(z)} \tag{6.47}$$

□

6.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- Using truth tables to prove the equivalence of propositions.
- Basic inference rules (Modus Ponens, Modus Tollens, and Hypothetical Syllogism) and their generalizations to fuzzy propositions (Generalized Modus Ponens, Generalized Modus Tollens, and Generalized Hypothetical Syllogism).
- The idea and applications of the compositional rule of inference.
- Determining the resulting membership functions from different implication rules and typical cases of premises.

A comprehensive treatment of many-valued logic was prepared by Rescher [1969]. The generalizations of classical logic principles to fuzzy logic were proposed in Zadeh [1973], Zadeh [1975] and other papers of Zadeh in the 1970s. The compositional rule of inference also can be found in these papers of Zadeh.

6.5 Exercises

Exercise 6.1. Use the truth table method to prove that the following are tautologies: (a) modus ponens (6.3), (b) modus tollens (6.4), and (c) hypothetical syllogism (6.5).

Exercise 6.2. Let $U = \{x_1, x_2, x_3\}$ and $V = \{y_1, y_2\}$, and assume that a fuzzy IF-THEN rule "IF x is A, THEN y is B" is given, where $A = .5/x_1 + 1/x_2 + .6/x_3$ and $B = 1/y_1 + .4/y_2$. Then, given a fact "x is A'," where $A' = .6/x_1 + .9/x_2 + .7/x_3$, use the generalized modus ponens (6.10) to derive a conclusion in the form "y is B'," where the fuzzy relation $A \to B$ is interpreted using:

(a) Dienes-Rescher implication (5.23),

(b) Lukasiewicz implication (5.24),

(c) Zadeh implication (5.25), and

(d) Mamdani Product implication (5.32).

Exercise 6.3. Repeat Exercise 6.2 with $A = .6/x_1 + 1/x_2 + .9/x_3$, $B = .6/y_1 + 1/y_2$, and $A' = .5/x_1 + .9/x_2 + 1/x_3$.

Exercise 6.4. Let U, V, A, and B be the same as in Exercise 6.2. Now given a fact "y is B'," where $B' = .9/y_1 + .7/y_2$, use the generalized modus tollens (6.11) to derive a conclusion "x is A'," where the fuzzy relation $A \rightarrow B$ is interpreted using:

(a) Lukasiewicz implication (5.24), and

(b) Mamdani Product implication (5.32).

Exercise 6.5. Use *min* for the t-norm and Lukasiewicz implication (5.24) for the $\mu_{A\rightarrow B}(x, y)$ in the generalized modus ponens (6.10), and determine the membership function $\mu_{B'}(y)$ in terms of $\mu_B(y)$ for: (a) $A' = A$, (b) $A' = very\ A$, (c) $A' = more\ or\ less\ A$, and (d) $A' = \bar{A}$.

Exercise 6.6. Use *min* for the t-norm and Dienes-Rescher implication (5.23) for the $\mu_{A\rightarrow B}(x, y)$ in the generalized modus ponens (6.10), and determine the membership function $\mu_{B'}(y)$ in terms of $\mu_B(y)$ for: (a) $A' = A$, (b) $A' = very\ A$, (c) $A' = more\ or\ less\ A$, and (d) $A' = \bar{A}$.

Exercise 6.7. With *min* as the t-norm and Mamdani minimum implication (5.31) for the $\mu_{A\rightarrow B}(x, y)$ in the generalized modus tollens (6.11), determine the membership function $\mu_{A'}(x)$ in terms of $\mu_A(x)$ for: (a) $B' = \bar{B}$, (b) $B' = not\ very\ B$, (c) $B' = not\ more\ or\ less\ B$, and (d) $B' = B$.

Exercise 6.8. Consider a fuzzy logic based on the standard operation $(min, max, 1 - a)$. For any two arbitrary propositions, A and B, in the logic, assume that we require that the equality

$$\overline{A \wedge \bar{B}} = B \vee (\bar{A} \wedge \bar{B}) \tag{6.48}$$

holds. Imposing such requirement means that pairs of truth values of A and B become restricted to a subset of $[0, 1]^2$. Show exactly how they are restricted.

Part II

Fuzzy Systems and Their Properties

We learned in Chapter 1 that a fuzzy system consists of four components: fuzzy rule base, fuzzy inference engine, fuzzifier and defuzzifier, as shown in Fig. 1.5. In this part (Chapters 7-11), we will study each of the four components in detail. We will see how the fuzzy mathematical and logic principles we learned in Part I are used in the fuzzy systems. We will derive the compact mathematical formulas for different types of fuzzy systems and study their approximation properties.

In Chapter 7, we will analyze the structure of fuzzy rule base and propose a number of specific fuzzy inference engines. In Chapter 8, a number of fuzzifiers and defuzzifiers will be proposed and analyzed in detail. In Chapter 9, the fuzzy inference engines, fuzzifiers, and defuzzifiers proposed in Chapters 7 and 8 will be combined to obtain some specific fuzzy systems that will be proven to have the universal approximation property. In Chapters 10 and 11, the approximation accuracy of fuzzy systems will be studied in detail and we will show how to design fuzzy systems to achieve any specified accuracy.

Chapter 7

Fuzzy Rule Base and Fuzzy Inference Engine

Consider the fuzzy system shown in Fig. 1.5, where $U = U_1 \times U_2 \times \cdots \times U_n \subset R^n$ and $V \subset R$. We consider only the multi-input-single-output case, because a multi-output system can always be decomposed into a collection of single-output systems. For example, if we are asked to design a 4-input-3-output fuzzy system, we can first design three 4-input-1-output fuzzy systems separately and then put them together as in Fig. 7.1.

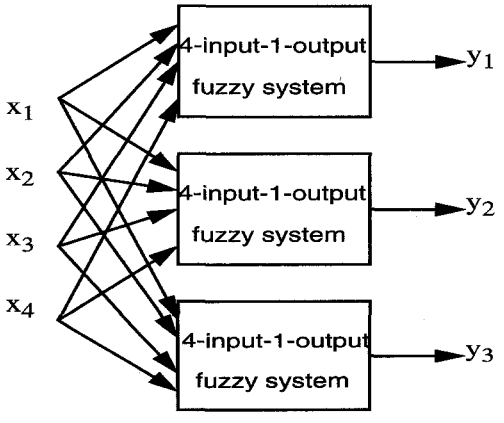


Figure 7.1. A multi-input-multi-output fuzzy system can be decomposed into a collection of multi-input-single-output fuzzy systems.

In this chapter, we will study the details inside the fuzzy rule base and fuzzy

inference engine; fuzzifiers and defuzzifiers will be studied in the next chapter.

7.1 Fuzzy Rule Base

7.1.1 Structure of Fuzzy Rule Base

A *fuzzy rule base* consists of a set of fuzzy IF-THEN rules. It is the heart of the fuzzy system in the sense that all other components are used to implement these rules in a reasonable and efficient manner. Specifically, the fuzzy rule base comprises the following fuzzy IF-THEN rules:

$$Ru^{(l)}: IF\ x_1\ is\ A_1^l\ and\ ...\ and\ x_n\ is\ A_n^l,\ THEN\ y\ is\ B^l \tag{7.1}$$

where A_i^l and B^l are fuzzy sets in $U_i \subset R$ and $V \subset R$, respectively, and $\mathbf{x} = (x_1, x_2, ..., x_n)^T \in U$ and $y \in V$ are the input and output (linguistic) variables of the fuzzy system, respectively. Let M be the number of rules in the fuzzy rule base; that is, $l = 1, 2, ..., M$ in (7.1). We call the rules in the form of (7.1) *canonical fuzzy IF-THEN rules* because they include many other types of fuzzy rules and fuzzy propositions as special cases, as shown in the following lemma.

Lemma 7.1. The canonical fuzzy IF-THEN rules in the form of (7.1) include the following as special cases:

(a) "Partial rules":

$$IF\ x_1\ is\ A_1^l\ and\ ...\ and\ x_m\ is\ A_m^l,\ THEN\ y\ is\ B^l \tag{7.2}$$

where $m < n$.

(b) "Or rules":

$$IF\ x_1\ is\ A_1^l\ and\ ...\ and\ x_m\ is\ A_m^l\ or\ x_{m+1}\ is\ A_{m+1}^l\ and\ ...\ and\ x_n\ is\ A_n^l,\ THEN\ y\ is\ B^l \tag{7.3}$$

(c) Single fuzzy statement:

$$y\ is\ B^l \tag{7.4}$$

(d) "Gradual rules," for example:

$$The\ smaller\ the\ x,\ the\ bigger\ the\ y \tag{7.5}$$

(e) Non-fuzzy rules (that is, conventional production rules).

Proof: The partial rule (7.2) is equivalent to

$$IF x_1\ is\ A_1^l\ and\ ...\ and\ x_m\ is\ A_m^l\ and\ x_{m+1}\ is\ I\ and\ ...\ and\ x_n\ is\ I,\ THEN\ y\ is\ B^l \tag{7.6}$$

where I is a fuzzy set in R with $\mu_I(x) \equiv 1$ for all $x \in R$. The preceding rule is in the form of (7.1); this proves (a). Based on intuitive meaning of the logic operator "or," the "Or rule" (7.3) is equivalent to the following two rules:

$$IF x_1 \text{ is } A_1^l \text{ and } ... \text{ and } x_m \text{ is } A_m^l, \text{ THEN } y \text{ is } B^l \tag{7.7}$$

$$IF x_{m+1} \text{ is } A_{m+1}^l \text{ and } ... \text{ and } x_n \text{ is } A_n^l, \text{ THEN } y \text{ is } B^l \tag{7.8}$$

From (a) we have that the two rules (7.7) and (7.8) are special cases of (7.1); this proves (b). The fuzzy statement (7.4) is equivalent to

$$IF\ x_1 \text{ is } I \text{ and } ... \text{ and } x_n \text{ is } I, \text{ THEN } y \text{ is } B^l \tag{7.9}$$

which is in the form of (7.1); this proves (c). For (d), let S be a fuzzy set representing "smaller," for example, $\mu_S(x) = 1/(1 + exp(5(x + 2)))$, and B be a fuzzy set representing "bigger," for example, $\mu_B(y) = 1/(1 + exp(-5(y - 2)))$, then the "Gradual rule" (7.5) is equivalent to

$$IF\ x \text{ is } S, \text{ THEN } y \text{ is } B \tag{7.10}$$

which is a special case of (7.1); this proves (d). Finally, if the membership functions of A_i^l and B^l can only take values 1 or 0, then the rules (7.1) become non-fuzzy rules. □

In our fuzzy system framework, human knowledge has to be represented in the form of the fuzzy IF-THEN rules (7.1). That is, we can only utilize human knowledge that can be formulated in terms of the fuzzy IF-THEN rules. Fortunately, Lemma 7.1 ensures that these rules provide a quite general knowledge representation scheme.

7.1.2 Properties of Set of Rules

Because the fuzzy rule base consists of a set of rules, the relationship among these rules and the rules as a whole impose interesting questions. For example, do the rules cover all the possible situations that the fuzzy system may face? Are there any conflicts among these rules? To answer these sorts of questions, we introduce the following concepts.

Definition 7.1. A set of fuzzy IF-THEN rules is *complete* if for any $\mathbf{x} \in U$, there exists at least one rule in the fuzzy rule base, say rule $Ru^{(l)}$ (in the form of (7.1)), such that

$$\mu_{A_i^l}(x_i) \neq 0 \tag{7.11}$$

for all $i = 1, 2, ..., n$.

Intuitively, the completeness of a set of rules means that at any point in the input space there is at least one rule that "fires"; that is, the membership value of the IF part of the rule at this point is non-zero.

Example 7.1. Consider a 2-input-1-output fuzzy system with $U = U_1 \times U_2 = [0,1] \times [0,1]$ and $V = [0,1]$. Define three fuzzy sets S_1, M_1 and L_1 in U_1, and two fuzzy sets S_2 and L_2 in U_2, as shown in Fig. 7.2. In order for a fuzzy rule base to be complete, it must contain the following six rules whose IF parts constitute all the possible combinations of S_1, M_1, L_1 with S_2, L_2:

$$\begin{aligned}
&IF\ x_1\ is\ S_1\ and\ x_2\ is\ S_2,\ THEN\ y\ is\ B^1\\
&IF\ x_1\ is\ S_1\ and\ x_2\ is\ L_2,\ THEN\ y\ is\ B^2\\
&IF\ x_1\ is\ M_1\ and\ x_2\ is\ S_2,\ THEN\ y\ is\ B^3\\
&IF\ x_1\ is\ M_1\ and\ x_2\ is\ L_2,\ THEN\ y\ is\ B^4\\
&IF\ x_1\ is\ L_1\ and\ x_2\ is\ S_2,\ THEN\ y\ is\ B^5\\
&IF\ x_1\ is\ L_1\ and\ x_2\ is\ L_2,\ THEN\ y\ is\ B^6
\end{aligned} \tag{7.12}$$

where B^l $(l = 1, 2, ..., 6)$ are fuzzy sets in V. If any rule in this group is missing, then we can find point $x^* \in U$, at which the IF part propositions of all the remaining rules have zero membership value. For example, if the second rule in (7.12) is missing, then this $x^* = (0,1)$ (Why?). □

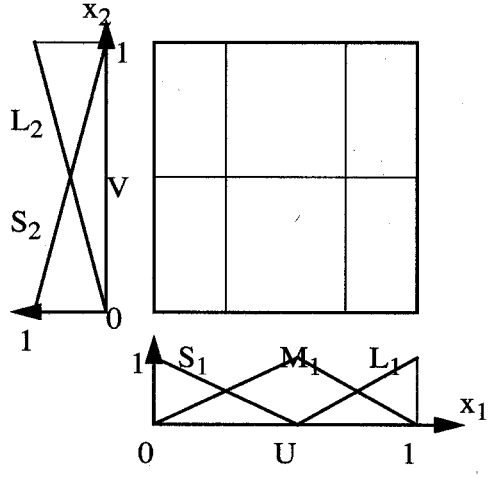


Figure 7.2. An example of membership functions for a two-input fuzzy system.

From Example 7.1 we see that if we use the triangular membership functions as in Fig. 7.2, the number of rules in a *complete* fuzzy rule base increases exponentially with the dimension of the input space U. This problem is called the *curse of dimensionality* and will be further discussed in Chapter 22.

Definition 7.2. A set of fuzzy IF-THEN rules is *consistent* if there are no rules with the same IF parts but different THEN parts.

For nonfuzzy production rules, consistence is an important requirement because it is difficult to continue the search if there are conflicting rules. For fuzzy rules, however, consistence is not critical because we will see later that if there are conflicting rules, the fuzzy inference engine and the defuzzifier will automatically average them out to produce a compromised result. Of course, it is better to have a consistent fuzzy rule base in the first place.

Definition 7.3. A set of fuzzy IF-THEN rules is *continuous* if there do not exist such neighboring rules whose THEN part fuzzy sets have empty intersection.

Intuitively, continuity means that the input-output behavior of the fuzzy system should be smooth. It is difficult to explain this concept in more detail at this point, because we have not yet derived the complete formulas of the fuzzy systems, but it will become clear as we move into Chapter 9.

7.2 Fuzzy Inference Engine

In a *fuzzy inference engine*, fuzzy logic principles are used to combine the fuzzy IF-THEN rules in the fuzzy rule base into a mapping from a fuzzy set A' in U to a fuzzy set B' in V. In Chapter 6 we learned that a fuzzy IF-THEN rule is interpreted as a fuzzy relation in the input-output product space $U \times V$, and we proposed a number of implications that specify the fuzzy relation. If the fuzzy rule base consists of only a single rule, then the generalized modus ponens (6.10) specifies the mapping from fuzzy set A' in U to fuzzy set B' in V. Because any practical fuzzy rule base constitutes more than one rule, the key question here is how to infer with a set of rules. There are two ways to infer with a set of rules: composition based inference and individual-rule based inference, which we will discuss next.

7.2.1 Composition Based Inference

In *composition based inference*, all rules in the fuzzy rule base are combined into a single fuzzy relation in $U \times V$, which is then viewed as a single fuzzy IF-THEN rule. So the key question is how to perform this combination. We should first understand what a set of rules mean intuitively, and then we can use appropriate logic operators to combine them.

There are two opposite arguments for what a set of rules should mean. The first one views the rules as independent conditional statements. If we accept this point of view, then a reasonable operator for combining the rules is *union*. The second one views the rules as strongly coupled conditional statements such that the conditions of all the rules must be satisfied in order for the whole set of rules to have an impact. If we adapt this view, then we should use the operator *intersection*

to combine the rules. The second view may look strange, but for some implications, for example, the Gödel implication (5.26), it makes sense, as we will see very soon in this section. We now show the details of these two schemes.

Let $Ru^{(l)}$ be a fuzzy relation in $U \times V$, which represents the fuzzy IF-THEN rule (7.1); that is, $Ru^{(l)} = A_1^l \times \cdots \times A_n^l \rightarrow B^l$. From Chapter 5 we know that $A_1^l \times \cdots \times A_n^l$ is a fuzzy relation in $U = U_1 \times \cdots \times U_n$ defined by

$$\mu_{A_1^l \times \cdots \times A_n^l}(x_1, ..., x_n) = \mu_{A_1^l}(x_1) \star \cdots \star \mu_{A_n^l}(x_n) \tag{7.13}$$

where $\star$ represents any t-norm operator. The implication $\rightarrow$ in $Ru^{(l)}$ is defined according to various implications (5.23)-(5.26), (5.31), and (5.32). If we accept the first view of a set of rules, then the M rules in the form of (7.1) are interpreted as a single fuzzy relation Q_M in $U \times V$ defined by

$$Q_M = \bigcup_{l=1}^{M} Ru^{(l)} \tag{7.14}$$

This combination is called the *Mamdani combination.* If we use the symbol $\dot{+}$ to represent the s-norm, then (7.14) can be rewritten as

$$\mu_{Q_M}(\mathbf{x}, y) = \mu_{Ru^{(1)}}(\mathbf{x}, y) \dot{+} \cdots \dot{+} \mu_{Ru^{(M)}}(\mathbf{x}, y) \tag{7.15}$$

For the second view of a set of rules, the M fuzzy IF-THEN rules of (7.1) are interpreted as a fuzzy relation Q_G in $U \times V$, which is defined as

$$Q_G = \bigcap_{l=1}^{M} Ru^{(l)} \tag{7.16}$$

or equivalently,

$$\mu_{Q_G}(\mathbf{x}, y) = \mu_{Ru^{(1)}}(\mathbf{x}, y) \star \cdots \star \mu_{Ru^{(M)}}(\mathbf{x}, y) \tag{7.17}$$

where $\star$ denotes t-norm. This combination is called the *Gödel combination.*

Let A' be an arbitrary fuzzy set in U and be the input to the fuzzy inference engine. Then, by viewing Q_M or Q_G as a single fuzzy IF-THEN rule and using the generalized modus ponens (6.10), we obtain the output of the fuzzy inference engine as

$$\mu_{B'}(y) = \sup_{\mathbf{x} \in U} t[\mu_{A'}(\mathbf{x}), \mu_{Q_M}(\mathbf{x}, y)] \tag{7.18}$$

if we use the Mamdani combination, or as

$$\mu_{B'}(y) = \sup_{\mathbf{x} \in U} t[\mu_{A'}(\mathbf{x}), \mu_{Q_G}(\mathbf{x}, y)] \tag{7.19}$$

if we use the Gödel combination.

In summary, the computational procedure of the composition based inference is given as follows:

- **Step 1**: For the M fuzzy IF-THEN rules in the form of (7.1), determine the membership functions $\mu_{A_1^l \times \cdots \times A_n^l}(x_1, ..., x_n)$ for $l = 1, 2, ..., M$ according to (7.13).

- **Step 2**: View $A_1^l \times \cdots \times A_n^l$ as the FP_1 and B^l as the FP_2 in the implications (5.23)-(5.26), (5.31) and (5.32), and determine $\mu_{Ru^{(l)}}(x_1, ..., x_n, y) = \mu_{A_1^l \times \cdots \times A_n^l \to B^l}(x_1, ..., x_n, y)$ for $l = 1, 2, ..., M$ according to any one of these implications.

- **Step 3**: Determine $\mu_{Q_M}(\mathbf{x}, y)$ or $\mu_{Q_G}(\mathbf{x}, y)$ according to (7.15) or (7.17).

- **Step 4**: For given input A', the fuzzy inference engine gives output B' according to (7.18) or (7.19).

7.2.2 Individual-Rule Based Inference

In *individual-rule based inference*, each rule in the fuzzy rule base determines an output fuzzy set and the output of the whole fuzzy inference engine is the combination of the M individual fuzzy sets. The combination can be taken either by union or by intersection.

The computational procedure of the individual-rule based inference is summarized as follows:

- **Steps 1 and 2**: Same as the Steps 1 and 2 for the composition based inference.

- **Step 3**: For given input fuzzy set A' in U, compute the output fuzzy set B_l' in V for each individual rule $Ru^{(l)}$ according to the generalized modus ponens (6.10), that is,

$$\mu_{B_l'}(y) = \sup_{\mathbf{x} \in U} t[\mu_{A'}(\mathbf{x}), \mu_{Ru^{(l)}}(\mathbf{x}, y)] \tag{7.20}$$

  for $l = 1, 2, ..., M$.

- **Step 4**: The output of the fuzzy inference engine is the combination of the M fuzzy sets $\{B_1', ..., B_M'\}$ either by union, that is,

$$\mu_{B'}(y) = \mu_{B_1'}(y) \dot{+} \cdots \dot{+} \mu_{B_M'}(y) \tag{7.21}$$

  or by intersection, that is,

$$\mu_{B'}(y) = \mu_{B_1'}(y) \star \cdots \star \mu_{B_M'}(y) \tag{7.22}$$

  where $\dot{+}$ and $\star$ denote s-norm and t-norm operators, respectively.

7.2.3 The Details of Some Inference Engines

From the previous two subsections we see that there are a variety of choices in the fuzzy inference engine. Specifically, we have the following alternatives: (i) composition based inference or individual-rule based inference, and within the composition based inference, Mamdani inference or Gödel inference, (ii) Dienes-Rescher implication (5.23), Lukasiewicz implication (5.24), Zadeh implication (5.25), Gödel implication (5.26), or Mamdani implications (5.31)-(5.32), and (iii) different operations for the t-norms and s-norms in the various formulas. So a natural question is: how do we select from these alternatives?

In general, the following three criteria should be considered:

- *Intuitive appeal*: The choice should make sense from an intuitive point of view. For example, if a set of rules are given by a human expert who believes that these rules are independent of each other, then they should be combined by union.
- *Computational efficiency*: The choice should result in a formula relating B' with A', which is simple to compute.
- *Special properties*: Some choice may result in an inference engine that has special properties. If these properties are desirable, then we should make this choice.

We now show the detailed formulas of a number of fuzzy inference engines that are commonly used in fuzzy systems and fuzzy control.

- **Product Inference Engine**: In *product inference engine*, we use: (i) individual-rule based inference with union combination (7.21), (ii) Mamdani's product implication (5.32), and (iii) algebraic product for all the t-norm operators and *max* for all the s-norm operators. Specifically, from (7.20), (7.21), (5.32), and (7.13), we obtain the product inference engine as

$$\mu_{B'}(y) = \max_{l=1}^{M}[\sup_{\mathbf{x}\in U}(\mu_{A'}(\mathbf{x})\prod_{i=1}^{n}\mu_{A_i^l}(x_i)\mu_{B^l}(y))] \tag{7.23}$$

  That is, given fuzzy set A' in U, the product inference engine gives the fuzzy set B' in V according to (7.23).

- **Minimum Inference Engine**: In *minimum inference engine*, we use: (i) individual-rule based inference with union combination (7.21), (ii) Mamdani's minimum implication (5.31), and (iii) *min* for all the t-norm operators and *max* for all the s-norm operators. Specifically, from (7.20), (7.21), (5.31), and (7.13) we have

$$\mu_{B'}(y) = \max_{l=1}^{M}[\sup_{\mathbf{x}\in U} min(\mu_{A'}(\mathbf{x}), \mu_{A_1^l}(x_1), ..., \mu_{A_n^l}(x_n), \mu_{B^l}(y))] \tag{7.24}$$

That is, given fuzzy set A' in U, the minimum inference engine gives the fuzzy set B' in V according to (7.24).

The product inference engine and the minimum inference engine are the most commonly used fuzzy inference engines in fuzzy systems and fuzzy control. The main advantage of them is their computational simplicity; this is especially true for the product inference engine (7.23). Also, they are intuitively appealing for many practical problems, especially for fuzzy control. We now show some properties of the product and minimum inference engines.

Lemma 7.2. The product inference engine is unchanged if we replace "individual-rule based inference with union combination (7.21)" by "composition based inference with Mamdani combination (7.15)."

Proof: From (7.15) and (7.18) we have

$$\mu_{B'}(y) = \sup_{\mathbf{x}\in U}[\mu_{A'}(\mathbf{x}) \max_{l=1}^{M}(\mu_{Ru^{(l)}}(\mathbf{x}, y))] \tag{7.25}$$

Using (5.32) and (7.13), we can rewrite (7.25) as

$$\mu_{B'}(y) = \sup_{\mathbf{x}\in U} \max_{l=1}^{M}[\mu_{A'}(\mathbf{x}) \prod_{i=1}^{n} \mu_{A_i^l}(x_i)\mu_{B^l}(y)] \tag{7.26}$$

Because the $\max_{l=1}^{M}$ and $\sup_{\mathbf{x}\in U}$ are interchangeable, (7.26) is equivalent to (7.23). □

Lemma 7.3. If the fuzzy set A' is a fuzzy singleton, that is, if

$$\mu_{A'}(\mathbf{x}) = \begin{cases} 1 \; if \; \mathbf{x} = \mathbf{x}^* \\ 0 \; otherwise \end{cases} \tag{7.27}$$

where $\mathbf{x}^*$ is some point in U, then the product inference engine is simplified to

$$\mu_{B'}(y) = \max_{l=1}^{M}[\prod_{i=1}^{n} \mu_{A_i^l}(x_i^*)\mu_{B^l}(y)] \tag{7.28}$$

and the minimum inference engine is simplified to

$$\mu_{B'}(y) = \max_{l=1}^{M}[min(\mu_{A_1^l}(x_1^*), ..., \mu_{A_n^l}(x_n^*), \mu_{B^l}(y))] \tag{7.29}$$

Proof: Substituting (7.27) into (7.23) and (7.24), we see that the $\sup_{\mathbf{x}\in U}$ is achieved at $\mathbf{x} = \mathbf{x}^*$. Hence, (7.23) reduces (7.28) and (7.24) reduces (7.29). □

Lemma 7.2 shows that although the individual-rule based and composition based inferences are conceptually different, they produce the same fuzzy inference engine in certain important cases. Lemma 7.3 indicates that the computation within the

fuzzy inference engine can be greatly simplified if the input is a fuzzy singleton (the most difficult computation in (7.23) and (7.24) is the $\sup_{\mathbf{x}\in U}$, which disappears in (7.28) and (7.29)).

A disadvantage of the product and minimum inference engines is that if at some $\mathbf{x} \in U$ the $\mu_{A_i^l}(x_i)$'s are very small, then the $\mu_{B'}(y)$ obtained from (7.23) or (7.24) will be very small. This may cause problems in implementation. The following three fuzzy inference engines overcome this disadvantage.

- **Lukasiewicz Inference Engine**: In *Lukasiewicz inference engine*, we use: (i) individual-rule based inference with intersection combination (7.22), (ii) Lukasiewicz implication (5.24), and (iii) *min* for all the t-norm operators. Specifically, from (7.22), (7.20), (5.24) and (7.13) we obtain

$$\begin{aligned}\mu_{B'}(y) &= \min_{l=1}^{M}[\sup_{\mathbf{x}\in U} min(\mu_{A'}(\mathbf{x}), \mu_{Ru^{(l)}}(\mathbf{x}, y))] \\ &= \min_{l=1}^{M}\{\sup_{\mathbf{x}\in U} min[\mu_{A'}(\mathbf{x}), min(1, 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)) + \mu_{B^l}(y))]\} \\ &= \min_{l=1}^{M}\{\sup_{\mathbf{x}\in U} min[\mu_{A'}(\mathbf{x}), 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)) + \mu_{B^l}(y)]\} \qquad (7.30)\end{aligned}$$

  That is, for given fuzzy set A' in U, the Lukasiewicz inference engine gives the fuzzy set B' in V according to (7.30).

- **Zadeh Inference Engine**: In *Zadeh inference engine*, we use: (i) individual-rule based inference with intersection combination (7.22), (ii) Zadeh implication (5.25), and (iii) *min* for all the t-norm operators. Specifically, from (7.22), (7.20), (5.25), and (7.13) we obtain

$$\begin{aligned}\mu_{B'}(y) = \min_{l=1}^{M}\{\sup_{\mathbf{x}\in U} min[\mu_{A'}(\mathbf{x}), max(min(\mu_{A_1^l}(x_1), ..., \mu_{A_n^l}(x_n), \mu_{B^l}(y)), \\ 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)))]\} \qquad (7.31)\end{aligned}$$

- **Dienes-Rescher Inference Engine**: In *Dienes-Rescher inference engine*, we use the same operations as in the Zadeh inference engine, except that we replace the Zadeh implication (5.25) with the Dienes-Rescher implication (5.23). Specifically, we obtain from (7.22), (7.20), (5.23), and (7.13) that

$$\mu_{B'}(y) = \min_{l=1}^{M}\{\sup_{\mathbf{x}\in U} min[\mu_{A'}(\mathbf{x}), max(1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)), \mu_{B^l}(y))]\} \qquad (7.32)$$

Similar to Lemma 7.3, we have the following results for the Lukasiewicz, Zadeh and Dienes-Rescher inference engines.

Lemma 7.4: If A' is a fuzzy singleton as defined by (7.27), then the Lukasiewicz, Zadeh and Dienes-Rescher inference engines are simplified to

$$\mu_{B'}(y) = \min_{l=1}^{M}[1, 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i^*)) + \mu_{B^l}(y)] \tag{7.33}$$

$$\mu_{B'}(y) = \min_{l=1}^{M}\{max[min(\mu_{A_1^l}(x_1^*), ..., \mu_{A_n^l}(x_n^*), \mu_{B^l}(y)), 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_l^*))]\} \tag{7.34}$$

$$\mu_{B'}(y) = \min_{l=1}^{M}\{max[1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i^*)), \mu_{B^l}(y)]\} \tag{7.35}$$

respectively.

Proof: Using the same arguments as in the proof of Lemma 7.3, we can prove this lemma. □

We now have proposed five fuzzy inference engines. Next, we compare them through two examples.

Example 7.2: Suppose that a fuzzy rule base consists of only one rule

$$IF\ x_1\ is\ A_1\ and\ \cdots\ and\ x_n\ is\ A_n,\ THEN\ y\ is\ B \tag{7.36}$$

where

$$\mu_B(y) = \begin{cases} 1 - |y|\ if\ -1 \le y \le 1 \\ 0\ otherwise \end{cases} \tag{7.37}$$

Assume that A' is a fuzzy singleton defined by (7.27). We would like to plot the $\mu_{B'}(y)$ obtained from the five fuzzy inference engines. Let B'_P, B'_M, B'_L, B'_Z and B'_D be the fuzzy set B' using the product, minimum, Lukasiewicz, Zadeh and Dienes-Rescher inference engines, respectively, and let $min[\mu_{A_1}(x_1^*), ..., \mu_{A_n}(x_n^*)] = \mu_{A_p}(x_p^*)$ and $\prod_{i=1}^{n}\mu_{A_i}(x_i^*) = \mu_A(\mathbf{x}^*)$. Then from (7.28), (7.29), and (7.33)-(7.35) we have

$$\mu_{B'_P}(y) = \mu_A(\mathbf{x}^*)\mu_B(y) \tag{7.38}$$

$$\mu_{B'_M}(y) = min[\mu_{A_p}(x_p^*), \mu_B(y)] \tag{7.39}$$

$$\mu_{B'_L}(y) = min[1, 1 - \mu_{A_p}(x_p^*) + \mu_B(y)] \tag{7.40}$$

$$\mu_{B'_Z}(y) = max[min(\mu_{A_p}(x_p^*), \mu_B(y)), 1 - \mu_{A_p}(x_p^*)] \tag{7.41}$$

$$\mu_{B'_D}(y) = max[1 - \mu_{A_p}(x_p^*), \mu_B(y)] \tag{7.42}$$

For the case of $\mu_{A_p}(x_p^*) \ge 0.5$, $\mu_{B'_P}(y)$ and $\mu_{B'_M}(y)$ are plotted in Fig. 7.3, and $\mu_{B'_L}(y), \mu_{B'_Z}(y)$ and $\mu_{B'_D}(y)$ are plotted in Fig. 7.4. For the case of $\mu_{A_p}(x_p^*) < 0.5$, $\mu_{B'_P}(y)$ and $\mu_{B'_M}(y)$ are plotted in Fig. 7.5, and $\mu_{B'_L}(y), \mu_{B'_Z}(y)$ and $\mu_{B'_D}(y)$ are plotted in Fig. 7.6.

From Figs.7.3-7.6 we have the following observations: (i) if the membership value of the IF part at point $\mathbf{x}^*$ is small (say, $\mu_{A_p}(x_p^*) < 0.5$), then the product

and minimum inference engines give very small membership values, whereas the Lukasiewicz, Zadeh and Dienes-Rescher inference engines give very large membership values; (ii) the product and minimum inference engines are similar, while the Lukasiewicz, Zadeh and Dienes-Rescher inference engines are similar, but there are big differences between these two groups; and (iii) the Lukasiewicz inference engine gives the largest output membership function, while the product inference engine gives the smallest output membership function in all the cases; the other three inference engines are in between. □

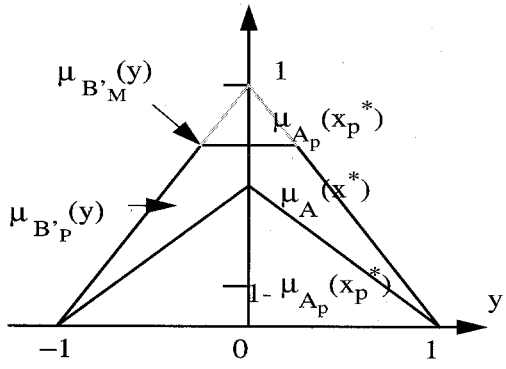


Figure 7.3. Output membership functions using the Lukasiewicz, Zadeh and Dienes-Rescher inference engines for the $\mu_{A_p^*}(x_p^*) \geq 0.5$ case.

Example 7.3: In this example, we consider that the fuzzy system contains two rules: one is the same as (7.36), and the other is

$$IF\ x_1\ is\ C_1\ and\ \cdots\ and\ x_n\ is\ C_n,\ THEN\ y\ is\ D \tag{7.43}$$

where

$$\mu_D(y) = \begin{cases} 1 - |y-1|\ if\ 0 \leq y \leq 2 \\ 0\ otherwise \end{cases} \tag{7.44}$$

Assume again that A' is the fuzzy singleton defined by (7.27). We would like to plot the $\mu_{B'}(y)$ using the product inference engine, that is, $\mu_{B'_P}(y)$. Fig. 7.7 shows the $\mu_{B'_P}(y)$, where $\mu_A(\mathbf{x}^*) = \prod_{i=1}^n \mu_{A_i}(x_i^*)$ and $\mu_C(\mathbf{x}^*) = \prod_{i=1}^n \mu_{C_i}(x_i^*)$. □

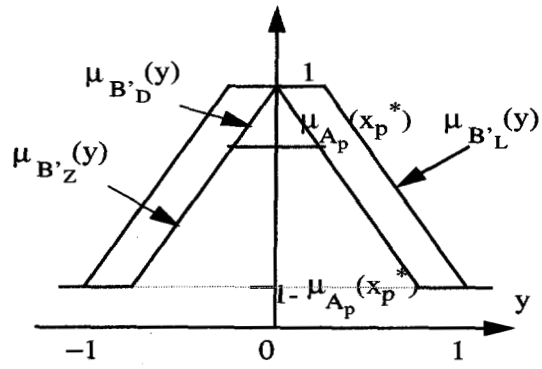


Figure 7.4. Output membership functions using the product and minimum inference engines for the $\mu_{A_p^*}(x_p^*) \geq 0.5$ case.

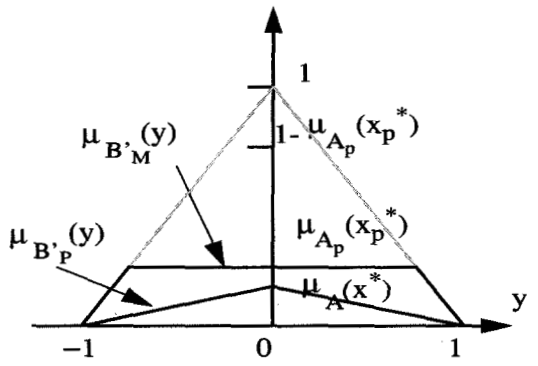


Figure 7.5. Output membership functions using the product and minimum inference engines for the $\mu_{A_p^*}(x_p^*) < 0.5$ case.

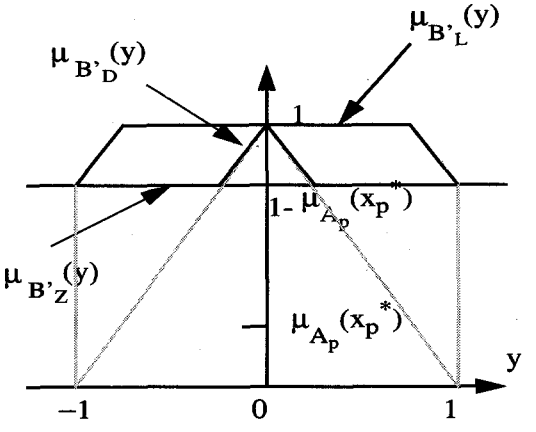


Figure 7.6. Output membership functions using the Lukasiewicz, Zadeh and Dienes-Rescher inference engines for the $\mu_{A_p^*}(x_p^*) < 0.5$ case.

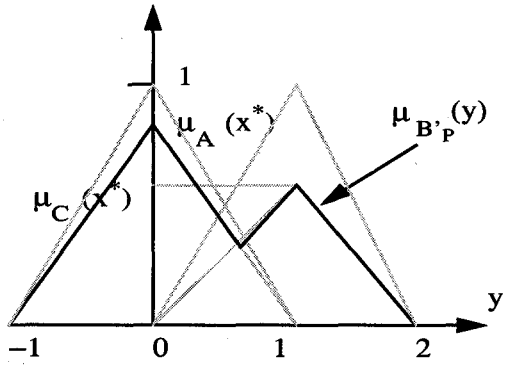


Figure 7.7. Output membership function using the product inference engine for the case of two rules.

7.3 Summary and Further Readings

In this chapter we have demonstrated the following:

- The structure of the canonical fuzzy IF-THEN rules and the criteria for evaluating a set of rules.
- The computational procedures for the composition based and individual-rule based inferences.
- The detailed formulas of the five specific fuzzy inference engines: product, minimum, Lukasiewicz, Zadeh and Dienes-Rescher inference engines, and their computations for particular examples.

Lee [1990] provided a very good survey on fuzzy rule bases and fuzzy inference engines. This paper gives intuitive analyses for various issues associated with fuzzy rule bases and fuzzy inference engines. A mathematical analysis of fuzzy inference engines, similar to the approach in this chapter, were prepared by Driankov, Hellendoorn and Reinfrank [1993].

7.4 Exercises

Exercise 7.1. If the third and sixth rules in (7.12) (Example 7.1) are missing, at what points do the IF part propositions of all the remaining rules have zero membership values?

Exercise 7.2. Give an example of fuzzy sets $B^1, ..., B^6$ such that the set of the six rules (7.12) is continuous.

Exercise 7.3. Suppose that a fuzzy rule base consists of only one rule (7.36) with

$$\mu_B(y) = exp(-y^2) \tag{7.45}$$

Let the input A' to the fuzzy inference engine be the fuzzy singleton (7.27). Plot the output membership functions $\mu_{B'}(y)$ using: (a) product, (b) minimum, (c) Lukasiewicz, (d) Zadeh, and (e) Dienes-Rescher inference engines.

Exercise 7.4. Consider Example 7.3 and plot the $\mu_{B'}(y)$ using: (a) Lukasiewicz inference engine, and (b) Zadeh inference engine.

Exercise 7.5. Use the Gödel implication to propose a so-called Gödel inference engine.

Chapter 8

Fuzzifiers and Defuzzifiers

We learned from Chapter 7 that the fuzzy inference engine combines the rules in the fuzzy rule base into a mapping from fuzzy set A' in U to fuzzy set B' in V. Because in most applications the input and output of the fuzzy system are real-valued numbers, we must construct interfaces between the fuzzy inference engine and the environment. The interfaces are the fuzzifier and defuzzifier in Fig. 1.5.

8.1 Fuzzifiers

The *fuzzifier* is defined as a mapping from a real-valued point $\mathbf{x}^* \in U \subset R^n$ to a fuzzy set A' in U. What are the criteria in designing the fuzzifier? First, the fuzzifier should consider the fact that the input is at the crisp point $\mathbf{x}^*$, that is, the fuzzy set A' should have large membership value at $\mathbf{x}^*$. Second, if the input to the fuzzy system is corrupted by noise, then it is desireable that the fuzzifier should help to suppress the noise. Third, the fuzzifier should help to simplify the computations involved in the fuzzy inference engine. From (7.23), (7.24) and (7.30)-(7.32) we see that the most complicated computation in the fuzzy inference engine is the $\sup_{\mathbf{x}\in U}$, therefore our objective is to simplify the computations involving $\sup_{\mathbf{x}\in U}$.

We now propose three fuzzifiers:

- **Singleton fuzzifier**: The *singleton fuzzifier* maps a real-valued point $\mathbf{x}^* \in U$ into a fuzzy singleton A' in U, which has membership value 1 at $\mathbf{x}^*$ and 0 at all other points in U; that is,

$$\mu_{A'}(\mathbf{x}) = \begin{cases} 1 \; if \; \mathbf{x} = \mathbf{x}^* \\ 0 \; otherwise \end{cases} \tag{8.1}$$

- **Gaussian fuzzifier**: The *Gaussian fuzzifier* maps $\mathbf{x}^* \in U$ into fuzzy set A' in U, which has the following Gaussian membership function:

$$\mu_{A'}(\mathbf{x}) = e^{-(\frac{x_1 - x_1^*}{a_1})^2} \star \cdots \star e^{-(\frac{x_n - x_n^*}{a_n})^2} \tag{8.2}$$

where a_i are positive parameters and the t-norm $\star$ is usually chosen as algebraic product or *min*.

- **Triangular fuzzifier**: The *triangular fuzzifier* maps $\mathbf{x}^* \in U$ into fuzzy set A' in U, which has the following triangular membership function

$$\mu_{A'}(\mathbf{x}) = \begin{cases} (1 - \frac{|x_1 - x_1^*|}{b_1}) \star \cdots \star (1 - \frac{|x_n - x_n^*|}{b_n}) \ if \ |x_i - x_i^*| \leq b_i, \ i = 1, 2, ..., n \\ 0 \ otherwise \end{cases} \tag{8.3}$$

where b_i are positive parameters and the t-norm $\star$ is usually chosen as algebraic product or *min*.

From (8.1)-(8.3) we see that all three fuzzifiers satisfy $\mu_{A'}(\mathbf{x}^*) = 1$; that is, they satisfy the first criterion mentioned before. From Lemmas 7.3 and 7.4 we see that the singleton fuzzifier greatly simplifies the computations involved in the fuzzy inference engine. Next, we show that if the fuzzy sets A_i^l in the rules (7.1) have Gaussian or triangular membership functions, then the Gaussian or triangular fuzzifier also will simplify some fuzzy inference engines.

Lemma 8.1. Suppose that the fuzzy rule base consists of M rules in the form of (7.1) and that

$$\mu_{A_i^l}(x_i) = e^{-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2} \tag{8.4}$$

where $\bar{x}_i^l$ and σ_i^l are constant parameters, $i = 1, 2, ..., n$ and $l = 1, 2, ..., M$. If we use the Gaussian fuzzifier (8.2), then:

(a) If we choose algebraic product for the t-norm $\star$ in (8.2), the product inference engine (7.23) is simplified to

$$\mu_{B'}(y) = \max_{l=1}^{M} [\prod_{i=1}^{n} e^{-(\frac{x_{iP}^l - \bar{x}_i^l}{\sigma_i^l})^2} e^{-(\frac{x_{iP}^l - x_i^*}{a_i})^2} \mu_{B^l}(y)] \tag{8.5}$$

where

$$x_{iP}^l = \frac{a_i^2 \bar{x}_i^l + (\sigma_i^l)^2 x_i^*}{a_i^2 + (\sigma_i^l)^2} \tag{8.6}$$

(b) If we choose *min* for the t-norm $\star$ in (8.2), the minimum inference engine (7.24) is simplified to

$$\mu_{B'}(y) = \max_{l=1}^{M} [min(e^{-(\frac{x_{1M}^l - \bar{x}_1^l}{\sigma_1^l})^2}, ..., e^{-(\frac{x_{nM}^l - \bar{x}_n^l}{\sigma_n^l})^2}, \mu_{B^l}(y))] \tag{8.7}$$

where

$$x_{iM}^l = \frac{a_i \bar{x}_i^l + \sigma_i^l x_i^*}{a_i + \sigma_i^l} \tag{8.8}$$

Proof: (a) Substituting (8.2) and (8.4) into (7.23) and noticing that the $\star$ is an algebraic product, we obtain

$$\mu_{B'}(y) = \max_{l=1}^{M}[\sup_{\mathbf{x}\in U}\prod_{i=1}^{n} e^{-(\frac{x_i-x_i^*}{a_i})^2} e^{-(\frac{x_i-\bar{x}_i^l}{\sigma_i^l})^2}\mu_{B^l}(y)]$$

$$= \max_{l=1}^{M}[\prod_{i=1}^{n}\sup_{\mathbf{x}\in U} e^{-(\frac{x_i-x_i^*}{a_i})^2-(\frac{x_i-\bar{x}_i^l}{\sigma_i^l})^2}\mu_{B^l}(y)] \tag{8.9}$$

Since

$$-(\frac{x_i-x_i^*}{a_i})^2-(\frac{x_i-\bar{x}_i^l}{\sigma_i^l})^2 = -k_1(x_i-\frac{a_i^2\bar{x}_i^l+(\sigma_i^l)^2x_i^*}{a_i^2+(\sigma_i^l)^2})^2+k_2 \tag{8.10}$$

where k_1 and k_2 are not functions of x_i, the $\sup_{\mathbf{x}\in U}$ in (8.9) is achieved at $x_{iP}\in U$ $(i=1,2,...,n)$, which is exactly (8.6).

(b) Substituting (8.2) and (8.4) into (7.24) and noticing that the $\star$ is *min*, we obtain

$$\mu_{B'}(y) = \max_{l=1}^{M}\{min[\sup_{\mathbf{x}\in U} min(e^{-(\frac{x_1-x_1^*}{a_1})^2}, e^{-(\frac{x_1-\bar{x}_1^l}{\sigma_1^l})^2}),\cdots,$$

$$\sup_{\mathbf{x}\in U} min(e^{-(\frac{x_n-x_n^*}{a_n})^2}, e^{-(\frac{x_n-\bar{x}_n^l}{\sigma_n^l})^2}), \mu_{B^l}(y)]\} \tag{8.11}$$

Clearly, the $\sup_{\mathbf{x}\in U} min$ is achieved at $x_i = x_{iM}^l$ when

$$e^{-(\frac{x_i-x_i^*}{a_i})^2} = e^{-(\frac{x_i-\bar{x}_i^l}{\sigma_i^l})^2} \tag{8.12}$$

which gives (8.8). Substituting $x_i = x_{iM}^l$ into (8.11) gives (8.7). □

We can obtain similar results as in Lemma 8.1 when the triangular fuzzifier is used. If $a_i = 0$, then from (8.6) and (8.8) we have $x_{iP}^l = x_{iM}^l = x_i^*$; that is, in this case the Gaussian fuzzifier becomes the singleton fuzzifier. If a_i is much larger than σ_i^l, then from (8.6) and (8.8) we see that x_{iP}^l and x_{iM}^l will be very close to $\bar{x}_i^l$; that is, x_{iP}^l and x_{iM}^l will be insensitive to the changes in the input x_i^*. Therefore, by choosing large a_i, the Gaussian fuzzifier can suppress the noise in the input x_i^*. More specifically, suppose that the input x_i^* is corrupted by noise, that is,

$$x_i^* = x_{i0}^* + n_i^* \tag{8.13}$$

where x_{io}^* is the useful signal and n_i^* is noise. Substituting (8.13) into (8.6), we have

$$x_{iP}^l = \frac{a_i^2\bar{x}_i^l+(\sigma_i^l)^2x_{io}^*}{a_i^2+(\sigma_i^l)^2} + \frac{(\sigma_i^l)^2}{a_i^2+(\sigma_i^l)^2}n_i^* \tag{8.14}$$

From (8.14) we see that after passing through the Gaussian fuzzifier, the noise is suppressed by the factor $\frac{(\sigma_i^l)^2}{a_i^2+(\sigma_i^l)^2}$. When a_i is much larger than σ_i^l, the noise will be greatly suppressed. Similarly, we can show that the triangular fuzzifier also has this kind of noise suppressing capability.

In summary, we have the following remarks about the three fuzzifiers:

- The singleton fuzzifier greatly simplifies the computation involved in the fuzzy inference engine for any type of membership functions the fuzzy IF-THEN rules may take.

- The Gaussian or triangular fuzzifiers also simplify the computation in the fuzzy inference engine, if the membership functions in the fuzzy IF-THEN rules are Gaussian or triangular, respectively.

- The Gaussian and triangular fuzzifiers can suppress noise in the input, but the singleton fuzzifier cannot.

8.2 Defuzzifiers

The *defuzzifier* is defined as a mapping from fuzzy set B' in $V \subset R$ (which is the output of the fuzzy inference engine) to crisp point $y^* \in V$. Conceptually, the task of the defuzzifier is to specify a point in V that best represents the fuzzy set B'. This is similar to the mean value of a random variable. However, since the B' is constructed in some special ways (see Chapter 7), we have a number of choices in determining this representing point. The following three criteria should be considered in choosing a defuzzification scheme:

- *Plausibility*: The point y^* should represent B' from an intuitive point of view; for example, it may lie approximately in the middle of the support of B' or has a high degree of membership in B'.

- *Computational simplicity*: This criterion is particularly important for fuzzy control because fuzzy controllers operate in real-time.

- *Continuity*: A small change in B' should not result in a large change in y^*.

We now propose three types of defuzzifiers. For all the defuzzifiers, we assume that the fuzzy set B' is obtained from one of the five fuzzy inference engines in Chapter 7, that is, B' is given by (7.23), (7.24), (7.30), (7.31), or (7.32). From these equations we see that B' is the union or intersection of M individual fuzzy sets.

8.2.1 center of gravity Defuzzifier

The *center of gravity defuzzifier* specifies the y^* as the center of the area covered by the membership function of B', that is,

$$y^* = \frac{\int_V y\mu_{B'}(y)dy}{\int_V \mu_{B'}(y)dy} \tag{8.15}$$

where $\int_V$ is the conventional integral. Fig. 8.1 shows this operation graphically.

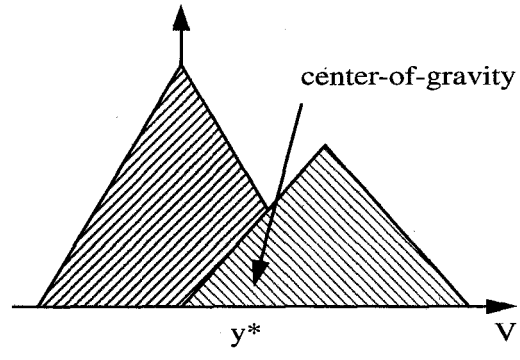


Figure 8.1. A graphical representation of the center of gravity defuzzifier.

If we view $\mu_{B'}(y)$ as the probability density function of a random variable, then the center of gravity defuzzifier gives the mean value of the random variable. Sometimes it is desirable to eliminate the $y \in V$, whose membership values in B' are too small; this results in the *indexed center of gravity defuzzifier*, which gives

$$y^* = \frac{\int_{V_\alpha} y\mu_{B'}(y)dy}{\int_{V_\alpha} \mu_{B'}(y)dy} \tag{8.16}$$

where V_α is defined as

$$V_\alpha = \{y \in V | \mu_{B'}(y) \geq \alpha\} \tag{8.17}$$

and α is a constant.

The advantage of the center of gravity defuzzifier lies in its intuitive plausibility. The disadvantage is that it is computationally intensive. In fact, the membership

function $\mu_{B'}(y)$ is usually irregular and therefore the integrations in (8.15) and (8.16) are difficult to compute. The next defuzzifier tries to overcome this disadvantage by approximating (8.15) with a simpler formula.

8.2.2 Center Average Defuzzifier

Because the fuzzy set B' is the union or intersection of M fuzzy sets, a good approximation of (8.15) is the weighted average of the centers of the M fuzzy sets, with the weights equal the heights of the corresponding fuzzy sets. Specifically, let $\bar{y}^l$ be the center of the *l'th* fuzzy set and w_l be its height, the *center average defuzzifier* determines y^* as

$$y^* = \frac{\sum_{l=1}^{M} \bar{y}^l w_l}{\sum_{l=1}^{M} w_l} \tag{8.18}$$

Fig. 8.2 illustrates this operation graphically for a simple example with $M = 2$.

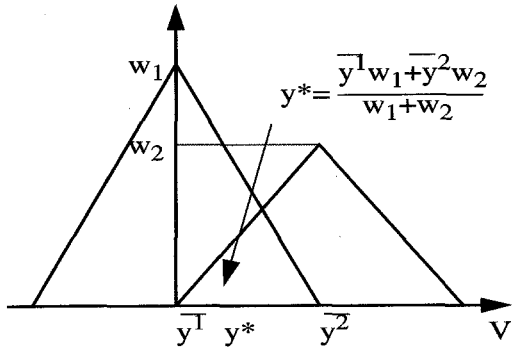


Figure 8.2. A graphical representation of the center average defuzzifier.

The center average defuzzifier is the most commonly used defuzzifier in fuzzy systems and fuzzy control. It is computationally simple and intuitively plausible. Also, small changes in $\bar{y}^l$ and w_l result in small changes in y^*. We now compare the center of gravity and center average defuzzifiers for a simple example.

Example 8.1. Suppose that the fuzzy set B' is the union of the two fuzzy sets shown in Fig. 8.2 with $\bar{y}^1 = 0$ and $\bar{y}^2 = 1$. Then the center average defuzzifier gives

$$y^* = \frac{w_2}{w_1 + w_2} \tag{8.19}$$

Table 8.1. Comparison of the center of gravity and center average defuzzifiers for Example 8.1.

w_1	w_2	y^* (center of gravity)	y^* (center average)	relative error
0.9	0.7	0.4258	0.4375	0.0275
0.9	0.5	0.5457	0.5385	0.0133
0.9	0.2	0.7313	0.7000	0.0428
0.6	0.7	0.3324	0.3571	0.0743
0.6	0.5	0.4460	0.4545	0.0192
0.6	0.2	0.6471	0.6250	0.0342
0.3	0.7	0.1477	0.1818	0.2308
0.3	0.5	0.2155	0.2500	0.1600
0.3	0.2	0.3818	0.4000	0.0476

We now compute the y^* resulting from the center of gravity defuzzifier. First, we notice that the two fuzzy sets intersect at $y = \frac{w_1}{w_1+w_2}$, hence,

$$\begin{aligned}\int_V \mu_{B'}(y)dy &= \textit{area of the first fuzzy set} + \textit{area of the second fuzzy set} \\ &\quad - \textit{their intersection} \\ &= w_1 + w_2 - \frac{1}{2}\frac{w_1 w_2}{w_1 + w_2} \end{aligned} \tag{8.20}$$

From Fig. 8.2 we have

$$\begin{aligned}\int_V y\mu_{B'}(y)dy &= \int_{-1}^{0} yw_1(1+y)dy + \int_{0}^{\frac{w_1}{w_1+w_2}} yw_1(1-y)dy + \int_{\frac{w_1}{w_1+w_2}}^{1} yw_2 y dy \\ &\quad + \int_1^2 yw_2(2-y)dy \\ &= w_1(\frac{1}{2}y^2 + \frac{1}{3}y^3)|_{-1}^{0} + w_1(\frac{1}{2}y^2 - \frac{1}{3}y^3)|_{0}^{\frac{w_1}{w_1+w_2}} + w_2\frac{1}{3}y^3|_{\frac{w_1}{w_1+w_2}}^{1} \\ &\quad + w_2(y^2 - \frac{1}{3}y^3)|_1^2 \\ &= -\frac{1}{6}w_1 + w_2 + \frac{1}{6}\frac{w_1^3}{(w_1+w_2)^2} \end{aligned} \tag{8.21}$$

Dividing (8.21) by (8.20) we obtain the y^* of the center of gravity defuzzifier. Table 8.1 shows the values of y^* using these two defuzzifiers for certain values of w_1 and w_2. We see that the computation of the center of gravity defuzzifier is much more intensive than that of the center average defuzzifier. □

8.2.3 Maximum Defuzzifier

Conceptually, the maximum defuzzifier chooses the y^* as the point in V at which $\mu_{B'}(y)$ achieves its maximum value. Define the set

$$hgt(B') = \{y \in V | \mu_{B'}(y) = \sup_{y \in V} \mu_{B'}(y)\} \tag{8.22}$$

that is, $hgt(B')$ is the set of all points in V at which $\mu_{B'}(y)$ achieves its maximum value. The *maximum defuzzifier* defines y^* as an arbitrary element in $hgt(B')$, that is,

$$y^* = any\ point\ in\ hgt(B') \tag{8.23}$$

If $hgt(B')$ contains a single point, then y^* is uniquely defined. If $hgt(B')$ contains more than one point, then we may still use (8.23) or use the smallest of maxima, largest of maxima, or mean of maxima defuzzifiers. Specifically, the *smallest of maxima defuzzifier* gives

$$y^* = inf\{y \in hgt(B')\} \tag{8.24}$$

the *largest of maxima defuzzifier* gives

$$y^* = sup\{y \in hgt(B')\} \tag{8.25}$$

and the *mean of maxima defuzzifier* is defined as

$$y^* = \frac{\int_{hgt(B')} y dy}{\int_{hgt(B')} dy} \tag{8.26}$$

where $\int_{hgt(B')}$ is the usual integration for the continuous part of $hgt(B')$ and is summation for the discrete part of $hgt(B')$. We feel that the mean of maxima defuzzifier may give results which are contradictory to the intuition of maximum membership. For example, the y^* from the mean of maxima defuzzifier may have very small membership value in B'; see Fig. 8.3 for an example. This problem is due to the nonconvex nature of the membership function $\mu_{B'}(y)$.

The maximum defuzzifiers are intuitively plausible and computationally simple. But small changes in B' may result in large changes in y^*; see Fig.8.4 for an example. If the situation in Fig.8.4 is unlikely to happen, then the maximum defuzzifiers are good choices.

8.2.4 Comparison of the Defuzzifiers

Table 8.2 compares the three types of defuzzifiers according to the three criteria: plausibility, computational simplicity, and continuity. From Table 8.2 we see that the center average defuzzifier is the best.

Finally, we consider an example for the computation of the defuzzifiers with some particular membership functions.

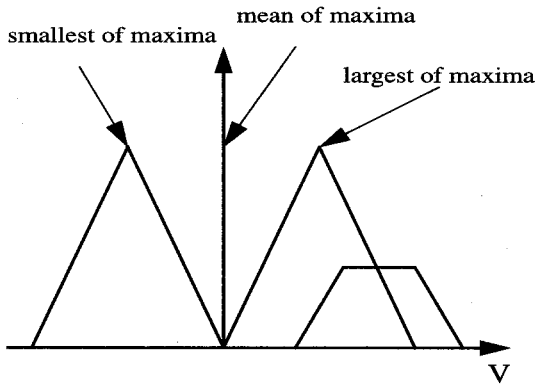


Figure 8.3. A graphical representation of the maximum defuzzifiers. In this example, the mean of maxima defuzzifier gives a result that is contradictory to the maximum-membership intuition.

Table 8.2. Comparison of the center of gravity, center average, and maximum defuzzifiers with respect to plausibility, computational simplicity, and continuity.

	center of gravity	center average	maximum
plausibility	yes	yes	yes
computational simplicity	no	yes	yes
continuity	yes	yes	no

Example 8.2. Consider a two-input-one-output fuzzy system that is constructed from the following two rules:

$$IF\ x_1\ is\ A_1\ and\ x_2\ is\ A_2,\ THEN\ y\ is\ A_1 \tag{8.27}$$

$$IF\ x_1\ is\ A_2\ and\ x_2\ is\ A_1,\ THEN\ y\ is\ A_2 \tag{8.28}$$

where A_1 and A_2 are fuzzy sets in R with membership functions

$$\mu_{A_1}(u) = \begin{cases} 1 - |u|\ if\ -1 \leq u \leq 1 \\ 0\ otherwise \end{cases} \tag{8.29}$$

$$\mu_{A_2}(u) = \begin{cases} 1 - |u-1|\ if\ 0 \leq u \leq 2 \\ 0\ otherwise \end{cases} \tag{8.30}$$

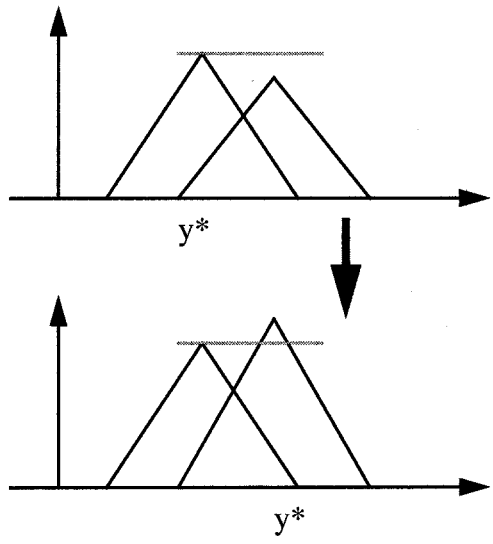


Figure 8.4. An example of maximum defuzzifier where small change in B' results in large change in y^*.

Suppose that the input to the fuzzy system is $(x_1^*, x_2^*) = (0.3, 0.6)$ and we use the singleton fuzzifier. Determine the output of the fuzzy system y^* in the following situations: (a) product inference engine (7.23) and center average defuzzifier (8.18); (b) product inference engine (7.23) and center of gravity defuzzifier (8.15); (c) Lakasiewicz inference engine (7.30) and mean of maxima defuzzifier (8.26); and (d) Lakasiewicz inference engine (7.30) and center average defuzzifier (8.18).

(a) Since we use singleton fuzzifier, from Lemma 7.3 ((7.28)) and (8.29)-(8.30) we have

$$\begin{aligned}\mu_{B'}(y) &= max[\mu_{A_1}(0.3)\mu_{A_2}(0.6)\mu_{A_1}(y), \mu_{A_2}(0.3)\mu_{A_1}(0.6)\mu_{A_2}(y)] \\ &= max[0.7 * 0.6 * \mu_{A_1}(y), 0.3 * 0.4 * \mu_{A_2}(y)] \\ &= max[0.42\mu_{A_1}(y), 0.12\mu_{A_2}(y)] \end{aligned} \tag{8.31}$$

which is shown in Fig. 8.2 with $\bar{y}^1 = 0, \bar{y}^2 = 1, w_1 = 0.42$, and $w_2 = 0.12$. Hence, from (8.19) we obtain that the center average defuzzifier gives

$$y^* = \frac{0.12}{0.42 + 0.12} = 0.2222 \tag{8.32}$$

(b) Following (a) and (8.20)-(8.21), we have

$$\int_V \mu_{B'}(y)dy = 0.42 + 0.12 - \frac{1}{2}\frac{0.42 * 0.12}{0.12 + 0.42} = 0.4933 \tag{8.33}$$

$$\int_V y\mu_{B'}(y)dy = -\frac{1}{6}0.42 + 0.12 + \frac{1}{6}\frac{(0.42)^3}{(0.42 + 0.12)^2} = 0.0923 \tag{8.34}$$

Hence, the y^* in this case is

$$y^* = 0.0923/0.4933 = 0.1871 \tag{8.35}$$

(c) If we use the Lukasiewicz inference engine, then from Lemma 7.4 ((7.33)) we have

$$\begin{aligned}\mu_{B'}(y) &= min\{1, 1 - min[\mu_{A_1}(0.3), \mu_{A_2}(0.6)] + \mu_{A_1}(y), \\ &\quad 1 - min[\mu_{A_2}(0.3), \mu_{A_1}(0.6)] + \mu_{A_2}(y)\} \\ &= min[1, 0.4 + \mu_{A_1}(y), 0.7 + \mu_{A_2}(y)]\end{aligned} \tag{8.36}$$

which is plotted in Fig. 8.5. From Fig. 8.5 we see that $\sup_{y \in V} \mu_{B'}(y)$ is achieved in the interval $[0.3, 0.4]$, so the mean of maxima defuzzifier gives

$$y^* = 0.35 \tag{8.37}$$

(d) From Fig. 8.5 we see that in this case $\bar{y}^1 = 0, \bar{y}^2 = 1, w_1 = 1$ and $w_2 = 1$. So the center average defuzzifier (8.18) gives

$$y^* = \frac{0 * 1 + 1 * 1}{1 + 1} = 0.5 \tag{8.38}$$

□

8.3 Summary and Further Readings

In this chapter we have demonstrated the following:

- The definitions and intuitive meanings of the singleton, Gaussian and triangular fuzzifiers, and the center of gravity, center average and maximum defuzzifiers.
- Computing the outputs of the fuzzy systems for different combinations of the fuzzifiers, defuzzifiers, and fuzzy inference engines for specific examples.

Different defuzzification schemes were studied in detail in the books Driankov, Hellendoorn and Reifrank [1993] and Yager and Filev [1994].

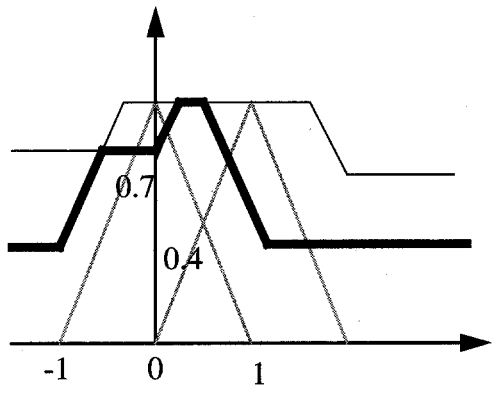


Figure 8.5. A graphical representation of the membership function $\mu_{B'}(y)$ of (8.36).

8.4 Exercises

Exercise 8.1. Suppose that a fuzzy rule base consists of the M rules (7.1) with

$$\mu_{A_i^l}(x_i) = \begin{cases} 1 - \frac{|x_i - \bar{x}_i^l|}{\sigma_i^l} \; if \; |x_i - \bar{x}_i^l| \leq \sigma_i^l \\ 0 \; otherwise \end{cases} \tag{8.39}$$

and that we use the triangular fuzzifier (8.3). Determine the output of the fuzzy inference engine $\mu_{B'}(y)$ for:

(a) product inference engine (7.23) with all $\star = algebraic\ product$, and

(b) minimum inference engine (7.24) with all $\star = min$.

Exercise 8.2. Consider Example 8.1 and determine the y^* using the indexed center of gravity defuzzifier with $\alpha = 0.1$. Compute the y^* for the specific values of w_1 and w_2 in Table 8.1.

Exercise 8.3. Consider Example 8.2 and determine the fuzzy system output y^* (with input $(x_1^*, x_2^*) = (0.3, 0.6)$) for:

(a) Zadeh inference engine (7.31) and maximum (or mean of maxima) defuzzifier, and

(b) Dienes-Rescher inference engine (7.32) with maximum (or mean of maxima) defuzzifier.

Exercise 8.4. Use a practical example, such as the mobile robot path planning problem, to show that the center of gravity and center average defuzzifiers may create problems when the fuzzy set B' is non-convex.

Exercsie 8.5. When the fuzzy set B' is non-convex, the so-called center of largest area defuzzifier is proposed. This method determines the convex part of B' that has the largest area and defines the crisp output y^* to be the center of gravity of this convex part. Create a specific non-convex B' and use the center of largest area defuzzifier to determine the defuzzified output y^*.

Chapter 9

Fuzzy Systems as Nonlinear Mappings

9.1 The Formulas of Some Classes of Fuzzy Systems

From Chapters 7 and 8 we see that there are a variety of choices in the fuzzy inference engine, fuzzifier, and defuzzifier modules. Specifically, we proposed five fuzzy inference engines (product, minimum, Lukasiewicz, Zadeh, and Dienes-Rescher), three fuzzifiers (singleton, Gaussian and triangular), and three types of defuzzifiers (center-of-gravity, center average, and maximum). Therefore, we have at least $5 * 3 * 3 = 45$ types of fuzzy systems by combining these different types of inference engines, fuzzifiers, and defuzzifiers. In this chapter, we will derive the detailed formulas of certain classes of fuzzy systems. We will see that some classes of fuzzy systems are very useful, while others do not make a lot of sense. That is, not every combination results in useful fuzzy systems. Because in Chapter 8 we showed that the center-of-gravity defuzzifier is computationally expensive and the center average defuzzifier is a good approximation of it, we will not consider the center-of-gravity defuzzifier in this chapter. We classify the fuzzy systems to be considered into two groups: fuzzy systems with center average defuzzifier, and fuzzy systems with maximum defuzzifier.

9.1.1 Fuzzy Systems with Center Average Defuzzifier

Lemma 9.1. Suppose that the fuzzy set B^l in (7.1) is normal with center $\bar{y}^l$. Then the fuzzy systems with fuzzy rule base (7.1), product inference engine (7.23), singleton fuzzifier (8.1), and center average defuzzifier (8.18) are of the following form:

$$f(x) = \frac{\sum_{l=1}^{M} \bar{y}^l (\prod_{i=1}^{n} \mu_{A_i^l}(x_i))}{\sum_{l=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^l}(x_i))} \tag{9.1}$$

where $x \in U \subset R^n$ is the input to the fuzzy system, and $f(x) \in V \subset R$ is the output of the fuzzy system.

Proof: Substituting (8.1) into (7.23), we have

$$\mu_{B'}(y) = \max_{l=1}^{M}[\prod_{i=1}^{n} \mu_{A_i^l}(x_i^*)\mu_{B^l}(y)] \tag{9.2}$$

Since for a given input x_i^*, the center of the $l'th$ fuzzy set in (9.2) (that is, the fuzzy set with membership function $\mu_{A_i^l}(x_i^*)\mu_{B^l}(y)$) is the center of B^l, we see that the $\bar{y}^l$ in (8.18) is the same $\bar{y}^l$ in this lemma. Additionally, the height of the $l'th$ fuzzy set in (9.2), denoted by w_l in (8.18), is $\prod_{i=1}^{n} \mu_{A_i^l}(x_i^*)\mu_{B^l}(\bar{y}^l) = \prod_{i=1}^{n} \mu_{A_i^l}(x_i^*)$ (since B^l is normal). Hence, using the center average defuzzifier (8.18) for (9.2), we obtain

$$y^* = \frac{\sum_{l=1}^{M} \bar{y}^l(\prod_{i=1}^{n} \mu_{A_i^l}(x_i^*))}{\sum_{l=1}^{M}(\prod_{i=1}^{n} \mu_{A_i^l}(x_i^*))} \tag{9.3}$$

Using the notion of this lemma, we have $x^* = x$ and $y^* = f(x)$; thus, (9.3) becomes (9.1). □

From Lemma 9.1 we see that the fuzzy system is a nonlinear mapping from $x \in U \subset R^n$ to $f(x) \in V \subset R$, and (9.1) gives the detailed formula of this mapping. The fuzzy systems in the form of (9.1) are the most commonly used fuzzy systems in the literature. They are computationally simple and intuitively appealing. From (9.1) we see that the output of the fuzzy system is a weighted average of the centers of the fuzzy sets in the THEN parts of the rules, where the weights equal the membership values of the fuzzy sets in the IF parts of the rules at the input point. Consequently, the more the input point agrees with the IF part of a rule, the larger weight this rule is given; this makes sense intuitively.

Lemma 9.1 also reveals an important contribution of fuzzy systems theory that is summarized as follows:

- **The dual role of fuzzy systems**: On one hand, fuzzy systems are rule-based systems that are constructed from a collection of linguistic rules; on the other hand, fuzzy systems are nonlinear mappings that in many cases can be represented by precise and compact formulas such as (9.1). *An important contribution of fuzzy systems theory is to provide a systematic procedure for transforming a set of linguistic rules into a nonlinear mapping.* Because nonlinear mappings are easy to implement, fuzzy systems have found their way into a variety of engineering applications.

By choosing different forms of membership functions for $\mu_{A_i^l}$ and μ_{B^l}, we obtain different subclasses of fuzzy systems. One choice of $\mu_{A_i^l}$ and μ_{B^l} is Gaussian

membership function. Specifically, if we choose the following *Gaussian membership function* for $\mu_{A_i^l}$ and μ_{B^l}:

$$\mu_{A_i^l}(x_i) = a_i^l exp[-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2] \tag{9.4}$$

$$\mu_{B^l}(y) = exp[-(y - \bar{y}^l)^2] \tag{9.5}$$

where $a_i^l \in (0, 1], \sigma_i^l \in (0, \infty)$ and $\bar{x}_i^l, \bar{y}^l \in R$ are real-valued parameters, then the fuzzy systems in Lemma 9.1 become

$$f(x) = \frac{\sum_{l=1}^M \bar{y}^l [\prod_{i=1}^n a_i^l exp(-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2)]}{\sum_{l=1}^M [\prod_{i=1}^n a_i^l exp(-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2)]} \tag{9.6}$$

We call the fuzzy systems in the form of (9.6) *fuzzy systems with product inference engine, singleton fuzzifier, center average defuzzifier, and Gaussian membership functions.* Other popular choices of $\mu_{A_i^l}$ and μ_{B^l} are triangular and trapezoid membership functions. We will study the fuzzy systems with these types of membership functions in detail in Chapters 10 and 11.

Another class of commonly used fuzzy systems is obtained by replacing the product inference engine in Lemma 9.1 by the minimum inference engine. Using the same procedure as in the proof of Lemma 9.1, we obtain that *the fuzzy systems with fuzzy rule base (7.1), minimum inference engine (7.24), singleton fuzzifier (8.1), and center average defuzzifier (8.18)* are of the following form:

$$f(x) = \frac{\sum_{l=1}^M \bar{y}^l (\min_{i=1}^n \mu_{A_i^l}(x_i))}{\sum_{l=1}^M (\min_{i=1}^n \mu_{A_i^l}(x_i))} \tag{9.7}$$

where the variables have the same meaning as in (9.1).

We showed in Chapter 8 (Lemma 8.1) that if the membership functions for A_i^l are Gaussian, then the Gaussian fuzzifier also significantly simplifies the fuzzy inference engine. What do the fuzzy systems look like in this case?

Lemma 9.2. The fuzzy systems with fuzzy rule base (7.1), product inference engine (7.23), Gaussian fuzzifier (8.2) with $\star$=product, center average defuzzifier (8.18), and Gaussian membership functions (9.4) and (9.5) (with $a_i^l = 1$) are of the following form:

$$f(x) = \frac{\sum_{l=1}^M \bar{y}^l [\prod_{i=1}^n exp(-\frac{(x_i - \bar{x}_i^l)^2}{a_i^2 + (\sigma_i^l)^2})]}{\sum_{l=1}^M [\prod_{i=1}^n exp(-\frac{(x_i - \bar{x}_i^l)^2}{a_i^2 + (\sigma_i^l)^2})]} \tag{9.8}$$

If we replace the product inference engine (7.23) with the minimum inference engine

(7.24) and use $\star = min$ in (8.2), then the fuzzy systems become

$$f(x) = \frac{\sum_{l=1}^{M} \bar{y}^l [\min_{i=1}^{n} exp(-(\frac{x_i - \bar{x}_i^l}{a_i + \sigma_i^l})^2)]}{\sum_{l=1}^{M} [\min_{i=1}^{n} exp(-(\frac{x_i - \bar{x}_i^l}{a_i + \sigma_i^l})^2)]} \tag{9.9}$$

Proof: Substituting (8.6) into (8.5) and use x for x^*, we have

$$\begin{aligned}\mu_{B'}(y) &= \max_{l=1}^{M}\{\prod_{i=1}^{n} exp[-(\frac{\frac{a_i^2 \bar{x}_i^l + (\sigma_i^l)^2 x_i}{a_i^2 + (\sigma_i^l)^2} - \bar{x}_i^l}{\sigma_i^l})^2 - (\frac{\frac{a_i^2 \bar{x}_i^l + (\sigma_i^l)^2 x_i}{a_i^2 + (\sigma_i^l)^2} - x_i}{a_i})^2]\mu_{B^l}(y)\} \\ &= \max_{l=1}^{M}\{\prod_{i=1}^{n} exp[-\frac{(x_i - \bar{x}_i^l)^2}{a_i^2 + (\sigma_i^l)^2}]\mu_{B^l}(y)\}\end{aligned} \tag{9.10}$$

Using the same arguments as in the proof of Lemma 9.1 and applying the center average defuzzifier (8.18) to (9.10), we obtain (9.8). Similarly, substituting (8.8) into (8.7), we have

$$\begin{aligned}\mu_{B'}(y) &= \max_{l=1}^{M}\{min[exp(-(\frac{\frac{a_1 \bar{x}_1^l + \sigma_1^l x_1}{a_1 + \sigma_1^l} - \bar{x}_1^l}{\sigma_1^l})^2), ..., exp(-(\frac{\frac{a_n \bar{x}_n^l + \sigma_n^l x_n}{a_n + \sigma_n^l} - \bar{x}_n^l}{\sigma_n^l})^2), \mu_{B^l}(y)]\} \\ &= \max_{l=1}^{M}\{min[exp(-(\frac{x_1 - \bar{x}_1^l}{a_1 + \sigma_1^l})^2), ..., exp(-(\frac{x_n - \bar{x}_n^l}{a_n + \sigma_n^l})^2), \mu_{B^l}(y)]\}\end{aligned} \tag{9.11}$$

Applying the center average defuzzifier (8.18) to (9.11), we obtain (9.9). □

In Chapter 7 we saw that the product and minimum inference engines are quite different from the Lukasiewicz, Zadeh and Dienes-Rescher inference engines. What do the fuzzy systems with these inference engines look like?

Lemma 9.3. If the fuzzy set B^l in (7.1) are normal with center $\bar{y}^l$, then the fuzzy systems with fuzzy rule base (7.1), Lukasiewicz inference engine (7.30) or Dienes-Rescher inference engine (7.32), singleton fuzzifier (8.1) or Gaussian fuzzifier (8.2) or triangular fuzzifier (8.3), and center average defuzzifier (8.18) are of the following form:

$$f(x) = \frac{1}{M} \sum_{l=1}^{M} \bar{y}^l \tag{9.12}$$

Proof: Since $\mu_{B^l}(\bar{y}^l) = 1$, we have $1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)) + \mu_{B^l}(\bar{y}^l) \geq 1$; therefore, the height of the $l'th$ fuzzy set in (7.30) is

$$\begin{aligned}w_l &= \sup_{x \in U} min[\mu_{A'}(x), 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)) + \mu_{B^l}(\bar{y}^l)] \\ &= \sup_{x \in U} \mu_{A'}(x) \\ &= 1\end{aligned} \tag{9.13}$$

where we use the fact that for all the three fuzzifiers (8.1)-(8.3) we have $\sup_{x \in U} \mu_{A'}(x) = 1$. Similarly, the height of the $l'th$ fuzzy set in (7.32) is also equal to one. Hence, with the center average defuzzifier (8.18) we obtain (9.12). □

The fuzzy systems in the form of (9.12) do not make a lot of sense because it gives a constant output no matter what the input is. Therefore, the combinations of fuzzy inference engine, fuzzifier, and defuzzifier in Lemma 9.3 do not result in useful fuzzy systems.

9.1.2 Fuzzy Systems with Maximum Defuzzifier

Lemma 9.4. Suppose the fuzzy set B^l in (7.1) is normal with center $\bar{y}^l$, then the fuzzy systems with fuzzy rule base (7.1), product inference engine (7.23), singleton fuzzifier (8.1), and maximum defuzzifier (8.23) are of the following form:

$$f(x) = \bar{y}^{l*} \tag{9.14}$$

where $l* \in \{1, 2, ..., M\}$ is such that

$$\prod_{i=1}^{n} \mu_{A_i^{l*}}(x_i) \geq \prod_{i=1}^{n} \mu_{A_i^l}(x_i) \tag{9.15}$$

for all $l = 1, 2, ..., M$.

Proof: From (7.28) (Lemma 7.3) and noticing that $x^* = x$ in this case, we have

$$\sup_{y \in V} \mu_{B'}(y) = \sup_{y \in V} \max_{l=1}^{M} [\prod_{i=1}^{n} \mu_{A_i^l}(x_i) \mu_{B^l}(y)] \tag{9.16}$$

Since $\sup_{y \in V}$ and $\max_{l=1}^{M}$ are interchangeable and B^l is normal, we have

$$\begin{aligned} \sup_{y \in V} \mu_{B'}(y) &= \max_{l=1}^{M} [\sup_{y \in V} \prod_{i=1}^{n} \mu_{A_i^l}(x_i) \mu_{B^l}(y)] \\ &= \max_{l=1}^{M} [\prod_{i=1}^{n} \mu_{A_i^l}(x_i)] \\ &= \prod_{i=1}^{n} \mu_{A_i^{l*}}(x_i) \end{aligned} \tag{9.17}$$

where $l*$ is defined according to (9.15). Since $\mu_{B^l}(\bar{y}^{l*}) \leq 1$ when $l \neq l*$ and $\mu_{B^{l*}}(\bar{y}^{l*}) = 1$, we have

$$\begin{aligned} \mu_{B'}(\bar{y}^{l*}) &= \max_{l=1}^{M} [\prod_{i=1}^{n} \mu_{A_i^l}(x_i) \mu_{B^l}(\bar{y}^{l*})] \\ &= \prod_{i=1}^{n} \mu_{A_i^{l*}}(x_i) \end{aligned} \tag{9.18}$$

Hence, the $\sup_{y\in V}$ in (9.16) is achieved at $\bar{y}^{l*}$. Using the maximum defuzzifier (8.23) we obtain (9.14). □

From Lemma 9.4 we see that the fuzzy systems in this case are simple functions, that is, they are piece-wise constant functions, and these constants are the centers of the membership functions in the THEN parts of the rules. From (9.15) we see that as long as the product of membership values of the IF-part fuzzy sets of the rule is greater than or equal to those of the other rules, the output of the fuzzy system remains unchanged. Therefore, these kinds of fuzzy systems are robust to small disturbances in the input and in the membership functions $\mu_{A_i^l}(x_i)$. However, these fuzzy systems are not continuous, that is, when $l*$ changes from one number to the other, $f(x)$ changes in a discrete fashion. If the fuzzy systems are used in decision making or other open-loop applications, this kind of abrupt change may be tolerated, but it is usually unacceptable in closed-loop control.

The next lemma shows that we can obtain a similar result if we use the minimum inference engine.

Lemma 9.5. If we change the product inference engine in Lemma 9.4 to the minimum inference engine (7.24) and keep the others unchanged, then the fuzzy systems are of the same form as (9.14) with $l*$ determined by

$$\min_{i=1}^{n}[\mu_{A_i^{l*}}(x_i)] \geq \min_{i=1}^{n}[\mu_{A_i^l}(x_i)] \tag{9.19}$$

where $l = 1, 2, ..., M$.

Proof: From (7.29) (Lemma 7.3) and using the facts that $\sup_{y\in V}$ and $\max_{l=1}^{M}$ are interchangeable and that B^l are normal, we have

$$\begin{aligned}\sup_{y\in V}\mu_{B^l}(y) &= \max_{l=1}^{M}[\sup_{y\in V} min(\mu_{A_1^l}(x_1), ..., \mu_{A_n^l}(x_n), \mu_{B^l}(y))] \\ &= \max_{l=1}^{M}[\min_{i=1}^{n}(\mu_{A_i^l}(x_i))] \\ &= \min_{i=1}^{n}(\mu_{A_i^{l*}}(x_i)) \end{aligned} \tag{9.20}$$

Also from (7.29) we have that $\mu_{B^l}(\bar{y}^{l*}) = \min_{i=1}^{n}(\mu_{A_i^{l*}}(x_i))$, thus the $\sup_{y\in V}$ in (9.20) is achieved at $\bar{y}^{l*}$. Hence, the maximum defuzzifier (8.23) gives (9.14). □

Again, we obtain a class of fuzzy systems that are simple functions.

It is difficult to obtain closed-form formulas for fuzzy systems with maximum defuzzifier and Lukasiewicz, Zadeh, or Dienes-Rescher inference engines. The difficulty comes from the fact that the $\sup_{y\in V}$ and *min* operators are not interchangeable in general, therefore, from (7.30)-(7.32) we see that the maximum defuzzification becomes an optimization problem for a non-smooth function. In these cases, for a given input x, the output of the fuzzy system has to be computed in a step-by-step fashion, that is, computing the outputs of fuzzifier, fuzzy inference engine, and defuzzifier in sequel. Note that the output of the fuzzy inference engine is a function,

not a single value, so the computation is very complex. We will not use this type of fuzzy systems (maximum defuzzifier with Lukasiewicz, Zadeh, or Dienes-rescher inference engine) in the rest of this book.

9.2 Fuzzy Systems As Universal Approximators

In the last section we showed that certain types of fuzzy systems can be written as compact nonlinear formulas. On one hand, these compact formulas simplify the computation of the fuzzy systems; on the other hand, they give us a chance to analyze the fuzzy systems in more details. We see that the fuzzy systems are particular types of nonlinear functions, so no matter whether the fuzzy systems are used as controllers or decision makers or signal processors or any others, it is interesting to know the capability of the fuzzy systems from a function approximation point of view. For example, what types of nonlinear functions can the fuzzy systems represent or approximate and to what degree of accuracy? If the fuzzy systems can approximate only certain types of nonlinear functions to a limited degree of accuracy, then the fuzzy systems would not be very useful in general applications. But if the fuzzy systems can approximate any nonlinear function to arbitrary accuracy, then they would be very useful in a wide variety of applications. In this section, we prove that certain classes of fuzzy systems that we studied in the last section have this universal approximation capability. Specifically, we have the following main theorem.

Theorem 9.1 (Universal Approximation Theorem). Suppose that the input universe of discourse U is a compact set in R^n. Then, for any given real continuous function $g(x)$ on U and arbitrary $\epsilon > 0$, there exists a fuzzy system $f(x)$ in the form of (9.6) such that

$$\sup_{x \in U} |f(x) - g(x)| < \epsilon \tag{9.21}$$

That is, the fuzzy systems with product inference engine, singleton fuzzifier, center average defuzzifier, and Gaussian membership functions are universal approximators.

One proof of this theorem is based on the following Stone-Weierstrass Theorem, which is well known in analysis.

Stone-Weierstrass Theorem (Rudin [1976]). Let Z be a set of real continuous functions on a compact set U. If (i) Z is an *algebra*, that is, the set Z is closed under addition, multiplication, and scalar multiplication; (ii) Z *separates points on* U, that is, for every $x, y \in U, x \neq y$, there exists $f \in Z$ such that $f(x) \neq f(y)$; and (iii) Z *vanishes at no point of* U, that is, for each $x \in U$ there exists $f \in Z$ such that $f(x) \neq 0$; then for any real continuous function $g(x)$ on U and arbitrary $\epsilon > 0$, there exists $f \in Z$ such that $\sup_{x \in U} |f(x) - g(x)| < \epsilon$.

Proof of Theorem 9.1: Let Y be the set of all fuzzy systems in the form of

(9.6). We now show that Y is an algebra, Y separates points on U, and Y vanishes at no point of U.

Let $f_1, f_2 \in Y$, so that we can write them as

$$f_1(x) = \frac{\sum_{l=1}^{M1} \bar{y1}^l [\prod_{i=1}^{n} a1_i^l exp(-(\frac{x_i - \bar{x1}_i^l}{\sigma 1_i^l})^2)]}{\sum_{l=1}^{M1} [\prod_{i=1}^{n} a1_i^l exp(-(\frac{x_i - \bar{x1}_i^l}{\sigma 1_i^l})^2)]} \tag{9.22}$$

$$f_2(x) = \frac{\sum_{l=1}^{M2} \bar{y2}^l [\prod_{i=1}^{n} a2_i^l exp(-(\frac{x_i - \bar{x2}_i^l}{\sigma 2_i^l})^2)]}{\sum_{l=1}^{M2} [\prod_{i=1}^{n} a2_i^l exp(-(\frac{x_i - \bar{x2}_i^l}{\sigma 2_i^l})^2)]} \tag{9.23}$$

Hence,

$$f_1(x)+f_2(x) = \frac{\sum_{l1=1}^{M1} \sum_{l2=1}^{M2} (\bar{y1}^{l1} + \bar{y2}^{l2}) [\prod_{i=1}^{n} a1_i^{l1} a2_i^{l2} exp(-(\frac{x_i - \bar{x1}_i^{l1}}{\sigma 1_i^{l1}})^2 - (\frac{x_i - \bar{x2}_i^{l2}}{\sigma 2_i^{l2}})^2)]}{\sum_{l1=1}^{M1} \sum_{l2=1}^{M2} [\prod_{i=1}^{n} a1_i^{l1} a2_i^{l2} exp(-(\frac{x_i - \bar{x1}_i^{l1}}{\sigma 1_i^{l1}})^2 - (\frac{x_i - \bar{x2}_i^{l2}}{\sigma 2_i^{l2}})^2)]} \tag{9.24}$$

Since $a1_i^{l1} a2_i^{l2} exp(-(\frac{x_i - \bar{x1}_i^{l1}}{\sigma 1_i^{l1}})^2 - (\frac{x_i - \bar{x2}_i^{l2}}{\sigma 2_i^{l2}})^2)$ can be represented in the form of (9.4) and $\bar{y1}^{l1} + \bar{y2}^{l2}$ can be viewed as the center of a fuzzy set in the form of (9.5), $f_1(x) + f_2(x)$ is in the form of (9.6); that is, $f_1 + f_2 \in Y$. Similarly,

$$f_1(x)f_2(x) = \frac{\sum_{l1=1}^{M1} \sum_{l2=1}^{M2} (\bar{y1}^{l1} \bar{y2}^{l2}) [\prod_{i=1}^{n} a1_i^{l1} a2_i^{l2} exp(-(\frac{x_i - \bar{x1}_i^{l1}}{\sigma 1_i^{l1}})^2 - (\frac{x_i - \bar{x2}_i^{l2}}{\sigma 2_i^{l2}})^2)]}{\sum_{l1=1}^{M1} \sum_{l2=1}^{M2} [\prod_{i=1}^{n} a1_i^{l1} a2_i^{l2} exp(-(\frac{x_i - \bar{x1}_i^{l1}}{\sigma 1_i^{l1}})^2 - (\frac{x_i - \bar{x2}_i^{l2}}{\sigma 2_i^{l2}})^2)]} \tag{9.25}$$

which also is in the form of (9.6), hence, $f_1 f_2 \in Y$. Finally, for arbitrary $c \in R$,

$$cf_1(x) = \frac{\sum_{l=1}^{M1} c\bar{y1}^l [\prod_{i=1}^{n} a1_i^l exp(-(\frac{x_i - \bar{x1}_i^l}{\sigma 1_i^l})^2)]}{\sum_{l=1}^{M1} [\prod_{i=1}^{n} a1_i^l exp(-(\frac{x_i - \bar{x1}_i^l}{\sigma 1_i^l})^2)]} \tag{9.26}$$

which is again in the form of (9.6), so $cf_1 \in Y$. Hence, Y is an algebra.

We show that Y separates points on U by constructing a required fuzzy system $f(x)$. Let $x^0, z^0 \in U$ be two arbitrary points and $x^0 \neq z^0$. We choose the parameters of the $f(x)$ in the form of (9.6) as follows: $M = 2, \bar{y}^1 = 0, \bar{y}^2 = 1, a_i^l = 1, \sigma_i^l = 1, \bar{x}_i^1 = x_i^0$ and $\bar{x}_i^2 = z_i^0$, where $i = 1, 2, ..., n$ and $l = 1, 2$. This specific fuzzy system is

$$f(x) = \frac{exp(-||x - z^0||_2^2)}{exp(-||x - x^0||_2^2) + exp(-||x - z^0||_2^2)} \tag{9.27}$$

from which we have

$$f(x^0) = \frac{exp(-||x^0 - z^0||_2^2)}{1 + exp(-||x^0 - z^0||_2^2)} \tag{9.28}$$

and

$$f(z^0) = \frac{1}{1 + exp(-||x^0 - z^0||_2^2)} \tag{9.29}$$

Since $x^0 \neq z^0$, we have $exp(-||x^0 - z^0||_2^2) \neq 1$ which, from (9.28) and (9.29), gives $f(x^0) \neq f(z^0)$. Hence, Y separates points on U.

To show that Y vanishes at no point of U, we simply observe that any fuzzy system $f(x)$ in the form of (9.6) with all $\bar{y}^l > 0$ has the property that $f(x) > 0, \forall x \in U$. Hence, Y vanishes at no point of U.

In summary of the above and the Stone-Weierstrass Theorem, we obtain the conclusion of this theorem. □

Theorem 9.1 shows that fuzzy systems can approximate continuous functions to arbitrary accuracy; the following corollary extends the result to discrete functions.

Corollary 9.1. For any square-integrable function $g(x)$ on the compact set $U \subset R^n$, that is, for any $g \in L_2(U) = \{g : U \to R | \int_U |g(x)|^2 dx < \infty\}$, there exists fuzzy system $f(x)$ in the form of (9.6) such that

$$(\int_U |f(x) - g(x)|^2 dx)^{1/2} < \epsilon \tag{9.30}$$

Proof: Since U is compact, $\int_U dx = E < \infty$. Since continuous functions on U form a dense subset of $L_2(U)$ (Rudin [1976]), for any $g \in L_2(U)$ there exists a continuous function $\bar{g}$ on U such that $(\int_U |g(x) - \bar{g}(x)|^2 dx)^{1/2} < \epsilon/2$. By Theorem 9.1, there exists $f \in Y$ such that $\sup_{x \in U} |f(x) - \bar{g}(x)| < \epsilon/(2E^{1/2})$. Hence, we have

$$\begin{aligned}(\int_U |f(x) - g(x)|^2 dx)^{1/2} &\leq (\int_U |f(x) - \bar{g}(x)|^2 dx)^{1/2} + (\int_U |\bar{g}(x) - g(x)|^2 dx)^{1/2} \\ &< (\int_U (\sup_{x \in U} |f(x) - \bar{g}(x)|)^2 dx)^{1/2} + \epsilon/2 \\ &< (\frac{\epsilon^2}{2^2 E} E)^{1/2} + \epsilon/2 = \epsilon \end{aligned} \tag{9.31}$$

□

Theorem 9.1 and Corollary 9.1 provide a justification for using fuzzy systems in a variety of applications. Specifically, they show that for any kind of nonlinear operations the problem may require, it is always possible to design a fuzzy system that performs the required operation with any degree of accuracy. They also provide a theoretical explanation for the success of fuzzy systems in practical applications.

However, Theorem 9.1 and Corollary 9.1 give only existence result; that is, they show that there *exists* a fuzzy system in the form of (9.6) that can approximate any function to arbitrary accuracy. They do not show how to find such a fuzzy system. For engineering applications, knowing the existence of an ideal fuzzy system is not enough; we must develop approaches that can find good fuzzy systems for the

particular applications. Depending upon the information provided, we may or may not find the ideal fuzzy system. In the next few chapters, we will develop a variety of approaches to designing the fuzzy systems.

9.3 Summary and Further Readings

In this chapter we have demonstrated the following:

- The compact formulas of some useful classes of fuzzy systems.
- How to derive compact formulas for any classes of fuzzy systems if such compact formulas exist.
- How to use the Stone-Weierstrass Theorem.

The derivations of the mathematical formulas of the fuzzy systems are new. A related reference is Wang [1994]. The Universal Approximation Theorem and its proof are taken from Wang [1992]. Other approaches to this problem can be found in Buckley [1992b] and Zeng and Singh [1994].

9.4 Exercises

Exercise 9.1. Derive the compact formula for the fuzzy systems with fuzzy rule base (7.1), Zadeh inference engine (7.31), singleton fuzzifier (8.1), and center average defuzzifier (8.18).

Exercise 9.2. Repeat Exercise 9.1 using Lukasiewicz inference engine rather than Zadeh inference engine.

Exercise 9.3. Show that the fuzzy systems in the form of (9.1) have the universal approximation property in Theorem 9.1.

Exercise 9.4. Can you use the Stone-Weierstrass Theorem to prove that fuzzy systems in the form of (9.7) or (9.6) with $a_i^l = 1$ are universal approximators? Explain your answer.

Exercise 9.5. Use the Stone-Weierstrass Theorem to prove that polynomials are universal appproximators.

Exercise 9.6. Plot the fuzzy systems $f_1(x)$ and $f_2(x)$ for $x \in U = [-1, 2] \times [-1, 2]$, where $f_1(x)$ is the fuzzy system with the two rules (8.27) and (8.28), product inference engine (7.23), singleton fuzzifier (8.1), and maximum defuzzifier (8.23), and $f_2(x)$ is the same as $f_1(x)$ except that the maximum defuzzifier is replaced by the center average defuzzifier (8.18).

Chapter 10

Approximation Properties of Fuzzy Systems I

In Chapter 9 we proved that fuzzy systems are universal approximators; that is, they can approximate any function on a compact set to arbitrary accuracy. However, this result showed only the existence of the optimal fuzzy system and did not provide methods to find it. In fact, finding the optimal fuzzy system is much more difficult than proving its existence. Depending upon the information provided, we may or may not find the optimal fuzzy system.

To answer the question of how to find the optimal fuzzy system, we must first see what information is available for the nonlinear function $g(x) : U \subset R^n \to R$, which we are asked to approximate. Generally speaking, we may encounter the following three situations:

- The analytic formula of $g(x)$ is known.
- The analytic formula of $g(x)$ is unknown, but for any $x \in U$ we can determine the corrspending $g(x)$. That is, $g(x)$ is a black box—we know the input-output behavior of $g(x)$ but do not know the details inside it.
- The analytic formula of $g(x)$ is unknown and we are provided only a limited number of input-output pairs $(x^j, g(x^j))$, where $x^j \in U$ cannot be arbitrarily chosen.

The first case is not very interesting because if the analytic formula of $g(x)$ is known, we can use it for whatever purpose the fuzzy system tries to achieve. In the rare case where we want to replace $g(x)$ by a fuzzy system, we can use the methods for the second case because the first case is a special case of the second one. Therefore, we will not consider the first case separately.

The second case is more realistic. We will study it in detail in this and the following chapters.

The third case is the most general one in practice. This is especially true for fuzzy control because stability requirements for control systems may prevent us from choosing the input values arbitrarily. We will study this case in detail in Chapters 12-15.

So, in this chapter we assume that the analytic formula of $g(x)$ is unknown but we can determine the input-output pairs $(x; g(x))$ for any $x \in U$. Our task is to design a fuzzy system that can approximate $g(x)$ in some optimal manner.

10.1 Preliminary Concepts

We first introduce some concepts.

Definition 10.1: *Pseudo-Trapezoid Membership Function.* Let $[a, d] \subset R$. The *pseudo-trapezoid membership function* of fuzzy set A is a continuous function in R given by

$$\mu_A(x; a, b, c, d, H) = \begin{cases} I(x), \ x \in [a, b) \\ H, \ x \in [b, c] \\ D(x), \ x \in (c, d] \\ 0, \ x \in R - (a, d) \end{cases} \tag{10.1}$$

where $a \leq b \leq c \leq d, a < d, 0 < H \leq 1, 0 \leq I(x) \leq 1$ is a nondecreasing function in $[a, b)$ and $0 \leq D(x) \leq 1$ is a nonincreasing function in $(c, d]$. When the fuzzy set A is normal (that is, $H = 1$), its membership function is simply written as $\mu_A(x; a, b, c, d)$.

Fig. 10.1 shows some examples of pseudo-trapezoid membership functions. If the universe of discourse is bounded, then a, b, c, d are finite numbers. Pseudo-trapezoid membership functions include many commonly used membership functions as special cases. For example, if we choose

$$I(x) = \frac{x - a}{b - a}, D(x) = \frac{x - d}{c - d} \tag{10.2}$$

then the pseudo-trapezoid membership functions become the *trapezoid membership functions*. If $b = c$ and $I(x)$ and $D(x)$ are as in (10.2), we obtain the *triangular membership functions*; a triangular membership function is denoted as $\mu_A(x; a, b, d)$. If we choose $a = \infty, b = c = \bar{x}, d = \infty$, and

$$I(x) = D(x) = exp(-(\frac{x - \bar{x}}{\sigma})^2) \tag{10.3}$$

then the pseudo-trapezoid membership functions become the *Gaussian membership functions*. Therefore, the pseudo-trapezoid membership functions constitute a very rich family of membership functions.

Definition 10.2: *Completeness of Fuzzy Sets.* Fuzzy sets $A^1, A^2, ..., A^N$ in $W \subset R$ are said to be *complete on* W if for any $x \in W$, there exists A^j such that $\mu_{A^j}(x) > 0$.

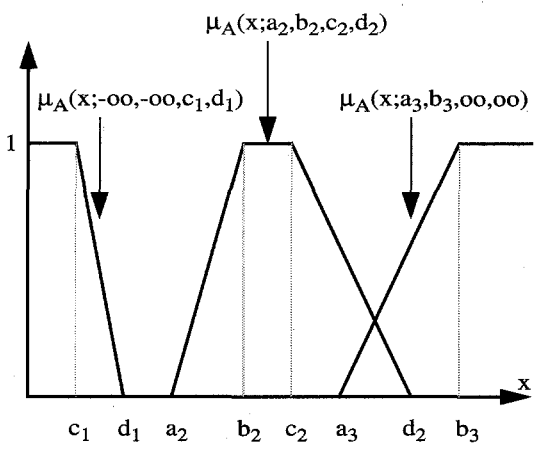


Figure 10.1. Examples of pseudo-trapezoid membership functions.

Definition 10.3: *Consistency of Fuzzy Sets.* Fuzzy sets $A^1, A^2, ..., A^N$ in $W \subset R$ are said to be *consistent on* W if $\mu_{A^i}(x) = 1$ for some $x \in W$ implies that $\mu_{A^i}(x) = 0$ for all $i \neq j$.

Definition 10.4: *High Set of Fuzzy Set.* The *high set* of a fuzzy set A in $W \subset R$ is a subset in W defined by

$$hgh(A) = \{x \in W | \mu_A(x) = \sup_{x' \in W} \mu_A(x')\} \tag{10.4}$$

If A is a normal fuzzy set with pseudo-trapezoid membership function $\mu_A(x; a, b, c, d)$, then $hgh(A) = [b, c]$.

Definition 10.5: *Order Between Fuzzy Sets.* For two fuzzy sets A and B in $W \subset R$, we say $A > B$ if $hgh(A) > hgh(B)$ (that is, if $x \in hgh(A)$ and $x' \in hgh(B)$, then $x > x'$).

We now show some properties of fuzzy sets with pseudo-trapezoid membership functions.

Lemma 10.1. If $A^1, A^2, ..., A^N$ are consistent and normal fuzzy sets in $W \subset R$ with pseudo-trapezoid membership functions $\mu_{A^i}(x; a_i, b_i, c_i, d_i)$ $(i = 1, 2, ..., N)$, then there exists a rearrangement $\{i_1, i_2, ..., i_N\}$ of $\{1, 2, ..., N\}$ such that

$$A^{i_1} < A^{i_2} < \cdots < A^{i_N} \tag{10.5}$$

Proof: For arbitrary $i, j \in \{1, 2, ..., N\}$, it must be true that $[b_i, c_i] \cap [b_j, c_j] = 0$,

since otherwise the fuzzy sets $A^1, ..., A^N$ would not be consistent. Thus, there exists a rearrangement $\{i_1, i_2, ..., i_N\}$ of $\{1, 2, ..., N\}$ such that

$$[b_{i_1}, c_{i_1}] < [b_{i_2}, c_{i_2}] < \cdots < [b_{i_N}, c_{i_N}] \tag{10.6}$$

which implies (10.5). □

Lemma 10.1 shows that we can always assume $A^1 < A^2 < \cdots < A^N$ without loss of generality.

Lemma 10.2. Let the fuzzy sets $A^1, A^2, ..., A^N$ in $W \subset R$ be normal, consistent and complete with pseudo-trapezoid membership functions $\mu_{A^i}(x; a_i, b_i, c_i, d_i)$. If $A^1 < A^2 < \cdots < A^N$, then

$$c_i \leq a_{i+1} < d_i \leq b_{i+1} \tag{10.7}$$

for $i = 1, 2, ..., N-1$.

Fig. 10.2 illustrates the assertion of Lemma 10.2. The proof of Lemma 10.2 is straightforward and is left as an exercise.

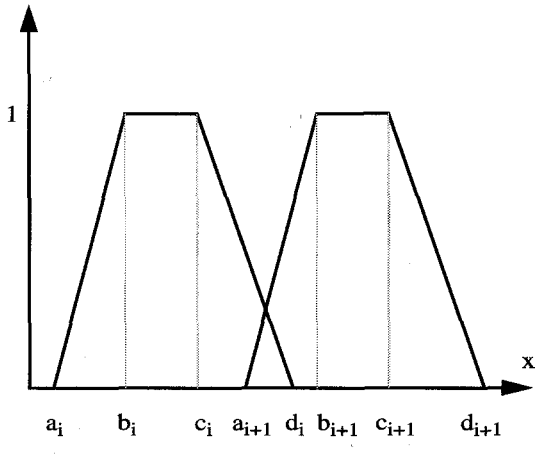


Figure 10.2. An example of Lemma 10.2: $c_i \leq a_{i+1} < d_i \leq b_{i+1}$.

10.2 Design of Fuzzy System

We are now ready to design a particular type of fuzzy systems that have some nice properties. For notational simplicity and ease of graphical explanation, we consider

two-input fuzzy systems; however, the approach and results are all valid for general n-input fuzzy systems. That is, we can use exactly the same procedure to design n-input fuzzy systems. We first specify the problem.

The Problem: Let $g(x)$ be a function on the compact set $U = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \subset R^2$ and the analytic formula of $g(x)$ be unknown. Suppose that for any $x \in U$, we can obtain $g(x)$. Our task is to design a fuzzy system that approximates $g(x)$.

We now design such a fuzzy system in a step-by-step manner.

Design of a Fuzzy System:

- **Step 1.** Define N_i $(i = 1, 2)$ fuzzy sets $A_i^1, A_i^2, ..., A_i^{N_i}$ in $[\alpha_i, \beta_i]$ which are normal, consistent, complete with pesudo-trapezoid membership functions $\mu_{A_i^1}(x_i; a_i^1, b_i^1, c_i^1, d_i^1)$, ...,$\mu_{A_i^{N_i}}(x_i; a_i^{N_i}, b_i^{N_i}, c_i^{N_i}, d_i^{N_i})$, and $A_i^1 < A_i^2 < \cdots < A_i^{N_i}$ with $a_i^1 = b_i^1 = \alpha_i$ and $c_i^{N_i} = d_i^{N_i} = \beta_i$. Define $e_1^1 = \alpha_1, e_1^{N_1} = \beta_1$, and $e_1^j = \frac{1}{2}(b_1^j + c_1^j)$ for $j = 2, 3, ..., N_1 - 1$. Similarly, define $e_2^1 = \alpha_2, e_2^{N_2} = \beta_2$, and $e_2^j = \frac{1}{2}(b_2^j + c_2^j)$ for $j = 2, 3, ..., N_2 - 1$. Fig. 10.3 shows an example with $N_1 = 3, N_2 = 4, \alpha_1 = \alpha_2 = 0$ and $\beta_1 = \beta_2 = 1$.

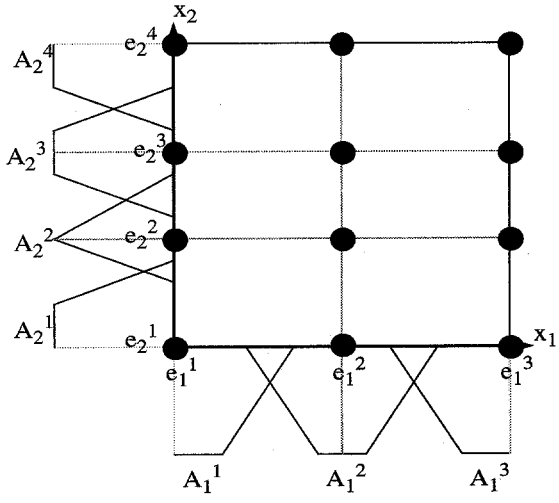


Figure 10.3. An example of the fuzzy sets defined in Step 1 of the design procedure.

- **Step 2.** Construct $M = N_1 \times N_2$ fuzzy IF-THEN rules in the following form:

$$Ru^{i_1 i_2} : \textit{IF } x_1 \textit{ is } A_1^{i_1} \textit{ and } x_2 \textit{ is } A_2^{i_2}, \textit{ THEN } y \textit{ is } B^{i_1 i_2} \qquad (10.8)$$

where $i_1 = 1, 2, ..., N_1, i_2 = 1, 2, ..., N_2$, and the center of the fuzzy set $B^{i_1 i_2}$, denoted by $\bar{y}^{i_1 i_2}$, is chosen as

$$\bar{y}^{i_1 i_2} = g(e_1^{i_1}, e_2^{i_2}) \tag{10.9}$$

For the example in Fig. 10.3, we have $3 \times 4 = 12$ rules, and the centers of $B^{i_1 i_2}$ are equal to the $g(x)$ evaluated at the 12 dark points shown in the figure.

- **Step 3**. Construct the fuzzy system $f(x)$ from the $N_1 \times N_2$ rules of (10.8) using product inference engine (7.23), singleton fuzzifier (8.1), and center average defuzzifier (8.18) (see Lemma 9.1):

$$f(x) = \frac{\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \bar{y}^{i_1 i_2} (\mu_{A_1^{i_1}}(x_1) \mu_{A_2^{i_2}}(x_2))}{\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} (\mu_{A_1^{i_1}}(x_1) \mu_{A_2^{i_2}}(x_2))} \tag{10.10}$$

Since the fuzzy sets $A_i^1, ... A_i^{N_i}$ are complete, at every $x \in U$ there exist i_1 and i_2 such that $\mu_{A_1^{i_1}}(x_1) \mu_{A_2^{i_2}}(x_2) \neq 0$. Hence, the fuzzy system (10.10) is well defined, that is, its denominator is always nonzero.

From Step 2 we see that the IF parts of the rules (10.8) constitute all the possible combinations of the fuzzy sets defined for each input variable. So, if we generalize the design procedure to n-input fuzzy systems and define N fuzzy sets for each input variable, then the total number of rules is N^n; that is, by using this design method, the number of rules increases exponentially with the dimension of the input space. This is called the *curse of dimensionality* and is a general problem for all high-dimensional approximation problems. We will address this issue again in Chapter 22.

The final observation of the design procedure is that we must know the values of $g(x)$ at $x = (e_1^{i_1}, e_2^{i_2})$ for $i_1 = 1, 2, ..., N_1$ and $i_2 = 1, 2, ..., N_2$. Since $(e_1^{i_1}, e_2^{i_2})$ can be arbitrary points in U, this is equivalent to say that we need to know the values of $g(x)$ at any $x \in U$.

Next, we study the approximation accuracy of the $f(x)$ designed above to the unknown function $g(x)$.

10.3 Approximation Accuracy of the Fuzzy System

Theorem 10.1. Let $f(x)$ be the fuzzy system in (10.10) and $g(x)$ be the unknown function in (10.9). If $g(x)$ is continuously differentiable on $U = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$, then

$$||g - f||_\infty \leq ||\frac{\partial g}{\partial x_1}||_\infty h_1 + ||\frac{\partial g}{\partial x_2}||_\infty h_2 \tag{10.11}$$

where the infinite norm $|| * ||_\infty$ is defined as $||d(x)||_\infty = \sup_{x \in U} |d(x)|$, and $h_i = \max_{1 \leq j \leq N_i - 1} |e_i^{j+1} - e_i^j|$ $(i = 1, 2)$.

Proof: Let $U^{i_1 i_2} = [e_1^{i_1}, e_1^{i_1+1}] \times [e_2^{i_2}, e_2^{i_2+1}]$, where $i_1 = 1, 2, ..., N_1 - 1$ and $i_2 = 1, 2, ..., N_2 - 1$. Since $[\alpha_i, \beta_i] = [e_i^1, e_i^2] \cup [e_i^2, e_i^3] \cup \cdots \cup [e_i^{N_i - 1}, e_i^{N_i}]$, $i = 1, 2$, we have

$$U = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] = \bigcup_{i_1=1}^{N_1-1} \bigcup_{i_2=1}^{N_2-1} U^{i_1 i_2} \tag{10.12}$$

which implies that for any $x \in U$, there exists $U^{i_1 i_2}$ such that $x \in U^{i_1 i_2}$. Now suppose $x \in U^{i_1 i_2}$, that is, $x_1 \in [e_1^{i_1}, e_1^{i_1+1}]$ and $x_2 \in [e_2^{i_2}, e_2^{i_2+1}]$ (since x is fixed, i_1 and i_2 are also fixed in the sequel). Since the fuzzy sets $A_1^1, ..., A_1^{N_1}$ are normal, consistent and complete, at least one and at most two $\mu_{A_1^{j_1}}(x_1)$ are nonzero for $j_1 = 1, 2, ..., N_1$. From the definition of $e_1^{j_1}$ ($j_1 = 1, 2, ..., N_1 - 1$), these two possible nonzero $\mu_{A_1^{j_1}}(x_1)$'s are $\mu_{A_1^{i_1}}(x_1)$ and $\mu_{A_1^{i_1+1}}(x_1)$. Similarly, the two possible nonzero $\mu_{A_2^{j_2}}(x_2)$'s (for $j_2 = 1, 2, ..., N_2$) are $\mu_{A_2^{i_2}}(x_2)$ and $\mu_{A_2^{i_2+1}}(x_2)$. Hence, the fuzzy system $f(x)$ of (10.10) is simplified to

$$\begin{aligned} f(x) &= \frac{\sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} \bar{y}^{j_1 j_2} (\mu_{A_1^{j_1}}(x_1) \mu_{A_2^{j_2}}(x_2))}{\sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} (\mu_{A_1^{j_1}}(x_1) \mu_{A_2^{j_2}}(x_2))} \\ &= \sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} \left[\frac{\mu_{A_1^{j_1}}(x_1) \mu_{A_2^{j_2}}(x_2)}{\sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} (\mu_{A_1^{j_1}}(x_1) \mu_{A_2^{j_2}}(x_2))} \right] g(e_1^{j_1}, e_2^{j_2}) \end{aligned} \tag{10.13}$$

where we use (10.9). Since

$$\sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} \left[\frac{\mu_{A_1^{j_1}}(x_1) \mu_{A_2^{j_2}}(x_2)}{\sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} (\mu_{A_1^{j_1}}(x_1) \mu_{A_2^{j_2}}(x_2))} \right] = 1 \tag{10.14}$$

we have

$$\begin{aligned} |g(x) - f(x)| &\le \sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} \left[\frac{\mu_{A_1^{j_1}}(x_1) \mu_{A_2^{j_2}}(x_2)}{\sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} (\mu_{A_1^{j_1}}(x_1) \mu_{A_2^{j_2}}(x_2))} \right] |g(x) - g(e_1^{j_1}, e_2^{j_2})| \\ &\le \max_{j_1=i_1, i_1+1; j_2=i_2, i_2+1} |g(x) - g(e_1^{j_1}, e_2^{j_2})| \end{aligned} \tag{10.15}$$

Using the Mean Value Theorem, we can further write (10.15) as

$$|g(x) - f(x)| \le \max_{j_1=i_1, i_1+1; j_2=i_2, i_2+1} (\| \frac{\partial g}{\partial x_1} \|_\infty |x_1 - e_1^{j_1}| + \| \frac{\partial g}{\partial x_2} \|_\infty |x_2 - e_2^{j_2}|) \tag{10.16}$$

Since $x \in U^{i_1 i_2}$, which means that $x_1 \in [e_1^{i_1}, e_1^{i_1+1}]$ and $x_2 \in [e_2^{i_2}, e_2^{i_2+1}]$, we have that $|x_1 - e_1^{j_1}| \le |e_1^{i_1+1} - e_1^{i_1}|$ and $|x_2 - e_2^{j_2}| \le |e_2^{i_2} - e_2^{i_2+1}|$ for $j_1 = i_1, i_1 + 1$ and $j_2 = i_2, i_2 + 1$. Thus, (10.16) becomes

$$|g(x) - f(x)| \le \| \frac{\partial g}{\partial x_1} \|_\infty |e_1^{i_1+1} - e_1^{i_1}| + \| \frac{\partial g}{\partial x_2} \|_\infty |e_2^{i_2+1} - e_2^{i_2}| \tag{10.17}$$

from which we have

$$\begin{aligned} ||g-f||_\infty &= \sup_{x\in U}|g(x)-f(x)| \\ &\le ||\frac{\partial g}{\partial x_1}||_\infty \max_{1\le i_1\le N_1-1}|e_1^{i_1+1}-e_1^{i_1}| + ||\frac{\partial g}{\partial x_2}||_\infty \max_{1\le i_2\le N_2-1}|e_2^{i_2+1}-e_2^{i_2}| \\ &= ||\frac{\partial g}{\partial x_1}||_\infty h_1 + ||\frac{\partial g}{\partial x_2}||_\infty h_2 \end{aligned} \tag{10.18}$$

□

Theorem 10.1 is an important theorem. We can draw a number of conclusions from it, as follows:

- From (10.11) we can conclude that fuzzy systems in the form of (10.10) are universal approximators. Specifically, since $||\frac{\partial g}{\partial x_1}||_\infty$ and $||\frac{\partial g}{\partial x_2}||_\infty$ are finite numbers (a continuous function on a compact set is bounded by a finite number), for any given $\epsilon > 0$ we can choose h_1 and h_2 small enough such that $||\frac{\partial g}{\partial x_1}||_\infty h_1 + ||\frac{\partial g}{\partial x_2}||_\infty h_2 < \epsilon$. Hence from (10.11) we have $\sup_{x\in U}|g(x) - f(x)| = ||g-f||_\infty < \epsilon$.

- From (10.11) and the definition of h_1 and h_2 we see that more accurate approximation can be obtained by defining more fuzzy sets for each x_i. This confirms the intuition that more rules result in more powerful fuzzy systems.

- From (10.11) we see that in order to design a fuzzy system with a prespecified accuracy, we must know the bounds of the derivatives of $g(x)$ with respect to x_1 and x_2, that is, $||\frac{\partial g}{\partial x_1}||_\infty$ and $||\frac{\partial g}{\partial x_2}||_\infty$. In the design process, we need to know the value of $g(x)$ at $x = (e_1^{i_1}, e_2^{i_2})$ for $i_1 = 1, 2, ..., N_1$ and $i_2 = 1, 2, ..., N_2$. Therefore, this approach requires these two pieces of information in order for the designed fuzzy system to achieve any prespecified degree of accuracy.

- From the proof of Theorem 10.1 we see that if we change $\mu_{A_1^{i_1}}(x_1)\mu_{A_2^{i_2}}(x_2)$ to $min[\mu_{A_1^{i_1}}(x_1), \mu_{A_2^{i_2}}(x_2)]$, the proof is still vaild. Therefore, if we use minimum inference engine in the design procedure and keep the others unchanged, the designed fuzzy system still has the approximation property in Theorem 10.1. Consequently, the fuzzy systems with minimum inference engine, singleton fuzzifier, center average defuzzifier and pseudo-trapezoid membership functions are universal approximators.

Theorem 10.1 gives the accuracy of $f(x)$ as an approximator to $g(x)$. The next lemma shows at what points $f(x)$ and $g(x)$ are exactly equal.

Lemma 10.3. Let $f(x)$ be the fuzzy system (10.10) and $e_1^{i_1}$ and $e_2^{i_2}$ be the points defined in the design procedure for $f(x)$. Then,

$$f(e_1^{i_1}, e_2^{i_2}) = g(e_1^{i_1}, e_2^{i_2}) \tag{10.19}$$

for $i_1 = 1, 2, ..., N_1$ and $i_2 = 1, 2, ..., N_2$.

Proof: From the definition of $e_1^{i_1}$ and $e_2^{i_2}$ and the fact that $A_i^{j_i}$'s are normal, we have $\mu_{A_1^{i_1}}(e_1^{i_1}) = \mu_{A_2^{i_2}}(e_2^{i_2}) = 1$. Since the fuzzy sets $A_i^1, A_i^2, ..., A_i^{N_i}$ $(i = 1, 2)$ are consistent, we have that $\mu_{A_1^{j_1}}(e_1^{i_1}) = \mu_{A_2^{j_2}}(e_2^{i_2}) = 0$ for $j_1 \neq i_1$ and $j_2 \neq i_2$. Hence from (10.10) and (10.9) we have $f(e_1^{i_1}, e_2^{i_2}) = \bar{y}^{i_1 i_2} = g(e_1^{i_1}, e_2^{i_2})$. □

Lemma 10.3 shows that the fuzzy system (10.10) can be viewed as an interpolation of function $g(x)$ at some regular points $(e_1^{i_1}, e_2^{i_2})$ $(i_1 = 1, 2, ..., N_1, i_2 = 1, 2, ..., N_2)$ in the universe of discourse U. This is intuitively appealing.

Finally, we show two examples of how to use Theorem 10.1 to design the required fuzzy system.

Example 10.1. Design a fuzzy system $f(x)$ to uniformly approximate the continuous function $g(x) = sin(x)$ defined on $U = [-3, 3]$ with a required accuracy of $\epsilon = 0.2$; that is, $\sup_{x \in U} |g(x) - f(x)| < \epsilon$.

Since $||\frac{\partial g}{\partial x}||_\infty = ||cos(x)||_\infty = 1$, from (10.11) we see that the fuzzy system with $h = 0.2$ meets our requirement. Therefore, we define the following 31 fuzzy sets A^j in $U = [-3, 3]$ with the triangular membership functions

$$\mu_{A^1}(x) = \mu_{A^1}(x; -3, -3, -2.8) \tag{10.20}$$

$$\mu_{A^{31}}(x) = \mu_{A^{31}}(x; 2.8, 3, 3) \tag{10.21}$$

and

$$\mu_{A^j}(x) = \mu_{A^j}(x; e^{j-1}, e^j, e^{j+1}) \tag{10.22}$$

where $j = 2, 3, ..., 30$, and $e^j = -3 + 0.2(j - 1)$. These membership functions are shown in Fig. 10.4. According to (10.10), the designed fuzzy system is

$$f(x) = \frac{\sum_{j=1}^{31} sin(e^j)\mu_{A^j}(x)}{\sum_{j=1}^{31} \mu_{A^j}(x)} \tag{10.23}$$

which is plotted in Fig.10.5 against $g(x) = sin(x)$. We see from Fig.10.5 that $f(x)$ and $g(x)$ are almost identical. □

Example 10.2. Design a fuzzy system to uniformly approximate the function $g(x) = 0.52 + 0.1x_1 + 0.28x_2 - 0.06x_1x_2$ defined on $U = [-1, 1] \times [-1, 1]$ with a required accuracy of $\epsilon = 0.1$.

Since $||\frac{\partial g}{\partial x_1}||_\infty = \sup_{x \in U} |0.1 - 0.06x_2| = 0.16$ and $||\frac{\partial g}{\partial x_2}||_\infty = \sup_{x \in U} |0.28 - 0.06x_1| = 0.34$, from (10.11) we see that $h_1 = h_2 = 0.2$ results in $||g - f||_\infty \leq 0.16 * 0.2 + 0.34 * 0.2 = 0.1$. Therefore, we define 11 fuzzy sets A^j $(j = 1, 2, ..., 11)$ in $[-1, 1]$ with the following triangular membership functions:

$$\mu_{A^1}(x) = \mu_{A^1}(x; -1, -1, -0.8) \tag{10.24}$$

$$\mu_{A^{11}}(x) = \mu_{A^{11}}(x; 0.8, 1, 1) \tag{10.25}$$

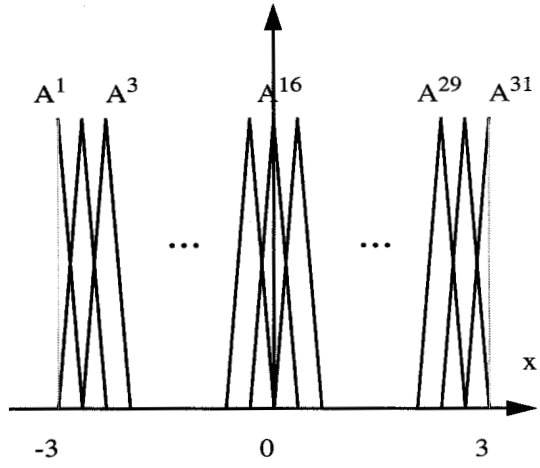


Figure 10.4. Membership functions in Example 10.1.

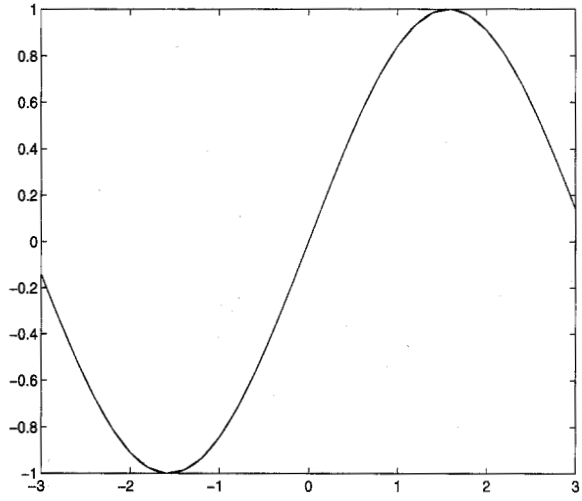


Figure 10.5. The designed fuzzy system $f(x)$ and the function $g(x) = six(x)$ (they are almost identical).

and

$$\mu_{A^j}(x) = \mu_{A^j}(x; e^{j-1}, e^j, e^{j+1}) \tag{10.26}$$

for $j = 2, 3, ..., 10$, where $e^j = -1 + 0.2(j-1)$. The fuzzy system is constructed from the following $11 \times 11 = 121$ rules:

$$IF\ x_1\ is\ A^{i_1}\ and\ x_2\ is\ A^{i_2},\ THEN\ y\ is\ B^{i_1 i_2} \tag{10.27}$$

where $i_1, i_2 = 1, 2, ..., 11$, and the center of $B^{i_1 i_2}$ is $\bar{y}^{i_1 i_2} = g(e^{i_1}, e^{i_2})$. The final fuzzy system is

$$f(x) = \frac{\sum_{i_1=1}^{11} \sum_{i_2=1}^{11} g(e^{i_1}, e^{i_2})(\mu_{A^{i_1}}(x_1)\mu_{A^{i_2}}(x_2))}{\sum_{i_1=1}^{11} \sum_{i_2=1}^{11} (\mu_{A^{i_1}}(x_1)\mu_{A^{i_2}}(x_2))} \tag{10.28}$$

□

From Example 10.2 we see that we need 121 rules to approximate the function $g(x)$. Are so many rules really necessary? In other words, can we improve the bound in (10.11) so that we can use less rules to approximate the same function to the same accuracy? The answer is yes and we will study the details in the next chapter.

10.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- The three types of approximation problems (classified according to the information available).
- The concepts of completeness, consistence, and order of fuzzy sets, and their application to fuzzy sets with pseudo-trapezoid membership functions.
- For a given accuracy requirement, how to design a fuzzy system that can approximate a given function to the required accuracy.
- The idea of proving the approximation bound (10.11).

Approximation accuracies of fuzzy systems were analyzed in Ying [1994] and Zeng and Singh [1995]. This is a relatively new topic and very few references are available.

10.5 Exercises

Exercise 10.1. Let fuzzy sets A^j in $U = [a, b]$ $(j = 1, 2, ..., N)$ be normal, consistent, and complete with pseudo-trapezoid membership functions $\mu_{A^j}(x) =$

$\mu_{A^j}(x; a_j, b_j, c_j, d_j)$. Suppose that $A^1 < A^2 < \cdots < A^N$. Define fuzzy sets B^j whose membership functions are given as

$$\mu_{B^j}(x) = \frac{\mu_{A^j}(x)}{\sum_{i=1}^{N} \mu_{A^i}(x)} \tag{10.29}$$

Show that:

(a) $\mu_{B^j}(x)$ $(j = 1, 2, ..., N)$ are also pesudo-trapezoid membership functions.

(b) The fuzzy sets B^j $(j = 1, 2, ..., N)$ are also normal, consistent, and complete.

(c) $B^1 < B^2 < \cdots < B^N$.

(d) If $\mu_{A^j}(x) = \mu_{A^j}(x; a_j, b_j, c_j, d_j)$ are trapezoid membership functions with $c_i = a_{i+1}, d_i = b_{i+1}$ for $i = 1, 2, ..., N-1$, then $\mu_{B^j}(x) = \mu_{A^j}(x)$ for $j = 1, 2, ..., N-1$.

Exercise 10.2. Design a fuzzy system to uniformly approximate the function $g(x) = sin(x\pi) + cos(x\pi) + sin(x\pi)cos(x\pi)$ on $U = [-1, 1]$ with a required accuracy of $\epsilon = 0.1$.

Exercise 10.3. Design a fuzzy system to uniformly approximate the function $g(x) = sin(x_1\pi) + cos(x_2\pi) + sin(x_1\pi)cos(x_2\pi)$ on $U = [-1, 1] \times [-1, 1]$ with a required accuracy of $\epsilon = 0.1$.

Exercise 10.4. Extend the design method in Section 10.2 to n-input fuzzy systems.

Exercise 10.5. Let the function $g(x)$ on $U = [0, 1]^3$ be given by

$$g(x_1, x_2, x_3) = 1 + \sum_{k_1 k_2 k_3 \in K} x_1^{k_1} x_2^{k_2} x_3^{k_3} \tag{10.30}$$

where $K = \{k_1 k_2 k_3 | k_i = 0, 1; i = 1, 2, 3 \text{ and } k_1 + k_2 + k_3 > 0\}$. Design a fuzzy system to uniformly approximate $g(x)$ with a required accuracy of $\epsilon = 0.05$.

Exercise 10.6. Show that if the fuzzy sets $A_i^1, A_i^2, ..., A_i^{N_i}$ in the design procedure of Section 10.2 are not complete, then the fuzzy system (10.10) is not well defined. If these fuzzy sets are not normal or not consistent, is the fuzzy system (10.10) well defined?

Exercise 10.7. Plot the fuzzy system (10.28) on $U = [-1, 1] \times [-1, 1]$ and compare it with $g(x) = 0.52 + 0.1x_1 + 0.28x_2 - 0.06x_1x_2$.

Chapter 11

Approximation Properties of Fuzzy Systems II

In Chapter 10 we saw that by using the bound in Theorem 10.1 a large number of rules are usually required to approximate some simple functions. For example, Example 10.2 showed that we need 121 rules to approximate a two-dimensional quadratic function. Observing (10.11) we note that the bound is a linear function of h_i. Since h_i are usually small, if a bound could be a linear function of h_i^2, then this bound would be much smaller than the bound in (10.11). That is, if we can obtain tighter bound than that used in (10.11), we may use less rules to approximate the same function with the same accuracy.

In approximation theory (Powell [1981]), if $g(x)$ is a given function on U and $U^{i_1 i_2}$ $(i_1 = 1, 2, ..., N_1, i_2 = 1, 2, ..., N_2)$ is a partition of U as in the proof of Theorem 10.1, then $f(x)$ is said to be the $k'th$ order accurate approximator for $g(x)$ if $||g - f||_\infty \leq M_g h^k$, where M_g is a constant that depends on the function g, and h is the module of the partition that in our case is $max(h_1, h_2)$. In this chapter, we first design a fuzzy system that is a second-order accurate approximator.

11.1 Fuzzy Systems with Second-Order Approximation Accuracy

We first design the fuzzy system in a step-by-step manner and then study its approximation accuracy. As in Chapter 10, we consider two-input fuzzy systems for notational simplicity; the approach and results are still valid for n-input fuzzy systems. The design problem is the same as in Section 10.2.

Design of Fuzzy System with Second-Order Accuracy:

- **Step 1**. Define N_i $(i = 1, 2)$ fuzzy sets $A_i^1, A_i^2, ..., A_i^{N_i}$ in $[\alpha_i, \beta_i]$, which are normal, consistent, and complete with the triangular membership functions

$$\mu_{A_i^1}(x_i) = \mu_{A_i^1}(x_i; e_i^1, e_i^1, e_i^2) \tag{11.1}$$

$$\mu_{A_i^j}(x_i) = \mu_{A_i^j}(x_i; e_i^{j-1}, e_i^j, e_i^{j+1}) \tag{11.2}$$

for $j = 2, 3, ..., N_i - 1$, and

$$\mu_{A_i^{N_i}}(x_i) = \mu_{A_i^{N_i}}(x_i; e_i^{N_i-1}, e_i^{N_i}, e_i^{N_i}) \tag{11.3}$$

where $i = 1, 2$, and $\alpha_i = e_i^1 < e_i^2 < \cdots < e_i^{N_i} = \beta_i$. Fig. 11.1 shows an example with $N_1 = 4, N_2 = 5, \alpha_1 = \alpha_2 = 0$ and $\beta_1 = \beta_2 = 1$.

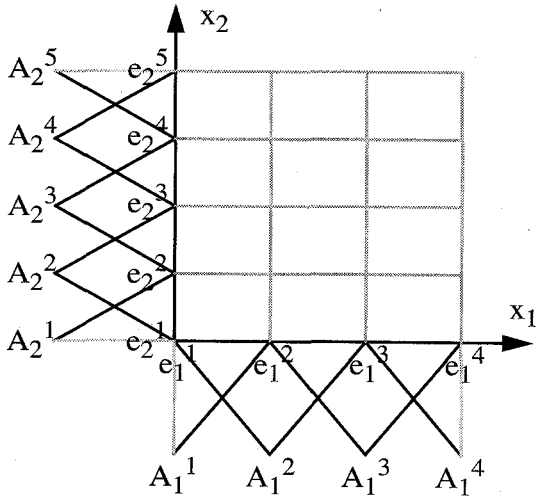


Figure 11.1. An example of fuzzy sets.

- **Steps 2 and 3**. The same as Steps 2 and 3 of the design procedure in Section 10.2. That is, the constructed fuzzy system is given by (10.10), where $\bar{y}^{i_1 i_2}$ are given by (10.9) and the $A_1^{i_1}$ and $A_2^{i_2}$ are given by (11.1)-(11.3).

Since the fuzzy system designed from the above steps is a special case of the fuzzy system designed in Section 10.2, Theorem 10.1 is still valid for this fuzzy system. The following theorem gives a stronger result.

Theorem 11.1. Let $f(x)$ be the fuzzy system designed through the above three steps. If the $g(x)$ is twice continuously differentiable on U, then

$$||g - f||_\infty \leq \frac{1}{8}\left[||\frac{\partial^2 g}{\partial x_1^2}||_\infty h_1^2 + ||\frac{\partial^2 g}{\partial x_2^2}||_\infty h_2^2\right] \tag{11.4}$$

where $h_i = \max_{1 \leq j \leq N_i - 1} |e_i^{j+1} - e_i^j|$ $(i = 1, 2)$.

Proof: As in the proof of Theorem 10.1, we partition U into $U = \bigcup_{i_1=1}^{N_1-1} \bigcup_{i_2=1}^{N_2-1} U^{i_1 i_2}$, where $U^{i_1 i_2} = [e_1^{i_1}, e_1^{i_1+1}] \times [e_2^{i_2}, e_2^{i_2+1}]$. So, for any $x \in U$, there exists $U^{i_1 i_2}$ such that $x \in U^{i_1 i_2}$. Now, suppose $x \in U^{i_1 i_2}$, then by the consistency and completeness of fuzzy sets $A_i^1, A_i^2, ..., A_i^{N_i}$ $(i = 1, 2)$, the fuzzy system can be simplified to (same as (10.13))

$$f(x) = \sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} \left[\frac{\mu_{A_1^{j_1}}(x_1)\mu_{A_2^{j_2}}(x_2)}{\sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} \mu_{A_1^{j_1}}(x_1)\mu_{A_2^{j_2}}(x_2)} \right] g(e_1^{i_1}, e_2^{i_2}) \tag{11.5}$$

Since $\mu_{A_i^{j_i}}(x_i)$ are the special triangular membership functions given by (11.1)-(11.3), we have

$$\mu_{A_i^{i_1}}(x_i) + \mu_{A_i^{i_1+1}}(x_i) = 1 \tag{11.6}$$

for $i = 1, 2$. Hence,

$$\sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} \mu_{A_1^{j_1}}(x_1)\mu_{A_2^{j_2}}(x_2) = \sum_{j_1=i_1}^{i_1+1} \mu_{A_1^{j_1}}(x_1)[\sum_{j_2=i_2}^{i_2+1} \mu_{A_2^{j_2}}(x_2)] = 1 \tag{11.7}$$

and the fuzzy system (11.5) is simplified to

$$f(x) = \sum_{j_1=i_1}^{i_1+1} \sum_{j_2=i_2}^{i_2+1} [\mu_{A_1^{j_1}}(x_1)\mu_{A_2^{j_2}}(x_2)] g(e_1^{i_1}, e_2^{i_2}) \tag{11.8}$$

Let $C^2(U^{i_1 i_2})$ be the set of all twice continuously differentiable functions on $U^{i_1 i_2}$ and define linear operators L_1 and L_2 on $C^2(U^{i_1 i_2})$ as follows:

$$(L_1 g)(x) = \sum_{j_1=i_1}^{i_1+1} (\mu_{A_1^{j_1}}(x_1)) g(e_1^{j_1}, x_2) \tag{11.9}$$

$$(L_2 g)(x) = \sum_{j_2=i_2}^{i_2+1} (\mu_{A_2^{j_2}}(x_2)) g(x_1, e_2^{j_2}) \tag{11.10}$$

Since $\mu_{A_1^{j_1}}(x_1)$ and $\mu_{A_2^{j_2}}(x_2)$ are linear functions in $U^{i_1 i_2}$, they are twice continuously differentiable. Hence, $g \in C^2(U^{i_1 i_2})$ implies $L_1 g \in C^2(U^{i_1 i_2})$ and $L_2 g \in C^2(U^{i_1 i_2})$. From (11.9) and (11.6) we have

$$||L_1 g||_\infty \leq ||g||_\infty | \sum_{j_1=i_1}^{i_1+1} \mu_{A_1^{j_1}}(x_1)| = ||g||_\infty \tag{11.11}$$

Combining (11.9) and (11.10) and observing (11.8), we have

$$(L_1 L_2 g)(x) = \sum_{j_1=i_1}^{i_1+1} \mu_{A_1^{j_1}}(x_1)[\sum_{j_2=i_2}^{i_2+1} \mu_{A_2^{j_2}}(x_2) g(e_1^{j_1}, e_2^{j_2})] = f(x) \tag{11.12}$$

Therefore, from (11.11) and (11.12) we obtain

$$\begin{aligned} ||g - f||_\infty &= ||g - L_1 L_2 g||_\infty \\ &\le ||g - L_1 g||_\infty + ||L_1(g - L_2 g)||_\infty \\ &\le ||g - L_1 g||_\infty + ||g - L_2 g||_\infty \end{aligned} \tag{11.13}$$

Since $x \in U^{i_1 i_2} = [e_1^{i_1}, e_1^{i_1+1}] \times [e_2^{i_2}, e_2^{i_2+1}]$ and using the result in univariate linear interpolation (Powell [1981]), we obtain

$$\begin{aligned} ||g - L_1 g||_\infty &= || \sum_{j_1=i_1}^{i_1+1} \mu_{A_1^{j_1}}(x_1)[g(x_1, x_2) - g(e_1^{j_1}, x_2)]||_\infty \\ &\le ||g(x_1, x_2) - g(e_1^{j_1}, x_2)||_\infty \\ &\le \frac{1}{8}(e_1^{i_1+1} - e_1^{i_1})^2 || \frac{\partial^2 g}{\partial x_1^2} ||_\infty \end{aligned} \tag{11.14}$$

Similarly, we have

$$||g - L_2 g||_\infty \le \frac{1}{8}(e_2^{i_2+1} - e_2^{i_2})^2 || \frac{\partial^2 g}{\partial x_2^2} ||_\infty \tag{11.15}$$

Substituting (11.14) and (11.15) into (11.13) and noticing the definition of h_i, we obtain (11.4). □

From Theorem 11.1 we see that if we choose the particular triangular membership functions, a second-order accurate approximator can be obtained. We now design fuzzy systems to approximate the same functions $g(x)$ in Examples 10.1 and 10.2 using the new bound (11.4). We will see that we can achieve the same accuracy with fewer rules.

Example 11.1. Same as Example 10.1 except that we now use the bound (11.4). Since $||\frac{\partial^2 g}{\partial x_2}||_\infty = 1$, we have from (11.4) that if we choose $h = 1$, then we have $||g - f||_\infty \le \frac{1}{8} < \epsilon$. Therefore, we define 7 fuzzy sets A^j in the form of (11.1)-(11.3) with $e^j = -3 + (j-1)$ for $j = 1, 2, ..., 7$. The designed fuzzy system is

$$f(x) = \frac{\sum_{j=1}^{7} sin(e^j)\mu_{A^j}(x)}{\sum_{j=1}^{7} \mu_{A^j}(x)} \tag{11.16}$$

Comparing (11.16) with (10.23) we see that the number of rules is reduced from 31 to 7, but the accuracy remains the same. The $f(x)$ of (11.16) is plotted in Fig. 11.2 against $g(x) = sin(x)$. □

Example 11.2. Same as Example 10.2 except that we now use the bound (11.4). Since $\frac{\partial^2 g}{\partial x_i^2} = 0$ $(i = 1, 2)$, we know from (11.4) that $f(x) = g(x)$ for all $x \in U$. In fact, choosing $h_i = 2, e_i^1 = -1$ and $e_i^2 = 1$ for $i = 1, 2$ (that is, $N_1 = N_2 = 2$), we

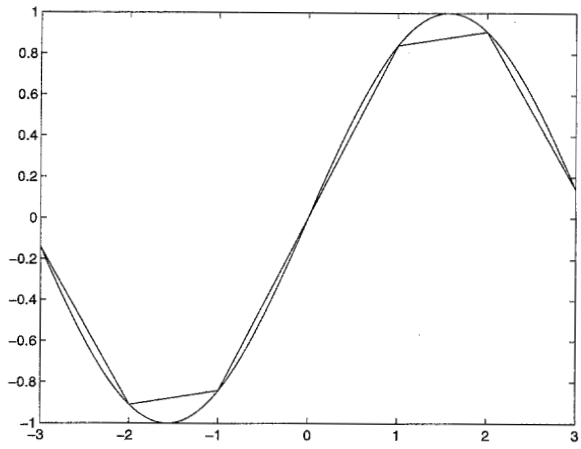


Figure 11.2. The designed fuzzy system $f(x)$ of (11.16) and the function $g(x) = sin(x)$ to be approximated.

obtain the designed fuzzy system as

$$f(x) = \frac{\sum_{i_1=1}^{2}\sum_{i_2=1}^{2} g(e_1^{i_1}, e_2^{i_2})(\mu_{A_1^{i_1}}(x_1)\mu_{A_2^{i_2}}(x_2))}{\sum_{i_1=1}^{2}\sum_{i_2=1}^{2}(\mu_{A_1^{i_1}}(x_1)\mu_{A_2^{i_2}}(x_2))} \tag{11.17}$$

where the membership functions are given by (11.1)-(11.3). For this particular case, we have for $i = 1, 2$ and $x \in U$ that

$$\mu_{A_i^1}(x_i) = \mu_{A_i^1}(x_i; -1, -1, 1) = \frac{1}{2}(1 - x_i) \tag{11.18}$$

$$\mu_{A_i^2}(x_i) = \mu_{A_i^2}(x_i; -1, 1, 1) = \frac{1}{2}(1 + x_i) \tag{11.19}$$

$g(e_1^1, e_2^1) = g(-1, -1) = 0.08, g(e_1^1, e_2^2) = g(-1, 1) = 0.76, g(e_1^2, e_2^1) = g(1, -1) = 0.4$, and $g(e_1^2, e_2^2) = g(1, 1) = 0.84$. Substituting these results into (11.17), we obtain

$$\begin{aligned} f(x) = [&\frac{0.08}{4}(1 - x_1)(1 - x_2) + \frac{0.76}{4}(1 - x_1)(1 + x_2) + \frac{0.4}{4}(1 + x_1)(1 - x_2) \\ &+ \frac{0.84}{4}(1 + x_1)(1 + x_2)]/[\frac{1}{4}(1 - x_1)(1 - x_2) + \frac{1}{4}(1 - x_1)(1 + x_2) \\ &+ \frac{1}{4}(1 + x_1)(1 - x_2) + \frac{1}{4}(1 + x_1)(1 + x_2)] \\ = &\ 0.52 + 0.1x_1 + 0.28x_2 - 0.06x_1x_2 \end{aligned} \tag{11.20}$$

which is exactly the same as $g(x)$. This confirms our conclusion that $f(x)$ exactly reproduces $g(x)$. In Example 10.2 we used 121 rules to construct the fuzzy system, whereas in this example we only use 4 rules to achieve zero approximation error. □

To generalize Example 11.2 , we observe from (11.4) that for any function with $||\frac{\partial^2 g}{\partial x_i^2}||_\infty = 0$, our fuzzy system $f(x)$ designed through the three steps reproduces the $g(x)$, that is, $f(x) = g(x)$ for all $x \in U$. This gives the following corollary of Theorem 11.1.

Corollary 11.1. Let $f(x)$ be the fuzzy system designed through the three steps in this section. If the function $g(x)$ is of the following form:

$$g(x) = \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} a_{k_1 k_2} x_1^{k_1} x_2^{k_2} \tag{11.21}$$

where $a_{k_1 k_2}$ are constants, then $f(x) = g(x)$ for all $x \in U$.

Proof: Since $\frac{\partial^2 g}{\partial x_1^2} = \frac{\partial^2 g}{\partial x_2^2} = 0$ for this class of $g(x)$, the conclusion follows immediately from (11.4). □

11.2 Approximation Accuracy of Fuzzy Systems with Maximum Defuzzifier

In Chapter 9 we learned that fuzzy systems with maximum defuzzifier are quite different from those with center average defuzzifier. In this section, we study the approximation properties of fuzzy systems with maximum defuzzifier.

Similar to the approach in Sections 11.1 and 10.2, we first design a particular fuzzy system and then study its properties.

Design of Fuzzy System with Maximum Defuzzifier:

- **Step 1**. Same as Step 1 of the design procedure in Section 11.1.
- **Step 2**. Same as Step 2 of the design procedure in Section 10.2.
- **Step 3**. Construct a fuzzy system $f(x)$ from the $N_1 \times N_2$ rules in the form of (10.8) using product inference engine (7.23), singleton fuzzifier (8.1), and maximum defuzzifier (8.23). According to Lemma 9.4, this fuzzy system is

$$f(x) = \bar{y}^{i_1^* i_2^*} = g(e_1^{i_1^*}, e_2^{i_2^*}) \tag{11.22}$$

where $i_1^* i_2^*$ is such that

$$\mu_{A_1^{i_1^*}}(x_1)\mu_{A_2^{i_2^*}}(x_2) \geq \mu_{A_1^{i_1}}(x_1)\mu_{A_2^{i_2}}(x_2) \tag{11.23}$$

for all $i_1 = 1, 2, ..., N_1$ and $i_2 = 1, 2, ..., N_2$.

The following theorem shows that the fuzzy system designed above is a first-order accurate approximator of $g(x)$.

Theorem 11.2. Let $f(x)$ be the fuzzy system (11.22) designed from the three steps above. If $g(x)$ is continuously differentiable on $U = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$, then

$$||g - f||_\infty \leq ||\frac{\partial g}{\partial x_1}||_\infty h_1 + ||\frac{\partial g}{\partial x_2}||_\infty h_2 \tag{11.24}$$

where $h_i = \max_{1 \leq j \leq N_i - 1} |e_i^{j+1} - e_i^j|$, $i = 1, 2$.

Proof: As in the proof of Theorem 11.1, we partition U into $U = \bigcup_{i_1=1}^{N_1-1} \bigcup_{i_2=1}^{N_2-1} U^{i_1 i_2}$, where $U^{i_1 i_2} = [e_1^{i_1}, e_1^{i_1+1}] \times [e_2^{i_2}, e_2^{i_2+1}]$. We now further partition $U^{i_1 i_2}$ into $U^{i_1 i_2} = U_{00}^{i_1 i_2} \cup U_{01}^{i_1 i_2} \cup U_{10}^{i_1 i_2} \cup U_{11}^{i_1 i_2}$, where $U_{00}^{i_1 i_2} = [e_1^{i_1}, \frac{1}{2}(e_1^{i_1} + e_1^{i_1+1})] \times [e_2^{i_2}, \frac{1}{2}(e_2^{i_2} + e_2^{i_2+1})]$, $U_{01}^{i_1 i_2} = [e_1^{i_1}, \frac{1}{2}(e_1^{i_1} + e_1^{i_1+1})] \times [\frac{1}{2}(e_2^{i_2} + e_2^{i_2}), e_2^{i_2+1}]$, $U_{10}^{i_1 i_2} = [\frac{1}{2}(e_1^{i_1} + e_1^{i_1+1}), e_1^{i_1+1}] \times [e_2^{i_2}, \frac{1}{2}(e_2^{i_2} + e_2^{i_2+1})]$, and $U_{11}^{i_1 i_2} = [\frac{1}{2}(e_1^{i_1} + e_1^{i_1+1}), e_1^{i_1+1}] \times [\frac{1}{2}(e_2^{i_2} + e_2^{i_2+1}), e_2^{i_2+1}]$; see Fig. 11.3 for an illustration. So for any $x \in U$, there exist $U_{pq}^{i_1 i_2}$ ($p, q = 0$ *or* 1) such that $x \in U_{pq}^{i_1 i_2}$. If x is in the interior of $U_{pq}^{i_1 i_2}$, then with the help of Fig. 11.3 we see that $\mu_{A_1^{i_1+p}}(x_1) > 0.5, \mu_{A_2^{i_2+q}}(x_2) > 0.5$, and all other membership values are less than 0.5. Hence, from (11.22) and (11.23) we obtain

$$f(x) = g(e_1^{i_1+p}, e_2^{i_2+q}) \tag{11.25}$$

Using the Mean Value Theorem and the fact that $x \in U^{i_1 i_2}$, we have

$$\begin{aligned} |g(x) - f(x)| &= |g(x_1, x_2) - g(e_1^{i_1+p}, e_2^{i_2+q})| \\ &\leq ||\frac{\partial g}{\partial x_1}||_\infty |x_1 - e_1^{i_1+p}| + ||\frac{\partial g}{\partial x_2}||_\infty |x_2 - e_2^{i_2+q}| \\ &\leq ||\frac{\partial g}{\partial x_1}||_\infty |e_1^{i_1+1} - e_1^{i_1}| + ||\frac{\partial g}{\partial x_2}||_\infty |e_2^{i_2+1} - e_2^{i_2}| \end{aligned} \tag{11.26}$$

If x is on the boundary of $U_{pq}^{i_1 i_2}$, then with the help of Fig. 11.3 we see that $f(x)$ may take any value from a set of at most the four elements $\{g(e_1^{i_1}, e_2^{i_2}), g(e_1^{i_1}, e_2^{i_2+1}), g(e_1^{i_1+1}, e_2^{i_2}), g(e_1^{i_1+1}, e_2^{i_2+1})\}$; so (11.26) is still true in this case. Finally, (11.24) follows from (11.26). □

From Theorem 11.2 we immediately see that the fuzzy systems with product inference engine, singleton fuzzifier, and maximum defuzzifier are universal approximators. In fact, by choosing the h_1 and h_2 sufficiently small, we can make $||g - f||_\infty < \epsilon$ for arbitrary $\epsilon > 0$ according to (11.24).

We now approximate the functions $g(x)$ in Examples 11.1 and 11.2 using the fuzzy system (11.22).

Example 11.3. Same as Example 10.1 except that we use the fuzzy system (11.22). Since $||\frac{\partial g}{\partial x}||_\infty = 1$, we choose $h = 0.2$ and define 31 fuzzy sets A^j in the form of (10.20)-(10.22) (Fig. 10.4). Let $e^j = -3 + 0.2(j-1)$, then the fuzzy system

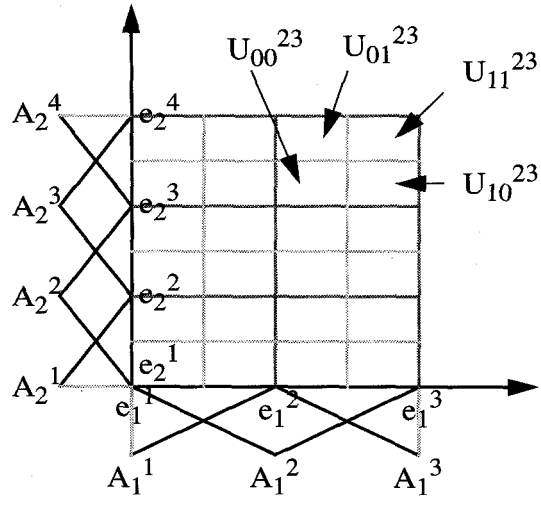


Figure 11.3. An example of partition of $U^{i_1 i_2}$ into subsets.

(11.22) becomes

$$f(x) = \begin{cases} sin(-3), \ x \in [-3, -2.8] \\ sin(e^j), \ x \in (e^j - 0.1, e^j + 0.1], \ j = 2, 3, ..., 30 \\ sin(3), \ x \in (2.8, 3] \end{cases} \tag{11.27}$$

which is plotted in Fig. 11.4. □

Example 11.4. Same as Example 10.2 except that we use the fuzzy system (11.22). As in Example 10.2, we choose $h_1 = h_2 = 0.2$ and define 11 fuzzy sets A^j on $[-1, 1]$ given by (10.24)-(10.26). We construct the fuzzy system using the 121 rules in the form of (10.27). For this example, $e^j = -1 + 0.2(j-1)$ $(j = 1, 2, ..., 11)$ and $U^{i_1 i_2} = [e^{i_1}, e^{i_1+1}] \times [e^{i_1}, e^{i_1+1}]$ $(i_1, i_2 = 1, 2, ..., 10)$. As shown in Fig. 11.3, we further decompose $U^{i_1 i_2}$ into $U^{i_1 i_2} = \bigcup_{p=0}^{1} \bigcup_{q=0}^{1} U_{pq}^{i_1 i_2}$, where $U_{00}^{i_1 i_2} = [e_1^{i_1}, \frac{1}{2}(e_1^{i_1} + e_1^{i_1+1})] \times [e_2^{i_2}, \frac{1}{2}(e_2^{i_2} + e_2^{i_2+1})]$, $U_{01}^{i_1 i_2} = [e_1^{i_1}, \frac{1}{2}(e_1^{i_1} + e_1^{i_1+1})] \times [\frac{1}{2}(e_2^{i_2} + e_2^{i_2}), e_2^{i_2+1}]$, $U_{10}^{i_1 i_2} = [\frac{1}{2}(e_1^{i_1} + e_1^{i_1+1}), e_1^{i_1+1}] \times [e_2^{i_2}, \frac{1}{2}(e_2^{i_2} + e_2^{i_2+1})]$, and $U_{11}^{i_1 i_2} = [\frac{1}{2}(e_1^{i_1} + e_1^{i_1+1}), e_1^{i_1+1}] \times [\frac{1}{2}(e_2^{i_2} + e_2^{i_2+1}), e_2^{i_2+1}]$. Then the fuzzy system (11.22) becomes

$$f(x) = g(e^{i_1+p}, e^{i_2+q}), \ x \in U_{pq}^{i_1 i_2} \tag{11.28}$$

which is computed through the following two steps: (i) for given $x \in U$, determine i_1, i_2, p, q such that $x \in U_{pq}^{i_1 i_2}$, and (ii) the $f(x)$ equals $g(e^{i_1+p}, e^{i_2+q})$. □

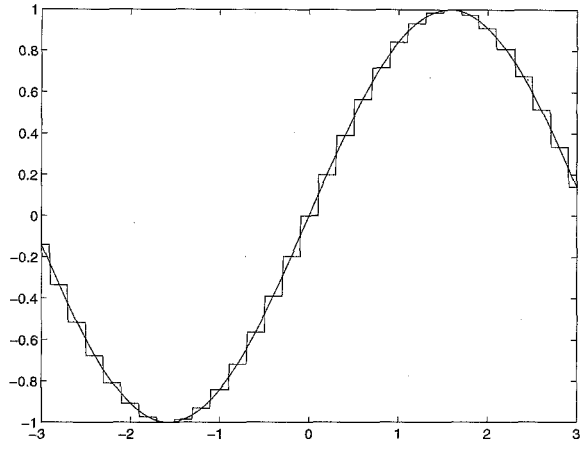


Figure 11.4. The designed fuzzy system (11.22) and the function $g(x) = sin(x)$ to be approximated in Example 11.3.

In Section 11.1 we showed that the fuzzy systems with center average defuzzifier are second-order accurate approximators. Theorem 11.2 shows that the fuzzy systems with maximum defuzzifier are first-order accurate approximators. So it is natural to ask whether they are second-order accurate approximators also? The following example shows, unfortunately, that they cannot be second-order accurate approximators.

Example 11.5. Let $g(x) = x$ on $U = [0, 1]$ and $\mu_{A^i}(x) = \mu_{A^i}(x; e^{i-1}, e^i, e^{i+1})$, where $e^0 = 0, e^{N+1} = 1, e^i = \frac{i-1}{N-1}$, $i = 1, 2, ..., N$, and N can be any positive integer. So, in this case we have N rules and $h = \frac{1}{N-1}$. Let $U^i = [e^i, e^{i+1}]$ $(i = 1, 2, ..., N-1)$ and $f(x)$ be the fuzzy system (11.22). If $x \in U^i$, then

$$\max_{x \in U^i} |g(x) - f(x)| = \max_{x \in [e^i, e^{i+1}]} |x - g(e^i) \text{ or } g(e^{i+1})| = \frac{1}{2}(e^{i+1} - e^i) = \frac{1}{2}h \quad (11.29)$$

Since $h(= \frac{1}{N-1}) \geq h^2$ for any positive integer N, the fuzzy system (11.22) cannot approximate the simple function $g(x) = x$ to second-order accuracy. Because of this counter-example, we conclude that fuzzy systems with maximum defuzzifier cannot be second-order accurate approximators. Therefore, fuzzy systems with center average defuzzifier are better approximators than fuzzy systems with maximum defuzzifier.

11.3 Summary and Further Readings

In this chapter we have demonstrated the following:

- Using the second-order bound (11.4) to design fuzzy systems with required accuracy.
- Designing fuzzy systems with maximum defuzzifier to approximate functions with required accuracy.
- The ideas of proving the second-order bound for fuzzy systems with center average defuzzifier (Theorem 11.1) and the first-order bound for fuzzy systems with maximum defuzzifier (Theorem 11.2).

Again, very few references are available on the topic of this chapter. The most relevant papers are Ying [1994] and Zeng and Singh [1995].

11.4 Exercises

Exercise 11.1. Use the first-order bound (10.11) and the second-order bound (11.4) to design two fuzzy systems with center average defuzzifier to uniformly approximate the function $g(x_1, x_2) = \frac{1}{3+x_1+x_2}$ on $U = [-1, 1] \times [-1, 1]$ to the accuracy of $\epsilon = 0.1$. Plot the designed fuzzy systems and compare them.

Exercise 11.2. Repeat Exercise 11.1 with $g(x_1, x_2) = \frac{1}{1+x_1^2+x_2^2}$.

Exercise 11.3. Design a fuzzy system with maximum defuzzifier to uniformly approximate the $g(x_1, x_2)$ in Exercise 11.1 on the same U to the accuracy of $\epsilon = 0.1$. Plot the designed fuzzy system.

Exercise 11.4. Repeat Exercise 11.3 with the $g(x_1, x_2)$ in Exercise 11.2.

Exercise 11.5. Generalize the design procedure in Section 11.1 to n-input fuzzy systems and prove that the designed fuzzy system $f(x)$ satisfies

$$||g - f||_\infty \leq \frac{1}{8}(\sum_{i=1}^{n} ||\frac{\partial^2 g}{\partial x_i^2}||_\infty h_i^2) \tag{11.30}$$

Exercise 11.6. Verify graphically that (11.29) is true.

Part III

Design of Fuzzy Systems from Input-Output Data

Fuzzy systems are used to formulate human knowledge. Therefore, an important question is: What forms does human knowledge usually take? Roughly speaking, human knowledge about a particular engineering problem may be classified into two categories: conscious knowledge and subconscious knowledge. By *conscious knowledge* we mean the knowledge that can be *explicitly* expressed in words, and by *subconscious knowledge* we refer to the situations where the human experts know what to do but cannot express exactly in words how to do it. For example, the experienced truck drivers know how to drive the truck in very difficult situations (they have subconscious knowledge), but it is difficult for them to express their actions in precise words. Even if they can express the actions in words, the description is usually incomplete and insufficient for accomplishing the task.

For conscious knowledge, we can simply ask the human experts to express it in terms of fuzzy IF-THEN rules and put them into fuzzy systems. For subconscious knowledge, what we can do is to ask the human experts to *demonstrate*, that is, to show what they do in some typical situations. When the expert is demonstrating, we view him/her as a black box and measure the inputs and the outputs; that is, we can collect a set of input-output data pairs. In this way, the subconscious knowledge is transformed into a set of input-output pairs; see Fig. 12.1. Therefore, a problem of fundamental importance is to construct fuzzy systems from input-output pairs.

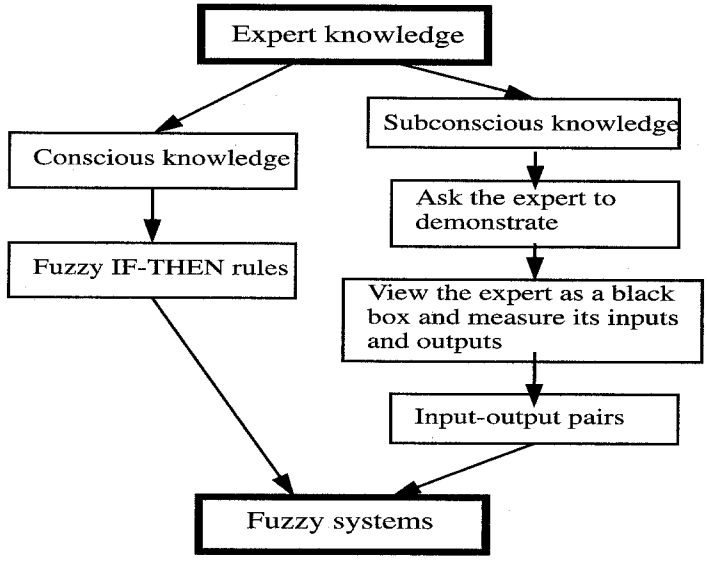


Figure 12.1. Converting expert knowledge into fuzzy systems.

In this part (Chapters 12-15), we will develop a number of methods for constructing fuzzy systems from input-output pairs. Because in many practical situations we are only provided with a limited number of input-output pairs and cannot obtain the outputs for arbitrary inputs, the design methods in Chapters 10 and 11 are not applicable (recall that the methods in Chapters 10 and 11 require that we can determine the output $g(x)$ for arbitrary input $x \in U$). That is, we are now considering the third case discussed in the beginning of Chapter 10. Our task is to design a fuzzy system that characterizes the input-output behavior represented by the input-output pairs.

In Chapter 12, we will develop a simple heuristic method for designing fuzzy systems from input-output pairs and apply the method to the truck backer-upper control and time series prediction problems. In Chapter 13, we will design the fuzzy system by first specifying its structure and then adjusting its parameters using a gradient descent training algorithm; we will use the designed fuzzy systems to identify nonlinear dynamic systems. In Chapter 14, the recursive least squares algorithm will be used to design the parameters of the fuzzy system and the designed fuzzy system will be used as an equalizer for nonlinear communication channels. Finally, Chapter 15 will show how to use clustering ideas to design fuzzy systems.

Chapter 12

Design of Fuzzy Systems Using A Table Look-Up Scheme

12.1 A Table Look-Up Scheme for Designing Fuzzy Systems from Input-Output Pairs

Suppose that we are given the following input-output pairs:

$$(x_0^p; y_0^p),\ p = 1, 2, ..., N \tag{12.1}$$

where $x_0^p \in U = [\alpha_1, \beta_1] \times \cdots \times [\alpha_n, \beta_n] \subset R^n$ and $y_0^p \in V = [\alpha_y, \beta_y] \subset R$. Our objective is to design a fuzzy system $f(x)$ based on these N input-output pairs. We now propose the following five step table look-up scheme to design the fuzzy system:

Step 1. Define fuzzy sets to cover the input and output spaces.

Specifically, for each $[\alpha_i, \beta_i]$, $i = 1, 2, ..., n$, define N_i fuzzy sets A_i^j $(j = 1, 2, ..., N_i)$, which are required to be complete in $[\alpha_i, \beta_i]$; that is, for any $x_i \in [\alpha_i, \beta_i]$, there exists A_i^j such that $\mu_{A_i^j}(x_i) \neq 0$. For example, we may choose $\mu_{A_i^j}(x_i)$ to be the pseudo-trapezoid membership functions: $\mu_{A_i^j}(x_i) = \mu_{A_i^j}(x_i; a_i^j, b_i^j, c_i^j, d_i^j)$, where $a_i^1 = b_i^1 = \alpha_i, c_i^j = a_i^{j+1} < b_i^{j+1} = d_i^j$ $(j = 1, 2, ..., N_i - 1)$, and $c_i^{N_i} = d_i^{N_i} = \beta_i$. Similarly, define N_y fuzzy sets B^j, $j = 1, 2, ..., N_y$, which are complete in $[\alpha_y, \beta_y]$. We also may choose $\mu_{B^j}(y)$ to be the pseudo-trapezoid membership functions: $\mu_{B^j}(y) = \mu_{B^j}(y; a^j, b^j, c^j, d^j)$, where $a^1 = b^1 = \alpha_y, c^j = a^{j+1} < b^{j+1} = d^j$ $(j = 1, 2, ..., N_y - 1)$, and $c^{N_y} = d^{N_y} = \beta_y$. Fig.12.2 shows an example for the $n = 2$ case, where $N_1 = 5, N_2 = 7, N_y = 5$, and the membership functions are all triangular.

Step 2. Generate one rule from one input-output pair.

First, for each input-output pair $(x_{01}^p, ..., x_{0n}^p; y_0^p)$, determine the membership values of x_{0i}^p $(i = 1, 2, ..., n)$ in fuzzy sets A_i^j $(j = 1, 2, ..., N_i)$ and the membership values of y_0^p in fuzzy sets B^l $(l = 1, 2, ..., N_y)$. That is, compute the following:

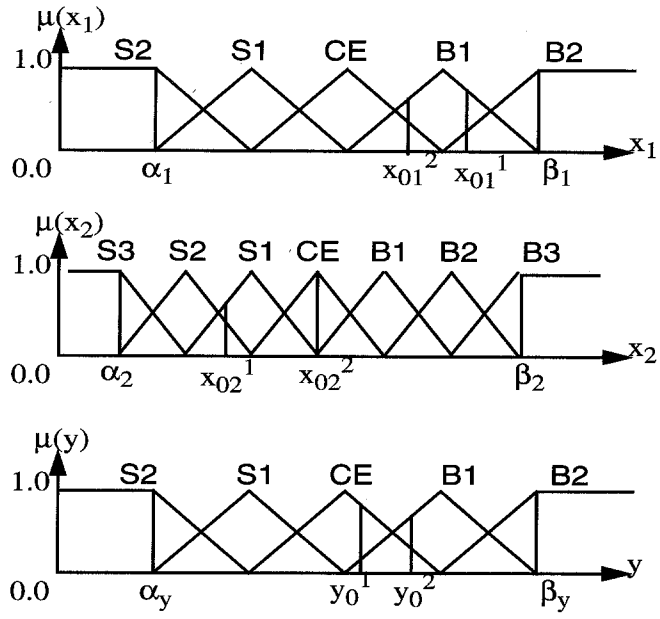


Figure 12.2. An example of membership functions and input-output pairs for the two-input case.

$\mu_{A_i^j}(x_{0i}^p)$ for $j = 1, 2, ..., N_i, i = 1, 2, ..., n$, and $\mu_{B^l}(y_0^p)$ for $l = 1, 2, ..., N_y$. For the example in Fig. 12.2, we have approximately that: x_{01}^1 has membership value 0.8 in B1, 0.2 in B2, and zero in other fuzzy sets; x_{02}^1 has membership value 0.6 in S1, 0.4 in S2, and zero in other fuzzy sets; and, y_0^1 has membership value 0.8 in CE, 0.2 in B1, and zero in other fuzzy sets.

Then, for each input variable x_i $(i = 1, 2, ..., n)$, determine the fuzzy set in which x_{0i}^p has the largest membership value, that is, determine A_i^{j*} such that $\mu_{A_i^{j*}}(x_{0i}^p) \geq \mu_{A_i^j}(x_{0i}^p)$ for $j = 1, 2, ..., N_i$. Similarly, determine B^{l*} such that $\mu_{B^{l*}}(y_0^p) \geq \mu_{B^l}(y_0^p)$ for $l = 1, 2, ..., N_y$. For the example in Fig. 12.2, the input-output pair $(x_{01}^1, x_{02}^1; y_0^1)$ gives $A_1^{j*} = B1, A_2^{j*} = S1$ and $B^{l*} = CE$, and the pair $(x_{01}^2, x_{02}^2; y_0^2)$ gives $A_1^{j*} = B1, A_2^{j*} = CE$ and $B^{l*} = B1$.

Finally, obtain a fuzzy IF-THEN rule as

$$IF\ x_1\ is\ A_1^{j*}\ and\ \cdots\ and\ x_n\ is\ A_n^{j*},\ THEN\ y\ is\ B^{l*} \tag{12.2}$$

For the example in Fig. 12.2, the pair $(x_{01}^1, x_{02}^1; y_0^1)$ gives the rule: $IF\ x_1\ is\ B1$

and x_2 *is* $S1$, *THEN* y *is* CE; and the pair $(x_{01}^2, x_{02}^2; y_0^2)$ gives the rule: *IF* x_1 *is* $B1$ *and* x_2 *is* CE, *THEN* y *is* $B1$.

Step 3. Assign a degree to each rule generated in Step 2.

Since the number of input-output pairs is usually large and with each pair generating one rule, it is highly likely that there are conflicting rules, that is, rules with the same IF parts but different THEN parts. To resolve this conflict, we assign a degree to each generated rule in Step 2 and keep only one rule from a conflicting group that has the maximum degree. In this way not only is the conflict problem resolved, but also the number of rules is greatly reduced.

The degree of a rule is defined as follows: suppose that the rule (12.2) is generated from the input-output pair $(x_0^p; y_0^p)$, then its degree is defined as

$$D(rule) = \prod_{i=1}^{n} \mu_{A_i^{j*}}(x_{0i}^p)\mu_{B^{l*}}(y_0^p) \tag{12.3}$$

For the example in Fig. 12.2, the rule generated by $(x_{01}^1, x_{02}^1; y_0^1)$ has degree

$$\begin{aligned} D(rule1) &= \mu_{B1}(x_{01}^1)\mu_{S1}(x_{02}^1)\mu_{CE}(y_0^1) \\ &= 0.8 * 0.6 * 0.8 = 0.384 \end{aligned} \tag{12.4}$$

and the rule generated by $(x_{01}^2, x_{02}^2; y_0^2)$ has degree

$$\begin{aligned} D(rule2) &= \mu_{B1}(x_{01}^2)\mu_{CE}(x_{02}^2)\mu_{B1}(y_0^2) \\ &= 0.6 * 1 * 0.7 = 0.42 \end{aligned} \tag{12.5}$$

If the input-output pairs have different reliability and we can determine a number to assess it, we may incorporate this information into the degrees of the rules. Specifically, suppose the input-output pair $(x_0^p; y_0^p)$ has reliable degree μ^p ($\in [0, 1]$), then the degree of the rule generated by $(x_0^p; y_0^p)$ is redefined as

$$D(rule) = \prod_{i=1}^{n} \mu_{A_i^{j*}}(x_{0i}^p)\mu_{B^{l*}}(y_0^p)\mu^p \tag{12.6}$$

In practice, we may ask an expert to check the data (if the number of input-output pairs is small) and estimate the degree μ^p. Or, if we know the characteristics of the noise in the data pair, we may choose μ^p to reflect the strength of the noise. If we cannot tell the difference among the input-output pairs, we simply choose all $\mu^p = 1$ so that (12.6) reduces to (12.3).

Step 4. Create the fuzzy rule base.

The fuzzy rule base consists of the following three sets of rules:

- The rules generated in Step 2 that do not conflict with any other rules.

- The rule from a conflicting group that has the maximum degree, where a group of conflicting rules consists of rules with the same IF parts.

- Linguistic rules from human experts (due to conscious knowledge).

Since the first two sets of rules are obtained from subconscious knowledge, the final fuzzy rule base combines both conscious and subconscious knowledge.

Intuitively, we can illustrate a fuzzy rule base as a look-up table in the two-input case. For example, Fig. 12.3 demonstrates a table-lookup representation of the fuzzy rule base corresponding to the fuzzy sets in Fig. 12.2. Each box represents a combination of fuzzy sets in $[\alpha_1, \beta_1]$ and fuzzy sets in $[\alpha_2, \beta_2]$ and thus a possible rule. A conflicting group consists of rules in the same box. This method can be viewed as filling up the boxes with appropriate rules; this is why we call this method a table look-up scheme.

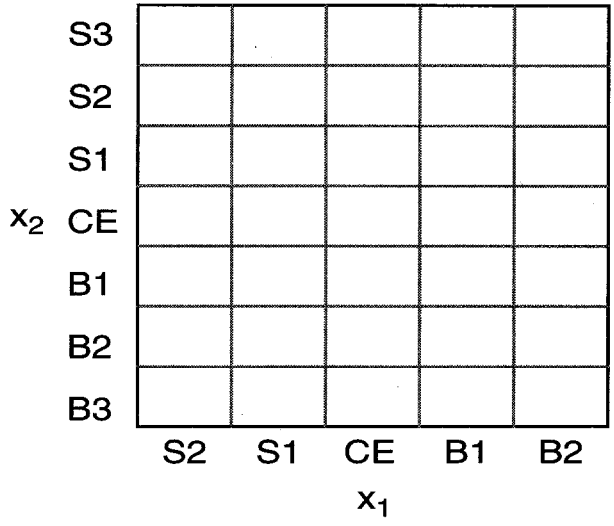


Figure 12.3. Table look-up illustration of the fuzzy rule base.

Step 5. Construct the fuzzy system based on the fuzzy rule base.

We can use any scheme in Chapter 9 to construct the fuzzy system based on the fuzzy rule base created in Step 4. For example, we may choose fuzzy systems with product inference engine, singleton fuzzifier, and center average defuzzifier (Lemma 9.1).

We now make a few remarks on this five step procedure of designing the fuzzy system from the input-output pairs.

- A fundamental difference between this method and the methods in Chapters 10 and 11 is that the methods in Chapters 10 and 11 require that we are able to determine the exact output $g(x)$ for any input $x \in U$, whereas in this method we cannot freely choose the input points in the given input-output pairs. Also, in order to design a fuzzy system with the required accuracy, the methods in Chapters 10 and 11 need to know the bounds of the first or second order derivatives of the function to be approximated, whereas this method does not require this information.

- If the input-output pairs happen to be the $(e_1^{i_1}, e_2^{i_2}; \bar{y}^{i_1 i_2})$ in (10.9), then it is easy to verify that the fuzzy system designed through these five steps is the same fuzzy system as designed in Section 10.2. Therefore, this method can be viewed as a generalization of the design method in Section 10.2 to the case where the input-output pairs cannot be arbitrarily chosen.

- The number of rules in the final fuzzy rule base is bounded by two numbers: N, the number of input-output pairs, and $\prod_{i=1}^{n} N_i$, the number of all possible combinations of the fuzzy sets defined for the input variables. If the dimension of the input space n is large, $\prod_{i=1}^{n} N_i$ will be a huge number and may be larger than N. Also, some input-output pairs may correspond to the same box in Fig. 12.3 and thus can contribute only one rule to the fuzzy rule base. Therefore, the number of rules in the fuzzy rule base may be much less than both $\prod_{i=1}^{n} N_i$ and N. Consequently, the fuzzy rule base generated by this method may not be complete. To make the fuzzy rule base complete, we may fill in the empty boxes in the fuzzy rule base by interpolating the existing rules; we leave the details to the reader to think about.

Next, we apply this method to a control problem and a time series prediction problem.

12.2 Application to Truck Backer-Upper Control

Backing up a truck to a loading dock is a nonlinear control problem. Using conventional control approach, we can first develop a mathematical model of the system and then design a controller based on nonlinear control theory (Walsh, Tilbury, Sastry, Murray, and Laumond [1994]). Another approach is to design a controller to emulate the human driver. We adapt the second approach. Assume that an experienced human driver is available and we can measure the truck's states and the corresponding control action of the human driver while he/she is backing the truck to the dock; that is, we can collect a set of input-output (state-control) pairs. We will design a fuzzy system based on these input-output pairs using the table

look-up scheme in the last section, and replace the human driver by the designed fuzzy system.

The simulated truck and loading zone are shown in Fig. 12.4. The truck position is determined by three state variables ϕ, x and y, where ϕ is the angle of the truck with respect to the horizontal line as shown in Fig. 12.4. Control to the truck is the steeling angle θ. Only backing up is permitted. The truck moves backward by a fixed unit distance every stage. For simplicity, we assume enough clearance between the truck and the loading dock such that y does not have to be considered as a state variable. The task is to design a controller whose inputs are (x, ϕ) and whose output is θ, such that the final state will be $(x_f, \phi_f) = (10, 90^o)$. We assume that $x \in [0, 20], \phi \in [-90^o, 270^o]$ and $\theta \in [-40^o, 40^o]$; that is, $U = [0, 20] \times [-90^o, 270^o]$ and $V = [-40^o, 40^o]$.

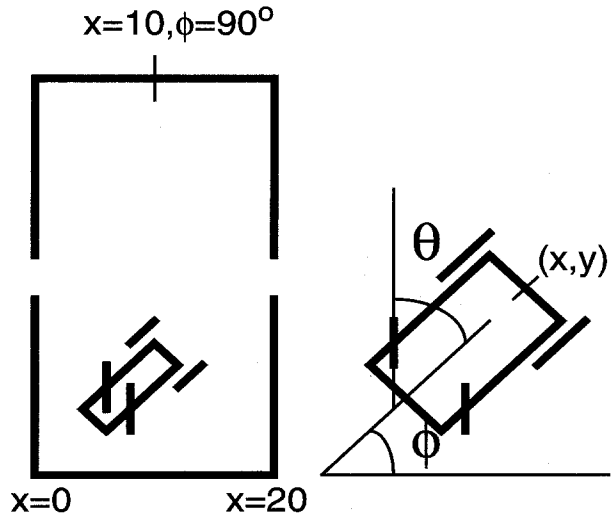


Figure 12.4. The simulated truck and loading zone.

First, we generate the input-output pairs $(x^p, \phi^p; \theta^p)$. We do this by trial and error: at every stage (given x and ϕ) starting from an initial state, we determine a control θ based on common sense (that is, our own experience of how to control the steering angle in the situation); after some trials, we choose the input-output pairs corresponding to the smoothest successful trajectory. The following 14 initial states were used to generate the desired input-output pairs: $(x_0, \phi_0^o) = (1, 0), (1, 90), (1, 270); (7, 0), (7, 90), (7, 180), (7, 270); (13, 0), (13, 90), (13, 180), (13, 270);$

$(19, 90), (19, 180), (19, 270)$. Table 12.1 shows the input-output pairs starting from the initial state $(x_0, \phi_0) = (1, 0^o)$. The input-output pairs starting from the other 13 initial states can be obtained in a similar manner. Totally, we have about 250 input-output pairs. We now design a fuzzy system based on these input-output pairs using the table look-up scheme developed in the last section.

Table 12.1. Ideal trajectory (x_t, ϕ_t^o) and the corresponding control θ_t^o starting from $(x_0, \phi_0) = (1, 0^o)$.

t	x_t	ϕ_t^o	θ_t^o
0	1.00	0.00	-19.00
1	1.95	9.37	-17.95
2	2.88	18.23	-16.90
3	3.79	26.57	-15.85
4	4.65	34.44	-14.80
5	5.45	41.78	-13.75
6	6.18	48.60	-12.70
7	7.48	54.91	-11.65
8	7.99	60.71	-10.60
9	8.72	65.99	-9.55
10	9.01	70.75	-8.50
11	9.28	74.98	-7.45
12	9.46	78.70	-6.40
13	9.59	81.90	-5.34
14	9.72	84.57	-4.30
15	9.81	86.72	-3.25
16	9.88	88.34	-2.20
17	9.91	89.44	0.00

In Step 1, we define 7 fuzzy sets in $[-90^o, 270^o]$, 5 fuzzy sets in $[0, 20]$ and 7 fuzzy sets in $[-40^o, 40^o]$, where the membership functions are shown in Fig.12.5. In Steps 2 and 3, we generate one rule from one input-output pair and compute the corresponding degrees of the rules. Table 12.2 shows the rules and their degrees generated by the corresponding input-output pairs in Table 12.1. The final fuzzy rule base generated in Step 4 is shown in Fig.12.6 (we see that some boxes are empty, so the input-output pairs do not cover all the state space; however, we will see that the rules in Fig. 12.6 are sufficient for controlling the truck to the desired position starting from a wide range of initial positions). Finally, in Step 5 we use the fuzzy system with product inference engine, singleton fuzzifier, and center average defuzzifier; that is, the designed fuzzy system is in the form of (9.1) with the rules in Fig. 12.6.

We now use the fuzzy system designed above as a controller for the truck. To simulate the control system, we need a mathematical model of the truck. We use

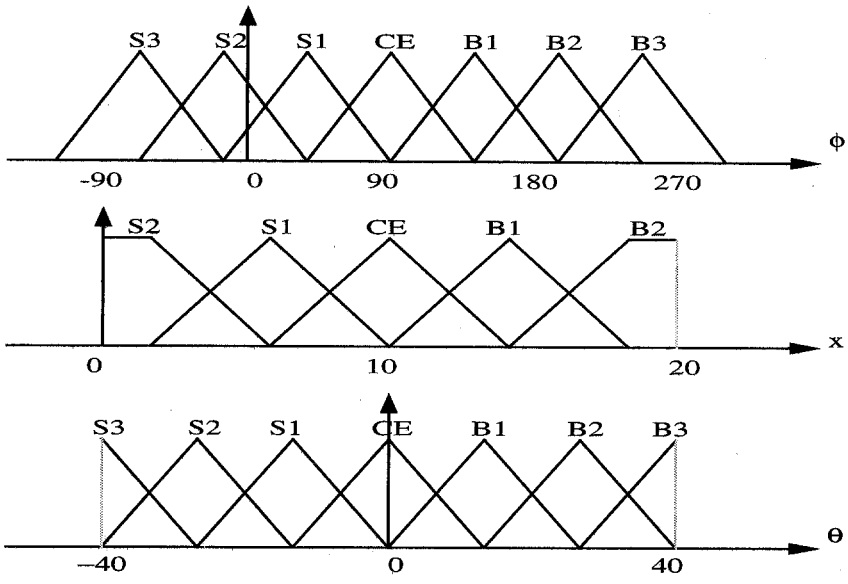


Figure 12.5. Membership functions for the truck backer-upper control problem.

ϕ \ x	S2	S1	CE	B1	B2
S3	S2	S3			
S2	S2	S3	S3	S3	
S1	B1	S1	S2	S3	S2
CE	B2	B2	CE	S2	S2
B1	B2	B3	B2	B1	S1
B2		B3	B3	B3	B2
B3				B3	B2

Figure 12.6. The final fuzzy rule base for the truck backer-upper control problem.

Table 12.2. Fuzzy IF-THEN rules generated from the input-output pairs in Table 12.1 and their degrees.

x *is*	ϕ *is*	θ *is*	degree
S2	S2	S2	1.00
S2	S2	S2	0.92
S2	S2	S2	0.35
S2	S2	S2	0.12
S2	S2	S2	0.07
S1	S2	S1	0.08
S1	S1	S1	0.18
S1	S1	S1	0.53
S1	S1	S1	0.56
S1	S1	S1	0.60
CE	S1	S1	0.35
CE	S1	S1	0.21
CE	S1	CE	0.16
CE	CE	CE	0.32
CE	CE	CE	0.45
CE	CE	CE	0.54
CE	CE	CE	0.88
CE	CE	CE	0.92

the following approximate model (Wang and Mendel [1992b]):

$$x(t+1) = x(t) + cos[\phi(t) + \theta(t)] + sin[\theta(t)]sin[\phi(t)] \tag{12.7}$$

$$y(t+1) = y(t) + sin[\phi(t) + \theta(t)] - sin[\theta(t)]cos[\phi(t)] \tag{12.8}$$

$$\phi(t+1) = \phi(t) - sin^{-1}[\frac{2sin(\theta(t))}{b}] \tag{12.9}$$

where b is the length of the truck and we assume $b = 4$ in our simulations. Fig. 12.7 shows the truck trajectory using the designed fuzzy system as the controller for two initial conditions: $(x_0, \phi_0) = (3, -30^o)$ and $(13, 30^o)$. We see that the fuzzy controller can successfully control the truck to the desired position.

12.3 Application to Time Series Prediction

Time series prediction is an important practical problem. Applications of time series prediction can be found in the areas of economic and business planning, inventory and production control, weather forecasting, signal processing, control, and many other fields. In this section, we use the fuzzy system designed by the table look-up scheme to predict the Mackey-Glass chaotic time series that is generated by the

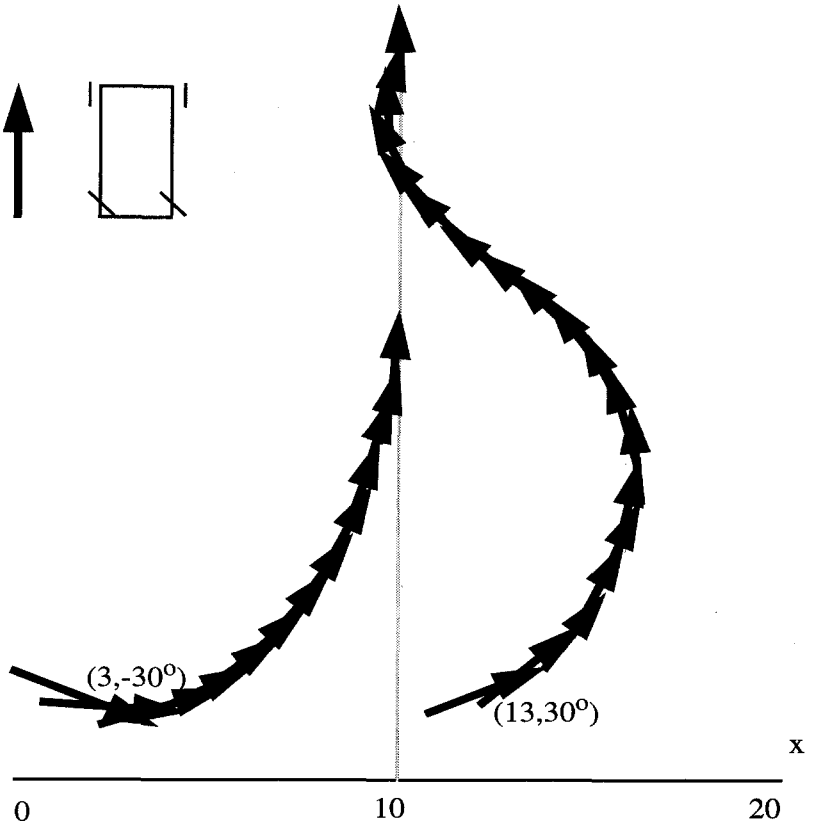


Figure 12.7. Truck trajectories using the fuzzy controller.

following delay differential equation:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{12.10}$$

When $\tau > 17$, (12.10) shows chaotic behavior. We choose $\tau = 30$.

Let $x(k)$ $(k = 1, 2, 3, ...)$ be the time series generated by (12.10) (sampling the continuous curve $x(t)$ generated by (12.10) with an interval of 1 sec.). Fig.12.8 shows 600 points of $x(k)$. The problem of time series prediction can be formulated as follows: given $x(k-n+1), x(k-n+2), ..., x(k)$, estimate $x(k+1)$, where n is a positive integer. That is, the task is to determine a mapping from $[x(k-n+1), x(k-n+2), ..., x(k)] \in R^n$ to $[x(k+1)] \in R$, and this mapping in our case is the designed

fuzzy system based on the input-output pairs. Assuming that $x(1), x(2), ..., x(k)$ are given with $k > n$, we can form $k - n$ input-output pairs as follows:

$$\begin{aligned} &[x(k-n), ..., x(k-1); x(k)] \\ &[x(k-n-1), ..., x(k-2); x(k-1)] \\ &... \\ &[x(1), ..., x(n); x(n+1)] \end{aligned} \tag{12.11}$$

These input-output pairs are used to design a fuzzy system $f(x)$ using the table lookup scheme in Section 12.2, and this $f(x)$ is then used to predict $x(k+l)$ for $l = 1, 2, ...$, where the input to $f(x)$ is $[x(k-n+l), ..., x(k-1+l)]$ when predicting $x(k+l)$.

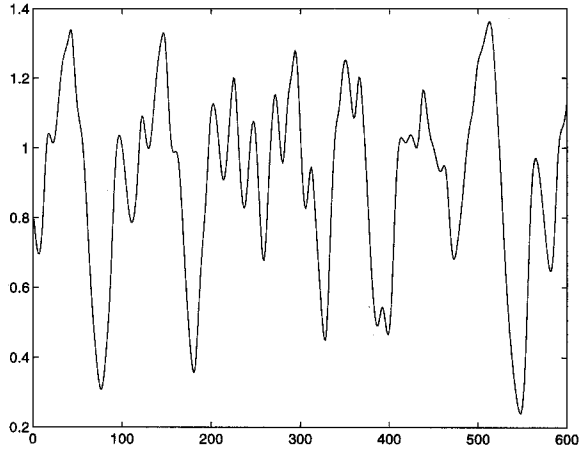


Figure 12.8. A section of the Mackey-Glass chaotic time series.

We now use the first 300 points in Fig. 12.8 to construct the input-output pairs and the designed fuzzy system is then used to predict the remaining 300 points. We consider two cases: (i) $n = 4$ and the 7 fuzzy sets in Fig.12.9 are defined for each input variable, and (ii) $n = 4$ and the 15 fuzzy sets in Fig.12.10 are used. The prediction results for these two cases are shown in Figs.12.11 and 12.12, respectively. Comparing Figs. 12.11 and 12.12, we see that the prediction accuracy is improved by defining more fuzzy sets for each input variable.

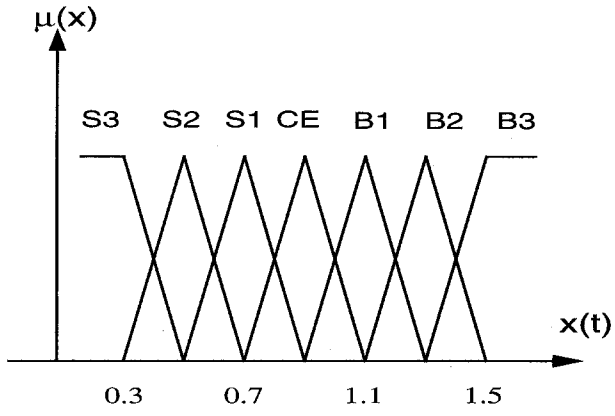


Figure 12.9. The first choice of membership functions for each input variable.

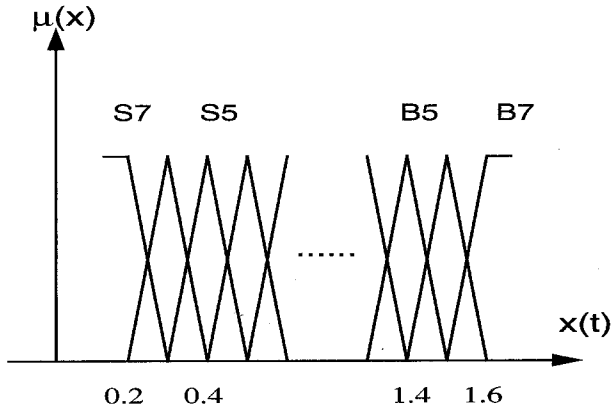


Figure 12.10. The second choice of membership functions for each input variable.

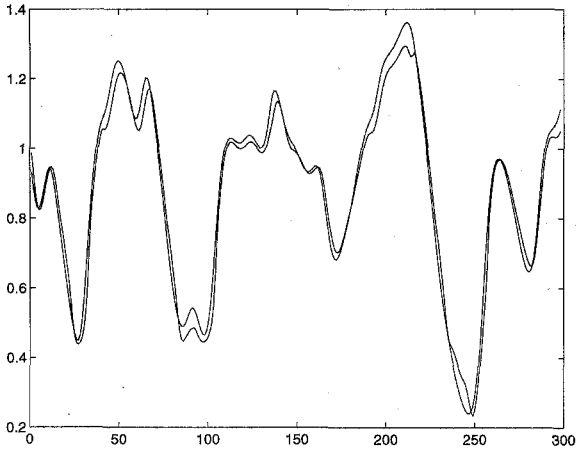


Figure 12.11. Prediction and the true values of the time series using the membership functions in Fig. 12.9.

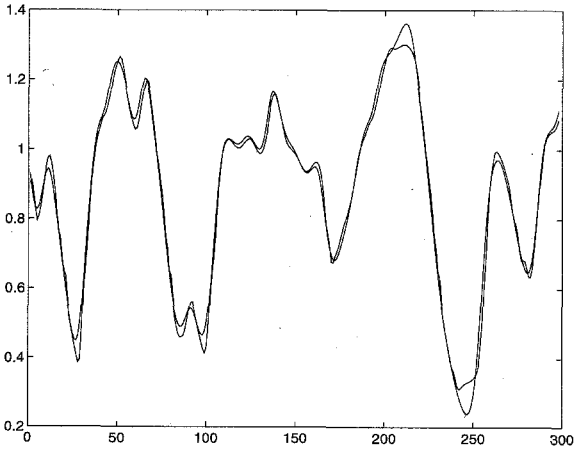


Figure 12.12. Prediction and the true values of the time series using the membership functions in Fig. 12.10.

12.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- The details of the table look-up method for designing fuzzy systems from input-output pairs.
- How to apply the method to the truck backer-upper control and the time series prediction problems.
- How to combine conscious and subconscious knowledge into fuzzy systems using this table look-up scheme.

This table look-up scheme is taken from Wang and Mendel [1992b] and Wang [1994], which discussed more details about this method and gave more examples. Application of the method to financial data prediction can be found in Cox [1994].

12.5 Exercises and Projects

Exercise 12.1. Consider the design of a 2-input-1-output fuzzy system using the table look-up scheme. Suppose that in Step 1 we define the fuzzy sets as shown in Fig. 12.2, where $\alpha_1 = \alpha_2 = \alpha_y = 0$ and $\beta_1 = \beta_2 = \beta_y = 1$, and the membership functions are triangular and equally spaced.

(a) What is the minimum number of input-output pairs such that every fuzzy sets in Fig. 12.2 will appear at least once in the generated rules? Give an example of this minimum set of input-output pairs.

(b) What is the minimum number of input-output pairs such that the generated fuzzy rule base is complete? Give an example of this minimum set of input-output pairs.

Exercise 12.2. Consider the truck backer-upper control problem in Section 12.3.

(a) Generate a set of input-output pairs by driving the truck from the initial state $(x_0, \phi_0) = (1, 90^o)$ to the final state $(x_f, \phi_f) = (10, 90^o)$ using common sense.

(b) Use the table look-up scheme to create a fuzzy rule base from the input-output pairs generated in (a), where the membership functions in Step 1 are given in Fig. 12.5.

(c) Construct a fuzzy system based on the fuzzy rule base in (b) and use it to control the truck from $(x_0, \phi_0) = (0, 90^o)$ and $(x_0, \phi_0) = (-3, 90^o)$. Comment on the simulation results.

Exercise 12.3. Let $f(x)$ be the fuzzy system designed using the table look-up scheme. Can you determine an error bound for $|f(x_0^p) - y_0^p|$? Explain your answer.

Exercise 12.4. Propose a method to fill up the empty boxes in the fuzzy rule base generated by the table look-up scheme. Justify your method and test it through examples.

Exercise 12.5. We are given 10 points $x(1), x(2), ..., x(10)$ of a time series and we want to predict $x(12)$.

(a) If we use the fuzzy system $f[x(10), x(8)]$ to predict $x(12)$, list all the input-output pairs for constructing this fuzzy system.

(b) If we use the fuzzy system $f[x(10), x(9), x(8)]$ to predict $x(12)$, list all the input-output pairs for constructing this fuzzy system.

12.6 (Project). Write a computer program to implement the table look-up scheme and apply your program to the time series prediction problem in Section 12.4. To make your codes generally applicable, you may have to include a method to fill up the empty boxes.

Chapter 13

Design of Fuzzy Systems Using Gradient Descent Training

13.1 Choosing the Structure of Fuzzy Systems

In the table look-up scheme of Chapter 12, the membership functions are fixed in the first step and do not depend on the input-output pairs; that is, the membership functions are not optimized according to the input-output pairs. In this chapter, we propose another approach to designing fuzzy systems where the membership functions are chosen in such a way that certain criterion is optimized.

From a conceptual point of view, the design of fuzzy systems from input-output pairs may be classified into two types of approaches. In the first approach, fuzzy IF-THEN rules are first generated from input-output pairs, and the fuzzy system is then constructed from these rules according to certain choices of fuzzy inference engine, fuzzifier, and defuzzifier. The table look-up scheme of Chapter 12 belongs to this approach. In the second approach, the structure of the fuzzy system is specified first and some parameters in the structure are free to change, then these free parameters are determined according to the input-output pairs. In this chapter, we adapt this second approach.

First, we specify the structure of the fuzzy system to be designed. Here we choose the fuzzy system with product inference engine, singleton fuzzifier, center average defuzzifier, and Gaussian membership function, given by (9.6). That is, we assume that the fuzzy system we are going to design is of the following form:

$$f(x) = \frac{\sum_{l=1}^{M} \bar{y}^l [\prod_{i=1}^{n} exp(-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2)]}{\sum_{l=1}^{M} [\prod_{i=1}^{n} exp(-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2)]} \tag{13.1}$$

where M is fixed, and $\bar{y}^l, \bar{x}_i^l$ and σ_i^l are free parameters (we choose $a_i^l = 1$). Although the structure of the fuzzy system is chosen as (13.1), the fuzzy system has not been designed because the parameters $\bar{y}^l, \bar{x}_i^l$ and σ_i^l are not specified. Once we specify the

parameters $\bar{y}^l, \bar{x}_i^l$ and σ_i^l, we obtain the designed fuzzy system; that is, designing the fuzzy system is now equivalent to determining the parameters $\bar{y}^l, \bar{x}_i^l$ and σ_i^l.

To determine these parameters in some optimal fashion, it is helpful to represent the fuzzy system $f(x)$ of (13.1) as a feedforward network. Specifically, the mapping from the input $x \in U \subset R^n$ to the output $f(x) \in V \subset R$ can be implemented according to the following operations: first, the input x is passed through a product Gaussian operator to become $z^l = \prod_{i=1}^n exp(-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2)$; then, the z^l are passed through a summation operator and a weighted summation operator to obtain $b = \sum_{l=1}^M z^l$ and $a = \sum_{l=1}^M \bar{y}^l z^l$; finally, the output of the fuzzy system is computed as $f(x) = a/b$. This three-stage operation is shown in Fig. 13.1 as a three-layer feedforward network.

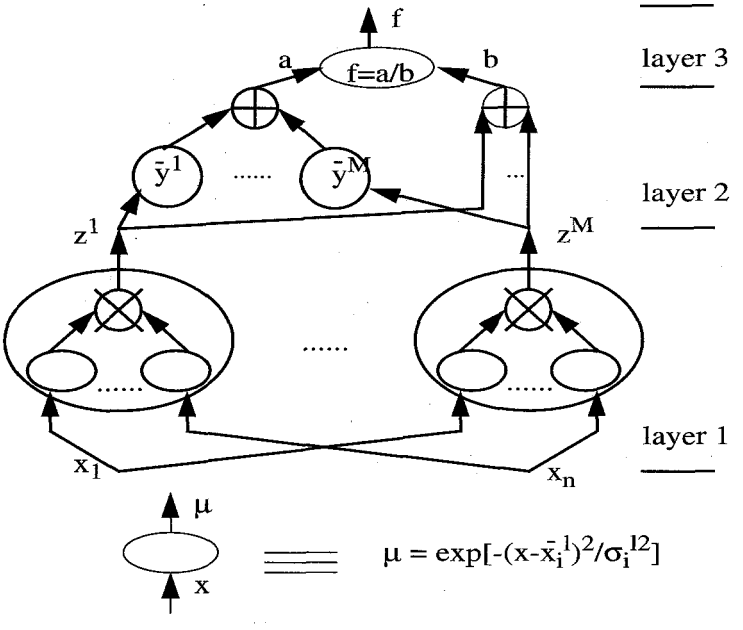


Figure 13.1. Network representation of the fuzzy system.

13.2 Designing the Parameters by Gradient Descent

As in Chapter 12, the data we have are the input-output pairs given by (12.1). Our task is to design a fuzzy system $f(x)$ in the form of (13.1) such that the matching

error

$$e^p = \frac{1}{2}[f(x_0^p) - y_0^p]^2 \tag{13.2}$$

is minimized. That is, the task is to determine the parameters $\bar{y}^l, \bar{x}_i^l$ and σ_i^l such that e^p of (13.2) is minimized. In the sequel, we use e, f and y to denote $e^p, f(x_0^p)$ and y_0^p, respectively.

We use the gradient descent algorithm to determine the parameters. Specifically, to determine $\bar{y}^l$, we use the algorithm

$$\bar{y}^l(q+1) = \bar{y}^l(q) - \alpha \frac{\partial e}{\partial \bar{y}^l}|_q \tag{13.3}$$

where $l = 1, 2, ..., M$, $q = 0, 1, 2, ...$, and α is a constant stepsize. If $\bar{y}^l(q)$ converges as q goes to infinity, then from (13.3) we have $\frac{\partial e}{\partial \bar{y}^l} = 0$ at the converged $\bar{y}^l$, which means that the converged $\bar{y}^l$ is a local minimum of e. From Fig. 13.1 we see that f (and hence e) depends on $\bar{y}^l$ only through a, where $f = a/b$, $a = \sum_{l=1}^M (\bar{y}^l z^l)$, $b = \sum_{l=1}^M z^l$, and $z^l = \prod_{i=1}^n exp(-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2)$; hence, using the Chain Rule, we have

$$\frac{\partial e}{\partial \bar{y}^l} = (f - y)\frac{\partial f}{\partial a}\frac{\partial a}{\partial \bar{y}^l} = (f - y)\frac{1}{b}z^l \tag{13.4}$$

Substituting (13.4) into (13.3), we obtain the training algorithm for $\bar{y}^l$:

$$\bar{y}^l(q+1) = \bar{y}^l(q) - \alpha \frac{f - y}{b} z^l \tag{13.5}$$

where $l = 1, 2, ..., M$, and $q = 0, 1, 2, ...$

To determine $\bar{x}_i^l$, we use

$$\bar{x}_i^l(q+1) = \bar{x}_i^l(q) - \alpha \frac{\partial e}{\partial \bar{x}_i^l}|_q \tag{13.6}$$

where $i = 1, 2, ..., n, l = 1, 2, ..., M$, and $q = 0, 1, 2, ...$ We see from Fig. 13.1 that f (and hence e) depends on $\bar{x}_i^l$ only through z^l; hence, using the Chain Rule, we have

$$\frac{\partial e}{\partial \bar{x}_i^l} = (f - y)\frac{\partial f}{\partial z^l}\frac{\partial z^l}{\partial \bar{x}_i^l} = (f - y)\frac{\bar{y}^l - f}{b} z^l \frac{2(x_{0i}^p - \bar{x}_i^l)}{\sigma_i^{l2}} \tag{13.7}$$

Substituting (13.7) into (13.6), we obtain the training algorithm for $\bar{x}_i^l$:

$$\bar{x}_i^l(q+1) = \bar{x}_i^l(q) - \alpha \frac{f - y}{b}(\bar{y}^l(q) - f) z^l \frac{2(x_{0i}^p - \bar{x}_i^l(q))}{\sigma_i^{l2}(q)} \tag{13.8}$$

where $i = 1, 2, ..., n, l = 1, 2, ..., M$, and $q = 0, 1, 2, ...$

Using the same procedure, we obtain the training algorithm for σ_i^l:

$$\sigma_i^l(q+1) = \sigma_i^l(q) - \alpha \frac{\partial e}{\partial \sigma_i^l}|_q$$
$$= \sigma_i^l(q) - \alpha \frac{f-y}{b}(\bar{y}^l(q) - f) z^l \frac{2(x_{0i}^p - \bar{x}_i^l(q))^2}{\sigma_i^{l3}(q)} \tag{13.9}$$

where $i = 1, 2, ..., n, l = 1, 2, ..., M$, and $q = 0, 1, 2, ...$

The training algorithm (13.5), (13.8), and (13.9) performs an error back-propagation procedure. To train $\bar{y}^l$, the "normalized" error $(f-y)/b$ is back-propagated to the layer of $\bar{y}^l$; then $\bar{y}^l$ is updated using (13.5) in which z^l is the input to $\bar{y}^l$ (see Fig. 13.1). To train $\bar{x}_i^l$ and σ_i^l, the "normalized" error $(f-y)/b$ times $(\bar{y}^l - f)$ and z^l is back-propagated to the processing unit of Layer 1 whose output is z^l; then $\bar{x}_i^l$ and σ_i^l are updated using (13.8) and (13.9), respectively, in which the remaining variables $\bar{x}_i^l, x_{0i}^p$, and σ_i^l (that is, the variables on the right-hand sides of (13.8) and (13.9), except the back-propagated error $\frac{f-y}{b}(\bar{y}^l - f)z^l$) can be obtained locally. Therefore, this algorithm is also called the *error back-propagation training algorithm.*

We now summarize this design method.

Design of Fuzzy Systems Using Gradient Descent Training:

- **Step 1. Structure determination and initial parameter setting.** Choose the fuzzy system in the form of (13.1) and determine the M. Larger M results in more parameters and more computation, but gives better approximation accuracy. Specify the initial parameters $\bar{y}^l(0), \bar{x}_i^l(0)$ and $\sigma_i^l(0)$. These initial parameters may be determined according to the linguistic rules from human experts, or be chosen in such a way that the corresponding membership functions uniformly cover the input and output spaces. For particular applications, we may use special methods; see Section 13.3 for an example.

- **Step 2. Present input and calculate the output of the fuzzy system.** For a given input-output pair $(x_0^p; y_0^p)$, $p = 1, 2, ...,$ and at the $q'th$ stage of training, $q = 0, 1, 2, ...,$ present x_0^p to the input layer of the fuzzy system in Fig. 13.1 and compute the outputs of Layers 1-3. That is, compute

$$z^l = \prod_{i=1}^{n} exp(-(\frac{x_{0i}^p - \bar{x}_i^l(q)}{\sigma_i^l(q)})^2) \tag{13.10}$$

$$b = \sum_{l=1}^{M} z^l \tag{13.11}$$

$$a = \sum_{l=1}^{M} \bar{y}^l(q) z^l \tag{13.12}$$

$$f = a/b \tag{13.13}$$

- **Step 3. Update the parameters.** Use the training algorithm (13.5), (13.8) and (13.9) to compute the updated parameters $\bar{y}^l(q+1)$, $\bar{x}_i^l(q+1)$ and $\sigma_i^l(q+1)$, where $y = y_0^p$, and z^l, b, a and f equal those computed in Step 2.

- **Step 4.** Repeat by going to Step 2 with $q = q + 1$, until the error $|f - y_0^p|$ is less than a prespecified number ϵ, or until the q equals a prespecified number.

- **Step 5.** Repeat by going to Step 2 with $p = p + 1$; that is, update the paramters using the next input-output pair $(x_0^{p+1}; y_0^{p+1})$.

- **Step 6.** If desirable and feasible, set $p = 1$ and do Steps 2-5 again until the designed fuzzy system is satisfactory. For on-line control and dynamic system identification, this step is not feasible because the input-output pairs are provided one-by-one in a real-time fashion. For pattern recognition problems where the input-output pairs are provided off-line, this step is usually desirable.

Because the training algorithm (13.5), (13.8) and (13.9) is a gradient descent algorithm, the choice of the initial parameters is crucial to the success of the algorithm. If the initial parameters are close to the optimal parameters, the algorithm has a good chance to converge to the optimal solution; otherwise, the algorithm may converge to a nonoptimal solution or even diverge. The advantage of using the fuzzy system is that the parameters $\bar{y}^l, \bar{x}_i^l$ and σ_i^l have clear physical meanings and we have methods to choose good initial values for them. Keep in mind that the parameters $\bar{y}^l$ are the centers of the fuzzy sets in the THEN parts of the rules, and the parameters $\bar{x}_i^l$ and σ_i^l are the centers and widths of the Gaussian fuzzy sets in the IF parts of the rules. Therefore, given a designed fuzzy system in the form of (13.1), we can recover the fuzzy IF-THEN rules that constitute the fuzzy system. These recovered fuzzy IF-THEN rules may help to explain the designed fuzzy system in a user-friendly manner.

Next, we apply this method to the problem of nonlinear dynamic system identification.

13.3 Application to Nonlinear Dynamic System Identification

13.3.1 Design of the Identifier

System identification is a process of determining an appropriate model for the system based on measurements from sensors. It is an important process because many approaches in engineering depend on the model of the system. Because the fuzzy systems are powerful universal approximators, it is reasonable to use them as identification models for nonlinear systems. In this section, we use the fuzzy system

(13.1) equipped with the training algorithm (13.5), (13.8) and (13.9) to approximate unknown nonlinear components in dynamic systems.

Consider the discrete time nonlinear dynamic system

$$y(k+1) = f(y(k), ..., y(k-n+1); u(k), ..., u(k-m+1)) \tag{13.14}$$

where f is an unknown function we want to identify, u and y are the input and output of the system, respectively, and n and m are positive integers. Our task is to identify the unknown function f based on fuzzy systems.

Let $\hat{f}(x)$ be the fuzzy system in the form of (13.1). We replace the $f(x)$ in (13.14) by $\hat{f}(x)$ and obtain the following identification model:

$$\hat{y}(k+1) = \hat{f}(y(k), ..., y(k-n+1); u(k), ..., u(k-m+1)) \tag{13.15}$$

Our task is to adjust the parameters in $\hat{f}(x)$ such that the output of the identification model $\hat{y}(k+1)$ converges to the output of the true system $y(k+1)$ as k goes to infinity. Fig. 13.2 shows this identification scheme.

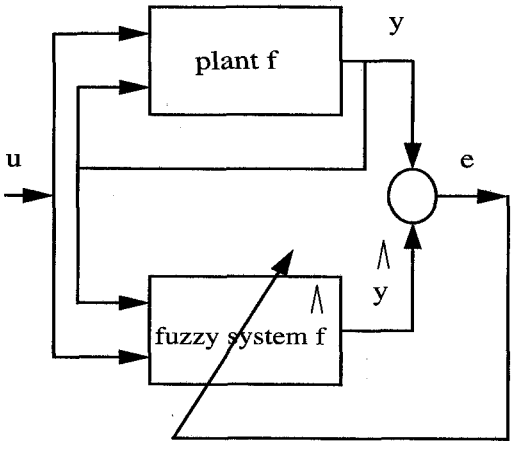


Figure 13.2. Basic scheme of identification model for the nonlinear dynamic system using the fuzzy system.

The input-output pairs in this problem are $(x_0^{k+1}; y_0^{k+1})$, where $x_0^{k+1} = (y(k), ..., y(k-n+1); u(k), ..., u(k-m+1))$, $y_0^{k+1} = y(k+1)$, and $k = 0, 1, 2, ...$ Because the system is dynamic, these input-output pairs are collected one at a time. The operation of the identification process is the same as the Steps 1-5 in Section 13.2. Note that the p there is the k in (13.14) and (13.15) and the n in (13.1) equals $n+m$.

13.3.2 Initial Parameter Choosing

As we discussed in Section 13.2, a good initial $\hat{f}$ is crucial for the success of the approach. For this particular identification problem, we propose the following method for on-line initial parameter choosing and provide a theoretical justification (Lemma 13.1) to explain why this is a good method.

An on-line initial parameter choosing method: Collect the input-output pairs $(x_0^{k+1}; y_0^{k+1}) = (y(k), ..., y(k-n+1), u(k), ..., u(k-m+1); y(k+1))$ for the first M time points $k = 0, 1, ..., M-1$, and do not start the training algorithm until $k = M-1$ (that is, set $q = k-M$ in (13.5), (13.8) and (13.9)). The initial parameters are chosen as: $\bar{y}^l(0) = y_0^l, \bar{x}_i^l(0) = x_{0i}^l$, and $\sigma_i^l(0)$ equals small numbers (see Lemma 13.1) or $\sigma_i^l(0) = [max(x_{oi}^l : l = 1, 2, ..., M) - min(x_{oi}^l : l = 1, 2, ..., M)]/M$, where $l = 1, 2, ..., M$ and $i = 1, 2, ..., n+m$. The second choice of $\sigma_i^l(0)$ makes the membership functions uniformly cover the range of x_{0i}^l from $l = 1$ to $l = M$.

We now show that by choosing the $\sigma_i^l(0)$ sufficiently small, the fuzzy system with the preceding initial parameters can match all the M input-output pairs $(x_0^{k+1}; y_0^{k+1})$, $k = 0, 1, ..., M-1$, to arbitrary accuracy.

Lemma 13.1. For arbitrary $\epsilon > 0$, there exists $\sigma^* > 0$ such that the fuzzy system $\hat{f}(x)$ of (13.1) with the preceding initial parameters $\bar{y}^l$ and $\bar{x}_i^l$ and $\sigma_i^l = \sigma^*$ has the property that

$$|\hat{f}(x_0^{k+1}) - y_0^{k+1}| < \epsilon \tag{13.16}$$

for $k = 0, 1, ..., M-1$.

Proof: Substituting the initial parameters $\bar{y}^l(0)$ and $\bar{x}_i^l(0)$ into (13.1) and setting $\sigma_i^l = \sigma^*$, we have

$$\hat{f}(x) = \frac{\sum_{l=1}^{M} y_0^l [\prod_{i=1}^{n+m} exp(-(\frac{x_i - x_{oi}^l}{\sigma^*})^2)]}{\sum_{l=1}^{M} [\prod_{i=1}^{n+m} exp(-(\frac{x_i - x_{oi}^l}{\sigma^*})^2)]} \tag{13.17}$$

Setting $x = x_0^{k+1}$ in (13.17) and noticing $1 \leq k+1 \leq M$, we have

$$\begin{aligned} \hat{f}(x_0^{k+1}) &= \frac{\sum_{l=1}^{M} y_0^l [\prod_{i=1}^{n+m} exp(-(\frac{x_{oi}^{k+1} - x_{oi}^l}{\sigma^*})^2)]}{\sum_{l=1}^{M} [\prod_{i=1}^{n+m} exp(-(\frac{x_{oi}^{k+1} - x_{oi}^l}{\sigma^*})^2)]} \\ &= \frac{y_0^{k+1} + \sum_{l=1, l\neq k+1}^{M} y_0^l [\prod_{i=1}^{n+m} exp(-(\frac{x_{oi}^{k+1} - x_{oi}^l}{\sigma^*})^2)]}{1 + \sum_{l=1, l\neq k+1}^{M} [\prod_{i=1}^{n+m} exp(-(\frac{x_{oi}^{k+1} - x_{oi}^l}{\sigma^*})^2)]} \end{aligned} \tag{13.18}$$

Hence,

$$|\hat{f}(x_0^{k+1}) - y_0^{k+1}| = \left| \frac{\sum_{l=1, l\neq k+1}^{M} (y_0^l - y_0^{k+1}) [\prod_{i=1}^{n+m} exp(-(\frac{x_{oi}^{k+1} - x_{oi}^l}{\sigma^*})^2)]}{1 + \sum_{l=1, l\neq k+1}^{M} [\prod_{i=1}^{n+m} exp(-(\frac{x_{oi}^{k+1} - x_{oi}^l}{\sigma^*})^2)]} \right| \tag{13.19}$$

If $x_0^{l_1} \neq x_0^{l_2}$ for $l_1 \neq l_2$, then we have $\prod_{i=1}^{n+m} exp(-(\frac{x_{oi}^{k+1} - x_{oi}^{l}}{\sigma^*})^2) \to 0$ as $\sigma^* \to 0$ for $l \neq k+1$. Therefore, by choosing σ^* sufficiently small, we can make $|\hat{f}(x_0^{k+1}) - y_0^{k+1}| < \epsilon$. Similarly, we can show that (13.16) is true in the case where $x_0^l = x_0^{k+1}$ for some $l \neq k+1$; this is left as an exercise. □

Based on Lemma 13.1, we can say that the initial parameter choosing method is a good one because the fuzzy system with these initial parameters can at least match the first M input-output pairs arbitrarily well. If these first M input-output pairs contain important features of the unknown nonlinear function $f(x)$, we can hope that after the training starts from time point M, the fuzzy identifier will converge to the unknown nonlinear system very quickly. In fact, based on our simulation results in the next subsection, this is indeed true. However, we cannot choose σ_i^l too small because, although a fuzzy system with small σ_i^l matches the first M pairs quite well, it may give large approximation errors for other input-output pairs. Therefore, in our following simulations we will use the second choice of σ_i^l described in the on-line initial parameter choosing method.

13.3.3 Simulations

Example 13.1. The plant to be identified is governed by the difference equation

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + g[u(k)] \tag{13.20}$$

where the unknown function has the form $g(u) = 0.6sin(\pi u) + 0.3sin(3\pi u) + 0.1sin(5\pi u)$. From (13.15), the identification model is governed by the difference equation

$$\hat{y}(k+1) = 0.3y(k) + 0.6y(k-1) + \hat{f}[u(k)] \tag{13.21}$$

where $\hat{f}[*]$ is in the form of (13.1) with $M = 10$. We choose $\alpha = 0.5$ in the training algorithm (13.5), (13.8) and (13.9) and use the on-line parameter choosing method. We start the training from time point $k = 10$, and adjust the parameters $\bar{y}^l$, $\bar{x}_i^l$, and σ_i^l for one cycle at each time point. That is, we use (13.5), (13.8) and (13.9) once at each time point. Fig. 13.3 shows the outputs of the plant and the identification model when the training stops at $k = 200$, where the input $u(k) = sin(2\pi k/200)$. We see from Fig. 13.3 that the output of the identification model follows the output of the plant almost immediately and still does so when the training stops at $k = 200$. □

Example 13.2. In this example, we show how the fuzzy identifier works for a multi-input-multi-output plant that is described by the equations

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} \frac{y_1(k)}{1+y_2^2(k)} \\ \frac{y_1(k)y_2(k)}{1+y_2^2(k)} \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{13.22}$$

The identification model consists of two fuzzy systems, $\hat{f}^1$ and $\hat{f}^2$, and is described

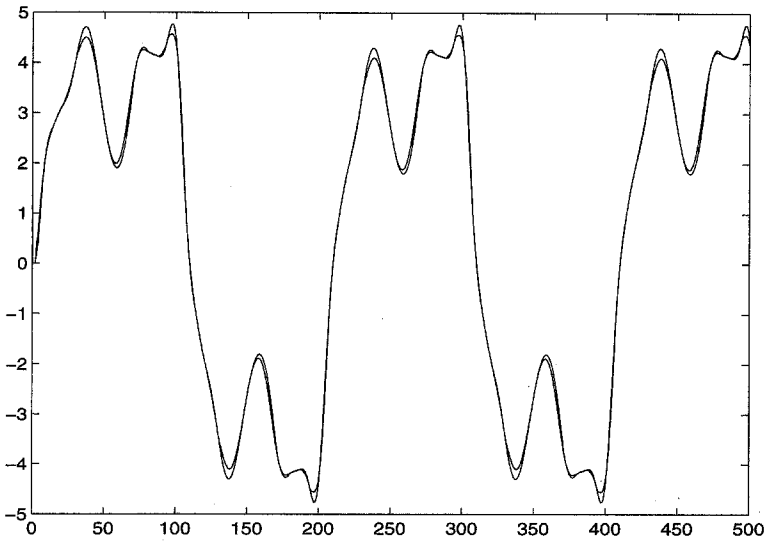


Figure 13.3. Outputs of the plant and the identification model for Example 13.1 when the training stops at $k = 200$.

by the equations

$$\begin{bmatrix} \hat{y}_1(k+1) \\ \hat{y}_2(k+1) \end{bmatrix} = \begin{bmatrix} \hat{f}^1(y_1(k), y_2(k)) \\ \hat{f}^2(y_1(k), y_2(k)) \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{13.23}$$

Both $\hat{f}^1$ and $\hat{f}^2$ are in the form of (13.1) with $M = 121$. The identification procedure is carried out for 5,000 time steps using random inputs $u_1(k)$ and $u_2(k)$ whose magnitudes are uniformly distributed over [-1,1], where we choose $\alpha = 0.5$, use the on-line initial parameter choosing method, and train the parameters for one cycle at each time point. The responses of the plant and the trained identification model for a vector input $[u_1(k), u_2(k)] = [sin(2\pi k/25), cos(2\pi k/25)]$ are shown in Figs. 13.4 and 13.5 for $y_1(k)$-$\hat{y}_1(k)$ and $y_2(k)$-$\hat{y}_2(k)$, respectively. We see that the fuzzy identifier follows the true system almost perfectly. □

13.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- The derivation of the gradient descent training algorithm.
- The method for choosing the initial parameters of the fuzzy identification model and its justification (Lemma 13.1).

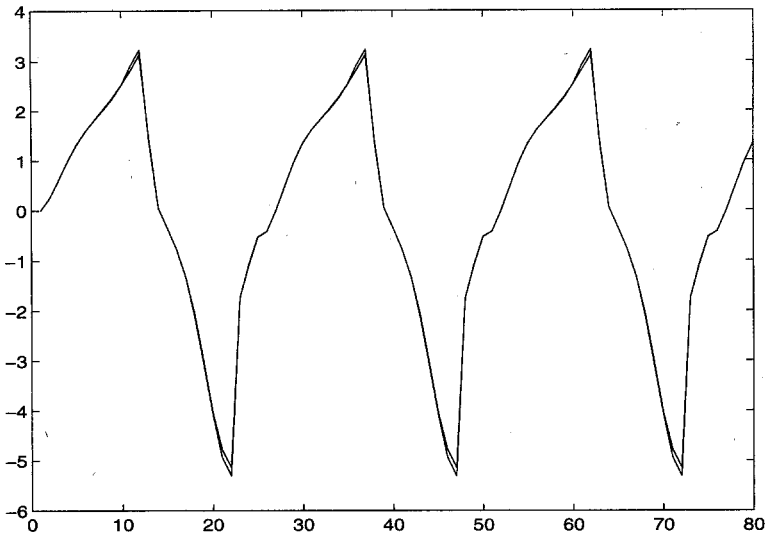


Figure 13.4. Outputs of the plant $y_1(k)$ and the identification model $\hat{y}_1(k)$ for Example 13.2 after 5,000 steps of training.

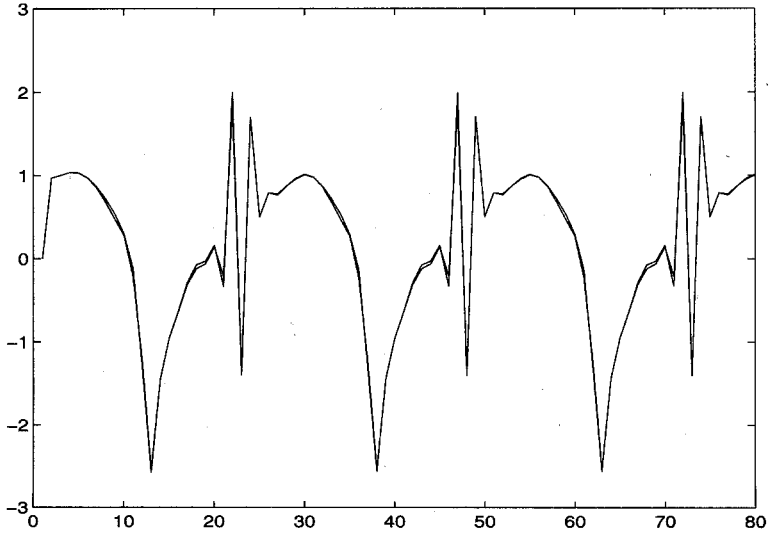


Figure 13.5. Outputs of the plant $y_2(k)$ and the identification model $\hat{y}_2(k)$ for Example 13.2 after 5,000 steps of training.

- Design of fuzzy systems based on the gradient descent training algorithm and the initial parameter choosing method.
- Application of this approach to nonlinear dynamic system identification and other problems.

The method in this chapter was inspired by the error back-propagation algorithm for neural networks (Werbos [1974]). Many similar methods were proposed in the literature; see Jang [1993] and Lin [1994]. More simulation results can be found in Wang [1994]. Narendra and Parthasarathy [1990] gave extensive simulation results of neural network identifiers for the nonlinear systems such as those in Examples 13.1 and 13.2.

13.5 Exercises and Projects

Exercise 13.1. Suppose that the parameters $\bar{x}_i^l$ and σ_i^l in (13.1) are fixed and only the $\bar{y}^l$ are free to change.

(a) Show that if the training algorithm (13.5) converges, then it converges to the global minimum of e^p of (13.2).

(b) Let $e^p(\bar{y}^l)$ be the e^p of (13.2). Find the optimal stepsize α by minimizing $e^p[\bar{y}^l(q) - \alpha\frac{f-y}{b}z^l]$ with respect to α.

Exercise 13.2. Why is the proof of Lemma 13.1 not valid if $x_0^l = x_0^{k+1}$ for some $l \neq k+1$? Prove Lemma 13.1 for this case.

Exercise 13.3. Suppose that we are given K input-output pairs $(x_0^p; y_0^p), p = 1, 2, ..., K$, and we want to design a fuzzy system $f(x)$ in the form of (13.1) such that the summation of squared errors

$$J = \sum_{p=1}^{K}[f(x_0^p) - y_0^p]^2 \tag{13.24}$$

is minimized. Let the parameters $\bar{x}_i^l$ and σ_i^l be fixed and only the $\bar{y}^l$ be free to change. Determine the optimal $\bar{y}^l$ $(l = 1, 2, ..., M)$ such that J is minimized (write the optimal $\bar{y}^l$ in a closed-form expression).

Exercise 13.4. Let $[x(k)]$ be a sequence of real-valued vectors generated by the gradient descent algorithm

$$x(k+1) = x(k) - \alpha\nabla e(x(k)), \tag{13.25}$$

where $e : R^n \to R$ is a cost function and $e \in C^2$ (i.e., e has continuous second derivative). Assume that all $x(k) \in D \subset R^n$ for some compact D, then there exist $\epsilon > 0$ and $L > 0$ such that if

$$0 < \epsilon \leq \alpha \leq \frac{2-\epsilon}{L}, \tag{13.26}$$

then:

(a) $e(x(k+1)) < e(x(k))$ if $\nabla e(x(k)) \neq 0$;

(b) $x(k) \to x^* \in D$ as $k \to \infty$ if $e(*)$ is bounded from below; and

(c) x^* is a local minimum of $e(x)$.

Exercise 13.5. It is a common perception that the gradient decent algorithm in Section 13.2 converges to local minima. Create a set of training samples that numerically shows this phenomenon; that is, show that a different initial condition may lead to a different final convergent solution.

Exercise 13.6. Explain how conscious and subconscious knowledge are combined into the fuzzy system using the design method in Section 13.2. Is it possible that conscious knowledge disappears in the final designed fuzzy system? Why? If we want to preserve conscious knowledge in the final fuzzy system, how to modify the design procedure in Section 13.2?

13.7 (Project). Write a computer program to implement the training algorithm (13.5), (13.8), and (13.9), and apply your codes to the time series prediction problem in Chapter 12.

Chapter 14

Design of Fuzzy Systems Using Recursive Least Squares

14.1 Design of the Fuzzy System

The gradient descent algorithm in Chapter 13 tries to minimum the criterion e^p of (13.2), which accounts for the matching error of only one input-output pair $(x_0^p; y_0^p)$. That is, the training algorithm updates the parameters to match one input-output pair at a time. In this chapter, we develop a training algorithm that minimizes the summation of the matching errors for all the input-output pairs up to p, that is, the objective now is to design a fuzzy system $f(x)$ such that

$$J_p = \sum_{j=1}^{p} [f(x_0^j) - y_0^j]^2 \tag{14.1}$$

is minimized. Additionally, we want to design the fuzzy system recursively; that is, if f_p is the fuzzy system designed to minimize J_p, then f_p should be represented as a function of f_{p-1}. We now use the recursive least squares algorithm to design the fuzzy system.

Design of the Fuzzy System by Recursive Least Squares:

- **Step 1**. Suppose that $U = [\alpha_1, \beta_1] \times \cdots \times [\alpha_n, \beta_n] \subset R^n$. For each $[\alpha_i, \beta_i]$ $(i = 1, 2, ..., n)$, define N_i fuzzy sets $A_i^{l_i}$ $(l_i = 1, 2, ..., N_i)$, which are complete in $[\alpha_i, \beta_i]$. For example, we may choose $A_i^{l_i}$ to be the pseudo-trapezoid fuzzy sets: $\mu_{A_i^{l_i}}(x_i) = \mu_{A_i^{l_i}}(x_i; a_i^{l_i}, b_i^{l_i}, c_i^{l_i}, d_i^{l_i})$, where $a_i^1 = b_i^1 = \alpha_i, c_i^j \leq a_i^{j+1} < d_i^j \leq b_i^{j+1}$ for $j = 1, 2, ..., N_i - 1$, and $c_i^{N_i} = d_i^{N_i} = \beta_i$.
- **Step 2**. Construct the fuzzy system from the following $\prod_{i=1}^{n} N_i$ fuzzy IF-THEN rules:

$$IF\ x_1\ is\ A_1^{l_1}\ and\ \cdots\ and\ x_n\ is\ A_n^{j_n},\ THEN\ y\ is\ B^{l_1 \cdots l_n} \tag{14.2}$$

where $l_i = 1, 2, ..., N_i$, $i = 1, 2, ..., n$ and $B^{l_1 \cdots l_n}$ is any fuzzy set with center at $\bar{y}^{l_1 \cdots l_n}$ which is free to change. Specifically, we choose the fuzzy system with product inference engine, singleton fuzzifier, and center average defuzzifier; that is, the designed fuzzy system is

$$f(x) = \frac{\sum_{l_1=1}^{N_1} \cdots \sum_{l_n=1}^{N_n} \bar{y}^{l_1 \cdots l_n} [\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)]}{\sum_{l_1=1}^{N_1} \cdots \sum_{l_n=1}^{N_n} [\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)]} \tag{14.3}$$

where $\bar{y}^{l_1 \cdots l_n}$ are free parameters to be designed, and $A_i^{l_i}$ are designed in Step 1. Collect the free parameters $\bar{y}^{l_1 \cdots l_n}$ into the $\prod_{i=1}^{n} N_i$-dimensional vector

$$\theta = (\bar{y}^{1 \cdots 1}, ..., \bar{y}^{N_1 1 \cdots 1}, \bar{y}^{121 \cdots 1}, ..., \bar{y}^{N_1 21 \cdots 1}, ..., \bar{y}^{1 N_2 \cdots N_n}, ..., \bar{y}^{N_1 N_2 \cdots N_n})^T \tag{14.4}$$

and rewrite (14.3) as

$$f(x) = b^T(x)\theta \tag{14.5}$$

where

$$b(x) = (b^{1 \cdots 1}(x), ..., b^{N_1 1 \cdots 1}(x), b^{121 \cdots 1}(x), ..., b^{N_1 21 \cdots 1}(x), ..., b^{1 N_2 \cdots N_n}(x), ..., b^{N_1 N_2 \cdots N_n}(x))^T \tag{14.6}$$

$$b^{l_1 \cdots l_n}(x) = \frac{\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{N_1} \cdots \sum_{l_n=1}^{N_n} [\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)]} \tag{14.7}$$

- **Step 3.** Choose the initial parameters $\theta(0)$ as follows: if there are linguistic rules from human experts (conscious knowledge) whose IF parts agree with the IF parts of (14.2), then choose $\bar{y}^{l_1 \cdots l_n}(0)$ to be the centers of the THEN part fuzzy sets in these linguistic rules; otherwise, choose $\theta(0)$ arbitrarily in the output space $V \subset R$ (for example, choose $\theta(0) = 0$ or the elements of $\theta(0)$ uniformly distributed over V). In this way, we can say that the initial fuzzy system is constructed from conscious human knowledge.

- **Step 4.** For $p = 1, 2, ...$, compute the parameters θ using the following recursive least squares algorithm:

$$\theta(p) = \theta(p-1) + K(p)[y_0^p - b^T(x_0^p)\theta(p-1)] \tag{14.8}$$

$$K(p) = P(p-1)b(x_0^p)[b^T(x_0^p)P(p-1)b(x_0^p) + 1]^{-1} \tag{14.9}$$

$$P(p) = P(p-1) - P(p-1)b(x_0^p) [b^T(x_0^p)P(p-1)b(x_0^p) + 1]^{-1} b^T(x_0^p)P(p-1) \tag{14.10}$$

where $\theta(0)$ is chosen as in Step 3, and $P(0) = \sigma I$ where σ is a large constant. The designed fuzzy system is in the form of (14.3) with the parameters $\bar{y}^{l_1 \cdots l_n}$ equal to the corresponding elements in $\theta(p)$.

The recursive least squares algorithm (14.8)-(14-10) is obtained by minimizing J_p of (14.1) with $f(x_0^j)$ in the form of (14.3); its derivation is given next.

14.2 Derivation of the Recursive Least Squares Algorithm

Let $Y_0^{p-1} = (y_0^1, ..., y_0^{p-1})^T$ and $B_{p-1} = (b(x_0^1), ..., b(x_0^{p-1}))^T$, then from (14.5) we can rewrite J_{p-1} as

$$\begin{aligned} J_{p-1} &= \sum_{j=1}^{p-1} [f(x_0^j) - y_0^j]^2 \\ &= (B_{p-1}\theta - Y_0^{p-1})^T (B_{p-1}\theta - Y_0^{p-1}) \end{aligned} \tag{14.11}$$

Since J_{p-1} is a quadratic function of θ, the optimal θ that minimizes J_{p-1}, denoted by $\theta(p-1)$, is

$$\theta(p-1) = (B_{p-1}^T B_{p-1})^{-1} B_{p-1}^T Y_0^{p-1} \tag{14.12}$$

When the input-output pair $(x_0^p; y_0^p)$ becomes available, the criterion changes to J_p of (14.1) which can be rewritten as

$$J_p = \left[\begin{pmatrix} B_{p-1} \\ b^T(x_0^p) \end{pmatrix} \theta - \begin{pmatrix} Y_0^{p-1} \\ y_0^p \end{pmatrix} \right]^T \left[\begin{pmatrix} B_{p-1} \\ b^T(x_0^p) \end{pmatrix} \theta - \begin{pmatrix} Y_0^{p-1} \\ y_0^p \end{pmatrix} \right] \tag{14.13}$$

Similar to (14.12), the optimal θ which minimizes J_p, denoted by $\theta(p)$, is obtained as

$$\begin{aligned} \theta(p) &= [(B_{p-1}^T b(x_0^p))(\begin{matrix} B_{p-1} \\ b^T(x_0^p) \end{matrix})]^{-1} (B_{p-1}^T b(x_0^p))(\begin{matrix} Y_0^{p-1} \\ y_0^p \end{matrix}) \\ &= [B_{p-1}^T B_{p-1} + b(x_0^p) b^T(x_0^p)]^{-1} [B_{p-1}^T Y_0^{p-1} + b(x_0^p) y_0^p] \end{aligned} \tag{14.14}$$

To further simplify (14.14), we need to use the matrix identity

$$(P^{-1} + bb^T)^{-1} = P - Pb(b^T Pb + 1)^{-1} b^T P \tag{14.15}$$

Defining $P(p-1) = (B_{p-1}^T B_{p-1})^{-1}$ and using (14.15), we can rewrite (14.14) as

$$\begin{aligned} \theta(p) = \{&P(p-1) - P(p-1)b(x_0^p)[b^T(x_0^p)P(p-1)b(x_0^p) + 1]^{-1} b^T(x_0^p)P(p-1)\} \\ &[B_{p-1}^T Y_0^{p-1} + b(x_0^p) y_0^p] \end{aligned} \tag{14.16}$$

Since $P(p-1)B_{p-1}^T Y_0^{p-1} = (B_{p-1}^T B_{p-1})^{-1} B_{p-1}^T Y_0^{p-1} = \theta(p-1)$ (see (14.12)), we can simplify (14.16) to

$$\begin{aligned} \theta(p) &= \theta(p-1) - P(p-1)b(x_0^p)[b^T(x_0^p)P(p-1)b(x_0^p) + 1]^{-1} b^T(x_0^p)\theta(p-1) \\ &\quad + P(p-1)b(x_0^p)[1 - (b^T(x_0^p)P(p-1)b(x_0^p) + 1)^{-1} b^T(x_0^p)P(p-1)b(x_0^p)]y_0^p \\ &= \theta(p-1) + P(p-1)b(x_0^p)[b^T(x_0^p)P(p-1)b(x_0^p) + 1]^{-1} \\ &\quad (y_0^p - b^T(x_0^p)\theta(p-1)) \end{aligned} \tag{14.17}$$

Defining $K(p) = P(p-1)b(x_0^p)[b^T(x_0^p)P(p-1)b(x_0^p)+1]^{-1}$, we obtain (14.8) and (14.9).

Finally, we derive (14.10). By definition, we have

$$\begin{aligned} P(p) &= [(B_{p-1}^T b(x_0^p))(\begin{matrix} B_{p-1} \\ b^T(x_0^p) \end{matrix})]^{-1} \\ &= [B_{p-1}^T B_{p-1} + b(x_0^p)b^T(x_0^p)]^{-1} \end{aligned} \tag{14.18}$$

Using the matrix identity (14.15) and the fact that $B_{p-1}^T B_{p-1} = P^{-1}(p-1)$, we obtain (14.10) from (14.18).

14.3 Application to Equalization of Nonlinear Communication Channels

14.3.1 The Equalization Problem and Its Geometric Formulation

Nonlinear distortion over a communication channel is now a significant factor hindering further increase in the attainable data rate in high-speed data transmission (Biglieri, E., A. Gersho, R.D. Gitlin, and T.L. Lim [1984]). Because the received signal over a nonlinear channel is a nonlinear function of the past values of the transmitted symbols and the nonlinear distortion varies with time and from place to place, effective equalizers for nonlinear channels should be nonlinear and adaptive. In this section, we use the fuzzy system designed from the recursive least squares algorithm as an equalizer for nonlinear channels.

The digital communication system considered here is shown in Fig. 14.1, where the *channel* includes the effects of the transmitter filter, the transmission medium, the receiver matched filter, and other components. The transmitted data sequence $s(k)$ is assumed to be an independent sequence taking values from $\{-1, 1\}$ with equal probability. The inputs to the equalizer, $x(k), x(k-1), \cdots, x(k-n+1)$, are the channel outputs corrupted by an additive noise $e(k)$. The task of the equalizer at the sampling instant k is to produce an estimate of the transmitted symbol $s(k-d)$ using the information contained in $x(k), x(k-1), \cdots, x(k-n+1)$, where the integers n and d are known as the order and the lag of the equalizer, respectively.

We use the geometric formulation of the equalization problem due to Chen, S., G.J. Gibson, C.F.N. Cowan and P.M. Grand [1990]. Define

$$P_{n,d}(1) = \{\hat{\mathbf{x}}(k) \in R^n | s(k-d) = 1\} \tag{14.19}$$

$$P_{n,d}(-1) = \{\hat{\mathbf{x}}(k) \in R^n | s(k-d) = -1\} \tag{14.20}$$

where

$$\hat{\mathbf{x}}(k) = [\hat{x}(k), \hat{x}(k-1), \cdots, \hat{x}(k-n+1)]^T, \tag{14.21}$$

$\hat{x}(k)$ is the noise-free output of the channel (see Fig. 14.1), and $P_{n,d}(1)$ and $P_{n,d}(-1)$ represent the two sets of possible channel noise-free output vectors $\hat{\mathbf{x}}(k)$ that can

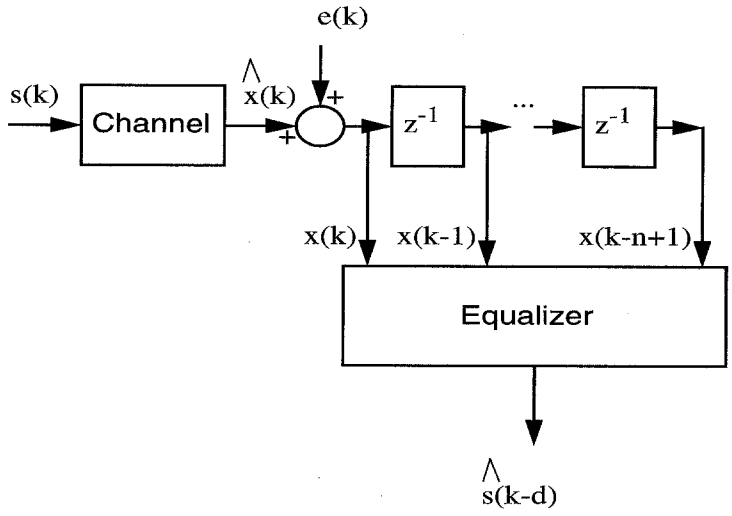


Figure 14.1. Schematic of data transmission system.

be produced from sequences of the channel inputs containing $s(k-d) = 1$ and $s(k-d) = -1$, respectively. The equalizer can be characterized by the function

$$g_k : R^n \rightarrow \{-1, 1\} \tag{14.22}$$

with

$$\hat{s}(k-d) = g_k(\mathbf{x}(k)), \tag{14.23}$$

where

$$\mathbf{x}(k) = [x(k), x(k-1), \cdots, x(k-n+1)]^T \tag{14.24}$$

is the observed channel output vector. Let $p_1[\mathbf{x}(k)|\hat{\mathbf{x}}(k) \in P_{n,d}(1)]$ and $p_{-1}[\mathbf{x}(k)|\hat{\mathbf{x}}(k) \in P_{n,d}(-1)]$ be the conditional probability density functions of $\mathbf{x}(k)$ given $\hat{\mathbf{x}}(k) \in P_{n,d}(1)$ and $\hat{\mathbf{x}}(k) \in P_{n,d}(-1)$, respectively. It was shown in Chen, S., G.J. Gibson, C.F.N. Cowan and P.M. Grand [1990] that the equalizer that is defined by

$$f_{opt}(\mathbf{x}(k)) = sgn[p_1(\mathbf{x}(k)|\hat{\mathbf{x}}(k) \in P_{n,d}(1)) - p_{-1}(\mathbf{x}(k)|\hat{\mathbf{x}}(k) \in P_{n,d}(-1))] \tag{14.25}$$

achieves the minimum bit error rate for the given order n and lag d, where $sgn(y) = 1(-1)$ if $y \geq 0$ ($y < 0$). If the noise $e(k)$ is zero-mean and Gaussian with covariance matrix

$$Q = E[(e(k), ..., e(k-n+1))(e(k), ..., e(k-n+1))^T] \tag{14.26}$$

then from $x(k) = \hat{x}(k) + e(k)$ we have that

$$\begin{aligned} & p_1[\mathbf{x}(k)|\hat{\mathbf{x}}(k) \in P_{n,d}(1)] - p_{-1}[\mathbf{x}(k)|\hat{\mathbf{x}}(k) \in P_{n,d}(-1)] \\ &= \sum exp[-\frac{1}{2}(\mathbf{x}(k) - \hat{\mathbf{x}}_+)^T Q^{-1}(\mathbf{x}(k) - \hat{\mathbf{x}}_+)] \\ &\quad - \sum exp[-\frac{1}{2}(\mathbf{x}(k) - \hat{\mathbf{x}}_-)^T Q^{-1}(\mathbf{x}(k) - \hat{\mathbf{x}}_-)] \end{aligned} \tag{14.27}$$

where the first (second) sum is over all the points $\hat{\mathbf{x}}_+ \in P_{n,d}(1)$ ($\hat{\mathbf{x}}_- \in P_{n,d}(-1)$).

Now consider the nonlinear channel

$$\hat{x}(k) = s(k) + 0.5s(k-1) - 0.9[s(k) + 0.5s(k-1)]^3 \tag{14.28}$$

and white Gaussian noise $e(k)$ with $E[e^2(k)] = 0.2$. For this case, the optimal decision region for $n = 2$ and $d = 0$,

$$[\mathbf{x}(k) \in R^2 | p_1[\mathbf{x}(k)|\hat{\mathbf{x}}(k) \in P_{2,0}(1)] - p_{-1}[\mathbf{x}(k)|\hat{\mathbf{x}}(k) \in P_{2,0}(-1)] \geq 0] \tag{14.29}$$

is shown in Fig. 14.2 as the shaded area. The elements of the sets $P_{2,0}(1)$ and $P_{2,0}(-1)$ are illustrated in Fig. 14.2 by the "o" and "*", respectively. From Fig. 14.2 we see that the optimal decision boundary for this case is severely nonlinear.

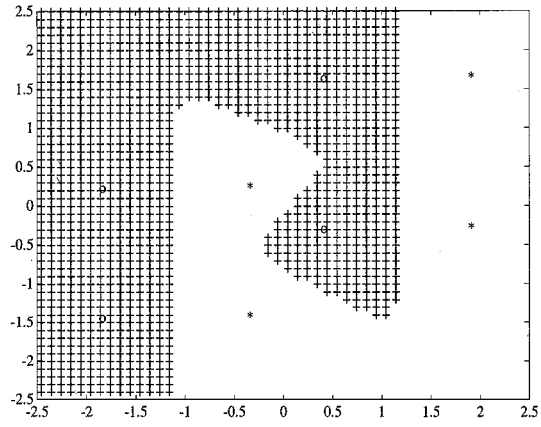


Figure 14.2. Optimal decision region for the channel (14.28), Gaussian white noise with variance $\sigma_e^2 = 0.2$, and equalizer order $n = 2$ and lag $d = 0$, where the horizontal axis denotes $x(k)$ and the vertical axis denotes $x(k-1)$.

14.3.2 Application of the Fuzzy System to the Equalization Problem

We use the fuzzy system (14.3) as the equalizer in Fig. 14.1. The operation consists of the following two phases:

- **Training Phase**: In this phase, the transmitted signal $s(k)$ is known and the task is to design the equalizer (that is, the fuzzy system (14.3)). We use the design method in Section 14.1 to specify the structure and the parameters of the equalizer. The input-output pairs for this problem are: $x_0^k = (x(k), ..., x(k-n+1))^T$ and $y_0^k = s(k-d)$ (the index p in Section 14.1 becomes the time index k here).

- **Application Phase**: In this phase, the transmitted signal $s(k)$ is unknown and the designed equalizer (the fuzzy system (14.3)) is used to estimate $s(k-d)$. Specifically, if the output of the fuzzy system is greater than or equal to zero, the estimate $\hat{s}(k-d) = 1$; otherwise, $\hat{s}(k-d) = -1$.

Example 14.1. Consider the nonlinear channel (14.28). Suppose that $n = 2$ and $d = 0$, so that the optimal decision region is shown in Fig. 14.2. Our task is to design a fuzzy system whose input-output behavior approximates that in Fig. 14.2, where the output of the fuzzy system is quantized as in the Application Phase. We use the design procedure in Section 14.1. In Step 1, we choose $N_1 = N_2 = 9$ and $\mu_{A_i^l}(x_i) = exp(-(\frac{x_i - \bar{x}_i^l}{0.3})^2)$, where $i = 1, 2$ and $\bar{x}_i^l = -2 + 0.5(l-1)$ for $l = 1, 2, ..., 9$. In Step 3, we choose the initial parameters $\theta(0)$ randomly in the interval $[-0.3, 0.3]$. In Step 4, we choose $\sigma = 0.1$. Figs.14.3-14.5 show the decision regions resulting from the designed fuzzy system when the training in Step 4 stops at $k = 30, 50$ and 100 (that is, when the p in (14.8)-(14.10) equals $30, 50$ and 100), respectively. From Figs.14.3-14.5 we see that the decision regions tend to converge to the optimal decision region as more training is performed. □

Example 14.2. In this example, we consider the same situation as in Example 14.1 except that we choose $d = 1$ rather than $d = 0$. The optimal decision region for this case is shown in Fig.14.6. Figs.14.7 and 14.8 show the decision regions resulting from the fuzzy system equalizer when the training in Step 4 stops at $k = 20$ and $k = 50$, respectively. We see, again, that the decision regions tend to converge to the optimal decision region. □

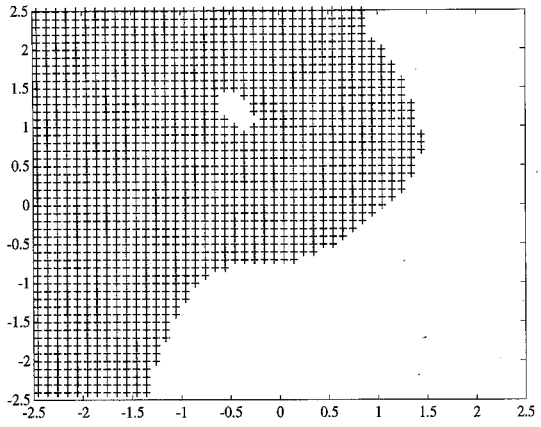


Figure 14.3. Decision region of the fuzzy system equalizer when the training stops at $k = 30$, where the horizontal axis denotes $x(k)$ and the vertical axis denotes $x(k-1)$.

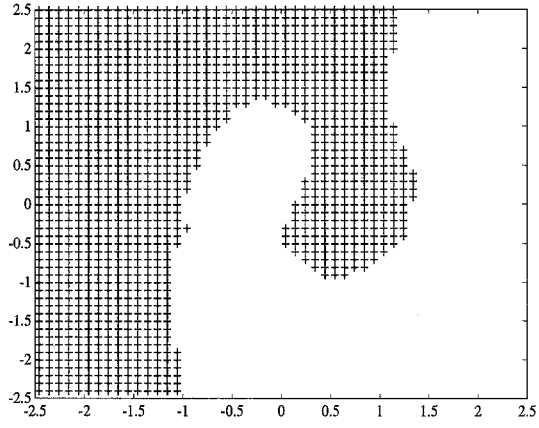


Figure 14.4. Decision region of the fuzzy system equalizer when the training stops at $k = 50$, where the horizontal axis denotes $x(k)$ and the vertical axis denotes $x(k-1)$.

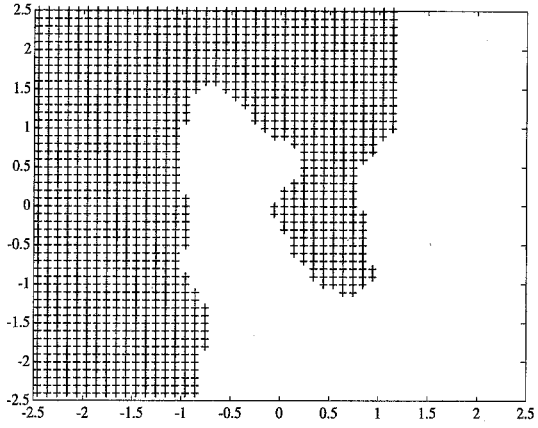


Figure 14.5. Decision region of the fuzzy system equalizer when the training stops at $k = 100$, where the horizontal axis denotes $x(k)$ and the vertical axis denotes $x(k-1)$.

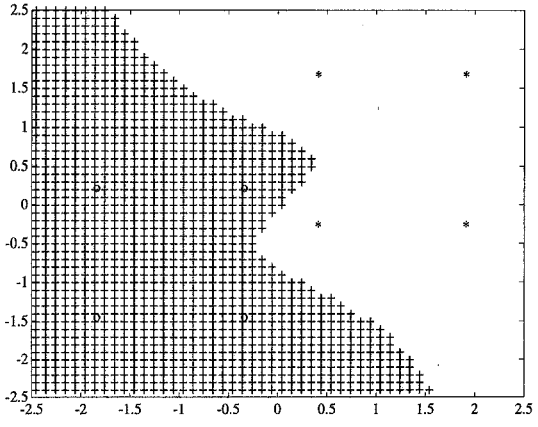


Figure 14.6. Optimal decision region for the channel (14.28), Gaussian white noise with variance $\sigma_e^2 = 0.2$, and equalizer order $n = 2$ and lag $d = 1$, where the horizontal axis denotes $x(k)$ and the vertical axis denotes $x(k-1)$.

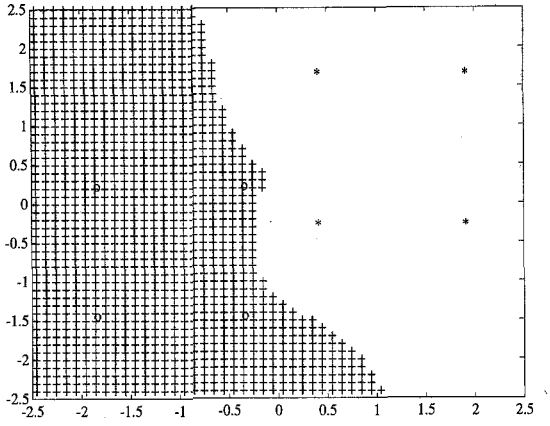


Figure 14.7. Decision region of the fuzzy system equalizer in Example 14.2 when the training stops at $k = 20$, where the horizontal axis denotes $x(k)$ and the vertical axis denotes $x(k-1)$.

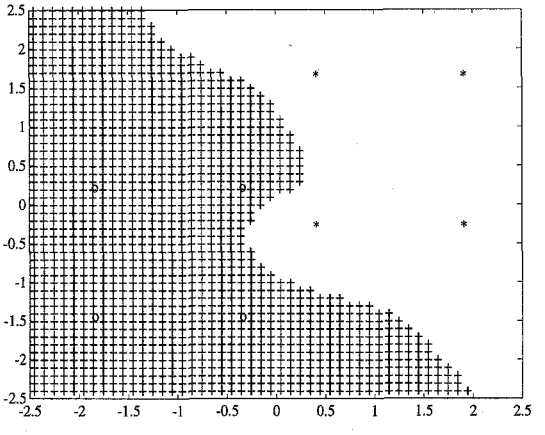


Figure 14.8. Decision region of the fuzzy system equalizer in Example 14.2 when the training stops at $k = 50$, where the horizontal axis denotes $x(k)$ and the vertical axis denotes $x(k-1)$.

14.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- Using the recursive least squares algorithm to design the parameters of the fuzzy system.
- The techniques used in the derivation of the recursive least squares algorithm.
- Formulation of the channel equalization problem as a pattern recognition problem and the application of the fuzzy system with the recursive least squares algorithm to the equalization and similar pattern recognition problems.

The recursive least squares algorithm was studied in detail in many standard textbooks on estimation theory and adaptive filters; for example, Mendel [1994] and Cowan and Grant [1985]. The method in this chapter is taken from Wang [1994a] and Wang and Mendel [1993] where more simulation results can be found. For a similar approach using neural networks, see Chen, S., G.J. Gibson, C.F.N. Cowan and P.M. Grand [1990].

14.5 Exercises and Projects

Exercise 14.1. The XoR function is defined as

x_1	x_2	$x_1 XoR x_2$
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1

(a) Design a fuzzy system $f(x_1, x_2)$ in the form of (14.3) such that $sgn[f(x_1, x_2)]$ implements the XoR function, where $sgn(f) = 1$ if $f \geq 0$ and $sgn(f) = -1$ if $f < 0$.

(b) Plot the decision region $\{x \in U | sgn[f(x)] \geq 0\}$, where $U = [-2, 2] \times [-2, 2]$ and $f(x)$ is the fuzzy system you designed in (a).

Exercise 14.2. Discuss the physical meaning of each of (14.8)-(14.10). Explain why the initial $P(0) = \sigma I$ should be large.

Exercise 14.3. Prove the matrix identity (14.15).

Exercise 14.4. Suppose that we change the criterion J_k of (14.1) to

$$J_k' = \sum_{j=1}^{k} \lambda^{k-j} [f(x_0^j) - y_0^j]^2 \tag{14.30}$$

where $\lambda \in (0, 1]$ is a forgetting factor, and that we still use the fuzzy system in the form of (14.5). Derive the recursive least squares algorithm similar to (14.8)-(14.10) for this new criterion J'_k.

Exercise 14.5. The objective to use the J'_k of (14.30) is to discount old data by putting smaller weights for them. Another way is to consider only the most recent N data pairs, that is, the criterion now is

$$J''_k = \sum_{j=k-N+1}^{k} [f(x_0^j) - y_0^j]^2 \tag{14.31}$$

Let $f(x_0^j)$ be the fuzzy system in the form of (14.5). Derive the recursive least squares algorithm similar to (14.8)-(14.10) which minimizes J''_k.

Exercise 14.6. Determine the exact locations of the sets of points: (a) $P_{2,0}(1)$ and $P_{2,0}(-1)$ in Fig. 14.2, and (b) $P_{2,1}(1)$ and $P_{2,1}(-1)$ in Fig. 14.6.

14.7 (Project). Write a computer program to implement the design method in Section 14.1 and apply your program to the nonlinear system identification problems in Chapter 13.

Chapter 15

Design of Fuzzy Systems Using Clustering

In Chapters 12-14, we proposed three methods for designing fuzzy systems. In all these methods, we did not propose a systematic procedure for determining the number of rules in the fuzzy systems. More specifically, the gradient descent method of Chapter 13 fixes the number of rules before training, while the table look-up scheme of Chapter 12 and the recursive least squares method of Chapter 14 fix the IF-part fuzzy sets, which in turn sets a bound for the number of rules. Choosing an appropriate number of rules is important in designing the fuzzy systems, because too many rules result in a complex fuzzy system that may be unnecessary for the problem, whereas too few rules produce a less powerful fuzzy system that may be insufficient to achieve the objective.

In this chapter, we view the number of rules in the fuzzy system as a design parameter and determine it based on the input-output pairs. The basic idea is to group the input-output pairs into clusters and use one rule for one cluster; that is, the number of rules equals the number of clusters. We first construct a fuzzy system that is optimal in the sense that it can match all the input-output pairs to arbitrary accuracy; this optimal fuzzy system is useful if the number of input-output pairs is small. Then, we determine clusters of the input-output pairs using the nearest neighborhood clustering algorithm, view the clusters as input-output pairs, and use the optimal fuzzy system to match them.

15.1 An Optimal Fuzzy System

Suppose that we are given N input-output pairs $(x_0^l; y_0^l), l = 1, 2, ..., N$, and N is small, say, $N = 20$. Our task is to construct a fuzzy system $f(x)$ that can match all the N pairs to any given accuracy; that is, for any given $\epsilon > 0$, we require that $|f(x_0^l) - y_0^l| < \epsilon$ for all $l = 1, 2, ..., N$.

This optimal fuzzy system is constructed as

$$f(x) = \frac{\sum_{l=1}^{N} y_0^l exp(-\frac{|x-x_0^l|^2}{\sigma^2})}{\sum_{l=1}^{N} exp(-\frac{|x-x_0^l|^2}{\sigma^2})} \tag{15.1}$$

Clearly, the fuzzy system (15.1) is constructed from the N rules in the form of (7.1) with $\mu_{A_i^l}(x_i) = exp(-\frac{|x_i-x_{0i}^l|^2}{\sigma^2})$ and the center of B^l equal to y_0^l, and using the product inference engine, singleton fuzzifier, and center average defuzzifier. The following theorem shows that by properly choosing the parameter σ, the fuzzy system (15.1) can match all the N input-output pairs to any given accuracy.

Theorem 15.1: For arbitrary $\epsilon > 0$, there exists $\sigma^* > 0$ such that the fuzzy system (15.1) with $\sigma = \sigma^*$ has the property that

$$|f(x_0^l) - y_0^l| < \epsilon \tag{15.2}$$

for all $l = 1, 2, ..., N$.

Proof: Viewing x_0^l and y_0^l as the x_0^{k+1} and y_0^{k+1} in Lemma 13.1 and using exactly the same method as in the proof of Lemma 13.1, we can prove this theorem. □

The σ is a smoothing parameter: the smaller the σ, the smaller the matching error $|f(x_0^l) - y_0^l|$, but the less smooth the $f(x)$ becomes. We know that if $f(x)$ is not smooth, it may not generalize well for the data points not in the training set. Thus, the σ should be properly chosen to provide a balance between matching and generalization. Because the σ is a one-dimensional parameter, it is usually not difficult to determine an appropriate σ for a practical problem. Sometimes, a few trial-and-error procedures may determine a good σ. As a general rule, large σ can smooth out noisy data, while small σ can make $f(x)$ as nonlinear as is required to approximate closely the training data.

The $f(x)$ is a general nonlinear regression that provides a smooth interpolation between the observed points $(x_0^l; y_0^l)$. It is well behaved even for very small σ.

Example 15.1. In this example, we would like to see the influence of the parameter σ on the smoothness and the matching accuracy of the optimal fuzzy system. We consider a simple single-input case. Suppose that we are given five input-output pairs: $(-2, 1), (-1, 0), (0, 2), (1, 2)$ and $(2, 1)$. The optimal fuzzy system $f(x)$ is in the form of (15.1) with $(x_0^l; y_0^l) = (-2, 1), (-1, 0), (0, 2), (1, 2), (2, 1)$ for $l = 1, 2, ..., 5$, respectively. Figs. 15.1-15.3 plot the $f(x)$ for $\sigma = 0.1, 0.3$ and 0.5, respectively. These plots confirm our early comment that smaller σ gives smaller matching errors but less smoothing functions. □

15.2 Design of Fuzzy Systems By Clustering

The optimal fuzzy system (15.1) uses one rule for one input-output pair, thus it is no longer a practical system if the number of input-output pairs is large. For

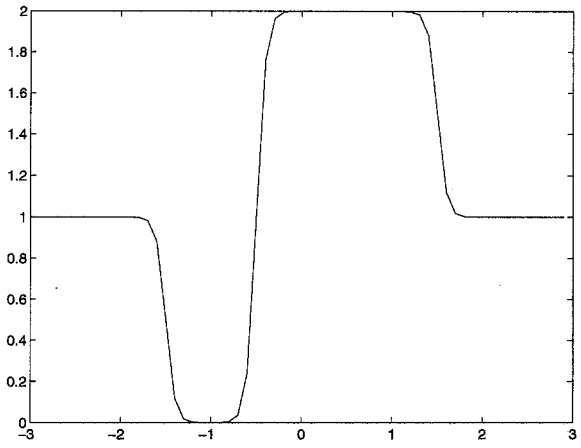


Figure 15.1. The optimal fuzzy system in Example 15.1 with $\sigma = 0.1$.

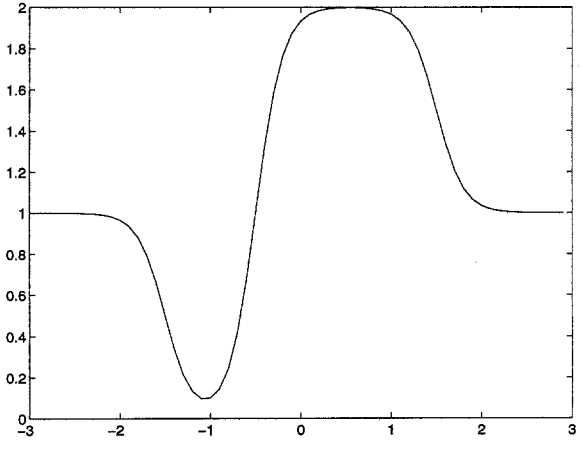


Figure 15.2. The optimal fuzzy system in Example 15.1 with $\sigma = 0.3$.

these large sample problems, various clustering techniques can be used to group the input-output pairs so that a group can be represented by one rule.

From a general conceptual point of view, clustering means partitioning of a collection of data into disjoint subsets or clusters, with the data in a cluster having

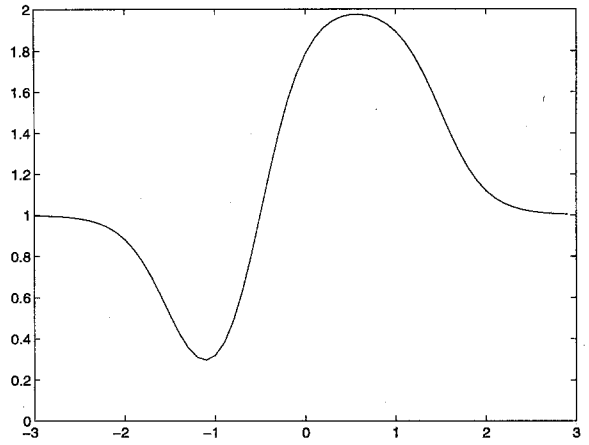


Figure 15.3. The optimal fuzzy system in Example 15.1 with $\sigma = 0.5$.

some properties that distinguish them from the data in the other clusters. For our problem, we first group the input-output pairs into clusters according to the distribution of the input points, and then use one rule for one cluster. Fig. 15.4 illustrates an example where six input-output pairs are grouped into two clusters and the two rules in the figure are used to construct the fuzzy system. The detailed algorithm is given next.

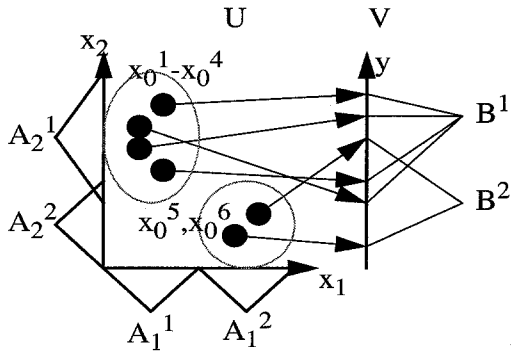
One of the simplest clustering algorithms is the nearest neighborhood clustering algorithm. In this algorithm, we first put the first datum as the center of the first cluster. Then, if the distances of a datum to the cluster centers are less than a prespecified value, put this datum into the cluster whose center is the closest to this datum; otherwise, set this datum as a new cluster center. The details are given as follows.

Design of the Fuzzy System Using Nearest Neighborhood Clustering:

- **Step 1**. Starting with the first input-output pair $(x_0^1; y_0^1)$, establish a cluster center x_c^1 at x_0^1, and set $A^1(1) = y_0^1$, $B^1(1) = 1$. Select a radius r.

- **Step 2**. Suppose that when we consider the $k'th$ input-output pair $(x_0^k; y_0^k)$, $k = 2, 3, ...$, there are M clusters with centers at $x_c^1, x_c^2, ..., x_c^M$. Compute the distances of x_0^k to these M cluster centers, $|x_0^k - x_c^l|$, $l = 1, 2, ..., M$, and let the smallest distances be $|x_0^k - x_c^{l_k}|$, that is, the nearest cluster to x_0^k is $x_c^{l_k}$. Then:

  a) If $|x_0^k - x_c^{l_k}| > r$, establish x_0^k as a new cluster center $x_c^{M+1} = x_0^k$, set



IF x_1 is A_1^1 and x_2 is A_2^1, THEN y is B^1
IF x_1 is A_1^2 and x_2 is A_2^2, THEN y is B^2

Figure 15.4. An example of constructing fuzzy IF-THEN rules from input-output pairs, where six input-output pairs $(x_0^1; y_0^1), ..., (x_0^6; y_0^6)$ are grouped into two clusters from which the two rules are generated.

$A^{M+1}(k) = y_0^k, B^{M+1}(k) = 1$, and keep $A^l(k) = A^l(k-1), B^l(k) = B^l(k-1)$ for $l = 1, 2, ..., M$.

b) If $|x_0^k - x_c^{l_k}| \leq r$, do the following:

$$A^{l_k}(k) = A^{l_k}(k-1) + y_0^k \tag{15.3}$$
$$B^{l_k}(k) = B^{l_k}(k-1) + 1 \tag{15.4}$$

and set

$$A^l(k) = A^l(k-1) \tag{15.5}$$
$$B^l(k) = B^l(k-1) \tag{15.6}$$

for $l = 1, 2, ..., M$ with $l \neq l_k$.

- **Step 3**. If x_0^k does not establish a new cluster, then the designed fuzzy system based on the k input-output pairs $(x_0^j; y_0^j), j = 1, 2, ..., k$, is

$$f_k(x) = \frac{\sum_{l=1}^{M} A^l(k) exp(-\frac{|x-x_c^l|^2}{\sigma})}{\sum_{l=1}^{M} B^l(k) exp(-\frac{|x-x_c^l|^2}{\sigma})} \tag{15.7}$$

If x_0^k establishes a new cluster, then the designed fuzzy system is

$$f_k(x) = \frac{\sum_{l=1}^{M+1} A^l(k) exp(-\frac{|x-x_c^l|^2}{\sigma})}{\sum_{l=1}^{M+1} B^l(k) exp(-\frac{|x-x_c^l|^2}{\sigma})} \tag{15.8}$$

- **Step 4**. Repeat by going to Step 2 with $k = k + 1$.

From (15.3)-(15.6) we see that the variable $B^l(k)$ equals the number of input-output pairs in the $l'th$ cluster after k input-output pairs have been used, and $A^l(k)$ equals the summation of the output values of the input-output pairs in the $l'th$ cluster. Therefore, if each input-output pair establishes a cluster center, then the designed fuzzy system (15.8) becomes the optimal fuzzy system (15.1). Because the optimal fuzzy system (15.1) can be viewed as using one rule to match one input-output pair, the fuzzy system (15.7) or (15.8) can be viewed as using one rule to match one cluster of input-output pairs. Since a new cluster may be introduced whenever a new input-output pair is used, the number of rules in the designed fuzzy system also is changing during the design process. The number of clusters (or rules) depends on the distribution of the input points in the input-output pairs and the radius r.

The radius r determines the complexity of the designed fuzzy system. For smaller r, we have more clusters, which result in a more sophisticated fuzzy system. For larger r, the designed fuzzy system is simpler but less powerful. In practice, a good radius r may be obtained by trials and errors.

Example 15.2. Consider the five input-output pairs in Example 15.1. Our task now is to design a fuzzy system using the design procedure in this section. If $r < 1$, then each of the five input-output pair establishes a cluster center and the designed fuzzy system $f_5(x)$ is the same as in Example 15.1. We now design the fuzzy system with $r = 1.5$.

In Step 1, we establish the center of the first cluster $x_c^1 = -2$ and set $A^1(1) = y_0^1 = 1, B^1(1) = 1$. In Step 2 with $k = 2$, since $|x_0^2 - x_c^{l_2}| = |x_0^2 - x_c^1| = |-1-(-2)| = 1 < r = 1.5$, we have $A^1(2) = A^1(1) + y_0^2 = 1 + 0 = 1$ and $B^1(2) = B^1(1) + 1 = 2$. For $k = 3$, since $|x_0^3 - x_c^{l_3}| = |x_0^3 - x_c^1| = |0-(-2)| = 2 > r$, we establish a new cluster center $x_c^2 = x^3 = 0$ together with $A^2(3) = y_0^3 = 2$ and $B^2(3) = 1$. The A^1 and B^1 remain the same, that is, $A^1(3) = A^1(2) = 1$ and $B^1(3) = B^1(2) = 2$. For $k = 4$, since $|x_0^4 - x_c^{l_4}| = |x_0^4 - x_c^2| = |1 - 0| = 1 < r$, we have $A^2(4) = A^2(3) + y_0^4 = 2 + 2 = 4, B^2(4) = B^2(3) + 1 = 2, A^1(4) = A^1(3) = 1$ and $B^1(4) = B^1(3) = 2$. Finally, for $k = 5$, since $|x_0^5 - x_c^{l_5}| = |x_0^5 - x_c^2| = |2-0| = 2 > r$, a new cluster center $x_c^3 = x_0^5 = 2$ is established with $A^3(5) = y_0^5 = 1$ and $B^3(5) = 1$. The other variables remain the same, that is, $A^1(5) = A^1(4) = 1, B^1(5) = B^1(4) = 2, A^2(5) = A^2(4) = 4$ and $B^2(5) = B^2(4) = 2$. The final fuzzy system is

$$f_5(x) = \frac{\sum_{l=1}^{3} A^l(5) exp(-\frac{(x-x_c^l)^2}{\sigma})}{\sum_{l=1}^{3} B^l(5) exp(-\frac{(x-x_c^l)^2}{\sigma})}$$

$$= \frac{exp(-\frac{(x+2)^2}{\sigma}) + 4exp(-\frac{x^2}{\sigma}) + exp(-\frac{(x-2)^2}{\sigma})}{2exp(-\frac{(x+2)^2}{\sigma}) + 2exp(-\frac{x^2}{\sigma}) + exp(-\frac{(x-2)^2}{\sigma})} \tag{15.9}$$

which is plotted in Fig. 15.5 with $\sigma = 0.3$. Comparing Figs. 15.5 with 15.2 we see, as expected, that the matching errors of the fuzzy system (15.9) at the five input-output pairs are larger than those of the optimal fuzzy system. □

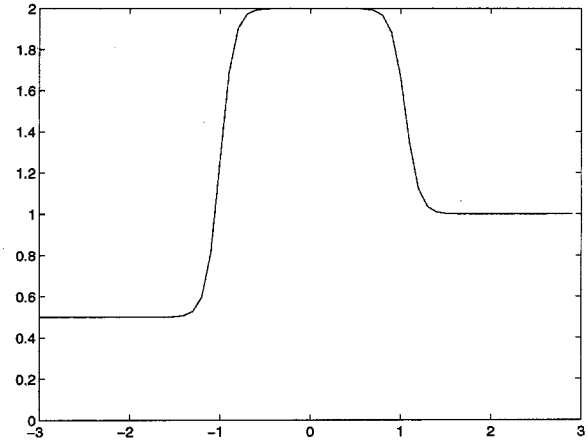


Figure 15.5. The designed fuzzy system $f_5(x)$ (15.9) in Example 15.2 with $\sigma = 0.3$.

Since the $A^l(k)$ and $B^l(k)$ coefficients in (15.7) and (15.8) are determined using the recursive equations (15.3)-(15.6), it is easy to add a forgetting factor to (15.3)-(15.6). This is desirable if the fuzzy system is being used to model systems with changing characteristics. For these cases, we replace (15.3) and (15.4) with

$$A^{l_k}(k) = \frac{\tau - 1}{\tau} A^{l_k}(k-1) + \frac{1}{\tau} y_0^k \tag{15.10}$$

$$B^{l_k}(k) = \frac{\tau - 1}{\tau} B^{l_k}(k-1) + \frac{1}{\tau} \tag{15.11}$$

and replace (15.5) and (15.6) with

$$A^l(k) = \frac{\tau - 1}{\tau} A^l(k-1) \tag{15.12}$$

$$B^l(k) = \frac{\tau - 1}{\tau} B^l(k-1) \tag{15.13}$$

where τ can be considered as a time constant of an exponential decay function. For practical considerations, there should be a lower threshold for $B^l(k)$ so that when sufficient time has elapsed without update for a particular cluster (which results in the $B^l(k)$ to be smaller than the threshold), that cluster would be eliminated.

15.3 Application to Adaptive Control of Nonlinear Systems

In this section, we use the designed fuzzy system (15.7) or (15.8) as a basic component to construct adaptive fuzzy controllers for discrete-time nonlinear dynamic systems. We consider two examples, but the approach can be generalized to other cases.

Example 15.3. Consider the discrete-time nonlinear system described by the difference equation

$$y(k+1) = g[y(k), y(k-1)] + u(k) \tag{15.14}$$

where the nonlinear function

$$g[y(k), y(k-1)] = \frac{y(k)y(k-1)[y(k)+2.5]}{1+y^2(k)+y^2(k-1)} \tag{15.15}$$

is assumed to be unknown. The objective here is to design a controller $u(k)$ (based on the fuzzy system (15.7) or (15.8)) such that the output $y(k)$ of the closed-loop system follows the output $y_m(k)$ of the reference model

$$y_m(k+1) = 0.6y_m(k) + 0.2y_m(k-1) + r(k) \tag{15.16}$$

where $r(k) = sin(2\pi k/25)$. That is, we want $e(k) = y(k) - y_m(k)$ converge to zero as k goes to infinity.

If the function $g[y(k), y(k-1)]$ is known, we can construct the controller as

$$u(k) = -g[y(k), y(k-1)] + 0.6y(k) + 0.2y(k-1) + r(k) \tag{15.17}$$

which, when applied to (15.14), results in

$$y(k+1) = 0.6y(k) + 0.2y(k-1) + r(k) \tag{15.18}$$

Combining (15.16) and (15.18), we have

$$e(k+1) = 0.6e(k) + 0.2e(k-1) \tag{15.19}$$

from which it follows that $lim_{k\to\infty} e(k) = 0$. However, since $g[y(k), y(k-1)]$ is unknown, the controller (15.17) cannot be implemented. To solve this problem, we replace the $g[y(k), y(k-1)]$ in (15.17) by the fuzzy system (15.7) or (15.8); that is, we use the following controller

$$u(k) = -f_k[y(k), y(k-1)] + 0.6y(k) + 0.2y(k-1) + r(k) \tag{15.20}$$

where $f_k[y(k), y(k-1)]$ is in the form of (15.7) or (15.8) with $x = (y(k), y(k-1))^T$. This results in the nonlinear difference equation

$$y(k+1) = g[y(k), y(k-1)] - f_k[y(k), y(k-1)] + 0.6y(k) + 0.2y(k-1) + r(k) \tag{15.21}$$

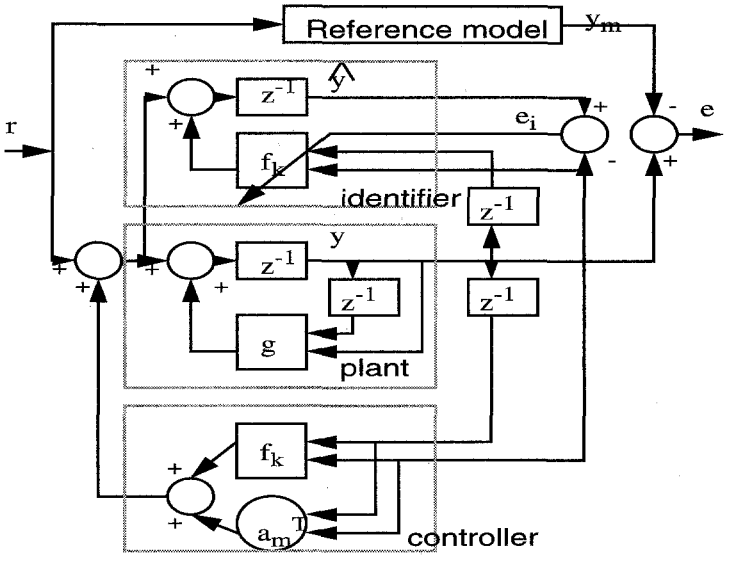


Figure 15.6. Overall adaptive fuzzy control system for Example 15.3.

governing the behavior of the closed-loop system. The overall control system is shown in Fig. 15.6. From Fig. 15.6 we see that the controller consists of two parts: an identifier and a controller. The identifier uses the fuzzy system f_k to approximate the unknown nonlinear function g, and this f_k is then copied to the controller.

We simulated the following two cases for this example:

- *Case 1*: The controller in Fig. 15.6 was first disconnected and only the identifier was operating to identify the unknown plant. In this identification phase, we chose the input $u(k)$ to be an i.i.d. random signal uniformly distributed in the interval $[-3, 3]$. After the identification procedure was terminated, (15.20) was used to generate the control input; that is, the controller in Fig. 15.6 began operating with f_k copied from the final f_k in the identifier. Figs.15.7 and 15.8 show the output $y(k)$ of the closed-loop system with this controller together with the reference model output $y_m(k)$ for the cases where the identification procedure was terminated at $k = 100$ and $k = 500$, respectively. In these simulations, we chose $\sigma = 0.3$ and $r = 0.3$. From these simulation results we see that: (i) with only 100 steps of training the identifier could produce an accurate model that resulted in a good tracking performance, and (ii) with more steps of training the control performance was improved.

- *Case 2*: The identifier and the controller operated simultaneously (as shown

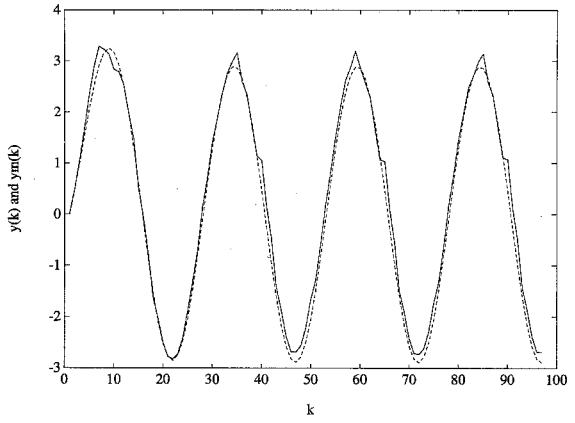


Figure 15.7. The output $y(k)$ (solid line) of the closed-loop system and the reference trajectory $y_m(k)$ (dashed line) for Case 1 in Example 15.3 when the identification procedure was terminated at $k = 100$.

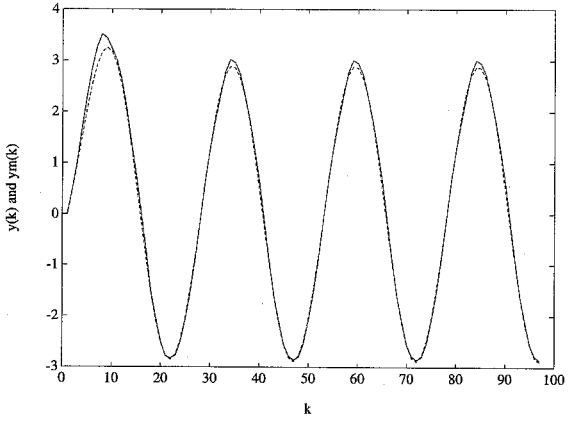


Figure 15.8. The output $y(k)$ (solid line) of the closed-loop system and the reference trajectory $y_m(k)$ (dashed line) for Case 1 in Example 15.3 when the identification procedure was terminated at $k = 500$.

in Fig. 15.6) from $k = 0$. We still chose $\sigma = 0.3$ and $r = 0.3$. Fig. 15.9 shows $y(k)$ and $y_m(k)$ for this simulation. □

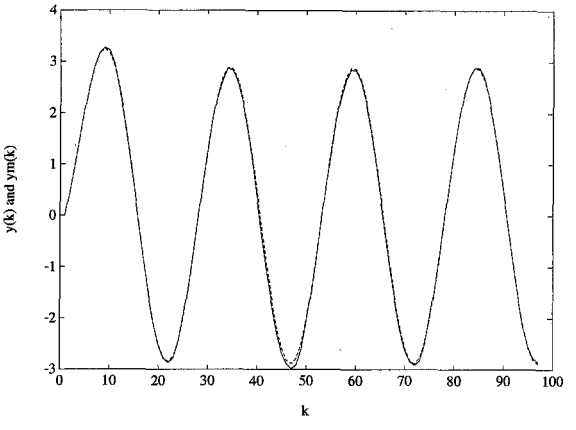


Figure 15.9. The output $y(k)$ (solid line) of the closed-loop system and the reference trajectory $y_m(k)$ (dashed line) for Case 2 in Example 15.3.

Example 15.4. In this example we consider the plant

$$y(k+1) = \frac{5y(k)y(k-1)}{1+y^2(k)+y^2(k-1)+y^2(k-2)} + u(k) + 0.8u(k-1) \qquad (15.22)$$

where the nonlinear function is assumed to be unknown. The aim is to design a controller $u(k)$ such that $y(k)$ will follow the reference model

$$y_m(k+1) = 0.32y_m(k) + 0.64y_m(k-1) - 0.5y_m(k-2) + sin(2\pi k/25) \qquad (15.23)$$

Using the same idea as in Example 15.3, we choose

$$u(k) = -f_k[y(k), y(k-1), y(k-2)] - 0.8u(k-1) + 0.32y(k) + 0.64y(k-1) \\ -0.5y(k-2) + sin(2\pi k/25) \qquad (15.24)$$

where $f_k[y(k), y(k-1), y(k-2)]$ is in the form of (15.7) or (15.8). The basic configuration of the overall control scheme is the same as Fig.15.6. Fig.15.10 shows $y(k)$ and $y_m(k)$ when both the identifier and the controller began operating from $k = 0$. We chose $\sigma = 0.3$ and $r = 0.3$ in this simulation. □

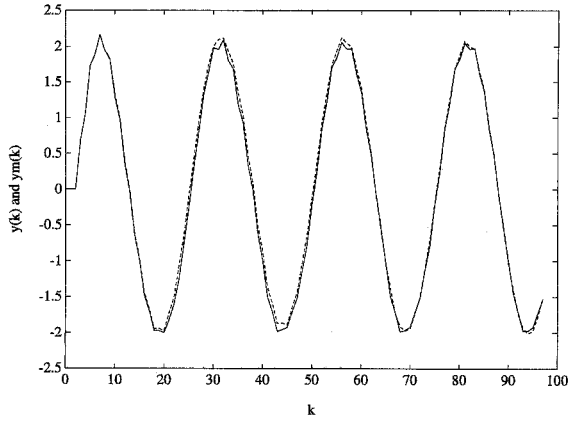


Figure 15.10. The output $y(k)$ (solid line) of the closed-loop system and the reference trajectory $y_m(k)$ (dashed line) for Example 15.4.

15.4 Summary and Further Readings

In this chapter we have demonstrated the following:

- The idea and construction of the optimal fuzzy system.
- The detailed steps of using the nearest neighborhood clustering algorithm to design the fuzzy systems from input-output pairs.
- Applications of the designed fuzzy system to the adaptive control of discrete-time dynamic systems and other problems.

Various clustering algorithms can be found in the textbooks on pattern recognition, among which Duda and Hart [1973] is still one of the best. The method in this chapter is taken from Wang [1994a], where more examples can be found.

15.5 Exercises and Projects

Exercise 15.1. Repeat Example 15.2 with $r = 2.2$.

Exercise 15.2. Modify the design method in Section 15.2 such that a cluster center is the average of inputs of the points in the cluster, the $A^l(k)$ parameter is the average of outputs of the points in the cluster, and the $B^l(k)$ parameter is deleted.

Exercise 15.3. Create an example to show that even with the same set of input-output pairs, the clustering method in Section 15.2 may create different fuzzy systems if the ordering of the input-output pairs used is different.

Exercise 15.4. The basic idea of *hierarchical clustering* is illustrated in Fig. 15.11. Propose a method to design fuzzy systems using the hierarchical clustering idea. Show your method in detail in a step-by-step manner and demonstrate it through a simple example.

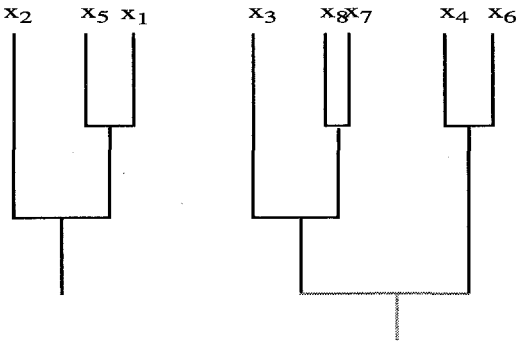


Figure 15.11. The basic idea of hierarchical clustering.

Exercise 15.5. Consider the two-input-two-output system

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} \frac{y_1(k)}{1+y_2^2(k)} \\ \frac{y_1(k)y_2(k)}{1+y_2^2(k)} \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{15.25}$$

where the nonlinear functions are assumed to be unknown.

(a) Design an identifier for the system using the fuzzy system (15.7) or (15.8) as basic block. Explain the working procedure of the identifier.

(b) Design a controller for the system such that the closed-loop system outputs follow the reference model

$$\begin{bmatrix} y_{m1}(k+1) \\ y_{m2}(k+1) \end{bmatrix} = \begin{bmatrix} y_{m1}(k) + y_{m2}(k) \\ y_{m2}(k) \end{bmatrix} + \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix} \tag{15.26}$$

where $r_1(k)$ and $r_2(k)$ are known reference signals. Under what conditions will the tracking error converge to zero?

15.6 (Project). Write a computer program to implement the design method in Section 15.2 and apply your program to the time series prediction and nonlinear system identification problems in Chapters 12 and 13.