6 Introduction to Fuzzy Logic

John K. Williams

6.1 Introduction

In the early 1990s, scientists at the National Center for Atmospheric Research were asked to help the Federal Aviation Administration objectively determine which flight service stations throughout the United States handled the most hazardous weather conditions, and hence should be spared from congressional budget cuts. They had access to 15 years of meteorological data from each location, including winds, temperature, fog, rain, and snow at 1-min intervals, as well as information about the air traffic density and a number of other factors. However, the level of aviation hazard was not indicated by any single statistic, but by the nature, frequency and duration of the conditions and their combinations. How could the stations be ranked in a reasonable, objective way?

The scientists began by surveying a group of subject domain experts – pilots, meteorologists, and airline dispatchers – quizzing them on what factors, or combinations of factors, they considered most dangerous. Using the results of these surveys along with the repository of historical data, they then computed a hazard score for each flight service station, and ranked them. When the final report was presented to the group of experts, they opened it and began to laugh – everyone agreed that the stations with the most hazardous weather were at the top of the list. Unwittingly, the NCAR scientists had created a fuzzy logic algorithm, efficiently encoding the experts' knowledge in a set of rules that reproduced their approach to assessing the level of hazard presented by each unique set of weather conditions.

The purpose of artificial intelligence algorithms, broadly stated, is to perform tasks in a way that mimics human intelligence. Many of the approaches to creating artificial intelligence algorithms presented in this book are *machine learning* algorithms that automatically recognize patterns or functional relationships in data. By training on a historical dataset or incrementally adapting as new examples are gathered, these approaches mimic a human's ability to draw inferences from data. Of course, a machine doesn't truly "learn" as a human does; rather, it uncovers statistical relationships. To achieve meaningful results, a human expert is still required to select an appropriate machine learning technique for a given problem, determine a representative data set, design data preprocessing or feature representation, set the evaluation criteria, and experiment with algorithm parameters to obtain optimal learning behavior.

Fuzzy logic presents an alternative approach to artificial intelligence by providing a framework for imitating a human expert's approach to solving a particular problem. This approach falls under the class of artificial intelligence algorithms called "expert systems" that encode expert knowledge as a set of heuristics, or rules. For instance, a medical expert system might help diagnose a disease based on a patient's answers to a series of questions. With each answer, the number of likely possibilities would diminish until finally the most probable diagnosis was determined. A key motivation for developing fuzzy logic expert systems is the recognition that human experts rarely make decisions based on a sequence of well-defined logical

John K. Williams (✉)
Research Applications Laboratory
National Center for Atmospheric Research
P.O. Box 3000, Boulder, CO 80307, USA
Phone: 303-497-2822; Fax: 303-497-8401;
Email: jkwillia@ucar.edu

statements. For instance, many questions that might be asked by a doctor diagnosing an illness may not have clear or definitive answers, and even the questions themselves might be ambiguous. Rather, humans often draw inferences from a preponderance of the evidence available to them – even if it is incomplete – with each source of information weighed according to its reliability and the strength of its connection to the outcome being evaluated.

A distinctive contribution of fuzzy logic is that it provides a conceptual framework for representing natural language concepts and reasoning by expanding on the traditional notion of a *set*, or collection of objects. For instance, the noun "chair" is in a sense equivalent to the set of objects identified as chairs. The set of green chairs is the *intersection* of the set of objects that are chairs and objects that are green. A proposition like "all chairs have legs" is a claim that the set of chairs is a *subset* of the set of objects having legs. The "fuzzy" part of fuzzy logic arises from a recognition that the sets of objects referred to by words are not "crisp" like the sets of traditional mathematics in which a given object is either in a particular set or outside it. Rather, in everyday language – and, by extension, in human reasoning – concepts often have "fuzzy" boundaries. For instance, take the concept of being a chair. Do thrones, or benches, or bean bags, or step stools, or tree stumps count as "chairs"? In traditional logic, each of these objects is either a chair or it isn't. In fuzzy logic, on the other hand, set membership is allowed to be partial. Thus, on a scale of 0 to 1, we might say that a throne is a chair to the degree 0.9, while a tree stump might be a chair to the degree 0.1. The proposition "all chairs have legs" can then be the understood as applying to different objects only to the degree that they are members in the set of "chairs". Reasoning with fuzzy logic accommodates and even exploits ambiguity, a feature that turns out to be immensely powerful for expert systems.

By formally accommodating the ambiguity of natural language, fuzzy logic makes it possible to encode human knowledge into relationships between concepts represented as fuzzy sets. Reasoning involves logical manipulation of these statements, and multiple lines of reasoning may be aggregated to form a conclusion. Thus, fuzzy logic can be viewed as the extension of classical logic to fuzzy sets and their manipulation. In addition, in many research communities the term "fuzzy logic" has come to be applied more broadly to the theory and practice of computing with fuzzy sets, or indeed any expert system or set of heuristics that preserves the uncertainty or ambiguity in data until a final "decision point." Fuzzy logic systems have become increasingly popular, both in industrial and environmental science applications, because they use information efficiently, are relatively simple to design and implement, and often perform impressively well. They do not require training data, models, or conditional probability distributions; they are robust to uncertain, missing and corrupted data; they naturally constrain the possible output to "reasonable" values; and they often have a "common sense" structure that makes them relatively easy to interpret and modify. While they may not always produce the optimal solution, fuzzy logic algorithms fall within the increasingly popular domain of "soft computing" methods that produce very good, practical and relatively inexpensive solutions to many problems.

This chapter presents a brief history of the development of fuzzy logic; describes fuzzy sets, fuzzy set theory, fuzzy numbers and fuzzy logical operations; discusses two types of fuzzy inference and a couple of variants; presents a fuzzy clustering method; and explains how these pieces can be used to create fuzzy logic algorithms that may be "tuned" using empirical data if desired. The field of fuzzy logic theory and techniques is very broad, though, and this short chapter is only able to introduce a brief overview of elements that have been found most useful to the author. Interested readers are encouraged to consult the texts by Chi et al. (1996), Klir and Folger (1988), McNiell and Freiberger (1993), Tanaka (1996), or Zimmerman (1996) and the collections of original papers in Klir and Yuan (1996) or Yager et al. (1987) for a more thorough exposition.

6.2 A Brief History

Fuzzy logic owes its existence to Lotfi Zadeh, who in 1965 published an article entitled "Fuzzy Sets" in a journal he edited, *Information and Control*, after having it rejected from several other journals (Zadeh 1965). Faced with skepticism from his contemporaries, Zadeh struggled for years to gain the recognition that his revolutionary ideas deserved. However, some researchers were quick to see the practical utility of his work. In 1973, Ebrahim Mamdami and Sedrak Assilian successfully used fuzzy logic to build a controller for a steam engine. In 1978, Laritz Peter Holmblad and Jens-Jorgen Østergaard developed and commercialized a fuzzy logic controller for industrial cement kilns. And in 1979, Hans Berliner's BKG 9.8 program beat the backgammon world champion by the convincing score of 7-1. The use of fuzzy logic developed particularly rapidly in Japan, where a subway system developed by Hitachi using predictive fuzzy controllers entered service in Sendai in 1987. These early successes were followed by an explosion of applications in consumer electronics and appliances, finance, and industrial process control, first in Asia and eventually also in Europe and the United States.

The value of fuzzy logic algorithms in the environmental sciences gained recognition beginning in the early 1990s. In 1993, MIT Lincoln Laboratory developed a fuzzy logic algorithm for detecting gust fronts from Doppler radar data (Delanoy and Troxel 1993). In 1994, researchers at the National Center for Atmospheric Research created an algorithm for detecting microbursts – wind events associated with thunderstorms that had been responsible for a number of airplane crashes – based on Doppler radar data (Albo 1994, 1996; see also Merritt 1991). Fuzzy logic was soon employed in algorithms for processing sensor data, recognizing hazardous weather conditions, producing short-range forecasts, and providing weather decision support information to aviation, state departments of transportation, and other users. By the early 21st century, a large number of the decision support systems produced by NCAR's Research Applications Laboratory incorporated fuzzy logic in one way or another. Some of these applications are described in detail in Chapter 17.

6.3 Classical and Fuzzy Sets

A fundamental concept of fuzzy logic is that of the fuzzy set. As mentioned earlier, a fuzzy set is an extension of the classical notion of a set: whereas a classical set divides the universe of objects into two distinct categories, those in the set and those outside it, a fuzzy set permits intermediate degrees of membership. More formally, every classical set is determined by its *characteristic function*, a mapping that assigns every object in the set a membership value of 1, and every object outside it a value of 0. For instance, the set of "tall people" might have a characteristic function that assigns 0 to all people having heights less than 5′ 9″ and 1 to those with height 5′ 9″ or greater. The equivalent concept for fuzzy sets is that of a *membership function*, which assigns every object a value between 0 and 1 representing its *degree* of membership in a set. The membership function for the fuzzy set of "tall people" might be 0 for people less than 5′ 4″ tall, then rise gradually to 0.5 for people 5′ 9″ tall, and then continue to rise to a value of 1 for people 6′ 3″ tall or taller. Most of us would agree that this representation more accurately represents the concept of being a "tall person": rather than enforcing a discontinuous cutoff at some particular height, the fuzzy set quantifies the ambiguity (Fig. 6.1).

The reader may have noticed that the concept of fuzzy membership is reminiscent of probability theory,

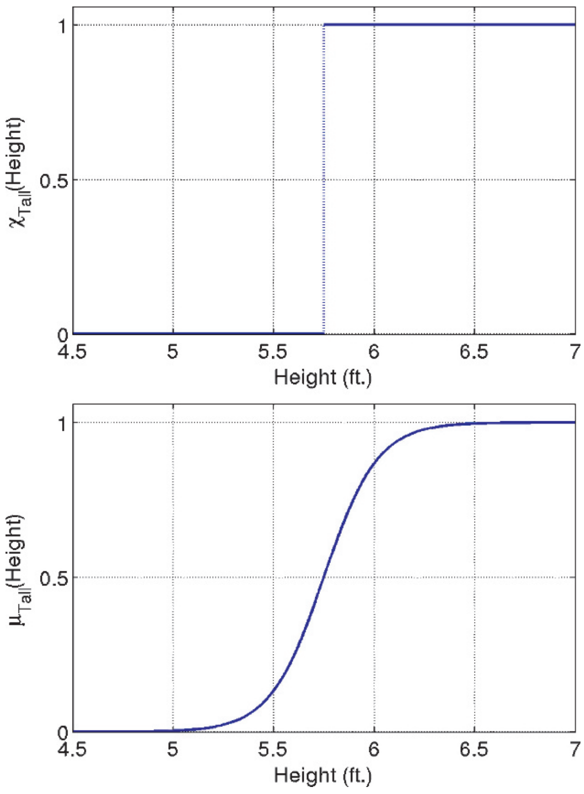


Fig. 6.1 (Top) A classical set characteristic function (0 or 1) for "tall" as a function of a person's height. (Bottom) A fuzzy set membership function representing continuous "degrees of tallness" as a function of height.

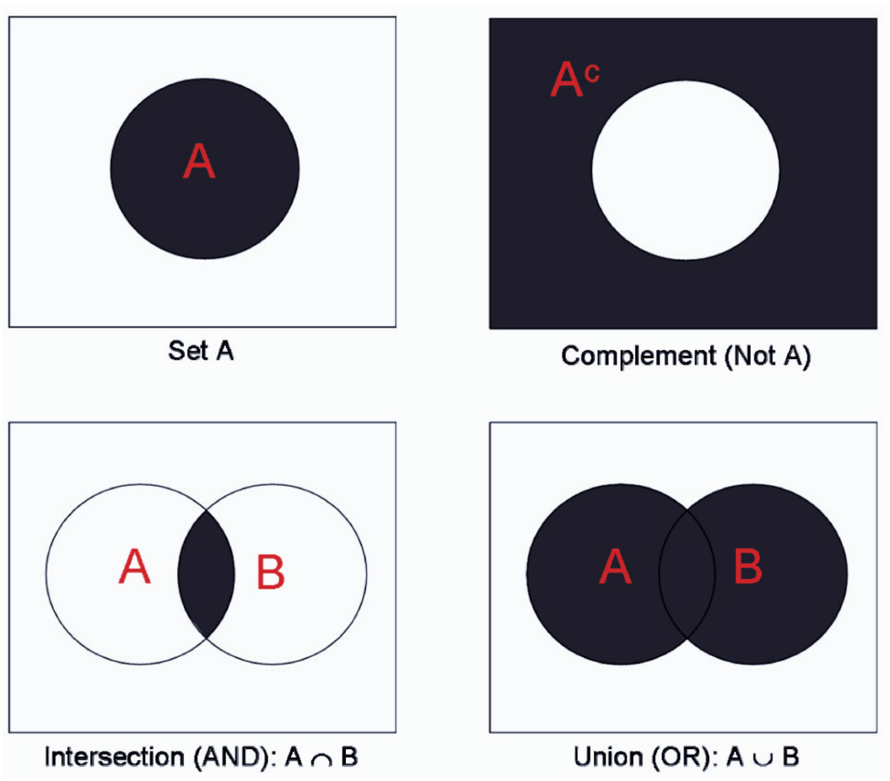


Fig. 6.2 Depiction of the classical set operations complement, intersection and union. Black shading represents elements in the resultant set.

in which "events" are assigned values between 0 and 1 representing their likelihood. Both fuzzy memberships and probabilities can be interpreted as "measures" and manipulated mathematically. However, probabilities and fuzzy set memberships are really quite different concepts. While probabilities capture summary information about random events – for instance, the likelihood that a person chosen randomly from a crowd will have a height above 5′ 9″ – a fuzzy membership captures the ambiguity in a concept itself, e.g., what we mean by the very notion of "being tall".

Classical set theory defines several important operations on sets that are useful in formal logic. For instance, the *complement* of a set A, denoted A^c, is the set of all objects not in A (Fig. 6.2). The characteristic function for A^c is the binary opposite of that for A; that is, when the characteristic function for A is 1, the characteristic function for A^c is 0, and vice versa. Writing the characteristic function for A in the traditional fashion as χ_A (here χ is the Greek letter "Chi") and that of A^c as χ_{A^c}, this relationship can be expressed in an equation as $\chi_{A^c} = 1 - \chi_A$. (Note that this is a *functional* equation, that is, one that involves a relationship between functions rather than numbers. It is equivalent to saying that the functions χ_{A^c} and χ_A share the same domain, and that $\chi_{A^c}(a) = 1 - \chi_A(a)$ for all elements a in that domain.) As an example, suppose A is the classical set of "tall people" as defined in the previous paragraph, and suppose Peter is 5′ 8″. Then $\chi_A(\text{Peter}) = 0$ and $\chi_{A^c}(\text{Peter}) = 1 - \chi_A(\text{Peter}) = 1$. The complement of "tall people" is the set of people who are not tall – a set that includes Peter.

In fuzzy set theory, the binary characteristic function is replaced by a continuous *membership function* that can take on any value between 0 and 1 (Fig. 6.3). Writing the membership function for the fuzzy set B as μ_B (μ is the Greek letter "Mu") and that of its fuzzy complement B^c as μ_{B^c}, then $\mu_{B^c} = 1 - \mu_B$. Since fuzzy sets and membership functions are equivalent, this functional equation *defines* the fuzzy set complement. Now suppose that the fuzzy set B of "tall people" is defined so that $\mu_B(\text{Peter}) = 0.4$. Then $\chi_{B^c}(\text{Peter}) = 1 - \chi_B(\text{Peter}) = 1 - 0.4 = 0.6$. In words, one could say that Peter is a member of "tall people" to the degree 0.4, and a member of "not tall people" to degree 0.6. As another

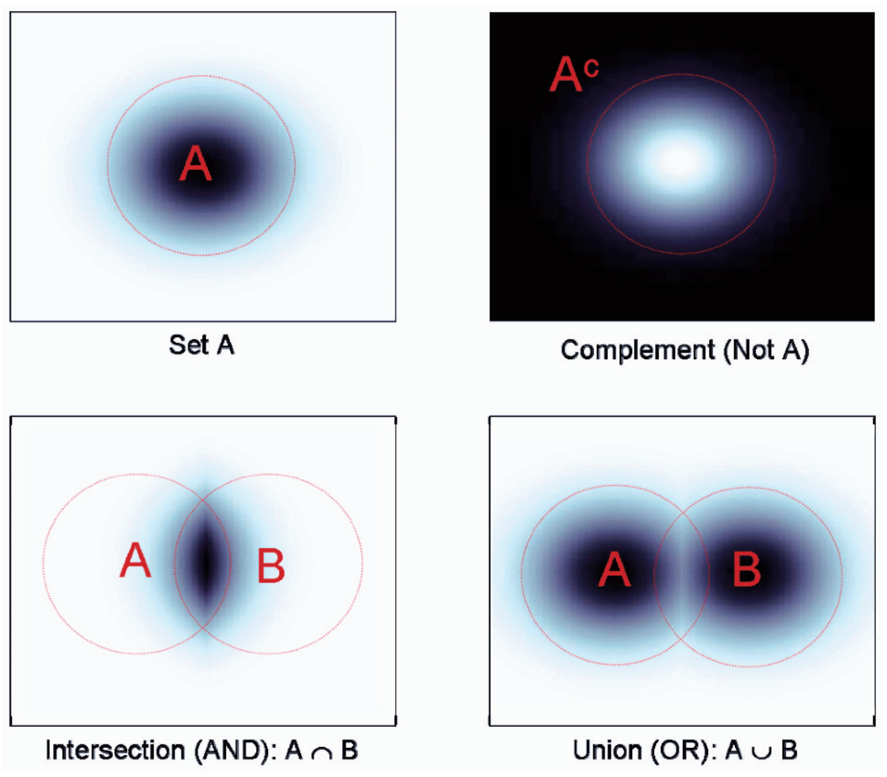
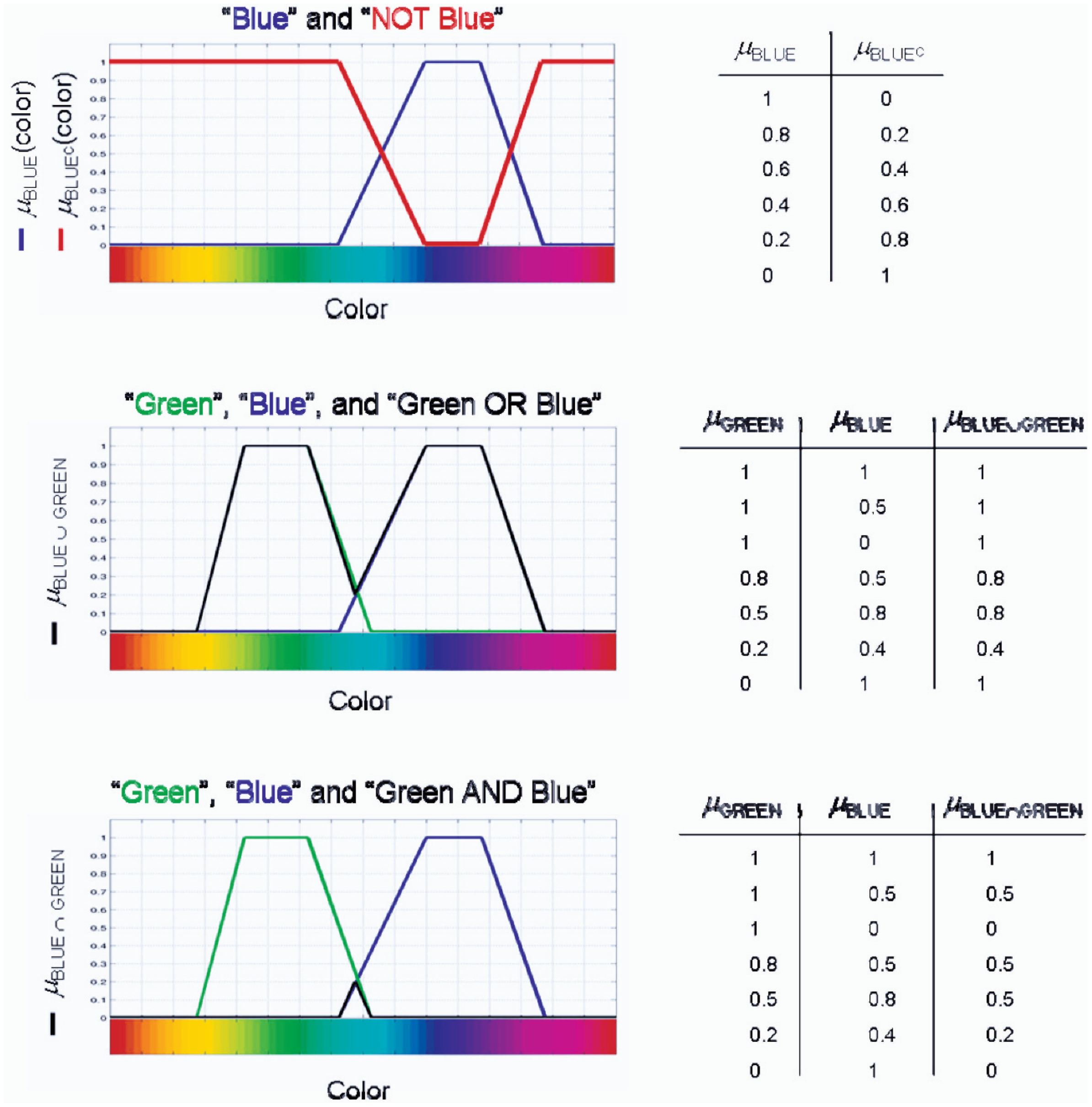


Fig. 6.3 Depictions of fuzzy set operations. The degree of membership in the resultant set is represented by shades of gray, with white points having membership 0 and black points having membership 1. Dotted red circles represent the 0.1 membership contours of the original fuzzy sets.

example, consider colors. When an object is identified as "blue", there is really quite a range of frequency values on the electromagnetic spectrum that could be referred to, ranging from blue-green to almost purple. Figure 6.4 (top) illustrates membership functions for the concept "blue" and its complement, "not blue".

Other elements of classical set theory have similarly natural analogues for fuzzy sets. The *union* of two classical sets A and B, written $A \cup B$, is the set of all elements in either set A or set B, or both. In terms of the characteristic function, $\chi_{A \cup B} = \max(\chi_A, \chi_B)$; that is, $\chi_{A \cup B} = 1$ when either $\chi_A = 1$ or $\chi_B = 1$ (or both). For fuzzy sets A and B, the union is defined by the membership function equation $\mu_{A \cup B} = \max(\mu_A, \mu_B)$, as illustrated using the color concepts "blue" and "green" in Fig. 6.4 (middle). The *intersection* of two classical sets A and B, written $A \cap B$, is the set of all objects in both sets A and set B. In terms of the characteristic function, $\chi_{A \cap B} = \min(\chi_A, \chi_B)$; that is, χ_A and χ_B must both be 1 for $\chi_{A \cap B}$ to be 1. The intersection of two fuzzy sets A and B is defined by the membership function equation $\mu_{A \cap B} = \min(\mu_A, \ \mu_B)$. This is illustrated in Fig. 6.4 (bottom). For two classical sets A and B, we say that A is a *subset* of B and write $A \subset B$ if all objects (or *elements*) of A are also in B. That is the same as saying that the characteristic function of A can be 1 only when the characteristic function of B is 1, that is, $\chi_A \le \chi_B$. Similarly, for fuzzy sets A and B, A is a *subset* of B if all elements of A are also in B to at least an equal degree, that is, if $\mu_A \le \mu_B$ (see Fig. 6.5).

Several consequences of these definitions that hold for classical sets also hold for fuzzy sets. For instance, if $A \subset B$ and $B \subset C$, then $A \subset C$ since $\mu_A \le \mu_B$ and $\mu_B \le \mu_C$ implies $\mu_A \le \mu_C$. And if $A \subset B$ and $B \subset A$, then $A = B$ because $\mu_A \le \mu_B$ and $\mu_B \le \mu_A$ implies $\mu_A = \mu_B$. The reader may similarly verify that many properties of classical set operations also hold true for fuzzy sets, including idempotency: $A \cup A = A$ and $A \cap A = A$, commutativity: $A \cup B = B \cup A$ and $A \cap B = B \cap A$, associativity: $(A \cup B) \cup C = A \cup (B \cup C)$ and $(A \cap B) \cap C = A \cap (B \cap C)$, double negation: $(A^c)^c = A$, and De Morgan's Laws: $(A \cup B)^c = A^c \cap B^c$ and $(A \cap B)^c = A^c \cup B^c$. However, the law of contradiction, $A \cap A^c = \emptyset$ (where $\emptyset$ is the empty set, having membership function 0 everywhere)



μ_{BLUE}	μ_{BLUE^c}
1	0
0.8	0.2
0.6	0.4
0.4	0.6
0.2	0.8
0	1

μ_{GREEN}	μ_{BLUE}	$\mu_{BLUE \cup GREEN}$
1	1	1
1	0.5	1
1	0	1
0.8	0.5	0.8
0.5	0.8	0.8
0.2	0.4	0.4
0	1	1

μ_{GREEN}	μ_{BLUE}	$\mu_{BLUE \cap GREEN}$
1	1	1
1	0.5	0.5
1	0	0
0.8	0.5	0.5
0.5	0.8	0.5
0.2	0.4	0.2
0	1	0

Fig. 6.4 An illustration of the fuzzy set operations complement, union and intersection using color concepts. The tables to the right provide numerical examples for several sample membership function values. Note that for the binary (0 or 1) cases, the fuzzy set operations produce the same results as those used in classical logic.

does *not* hold for fuzzy sets since an element may have nonzero membership in both a fuzzy set and its complement.

The logical operations of complement (not), intersection (and), and union (or) have been defined above by a mathematical operation on the sets' membership functions, as "one minus", "min", and "max", respectively. This is intriguing, since it suggests that other mathematical operations on membership functions may also be interpreted as logical operations on fuzzy sets. Indeed, the arithmetic mean of two membership functions, $(\mu_A + \mu_B)/2$, or geometric mean, $(\mu_A \cdot \mu_B)^{1/2}$ are both frequently used in fuzzy logic algorithms, offering "softened" versions of "or" and "and" as illustrated in Fig. 6.6. Using different exponents and weights can allow further manipulation of

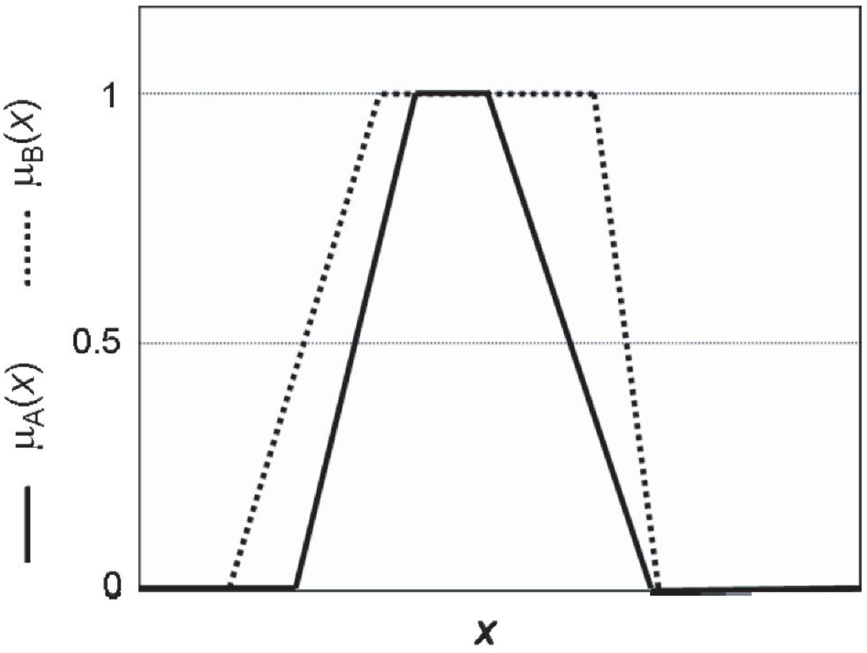


Fig. 6.5 Illustration of fuzzy sets A and B defined over a common domain with $A \subset B$.

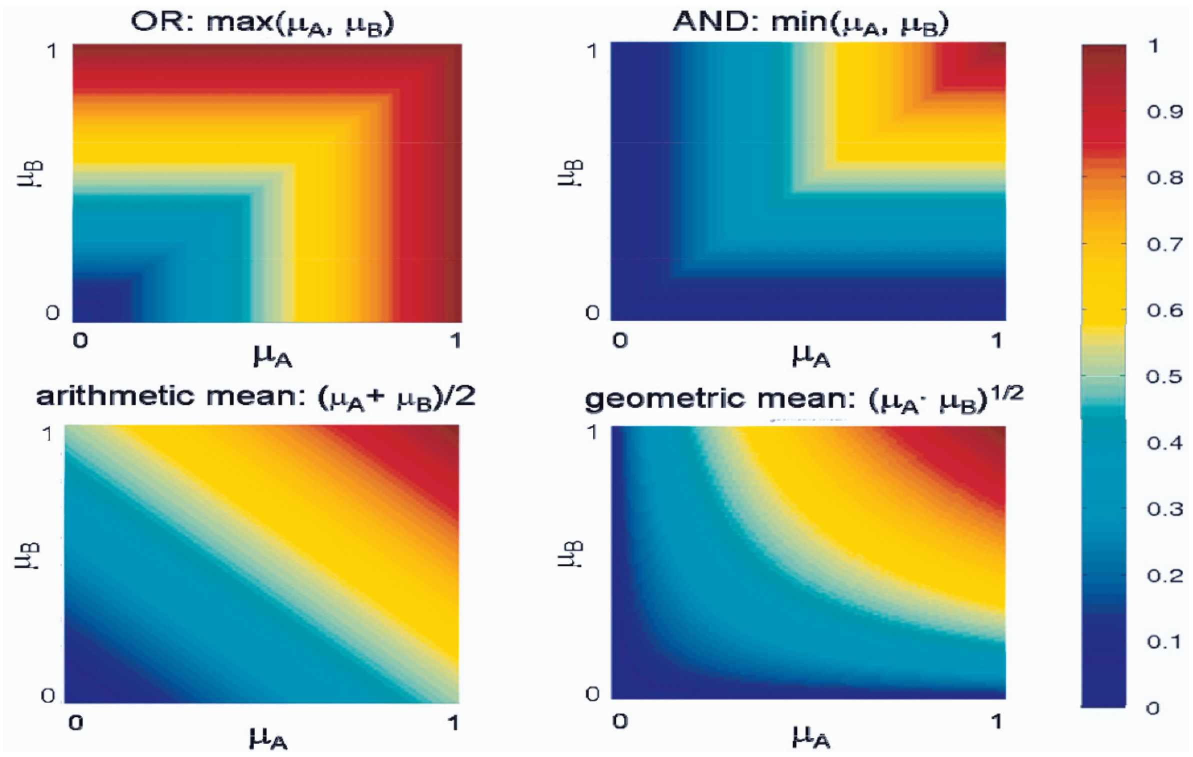


Fig. 6.6 Some functions of two variables that can be used as fuzzy-logical operations. The functions' output values are depicted by colors ranging from 0 (blue) to 1 (red) as illustrated by the colorbar at right.

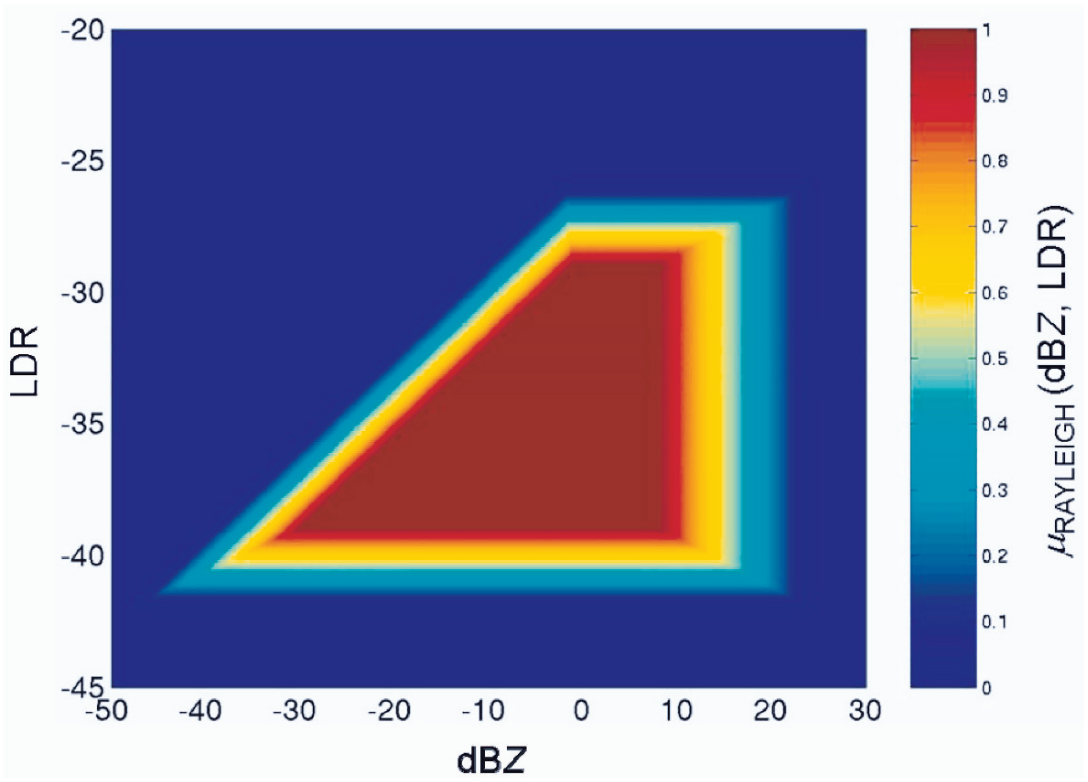


Fig. 6.7 An example 2-D membership function for "W-band radar Rayleigh scattering" as a function of reflectivity and linear depolarization ratio (LDR).

these means' behavior. Many other manipulations of fuzzy sets are possible, leading to great flexibility in the design of fuzzy logic algorithms. However, it is worth reiterating that for fuzzy sets having only binary (0 or 1) membership values, fuzzy logic's standard logical operations are completely equivalent to those for classical sets. Thus, fuzzy logic is equivalent to classical logic for "crisp" fuzzy sets, and fuzzy set theory is truly an *extension* of classical set theory.

Finally, it should be noted that while it is convenient to illustrate fuzzy membership functions as being functions of a single variable, they may actually be functions of arbitrarily many. In environmental science applications, the variables might be multiple attributes or measurements of the environmental state. As an example, consider the problem of determining whether the droplets in a cloud are small enough that they produce Rayleigh scattering of a radar signal, which is important for some remote sensing applications. A possible membership function for "W-band Rayleigh scattering" (W-band is a radar frequency near 95 GHz), based on the radar-measured linear depolarization ratio (LDR) and reflectivity (dBZ), is shown in Fig. 6.7. This fuzzy set is most conveniently defined in 2-D because of the complicated interaction of these two quantities (Vivekanandan et al. 1999).

6.4 Fuzzy Numbers

Although we usually think of numbers as being inherently precise, they often actually represent approximations or are subject to uncertainty. For instance, someone might say "I'll be ready at 7 pm" or "it will take 15 minutes to get there", but the first sentence is often understood to be approximate and the second is an estimate whose accuracy may vary considerably based on travel conditions. One way to propagate this sort of "fuzziness" through a calculation or take it into account when comparing two quantities is through the concept of *fuzzy numbers*.

Before defining fuzzy numbers, it is helpful to establish some descriptors for different kinds of fuzzy sets. We say that a fuzzy set A is *convex* if its membership function contains no local "dips", or, formally, if for any elements x_1 and x_2 and any number λ between 0 and 1, $\mu_A((1-\lambda)x_1 + \lambda x_2) \geq (1-\lambda)\,\mu_A(x_1) + \lambda\,\mu_A(x_2)$. A fuzzy set A is *normal* if there is at least one element x for which $\mu_A(x) = 1$. Using these concepts, a *fuzzy number* is defined to be a convex, normal fuzzy set with a continuous membership function having the property that $\{x : \mu_A(x) = 1\}$ is either a single element or a connected region (e.g., an interval). Common examples of fuzzy numbers include "triangle" functions and Gaussian bell-curve functions.

Any function f that operates on ordinary numbers may become a mapping of fuzzy numbers through the *extension principle*. If $f : \mathrm{X} \to \mathrm{Y}$, then the image of a fuzzy set A defined in the space X is a fuzzy set $f(A)$ in the space Y with membership function $\mu_{f(A)}(y) = \sup\{\mu_A(x) | f(x) = y\}$ when y is in the range of f (that is, there is at least one element $x \in \mathrm{X}$ for which $f(x) = y$), or 0 otherwise. Here "sup" stands for *supremum*, the smallest number that is greater than or equal to every member of the set of values within the curly brackets; it is the same as the *maximum* if the set in brackets is "closed." The extension principle basically says that the membership value of $f(A)$ at a point y is the biggest of the membership values $\mu_A(x)$ for the points x that are mapped to y. If f takes multiple arguments, so that $f : \mathrm{X}_1 \times \mathrm{X}_2 \times \ldots \mathrm{X}_n \to \mathrm{Y}$, then the fuzzy set $A = A_1 \times A_2 \times \ldots A_n$ may be defined by $\mu_A(x_1, x_2, \ldots, x_n) = \min(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \ldots, \mu_{A_n}(x_n))$ and the extension principle says $\mu_{f(A)}(y) = \sup\{\min(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \ldots, \mu_{A_n}(x_n)) | f(x_1, x_2, \ldots, x_n) = y\}$ when y is in the range of f, and 0 otherwise.

Computing the image of a fuzzy set can be computationally difficult, particularly for multivariate functions. However, there are some cases in which it is relatively easy. For instance, if f is an invertible function, and y is in the range of f, then $\mu_{f(A)}(y) = \mu_A\left(f^{-1}(y)\right)$. In particular, if f is linear (and nonzero) and A has a piecewise-linear membership function, then $f(A)$ is a piecewise-linear function determined by simply mapping the vertices of the membership function for A via f. For instance, the membership function for the fuzzy number "about 3" shown in Fig. 6.8 has vertices (2.5, 0), (2.875, 1), (3.125, 1), and (4.0, 0); its image under a linear function f is again piecewise linear with vertices (f(2.5), 0), (f(2.875), 1), (f(3.125), 1), and (f(4.0), 0). The image of "about 3" under a monotonic but nonlinear map g may not be piecewise linear, as the example in Fig. 6.8 shows. Nevertheless, a piecewise linear approximation with vertices (g(2.5), 0), (g(2.875), 1), (g(3.125), 1), and (g(4.0), 0) would probably be adequate for many practical applications.

Computing with fuzzy numbers provides a way to propagate uncertainty or ambiguity in input values through a formula or algorithm. For instance, fuzzy numbers may be used to represent environmental quantities subject to small-scale spatial or temporal variability or random measurement noise. If these are then used as inputs to a mathematical model or other formula, the answer will be a fuzzy number whose shape will indicate the spread or uncertainty in the result. For example, a method for remotely detecting the liquid water content L in clouds using measurements from two radars operating at different frequencies is based on a formula that relates L to the range derivative of the difference in measured reflectivity:

$$L(r) \approx \frac{(\mathrm{dB}Z_1(r+\Delta r) - \mathrm{dB}Z_2(r+\Delta r)) - (\mathrm{dB}Z_1(r-\Delta r) - \mathrm{dB}Z_2(r-\Delta r))}{2\,\Delta r\, A_L(r)} \tag{6.1}$$

where r represents a range along the radar beams, Δr is the range spacing between adjacent measurements along a beam, dBZ denotes reflectivity on a decibel scale, and A_L is a differential absorption coefficient whose value is a function of the two radar frequencies and the temperature (Williams and Vivekanandan 2007). The uncertainty in estimating the temperature at r leads to an uncertainty in A_L, and random noise affects each of the dBZ measurements. Using fuzzy numbers to represent each of these quantities yields a fuzzy number $L(r)$ that provides information about the distribution of possible true values

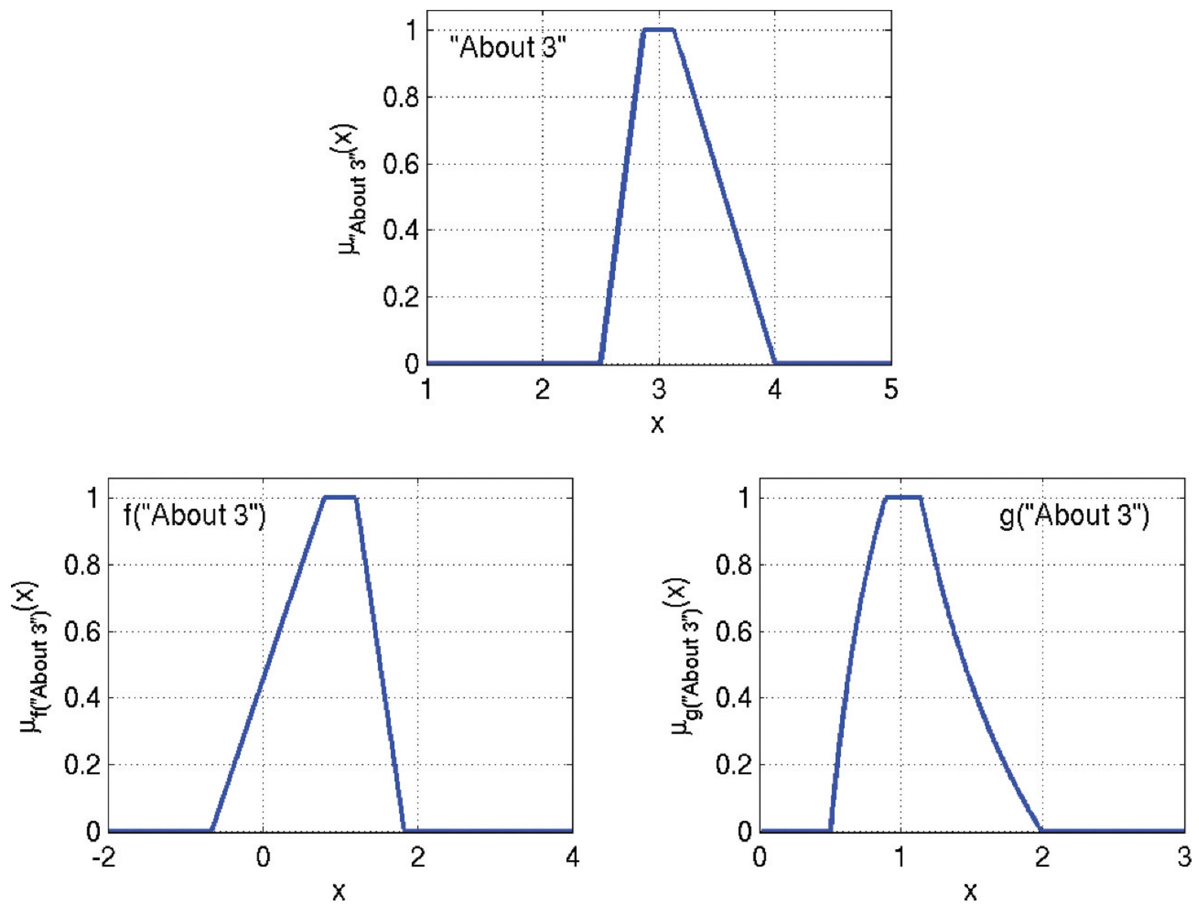


Fig. 6.8 (Top) One possible membership function for a fuzzy number "about 3". (Left) Membership function for f("about 3") where $f(x) = -5/3x + 6$. (Right) Membership function for g("about 3") where $g(x) = 1/(x - 2)$.

of liquid water content. This characterization of uncertainty might be very important to a decision support application – for instance, in assessing the risk that an aircraft flying through the measured region might experience hazardous icing conditions.

6.5 Fuzzy Logic

In classical logic, many logical statements may be represented in terms of set relations. For instance, a common form of reasoning is *modus ponens*, which states that if the sentence "if P, then Q" and the statement "P" both hold true, then it necessarily follows that the statement "Q" must also hold true. The statement "if P, then Q" is equivalent to the set inclusion $A \subset B$, where A is the set of elements for which P is true and B is the set of elements for which Q is true. Said another way, suppose x is an element, e.g., an environmental state, with attributes $a(x)$ and $b(x)$. Then the statement, "if $a(x) \in A$ then $b(x) \in B$" means that the set of elements x for which $a(x) \in A$ is a subset of the elements x for which $b(x) \in B$, that is, $A \subset B$. In terms of characteristic functions, if $\chi_A \leq \chi_B$ and $\chi_A(x) = 1$, then it follows from this statement that necessarily $\chi_B(x) = 1$; on the other hand, if $\chi_A(x) = 0$, nothing is known about $\chi_B(x)$. In fact, it is possible to write an equation for modus ponens in terms of characteristic functions that expresses the value of $\chi_B(x)$ given a value for $a(x)$ and the statement "if $a(x) \in A$ then $b(x) \in B$" ("$A \rightarrow B$" for short). One such equation is $\chi_{a(x),A\rightarrow B}(b(x)) = \min(1, 1 - \chi_A(x) + \chi_B(x))$. To see why, note that if

$a(x) \in A$, then $\chi_A(x) = 1$ and it follows that $\chi_{a(x),A\to B}(b(x)) = \min(1, \chi_B(x)) = \chi_B(x)$, that is, $b(x) \in B$. Otherwise, $\chi_A(x) = 0$ and it follows that $\chi_{a(x),A\to B}(b(x)) = \min(1, 1 + \chi_B(x)) = 1$, representing the convention in classical logic that a logical statement is always held to be true when the premise is not satisfied, or, equivalently, that the statement gives no information about whether $b(x) \in B$ in this case. Thus, the modus ponens equation shows that for each value $a(x)$ in the domain of A, the classical logic statement "$A \to B$" produces a *set* over $b(x)$ in the domain of B; this set may either be B itself or the entire domain of B. Since fuzzy logic is an extension of classical logic, the fuzzy logic equivalent to modus ponens must be defined to also generate such a set.

In fuzzy logic, the statements P and Q are not necessarily either true or false, but may be only partially true. If A and B are fuzzy sets with membership function μ_A representing the degree to which P is true and μ_B representing the degree to which Q is true, then by analogy with classical logic, the statement "if P, then Q" should mean $\mu_A \le \mu_B$. However, this tempting interpretation turns out to be nonsensical. For instance, it might happen that the fuzzy set B is not normal, that is, Q is a statement that is never fully true. Then μ_B is always less than one, but it should still be possible to form a meaningful fuzzy logical statement "$A \to B$" where A is normal, that is, μ_A may be 1 and the premise completely true! For instance, consider the statement "if a dessert includes chocolate, then it is delicious", so that P = "a dessert includes chocolate" and Q = "a dessert is delicious". The fuzzy set A of desserts including chocolate certainly has members with truth value one. On the other hand, the fuzzy set B of "delicious desserts" may not have any incontrovertible members – particularly if a cantankerous professional food critic is setting the scale. The crux of the issue is that fuzzy logic gives information about the consequent Q even when the premise P is not completely true or the consequent itself is ambiguous.

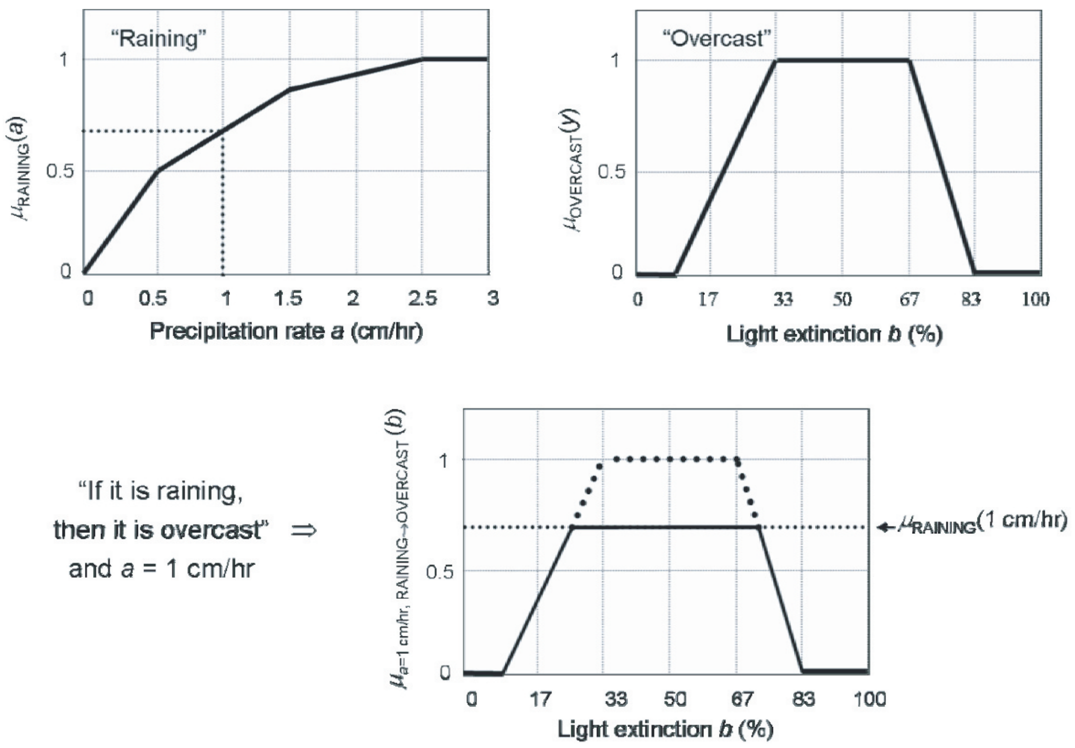


Fig. 6.9 Illustration of Mamdami's fuzzy implication. The top two plots provide possible definitions of the concepts "raining" in terms of precipitation rate and "overcast" in terms of solar light extinction, respectively; the lower plot illustrates the fuzzy set resulting from the statement "If it is raining, then it is overcast" given a measured precipitation rate value of 1 cm/h.

There are a number of different definitions for fuzzy modus ponens that define the resultant fuzzy set $\mu_{a(x),A\to B}(b(x))$ in a way that achieves this. One was proposed by Mamdami (1974): $\mu_{a(x),A\to B}(b(x)) = \min(\mu_A(x), \mu_B(x))$. For a given value of $a(x)$, $\mu_A(x)$ has a specific value between 0 and 1 and the right hand side of this equation represents a fuzzy set over attribute b of x that is equal to $\mu_B(x)$ but truncated at an upper bound of $\mu_A(x)$. In words, this definition of fuzzy logical implication means something like, "if attribute a of x is a member of set A to the extent $\mu_A(x)$ and $A \to B$, then attribute b of x is a member of set B to at most the extent $\mu_A(x)$." Mamdami-style inference is illustrated in Fig. 6.9 for the statement, "If it is raining, then it is overcast." Other definitions for $\mu_{a(x),A\to B}(b(x))$ include one proposed by Zadeh: $\min(1, 1 - \mu_A(x) + \mu_B(x))$; the product: $\mu_A(x)\mu_B(x)$; the bounded product: $\max(0, \mu_A(x) + \mu_B(x) - 1)$; and the Boolean logic implication: $\max(1 - \mu_A(x), \mu_B(x))$.

In most cases, additional pieces of evidence and associated logical statements are necessary to refine the fuzzy set resulting from fuzzy modus ponens. The process of combining different sources of evidence using logical rules and determining a final "crisp" output is called *fuzzy inference*.

6.6 Mamdami-Style Fuzzy Inference

The ultimate purpose of an artificial intelligence algorithm is to map a set of inputs to one or more outputs. In environmental science applications, the inputs often consist of environmental measurements or measurement-derived quantities, and the outputs typically represent an estimate of some attribute of the system's state or a recommended action to be taken. Such mappings may be complex, and will often be nonlinear. Fuzzy inference provides a way to build up a mapping of this sort using logical rules like those often employed in human natural language, as described in the previous section. Because an "if…then" statement involving fuzzy sets applies for any degree of truth of the antecedent and provides only weak information about the consequent, many such fuzzy rules may need to be combined to provide conclusive evidence. The ability to make use of multiple sources of ambiguous information is one of the great strengths of fuzzy logic inference. Not only does it allow information to be used efficiently, but the fact that many different rules are employed means that if some input data are missing, the fuzzy inference system can still function using the remaining rules. For this reason, fuzzy logic algorithms are generally quite robust.

To explore the structure of a fuzzy inference system more concretely, suppose that the goal of the system is to predict a value for a state parameter $u(x)$ based on a number of environmental attributes $a(x), b(x), \ldots, d(x)$. If a number of rules exist connecting these environmental attributes (measurements or derived quantities, for instance) to the state parameter being predicted, then the antecedents and consequents of these rules must be *fuzzified*, i.e., defined as appropriate fuzzy sets over the domain of the relevant attribute. The rules can then be expressed as a set of fuzzy logic statements like those discussed in the previous section, e.g., $A \to R, B \to S, \ldots D \to T$. Using the known values of the various environmental attributes in conjunction with these statements and an appropriate definition of fuzzy implication, each rule will yield a fuzzy set over the domain of the parameter $u(x)$. These resulting fuzzy sets may then be *aggregated* to form a final resultant fuzzy set over $u(x)$. The most common way to perform this aggregation is by taking the *maximum* of the membership functions resulting from the various rules. In this case. the membership function resulting from the input values and fuzzy rules is given by $\mu_{a(x),b(x),\ldots,d(x),A\to R,B\to S,\ldots,D\to T}(u(x)) = \max(\mu_{a(x),A\to R}(x), \mu_{b(x),B\to S}(x), \ldots, \mu_{d(x),D\to T}(x))$. If the Mamdami definition of fuzzy implication is used, for example, this becomes $\mu_{a(x),b(x),\ldots,d(x),A\to R,B\to S,\ldots,D\to T}(u(x)) = \max(\min(\mu_A(x), \mu_R(x)), \min(\mu_B(x), \mu_S(x)), \ldots, \min(\mu_D(x), \mu_T(x)))$ and is called Mamdami-style inference. Other methods for aggregation include taking the sum of the membership functions resulting from the various rules and either capping it at 1 or renormalizing the sum so that the maximum value is 1.

Once the resultant fuzzy set is obtained by aggregating the results of the fuzzy rules, it is often desirable to produce a final, "crisp" estimate of the value $u(x)$ through a process called *defuzzification*. One way to perform defuzzification is to compute the centroid of

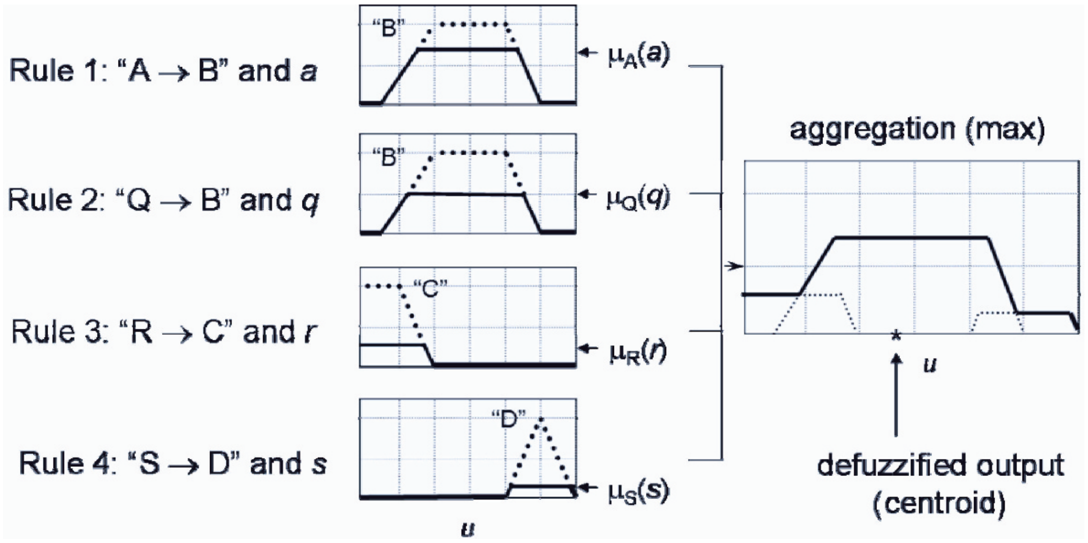


Fig. 6.10 Illustration of Mamdami-style inference for determining a quantity u given four fuzzy rules and input values a, q, r, and s. Aggregation is performed by taking the maximum of the four truncated sets and defuzzification is achieved by computing the centroid of the aggregated fuzzy set.

the aggregated fuzzy set:

$$\int u(x)\, \mu_{a(x),b(x),\ldots,d(x),A\to R,B\to S,\ldots,D\to T}\,(u(x))\,dx \Big/ \int \mu_{a(x),b(x),\ldots,d(x),A\to R,B\to S,\ldots,D\to T}\,(u(x))\,dx \quad (6.2)$$

This is the method illustrated in Fig. 6.10. Other methods for defuzzification include taking the value of $u(x)$ for which $\mu_{a(x),b(x),\ldots,d(x),A\to R,B\to S,\ldots,D\to T}(u(x))$ obtains its maximum, or computing the centroid over only those values for which it exceeds some threshold.

A concrete example may help clarify the Mamdami fuzzy inference procedure. Suppose that a certain teenager, who is in charge of watering the lawn each Saturday, appears to need some assistance in knowing how much to water each week. If you had a soil moisture sensor, you could devise a formula to prescribe a precise watering time. But without a sensor, you might come up with the following simple rules as a guide:

Rule 1: If the weather has been cool or wet, water a little.
Rule 2: If the weather has been hot or dry, water a lot.

These are precisely the kind of natural language rules that can be turned into a fuzzy logic expert system. The first step is *fuzzification*, that is, defining the appropriate fuzzy sets for the concepts involved. *Cool* and *hot* are weather attributes that might naturally be defined based on the average temperature measured over the past week, t; see Fig. 6.11. Similarly, *dry* and *wet* are precipitation concepts that could be defined in terms of the total precipitation over the past week, p. Finally, watering *a little* or *a lot* in this context could be defined by the number of hours the water is to be left on, h. Now on any given Saturday, a review of weather records over the past week will yield t and p. Recalling that the fuzzy set union ("or") may be represented by the maximum of the sets' membership functions, it follows that $\mu_{cool\ or\ wet}(\text{week}) = \max(\mu_{cool}(t), \mu_{wet}(p))$ and $\mu_{hot\ or\ dry}(\text{week}) = \max(\mu_{hot}(t), \mu_{dry}(p))$. To evaluate Rule 1 under Mamdami-style fuzzy inference, we compute the antecedent, $\mu_{cool\ or\ wet}(\text{week})$, and use it to "cap" the membership function of the consequent, $\mu_{a\ little}(h)$, resulting in the fuzzy set $\mu_1(h) = \max(\min(\mu_{cool\ or\ wet}(week), \mu_{a\ little}(h)))$. Similarly, evaluating Rule 2 yields the fuzzy set $\mu_2(h) = \max(\min(\mu_{hot\ or\ dry}(week), \mu_{a\ lot}(h)))$. Next we must *aggregate* the results of these two rules, for instance, by taking their maximum: $\max(\mu_1(h), \mu_2(h))$. Finally, we wish to determine a precise number of hours to water, which we do through *defuzzification*. This may be accomplished, for instance, by taking the centroid $\int h\ \max(\mu_1(h),\ \mu_2(h))\,dh \,/ \int \max(\mu_1(h), \mu_2(h))\,dh$,

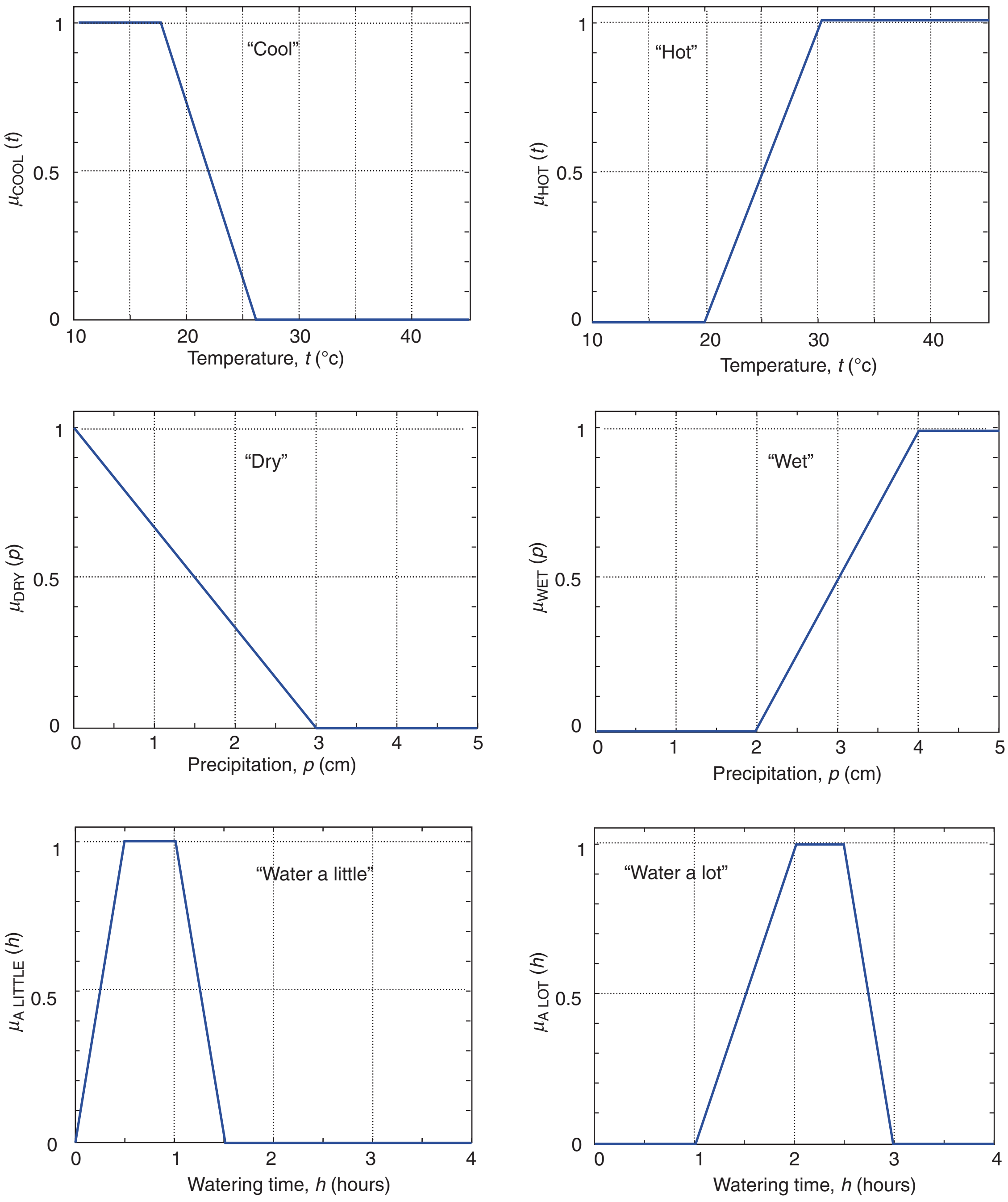


Fig. 6.11 Fuzzy sets for use in determining watering time: temperature concepts (top), precipitation concepts (middle), and watering time concepts (bottom).

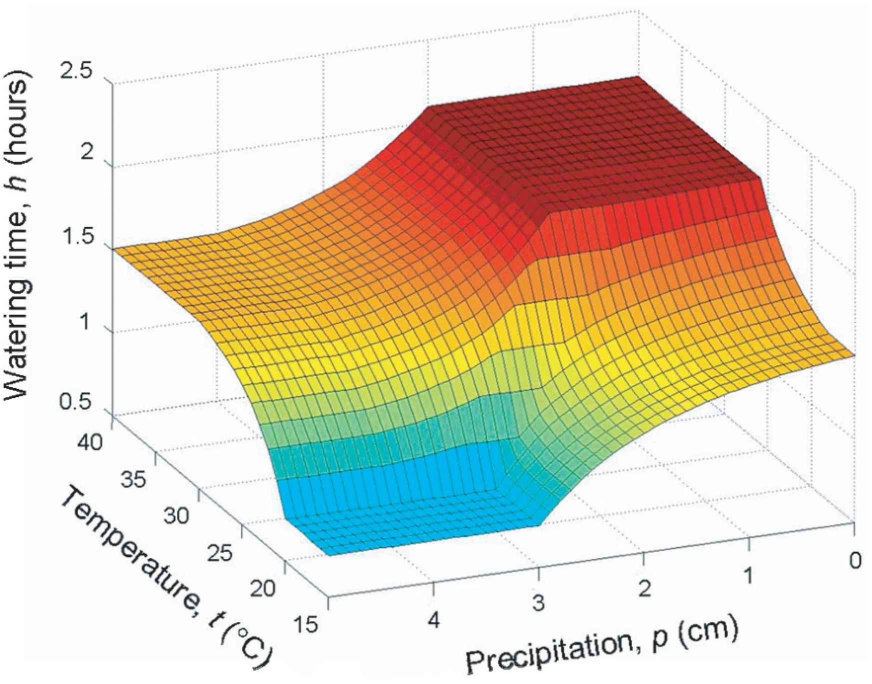


Fig. 6.12 Watering time as a function of the previous week's temperature and precipitation, produced by substituting pairs of values of t and p into the example Mamdami-style fuzzy logic inference system described in the text.

where the integrals are computed via an appropriate sum or quadrature. The ultimate result of this inference procedure is the *function* depicted in Fig. 6.12 that maps a week's average temperature t and total precipitation p into a prescribed number of hours h to water the lawn. The function is surprisingly complex given the simplicity of the rules, and is nonlinear despite the fact that all the membership functions are piecewise-linear. The addition of other rules could help refine it further. For instance, specifying that no watering is needed if it is cool and wet would allow the watering time to approach zero in those conditions, which the initial function does not. And including linguistic rules involving other factors, such as soil type, season of year, or type of grass used in the lawn, could make the function more general. Nevertheless, the fact that Mamdami-style fuzzy inference based on two simple rules produces a function having basically the correct behavior clearly illustrates the power of this approach.

6.7 Takagi-Sugeno Fuzzy Inference

The language of science is of course not only linguistic, but also mathematical. Thus, another form of human scientific reasoning involves determining when the environmental conditions are those in which a known physical relationship applies. For example, photosynthesis in plants may be limited by light, water, carbon dioxide or nutrient availability, so different models for photosynthesis may be applicable under different soil, weather and atmospheric conditions. One might apply each of these models to make a prediction of photosynthesis, then perform a weighted average of the predictions using weights that represent their degree of applicability based on the current environmental conditions. A version of fuzzy inference that accommodates this form of reasoning is called Takagi-Sugeno-style fuzzy inference, named for the researchers who proposed it (Takagi and Sugeno 1985).

In Takagi-Sugeno inference, a linguistic rule is used to determine the degree of applicability of a given formula for determining the variable of interest. The consequent of a Takagi-Sugeno fuzzy rule is not a fuzzy set as it is for Mamdami-style inference, but instead a direct estimate of the variable determined by the formula. So if the goal of the fuzzy inference system is to predict a value for a state parameter $u(x)$ based on a number of environmental attributes $a(x)$, $b(x), \ldots,$ $d(x)$, a fuzzy rule might then have the structure, "If A, then $u(x) = f(a(x), b(x), \ldots, d(x))$", where A is a fuzzy set dependent on some number of environmental attributes. (In a common formulation of Takagi-Sugeno fuzzy inference, the function f is assumed to be linear and to use only the variables involved in the linguistic rule, but these restrictions are not necessary.) Given n rules of the form, "If A_i, then $u(x) = f_i(a(x), b(x), \ldots, d(x))$", the resulting estimate of $u(x)$, denoted $\hat{u}(x)$, is then determined by the weighted average

$$\hat{u}(x) = \frac{\sum_{i=1}^{n} \mu_{A_i}(x)\, f_i(a(x), b(x), \ldots, d(x))}{\sum_{i=1}^{n} \mu_{A_i}(x)} \tag{6.3}$$

In some cases, there may be some rules which have greater relevance than others even when their antecedents are equally true. In this case, a researcher might assign an importance weight w_i to each rule and compute the estimate of $u(x)$ via the formula

$$\hat{u}(x) = \frac{\sum_{i=1}^{n} w_i\, \mu_{A_i}(x)\, f_i(a(x), b(x), \ldots, d(x))}{\sum_{i=1}^{n} w_i\, \mu_{A_i}(x)} \tag{6.4}$$

Note that no additional defuzzification step is needed for Takago-Sugeno fuzzy inference because the result is the numerical estimate $\hat{u}$, not a fuzzy set. This makes Takago-Sugeno fuzzy inference comparatively efficient, while the ability to use arbitrary functions f as the consequents of the fuzzy rules makes it quite flexible. In addition, algorithms using this architecture are inherently robust because if some data are missing, the remaining data may still be sufficient to compute (6.4) with some values of i left out.

A special form of Takagi-Sugeno fuzzy inference occurs when the functions f_i are simply constant values, that is, when the fuzzy rules have the form, "If A, then $u(x) = c$." In this case (6.3) may be interpreted as averaging a number of numerical predictions, with the weight of each prediction being given by the degree of truth of the antecedent to each rule. It is interesting to note that the final result in this case is the same as it would be if the consequents are interpreted as fuzzy sets and Mamdami-style fuzzy inference is employed along with the centroid method of defuzzification. For example, Takagi-Sugeno style rules for the watering time example from the previous section might be written as:

Rule 1: If the weather has been cool or wet, watering time $h_1 = 0.75$ h.
Rule 2: If the weather has been hot or dry, watering time $h_2 = 2.15$ h.

Then the resulting estimate of required watering time is given by the function:

$$\begin{aligned}\hat{h}(t,p) &= \frac{\mu_{cool\ or\ wet}(t,p)\,(0.75) + \mu_{hot\ or\ dry}(t,p)\,(2.15)}{\mu_{cool\ or\ wet}(t,p) + \mu_{hot\ or\ dry}(t,p)}\\ &= \frac{\max(\mu_{cool}(t), \mu_{wet}(p))(0.75) + \max(\mu_{hot}(t), \mu_{dry}(p))(2.15)}{\max(\mu_{cool}(t), \mu_{wet}(p)) + \max(\mu_{hot}(t), \mu_{dry}(p))}\end{aligned} \tag{6.5}$$

where the membership functions for *cool*, *wet*, *hot*, and *dry* are again defined by the plots in Fig. 6.11. This function $\hat{h}$ is nearly indistinguishable from the function resulting from the Mamdami-style inference system described in the previous section and plotted in Fig. 6.12. Indeed, the Mamdami-style system would be identical to (6.5) if the fuzzy sets for "water a little and "water a lot" were replaced by their "typical" values, that is, the singleton sets $\mu_{a\ little}(h) = \begin{cases} 1 \text{ if } h = 0.75 \\ 0 \text{ otherwise} \end{cases}$ and $\mu_{a\ lot}(h) = \begin{cases} 1 \text{ if } h = 2.15 \\ 0 \text{ otherwise} \end{cases}$, turning the centroid defuzzification step into the simple weighted mean. A Mamdami-style inference system can frequently be simplified using this technique without substantial loss of information, particularly if the consequent fuzzy sets are symmetric or nearly so and do not overlap substantially.

6.8 Fuzzy Consensus Methods

A somewhat simplified form of Takagi-Sugeno fuzzy inference may be interpreted as one of the class of artificial intelligence techniques called *consensus methods* or *mixtures of experts*. In this interpretation, each member of a group of "experts" evaluates various attributes of the environmental state and makes a prediction of some variable of interest. The results are then averaged, using weights for each expert's prediction based on some measure of his or her reputation or past skill. In addition, the experts might accompany each prediction with a 0 to 1 measure of their confidence in it, with greater confidence being expressed when the required data are available and of good quality and the theory or experience being applied by the expert is judged appropriate to the conditions at hand. If the reasoning of each expert from evidence to prediction is described by a function f, this procedure corresponds precisely to that described in (6.4), but with the fuzzy set membership values $\mu_{A_i}(x)$ replaced by each expert's "confidence", c_i:

$$\hat{u}(x) = \frac{\sum_{i=1}^{n} w_i\, c_i(x) f_i(x)}{\sum_{i=1}^{n} w_i\, c_i(x)} \tag{6.6}$$

Said another way, the fuzzy consensus reasoning formula (6.6) is a special version of Takagi-Sugeno fuzzy inference in which the antecedent A_i for each rule is taken to be the fuzzy set, "the required data are of good quality and the predictive function is appropriate for this scenario." Many applications in the environmental sciences are naturally handled using fuzzy consensus reasoning, taking as inputs several environmental measurements or features, using an assortment of methods to predict a variable of interest, and then combining the results. For instance, in forecasting thunderstorm initiation, researchers might utilize data from satellite-measured cloud-top growth rates, numerical weather models, and radar measurements, each of which can be used to give a prediction of whether initiation is likely to occur, and then combine them to obtain a more reliable forecast than any of the inputs could provide by themselves.

In many applications, assigning the confidences c_i is a very important part of the fuzzy consensus algorithm. In fact, the confidences are often computed using data quality values and assessments of relevance that are themselves determined by separate fuzzy logic algorithms. The assignment of a confidence to each prediction also allows the final output given by the fuzzy logic algorithm to itself be assigned a confidence value that will aid users in interpreting it for decision making purposes. For instance, if an atmospheric turbulence detection algorithm estimates severe turbulence in a region but has low confidence in this assessment, it might not make sense to re-route air traffic around the area without other confirming information that a hazard is present. The confidence associated with the estimate in (6.6) might be computed as

$$\hat{c}(x) = \frac{\sum_{i=1}^{n} w_i\ (c_i(x))^{m+1}}{\sum_{i=1}^{n} w_i\ (c_i(x))^{m}} \tag{6.7}$$

if any product $w_i c_i(x) > 0$, and 0 otherwise, where $m \geq 0$. For example, if one chooses $m = 0$, the formula (6.7) simply produces the weighted-mean confidence. If $m > 0$, then the confidences become part of the averaging weights, and as $m \to \infty$, $\hat{c}(x) \to \max_{\{i \mid w_i > 0\}} c_i(x)$. More generally, since confidences are equivalent to fuzzy membership function values, they may also be combined using fuzzy-logical operations, such as AND (min) or a weighted geometric average such as

$$\hat{c}(x) = \prod_{i=1}^{n} c_i(x)^{w_i / \sum_{k=1}^{n} w_k} \tag{6.8}$$

which yields a large final confidence $\hat{c}$ only if all of the input confidences c_i with $w_i > 0$ are not too small.

Returning to the example of determining the optimal lawn watering time based on the previous week's temperature and precipitation, predictive functions or *interest maps* based on t and p alone might be chosen as depicted in Fig. 6.13; although these happen to be piecewise-linear functions, that is for illustrative purposes and is not a requirement. Suppose that for a particular week both predictions have equal confidence (say 1), and that the temperature interest map is given a weight of 0.4 and the precipitation interest map a weight of 0.6. Then the function resulting from applying the fuzzy consensus reasoning formula (6.6) is displayed in Fig. 6.14. Note that Fig. 6.14 is quite similar to Fig. 6.12 except that it is piecewise linear (being a linear combination of piecewise linear functions in this example). If the confidence in t and p were not equal, then the interest map with higher confidence would begin to dominate. Of course, in a practical application there would ideally be many inputs providing somewhat redundant information so that loss of confidence in one or two input variables would only slightly degrade performance.

Two applications of the fuzzy consensus reasoning technique deserve special mention. The first is an

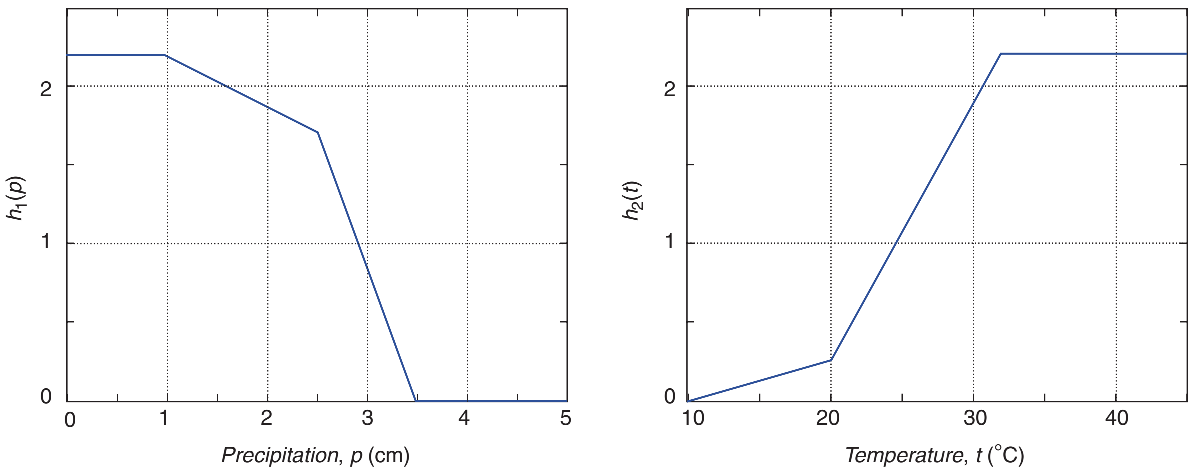


Fig. 6.13 Possible interest maps for determining watering time given precipitation (left) or temperature (right) for use in a fuzzy consensus reasoning system.

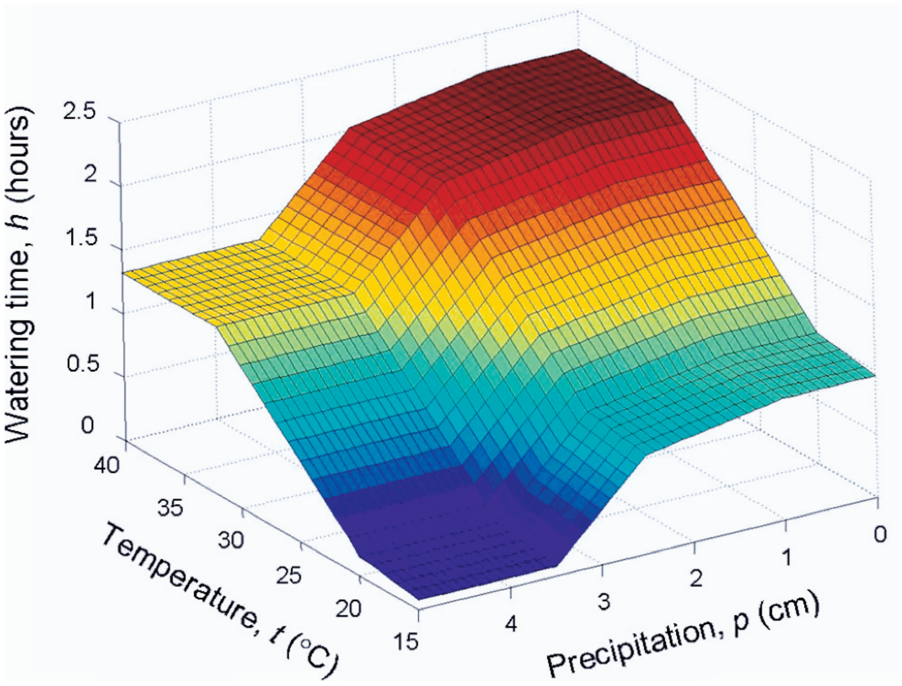


Fig. 6.14 Watering time as a function of t and p from the fuzzy consensus reasoning example described in the text.

application to data smoothing or interpolation. Suppose that a quantity of interest is available over some domain (usually spatial or spatio-temporal), and each value x_i is assigned a confidence estimate c_i, e.g., from a data quality control procedure. A standard smoothing or interpolation approach is to convolve a smoothing kernel, say a Gaussian, with the data field. Fuzzy consensus smoothing additionally makes use of the confidences to reduce the influence of lower-quality data, and also produces a confidence estimate for each smoothed point. For each target location, the smoothing kernel is "centered" at that point and the weight w_i for each data point x_i in the neighborhood is determined by the value of the smoothing kernel there. Then the smoothed value may be computed via an application of (6.6), with $f_i(x)$ replaced with the data value x_i

and $c_i(x)$ replaced with c_i; the confidence associated with the smoothed value may be computed via (6.7). When the quality control procedure is imperfect so that some outliers are assigned nonzero confidence, then the data points in the neighborhood may first be sorted and the top and bottom $p\%$ of the corresponding $w_i c_i$ set to 0 before (6.6) and (6.7) are applied, where p is between 0 and 50%. This method may be referred to as *fuzzy confidence-weighted trimmed-mean kernel smoothing*. If $p = 0$, no trimming occurs, while if $p = 50\%$, the method becomes a confidence-weighted median.

A closely related application is a fuzzy version of the popular k-nearest neighbor (k-NN) algorithm for generalizing from examples. In the classical k-NN, a set of labeled data are used to infer the class of a new example based on the class occurring most frequently among the k nearest labeled points. For instance, the data points might consist of vectors having as components measurements from various sensors (e.g., temperature, pressure, humidity) recorded at 9 am on a number of different days, and the classes could be what sort of weather was observed that afternoon (e.g., clear, cloudy, light rain, thunderstorm). Given a value of k, a distance metric, and a vector of sensor measurements from a subsequent morning, the k-NN algorithm would determine a weather prediction based on the historically best-matching observed class among the k nearest neighbors to that data vector. In the fuzzy version of k-NN, the class labels of the historical points are replaced by fuzzy class membership values between 0 and 1, and the membership of the new example is given by a distance- and membership-weighted consensus of the k nearest neighbors. More precisely, suppose a collection of data vectors $\{\mathbf{x}_j\}$ have fuzzy class memberships μ_{C_i} for classes $\{C_i | 1 \leq i \leq N\}$, k is chosen as the number of neighboring points to consider, and $m > 1$ (smaller values correspond to "tighter", more localized influence). A metric d must be specified so that $d(\mathbf{x}_i, \mathbf{x}_j)$ represents the "distance" between the two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Common choices for d are the standard Euclidean distance (the square root of the sum of squared differences of the vector elements) or the Manhattan distance (the sum of the absolute values of the vector element differences). It is sometimes useful to first re-scale the data so that the range of values for each vector component is approximately the same. If $\{\mathbf{y}_n | 1 \leq n \leq k\}$ are the k points from the set $\{\mathbf{x}_j\}$ that are closest to a new data vector $\mathbf{x}$, then fuzzy k-NN assigns the class membership of $\mathbf{x}$ for each class C_i via the formula

$$\mu_{C_i}(\mathbf{x}) = \frac{\sum_{n=1}^{k} \mu_{C_i}(\mathbf{y}_n)\, d(\mathbf{x}, \mathbf{y}_n)^{-\frac{2}{m-1}}}{\sum_{n=1}^{k} d(\mathbf{x}, \mathbf{y}_n)^{-\frac{2}{m-1}}} \tag{6.9}$$

If the original class memberships are defined so that their sum for each data point is 1, that is, $\sum_{i=1}^{N} \mu_{C_i}(\mathbf{x}_n) = 1$ for each $\mathbf{x}_n$, then it can be shown that $\sum_{i=1}^{N} \mu_{C_i}(\mathbf{x}) = 1$ as well. In contrast to the standard k-NN algorithm, which yields only the best-matching class, the class memberships $\mu_{C_i}(\mathbf{x})$ returned by fuzzy k-NN provide much richer information about the ambiguity of the similarities identified between the new data and the historical examples. In the weather forecasting example, the historical data class memberships could accommodate ambiguities in the observed weather (e.g., the *degree* of cloudiness), and the memberships assigned via (6.9) would provide some notion of the uncertainty in the nearest neighbor prediction. Moreover, even if the class with the highest membership is ultimately chosen as the final "defuzzified" prediction, the fuzzy k-NN method is somewhat more nuanced than the original version because it weighs the neighborhood data closest to the point being classified more strongly than more distant – and hence potentially less relevant – examples. For instance, in Fig. 6.15, classical k-NN with $k = 5$ assigns the point at "x" to the class "green", since the three of the five nearest points (shown in the dotted circle) are labeled as green. Fuzzy k-NN assigns non-zero memberships to both "blue" and "green", with the precise values depending on the distance weighting function and value of m chosen. If m is small, membership in "blue" could be greatest since a blue point is closest to the "x".

6.9 Fuzzy *c*-Means Clustering

Fuzzy c-means (FCM) clustering is an algorithm for partitioning a multivariate dataset into a prespecified number of fuzzy "clusters", each represented by a fuzzy set defined over the points in the dataset. It

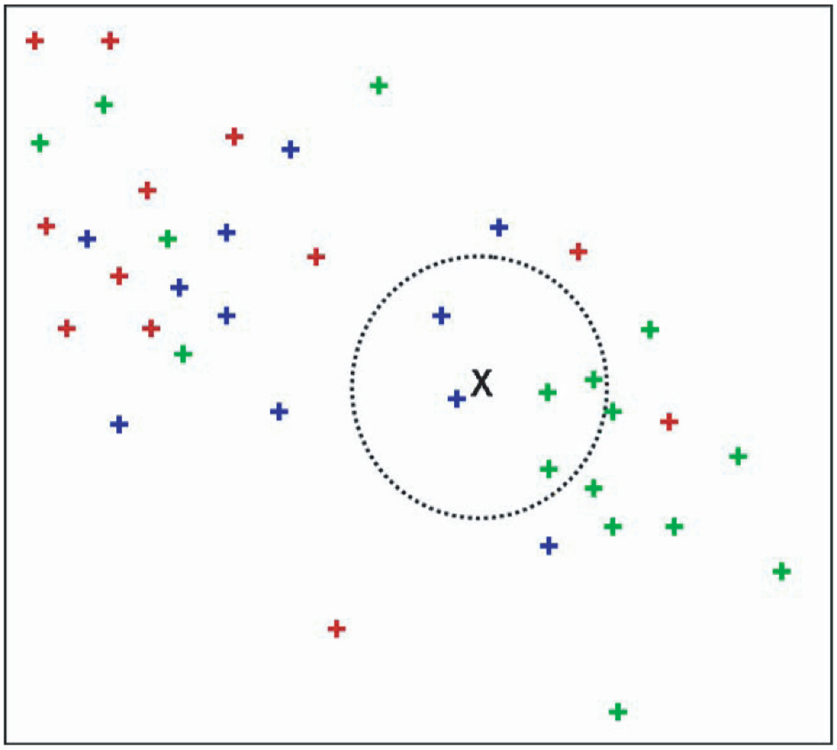


Fig. 6.15 k-nearest neighbor example for three classes (red, green and blue) and $k = 5$.

is a fuzzy variant of the familiar classical method called k-means clustering, which separates data into k "crisp" sets. While the result of k-means clustering is a partition of the data points into disjoint classical sets, under FCM clustering the data points may have partial membership in several different fuzzy sets. Both clustering methods are unsupervised learning methods that find patterns in data without the benefit of "labeling" by a human expert. In environmental science applications, the points being clustered will usually be vectors comprised of several measurements or derived quantities that represent features of the environmental state at a particular time and location. Clustering can reveal the different major domains or attractors in a dynamical system – weather patterns, for example – which may then be analyzed separately to determine their important characteristics. Using fuzzy sets in place of classical ones makes for a more robust clustering algorithm and may provide richer information about the data.

Fuzzy c-means clustering begins with c (a predetermined number) cluster centers, or *prototypes*, which may be arbitrarily initialized or chosen based on prior knowledge of the data. The distance from each of the dataset points to each of the cluster prototypes is computed, and each point is assigned a membership in each of the c clusters based on these distances. New prototypes for each fuzzy cluster are then computed by taking the cluster-membership-weighted centroid of all the points, and the process is repeated until the change in the prototypes becomes small. Figure 6.16 shows an example of fuzzy c-means clustering used to find two clusters in two-dimensional data.

More formally, suppose that the dataset consists of N vectors $\{\mathbf{x}_1, \mathbf{x}_{2,} \ldots, \mathbf{x}_N)$. As in the k-nearest neighbor technique described earlier, the FCM algorithm requires that a metric d is specified so that $d(\mathbf{x}, \mathbf{y})$ represents the "distance" between vectors $\mathbf{x}$ and $\mathbf{y}$. A number c is selected for the number of fuzzy clusters desired, and initial prototype points $\mathbf{v}_1, \ldots, \mathbf{v}_c$ are specified or chosen randomly. A parameter $m > 1$ is chosen to represent the "tightness" of the clusters: values of m near one will produce more distinct or crisp clusters, while larger values of m will allow more overlap or "fuzziness". Finally, a convergence threshold $\varepsilon > 0$ is chosen to determine when to stop the iteration, which proceeds as follows:

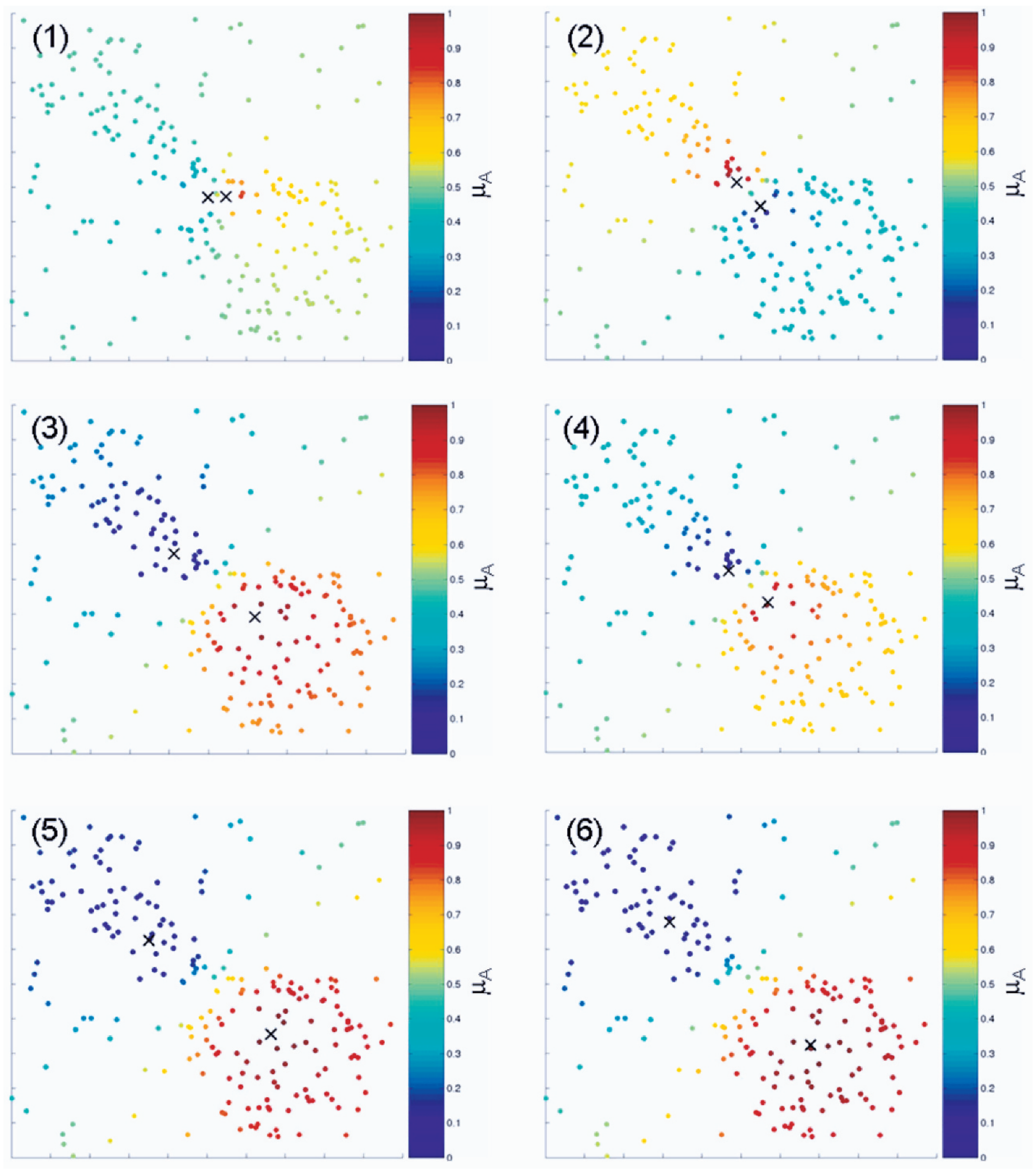


Fig. 6.16 Fuzzy *c*-means clustering example with two fuzzy sets and six iterations shown. The points are colored according to the membership in set "A", so that points with high membership in set "A" are colored red and points with low membership in set "A" (hence high membership in set "B") are colored blue. An "x" marks each cluster center.

1. Define the membership of each data point $\mathbf{x}_k$ in the ith cluster, C_i by

$$\mu_{C_i}(\mathbf{x}_k) = \left(\sum_{j=1}^{c} \left(\frac{d(\mathbf{x}_k, \mathbf{v}_i)}{d(\mathbf{x}_k, \mathbf{v}_j)} \right)^{\frac{1}{m-1}} \right)^{-1} . \tag{6.10}$$

2. Compute new cluster prototypes $\mathbf{v}_i$ via

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} \mathbf{x}_k \left(\mu_{C_i}(\mathbf{x}_k)\right)^m}{\sum_{k=1}^{N} \left(\mu_{C_i}(\mathbf{x}_k)\right)^m} . \tag{6.11}$$

3. If the absolute value of the change in each cluster prototype is less than ε for every vector element, then the iteration stops. Otherwise, return to step (1).

The final result is a set of fuzzy clusters C_i defined in such a way that for each data vector $\mathbf{x}_k$, $\sum_{i=1}^{c} \mu_{C_i}(\mathbf{x}_k) = 1$, that is, the total membership of $\mathbf{x}_k$ in all c fuzzy clusters is one.

As is true for many machine learning techniques, the choice of the number of clusters, c, and the "fuzziness" parameter m is a bit of an art form, and may require a trial-and-error approach to get meaningful results. And while there are formulas in the literature for determining the "goodness" of a given clustering, its utility may really be dependent on the final application. Note also that the FCM algorithm will not necessarily identify the same fuzzy clusters in different runs unless the initial prototypes are the same each time; thus, a careful choice of the initial prototypes or performing a number of independent runs may be worthwhile.

In addition to identifying structures in data, fuzzy clustering might also provide an important step to developing a fuzzy inference system when human expertise in solving a problem is incomplete. As an example, using clustering to identify different "domains" in weather sensor data might aid in creating a forecast based on those data by training a different predictive model (a multilinear fit, for example, or even a neural network) separately on each cluster. Then a Takagi-Sugeno-style fuzzy inference system could be constructed that combines the various models based on the degree of membership of a point in the antecedent fuzzy cluster.

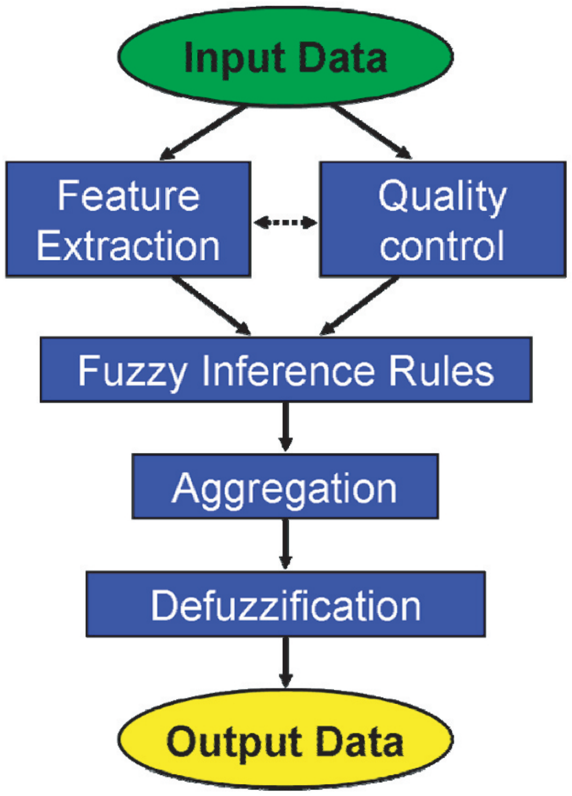


Fig. 6.17 Anatomy of a typical fuzzy logic algorithm.

6.10 Fuzzy Logic Algorithms

A typical fuzzy logic algorithm consists of several elements in addition to the fuzzy inference component (see Fig. 6.17). For instance, input data must often be pre-processed to derive desired features and perform appropriate quality control. Extracting features from the raw data may require complicated computations, such as convolving a kernel function with an image to identify boundaries, or computing contours, curvatures, derivatives or averages. These are then used to compute fuzzy set memberships or as inputs to interest maps. In conjunction with this process, input data should be quality controlled to ensure that corrupt data are censored or flagged with low confidence values so that their influence can be mitigated downstream. In environmental science applications, assessing data quality is often a vital component of a fuzzy logic algorithm that may even require a full-fledged fuzzy logic algorithm in its own right. After features and confidences have been computed, they may be used as input to fuzzy rules in the form of

Mamdami, Takagi-Sugeno, or fuzzy consensus reasoning. The outcomes of these rules are then aggregated and, if necessary, the result is "defuzzified" to produce a "crisp" output prediction or action. Fuzzy logic algorithms may also involve iterative components that run several times until a desired level of quality or certainty is achieved. And fuzzy logic algorithms may use combinations of different kinds of fuzzy reasoning, fuzzy numbers or even heuristics developed just for a particular problem; their common characteristic is that they mimic the human problem-solving approach of accommodating and exploiting ambiguity and postponing a "crisp" conclusion until the very last possible moment.

In the end, the fuzzy logic algorithm comprises a set of computations that provide a mapping from input data to an output prediction or action, and the methods described in this chapter are simply efficient ways for creating an appropriate mapping. If the human expert knowledge encoded in the fuzzy logic algorithm is of high quality and the algorithm is implemented correctly, the algorithm will usually do a good job on the problem it was designed to solve. On the other hand, if the human understanding of how to solve a given problem is inaccurate or incomplete, the fuzzy logic algorithm might not work as well as it potentially could.

One solution to this predicament is to use training data – pairs of input vectors and the associated ideal output, or "truth" values – to tune the fuzzy logic algorithm's parameters to optimize its performance. For instance, a training dataset might consist of the data used as input to a forecast algorithm along with subsequent measurements representing what actually happened. In this approach, the fuzzy logic algorithm is considered a function whose behavior can be modified by changing various parameters, which we may refer to collectively as the vector $\boldsymbol{\alpha}$. These might include values that control data preprocessing, parameters that describe each fuzzy set membership function (e.g., the vertices and values defining a piecewise-linear function), or the weights used in computing a fuzzy consensus. Indeed, tuning fuzzy logic algorithms is quite similar to training neural networks. In neural network case, the architecture of the network is defined by the number of hidden layers and nodes and the activation functions, and the parameters are the connection weights. Training occurs by modifying the weights to minimize the error between the neural network's outputs and the "truth" data, either by the gradient-descent backpropagation technique or some other method. A fuzzy logic algorithm has a different architecture, but it is still controlled by a set of parameters that can be adjusted to improve its performance (see Fig. 6.18). When fuzzy logic systems are optimized using training data, the result is sometimes called a *neuro-fuzzy system*. This approach to tuning a fuzzy logic system can be immensely powerful. It means that if a researcher has a good idea of what features are important and what the correct *form* (architecture) of an algorithm is but is not sure about some of the details – e.g., the ideal definition of each interest map – those details can be "filled in" using training data.

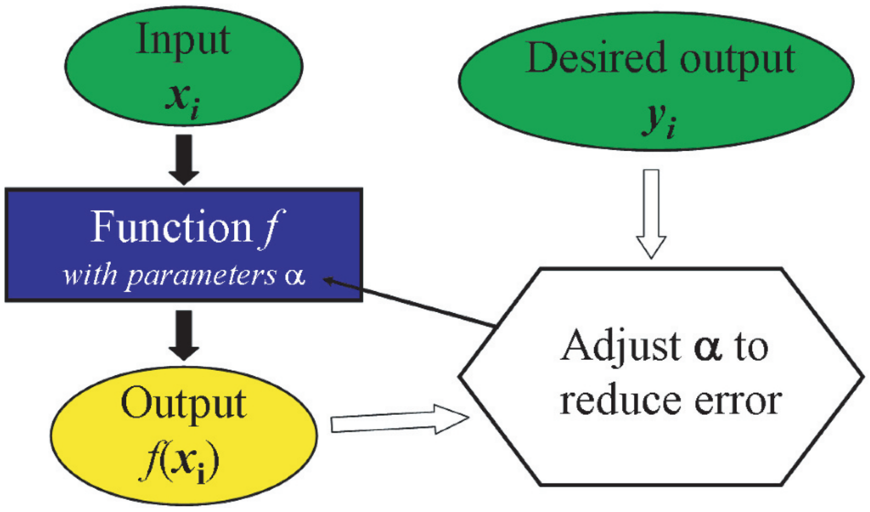

Input x_i
Desired output y_i
Function f *with parameters* α
Adjust α to reduce error
Output $f(x_i)$

Fig. 6.18 Diagram illustrating how a fuzzy logic algorithm, represented as a function f determined by a set of parameters $\boldsymbol{\alpha}$, may be tuned to improve its performance when training data in the form of x_i, y_i pairs are available.

Several considerations might inform how the fuzzy logic algorithm is tuned. First, the error function (e.g., sum of squared differences between the predicted and target values) will be easy to write down in closed form only for the simplest algorithms, so a gradient-descent style optimization method would usually require estimating the gradient using multiple evaluations. However, for some fuzzy logic systems the error function may not be differentiable or even continuous in the parameters $\boldsymbol{\alpha}$, or it might have numerous local minima besides the global minimum that could trap a gradient descent technique. Furthermore, minimizing a simple error metric like the sum of squared errors is not necessarily what is desired for environmental science applications. Often, a forecaster might be interested in maximizing the True Skill Score or another skill statistic, and a remote sensing system might be evaluated based on the area under the Receiver Operating Characteristic (ROC) curve (see Chapter 2). Evaluating a decision support system might involve a complicated simulation of the costs and benefits associated with using the system. Such evaluation functions are in general not differentiable with respect to the fuzzy logic algorithm's parameters. Therefore, a general and effective way to tune a fuzzy logic algorithm is to use a genetic algorithm, representing the vector of parameters as a chromosome (see Chapter 5). Genetic algorithms do not make any assumptions about the differentiability or even continuity of the evaluation function and have been shown to work well for tuning fuzzy logic algorithms. The most difficult task may be to define an evaluation function that fully captures the salient details of performance, carefully treating issues of rare events, for instance. Sometimes, in fact, the evaluation function itself might best be implemented as a fuzzy logic algorithm that appropriately balances performance tradeoffs in different situations.

When a fuzzy logic algorithm is tuned "on line" as it is operating, the result is called an *adaptive fuzzy system*. For instance, the combination weights w_i in a Takagi-Sugeno system (6.4) or fuzzy consensus system (6.6) may be modified based on the recent performance of the associated predictive function, giving those functions that are doing well a bit more weight when they are corroborated and reducing the weight for those that poorly match the verification data. Since the dynamics of many natural systems tend to transition between distinct domains (e.g., as they orbit strange attractors), each of which has different characteristics or phenomenology, this capability allows the fuzzy algorithm to adapt appropriately as the situation changes. However, a fuzzy system that relies heavily on this sort of dynamic tuning may not perform well when the environmental system being measured transitions suddenly from one domain to another, and it may mask a problem in a data source that might best be dealt with directly. Whenever possible, it is probably preferable to identify the different domains or hidden variables that underlie the changing performance and incorporate them into the fuzzy logic algorithm itself. On the other hand, many environmental systems are quite complicated, and treating them with an adaptive fuzzy system may be the only practical approach.

6.11 Conclusion

Fuzzy logic provides a framework for creating expert systems that use information efficiently, encode human knowledge and heuristics, and are relative straightforward to implement. A central concept is that of fuzzy sets, which may be used to represent unsharp concepts like those commonly used in human communication and reasoning. A mathematical definition for fuzzy sets, accompanied by rules for manipulating them in analogy to classical sets, has been presented. A discussion of how fuzzy membership functions may be formed and combined via logical operations was followed by a description of Mamdami, Takagi-Sugeno and fuzzy consensus methods of inference. The fuzzy consensus method provides a basis for the confidence-weighted smoothing of data and the fuzzy k-nearest neighbor method for classifying data based on a collection of examples. Fuzzy clustering was presented as a way to discover structure in datasets that could be used as a step in developing a Takagi-Sugeno style fuzzy inference system. Fuzzy logic provides a natural way to integrate data quality control and information about measurement uncertainty into algorithms. When training data are available, the performance of a fuzzy logic algorithm can be optimized by using a genetic algorithm or another technique to tune the parameters governing its behavior, though a careful choice of the objective function is necessary to obtain good results. If tuning is done during algorithm operation as verification data become available, the resulting adaptive fuzzy system can maintain good performance despite gradual changes in the environment or data sources. Fuzzy logic algorithms do not necessarily achieve the very optimum performance possible; rather, they fall into the category of "soft computing" methods that are robust, relatively easy to create and maintain, and perform very well on complex problems. These features make fuzzy logic a valuable tool for many environmental science prediction and decision support applications.

Acknowledgements Kent Goodrich first introduced me to fuzzy logic and is responsible for the story that begins this chapter, which describes how a project to produce weather hazard scores for flight service stations inadvertently resulted in a fuzzy logic algorithm. He, Jennifer Abernethy and Cathy Kessinger all read a draft of this chapter and offered helpful suggestions for improving it for which I am very grateful.

References

Albo, D. (1994). Microburst detection using fuzzy logic. *Terminal area surveillance system program project report* (49 pp.). Washington, DC: Federal Aviation Administration. [Available from the author at NCAR, P. O. Box 3000, Boulder, CO 80307.]

Albo, D. (1996). Enhancements to the microburst automatic detection algorithm. *Terminal area surveillance system program project report* (47 pp.). Washington, DC: Federal Aviation Administration. [Available from the author at NCAR, P. O. Box 3000, Boulder, CO 80307.]

Chi, Z., Hong, Y., & Tuan, P. (1996). *Fuzzy algorithms with applications to image processing and pattern recognition* (225 pp.). River Edge, NJ: World Scientific.

Delanoy, R. L., & Troxel, S. W. (1993). Machine intelligence gust front detection. *Lincoln Laboratory Journal*, *6*, 187–211.

Klir, G. J., & Folger, T. A. (1988). *Fuzzy sets, uncertainty and information* (355 pp.). Englewood Cliffs, NJ: Prentice Hall.

Klir, G. J., & Yuan, B. (Eds.). (1996). *Fuzzy sets, fuzzy logic and fuzzy systems: Selected papers by Lotfi A. Zadeh* (826 pp.). River Edge, NJ: World Scientific.

Mamdami, E. H. (1974). Applications of fuzzy logic algorithms for control of a simple dynamic plant. *Proceedings of IEEE*, *121*, 1585–1588.

McNiell, D., & Freiberger, P. (1993). *Fuzzy logic* (319 pp.). New York: Simon & Schuster.

Merritt, M. W. (1991). Microburst divergence detection for Terminal Doppler Weather Radar (TDWR). *MIT Lincoln Laboratory project report ATC-181* (174 pp). Lexington, MA: MIT Lincoln Laboratory.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Man and Cybernetics*, *15*, 116–132.

Tanaka, K. (1996). *An introduction to fuzzy logic for practical applications* (T. Tiimura, Trans., 138 pp.). New York: Springer.

Vivekanandan, J., Zrnic, D. S., Ellis, S. M., Oye, R., Ryzhkov, A. V., & Straka, J. (1999). Cloud microphysics retrieval using S-band dual polarization radar measurements. *Bulletin of American Meteorological Society*, *80*, 381–388.

Williams, J. K., & Vivekanandan, J. (2007). Sources of error in dual-wavelength radar remote sensing of cloud liquid water content. *Journal of Atmospheric and Oceanic Technology*, *24*, 1317–1336.

Yager, R. R., Ovchinnikov, S., Tong, R. M., & Nguyen, H. T. (Eds.) (1987). *Fuzzy sets and applications: Selected papers by L. A. Zadeh* (684 pp.). New York: Wiley.

Zadeh, L. H. (1965). Fuzzy sets. *Information and Control*, *8*, 338–353.

Zimmerman, H. J. (1996). *Fuzzy set theory and its applications* (435 pp.). Boston: Kluwer.