Fuzzy Logic

Lofti A. Zadeh
University of California, Berkeley

April 1988
some of these projects will be said in the section dealing with applications.

In most of the current applications of fuzzy logic, software is employed as a medium for the implementation of fuzzy algorithms and control rules. What is clear, however, is that it would be cheaper and more effective to use fuzzy logic chips and, eventually, fuzzy computers. The first logic chip was developed by Togai and Watanabe at Bell Telephone Laboratories in 1985, and it is likely to become available for commercial use in 1988 or 1989. On the heels of this important development came the announcement of a fuzzy computer designed by Yamakawa at Kumamoto University. These developments on the hardware front may lead to an expanded use of fuzzy logic not only in industrial applications but, more generally, in knowledge-based systems in which the deduction of an answer to a query requires the inference machinery of fuzzy logic.

One important branch of fuzzy logic may be called dispositional logic. This logic, as its name implies, deals with dispositions, that is, propositions that are preponderantly but not necessarily always true. For example, “snow is white” is a disposition, as are the propositions “Swedes are blond” and “high quality is expensive.” A disposition may be viewed as a usually-qualified proposition in which the qualifying quantifier “usually” is implicit rather than explicit. In this sense, the disposition “snow is white” may be viewed as the result of suppressing the fuzzy quantifier “usually” in the usability-qualified proposition usually (snow is white)

In this proposition, “usually” plays the role of a fuzzy proportion of the form shown in Figure 1.

![Figure 1. Representation of “usually" as a fuzzy proportion.](image)

The importance of dispositional logic stems from the fact that most of what is usually referred to as common sense knowledge may be viewed as a collection of dispositions. Thus, the main concern of dispositional logic lies in the development of rules of inference from common sense knowledge.

In what follows, I present a condensed exposition of some basic ideas underlying fuzzy logic and describe some representative applications. More detailed information regarding fuzzy logic and its applications may be found in the cited literature.

**Basic principles**

Fuzzy logic may be viewed as an extension of multivalued logic. Its uses and objectives, however, are quite different. Thus, the fact that fuzzy logic deals with approximate rather than precise modes of reasoning implies that, in general, the chains of reasoning in fuzzy logic are short in length, and rigor does not play as important a role as it does in classical logical systems. In a nutshell, in fuzzy logic everything, including truth, is a matter of degree.

The greater expressive power of fuzzy logic derives from the fact that it contains as special cases not only the classical two-valued and multivalued logical systems but also probability theory and probabilistic logic. The main features of fuzzy logic that differentiate it from traditional logical systems are the following:

1. In two-valued logical systems, a proposition p may be either true or false. In multivalued logical systems, a proposition may be true or false or have an intermediate truth value, which may be an element of a finite or infinite truth value set T. For example, if T is the unit interval, then a truth value in fuzzy logic, for example, “very true,” may be interpreted as a fuzzy subset of the unit interval. In this sense, a fuzzy truth value may be viewed as an imprecise characterization of a numerical truth value.

2. The predicates in two-valued logic are constrained to be crisp in the sense that the denotation of a predicate must be a nonfuzzy subset of the universe of discourse. In fuzzy logic, the predicates may be crisp—for example, “mortal,” “even,” and “father of”—or, more generally, fuzzy—for example, “ill,” “tired,” “large,” “tall,” “much heavier,” and “friend of.”

3. Two-valued as well as multivalued logics allow only two quantifiers: “all” and “some.” By contrast, fuzzy logic allows, in addition, the use of fuzzy quantifiers exemplified by “most,” “many,” “several,” “few,” “much of,” “frequently,” “occasionally,” “about ten,” and so on. Such quantifiers may be interpreted as fuzzy numbers that provide an imprecise characterization of the cardinality of one or more fuzzy or nonfuzzy sets. In this perspective, a fuzzy quantifier may be viewed as a second-order fuzzy predicate. Based on this view, fuzzy quantifiers may be used to represent the meaning of propositions containing fuzzy probabilities and thereby make it possible to manipulate probabilities within fuzzy logic.

4. Fuzzy logic provides a method for representing the meaning of both nonfuzzy and fuzzy predicate-modifiers exemplified by “not,” “very,” “more or less,” “extremely,” “slightly,” “much,” “a little,” and so on. This, in turn, leads to a system for computing with linguistic variables, that is, variables whose values are words or sentences in a natural or synthetic language. For example, “Age” is a linguistic variable when its values are assumed to be “young,” “old,” “very young,” “not very old,” and so forth. More about linguistic variables will be said at a later point.

5. In two-valued logical systems, a proposition p may be qualified, principally by associating with p a truth value, “true” or “false”; a modal operator such as “possible” or “necessary”; and an intensional operator such as “know” or “believe.” Fuzzy logic has three principal modes of qualification:

- **truth-qualification**, as in
  
  (Mary is young) is not quite true

- **probability-qualification**, as in
  
  (Mary is young) is unlikely

- **possibility-qualification**, as in
  
  (Mary is young) is almost impossible
bility is “almost impossible.”

An important issue in fuzzy logic relates to inference from qualified propositions, especially from probability-qualified propositions. This issue is of central importance in the management of uncertainty in expert systems and in the formalization of common sense reasoning. In the latter, it’s important to note the close connection between probability-qualification and usability-qualification and the role played by fuzzy quantifiers. For example, the disposition

Swedes are blond

may be interpreted as

most Swedes are blond;

or, equivalently, as

(Swede is blond) is likely,

where “likely” is a fuzzy probability that is numerically equal to the fuzzy quantifier “most”; or, equivalently, as

usually (a Swede is blond),

where “usually” qualifies the proposition “a Swede is blond.”

As alluded earlier, inference from propositions of this type is a main concern of dispositional logic. More about this logic will be said at a later point.

Meaning representation and inference

A basic idea serving as a point of departure in fuzzy logic is that a proposition \( p \) in a natural or synthetic language may be viewed as a collection of elastic constraints, \( C_1, \ldots, C_n \), which restrict the values of a collection of variables \( X = (X_1, \ldots, X_n) \). In general, the constraints as well as the variables they constrain are implicit rather than explicit in \( p \). Viewed in this perspective, representation of the meaning of \( p \) is, in essence, a process by which the implicit constraints and variables in \( p \) are made explicit. In fuzzy logic, this is accomplished by representing \( p \) in the so-called canonical form

\[ p \rightarrow X \text{ is } A \]

in which \( A \) is a fuzzy predicate or, equivalently, an \( n \)-ary fuzzy relation in \( U \), where \( U = U_1 \times U_2 \times \ldots \times U_n \), and \( U_i \), \( i = 1, \ldots, n \), is the domain of \( X_i \). Representation of \( p \) in its canonical form requires, in general, the construction of an explanatory database and a test procedure that tests and aggregates the test scores associated with the elastic constraints \( C_1, \ldots, C_n \).

In more concrete terms, the canonical form of \( p \) implies that the possibility distribution of \( X \) is equal to \( A \)—that is,

\[ \Pi_X = A \]

which in turn implies that

\[ \text{Poss} \{X = u\} = \mu_A(u), \quad u \in U \]

where \( \mu_A \) is the membership function of \( A \) and Poss \( \{X = u\} \) is the possibility that \( X \) may take \( u \) as its value. Thus, when the meaning of \( p \) is represented in the form of Equation 1, it signifies that \( p \) induces a possibility distribution \( \Pi_X \) that is equal to \( A \), with \( A \) playing the role of an elastic constraint on a variable \( X \) that is implicit in \( p \). In effect, the possibility distribution of \( X \), \( \Pi_X \), is the set of possible values of \( X \) with the understanding that possibility is a matter of degree. Viewed in this perspective, a proposition \( p \) constrains the possible values that \( X \) can take and thus defines its possibility distribution. This implies that the meaning of \( p \) is defined by (1) identifying the variable that is constrained and (2) characterizing the constraint to which the variable is subjected through its possibility distribution. Note that Equation 1 asserts that the possibility that \( X \) can take as its value is numerically equal to the grade of membership, \( \mu_A(u) \), of \( u \) in \( A \).

As an illustration, consider the proposition

\[ p \models \text{John is tall} \]

in which the symbol \( \models \) should be read as “denotes” or “is equal to by definition.” In this case, \( X = \text{Height(John)} \), \( A = \text{TALL} \), and the canonical form of \( p \) reads

\[ \text{Poss} \{\text{Height(John)} = u\} = \mu_{\text{TALL}}(u) \]

where \( \mu_{\text{TALL}} \) is the membership function of TALL and \( \text{TALL} \) is the grade of membership of \( u \) in TALL or, equivalently, the degree to which a numerical height \( u \) satisfies the constraint induced by the relation TALL.

When \( p \) is a conditional proposition, its canonical form may be expressed as “\( Y \text{ is } B \text{ if } X \text{ is } A \),” implying that \( p \) induces a conditional possibility distribution of \( Y \) given \( X \), written as \( \Pi_{X \rightarrow Y} \). In fuzzy logic, \( \Pi_{X \rightarrow Y} \) may be defined in a variety of ways, among which is a definition consistent with the definition of implication in Lakasiewicz’s \( L_{\text{fuzzy}} \) logic. In this case, the conditional possibility distribution function, \( \pi_{X \rightarrow Y} \), which defines \( \Pi_{X \rightarrow Y} \), may be expressed as

\[ \pi_{X \rightarrow Y}(u, v) = 1 \land (1 - \mu_A(u) + \mu_B(v)), \quad u \in U, \quad v \in V, \]

where

\[ \pi_{X \rightarrow Y}(u, v) \models \text{Poss} \{X = u, Y = v\} \]

\( \mu_A \) and \( \mu_B \) denote the membership functions of \( A \) and \( B \), respectively; and \( \land \) denotes the operator min.

When \( p \) is a quantified proposition of the form

\[ p \models Q \text{ A's are } B's \]

for example,

\[ p \models \text{most tall men are not very fat} \]

where \( Q \) is a fuzzy quantifier and \( A \) and \( B \) are fuzzy predicates, the constrained variable, \( X \), is the proportion of \( B \)'s in \( A \)'s, with \( Q \) representing an elastic constraint on \( X \). More specifically, if \( U \) is a finite set \( \{u_1, \ldots, u_n\} \), the proportion of \( B \)'s in \( A \)'s is defined as the relative sigma-count

\[ \Sigma \text{Count}(B/A) = \frac{\sum_j \mu_B(u_j) \land \mu_A(u_j)}{\sum_j \mu_A(u_j)} \quad (3) \]

where \( \mu_A(u) \) and \( \mu_B(u) \) denote the grades of membership of \( u \) in \( A \) and \( B \), respectively. Thus, expressed in its canonical form, Equation 3 may be written as

\[ \Sigma \text{Count}(B/A) = Q \]

which places in evidence the constrained variable, \( X \), in \( p \) and the elastic constraint, \( Q \), to which \( X \) is subjected. Note that \( X \) is the relative sigma-count of \( B \) in \( A \).

The concept of a canonical form pro-
vides an effective framework for formulating the problem of inference in expert systems. Specifically, consider a knowledge base, KB, which consists of a collection of propositions \{p_1, \ldots, p_n\}. Typically, a constituent proposition, \(p_i\), \(i = 1, \ldots, N\), may be (1) a fact that may be expressed in a canonical form as "\(X\) is \(A\)" or (2) a rule that may be expressed in a canonical form as "\(X\) is \(A\), where \(A\) is a fuzzy probability whose denotation as a fuzzy subset of the unit interval is the same as that of the fuzzy quantifier \(Q\) and \(X\) is chosen at random in \(U\)."

Now if \(p_i\) induces a possibility distribution \(\pi_{X_1, \ldots, X_n}\), where \(X_1, \ldots, X_n\) are the variables constrained by \(p_i\), then the possibility distribution \(\pi_{X_1, \ldots, X_n}\) which is induced by the totality of propositions in \(KB\) is given by the intersection of the constrained variables for each proposition.

Now suppose that we are interested in inferring the value of a specified function \(f(X_1, \ldots, X_n)\), \(f: U \rightarrow V\), of the variables constrained by the knowledge base. Because of the incompleteness and imprecision of the information resident in \(KB\), what we can deduce, in general, is not the exact value of \(f(X_1, \ldots, X_n)\) but its possibility distribution, \(\Pi_f\). By employing the extension principle, it can be shown that the possibility distribution function of \(f\) is given by the solution of the nonlinear program

\[
\Pi_f(x_1, \ldots, x_n) = \text{max} \{ \Pi_i(x_1, \ldots, x_n) \} \quad (4)
\]

subject to the constraint

\[
v = f(u_1, \ldots, u_n)
\]

where \(u_i \in U_i, i = 1, \ldots, n\), and \(v \in V\). The reduction to the solution of a nonlinear program constitutes the principal tool for inference in fuzzy logic.

**Fuzzy syllogisms.** A basic fuzzy syllogism in fuzzy logic that is of considerable relevance to the rules of combination of evidence in expert systems is the intersection/product syllogism—a syllogism that serves as a rule of inference for quantified propositions. This syllogism may be expressed as the inference rule

\[
Q_1, A's are B's
Q_2, (A and B)'s are C's
\]

\[
(Q_1 \otimes Q_2) A's are (B and C)'s
\]

in which \(Q_1\) and \(Q_2\) are fuzzy quantifiers, \(A, B,\) and \(C\) are fuzzy predicates, and \(Q_1 \otimes Q_2\) is the product of the fuzzy numbers \(Q_1\) and \(Q_2\) in fuzzy arithmetic. (See Figure 2.) For example, as a special case of Equation 5, we may write

most students are single

a little more than a half of single students are male

(most \(\otimes\) a little more than a half) of students are single and male

Since the intersection of \(B\) and \(C\) is contained in \(C\), the following corollary of Equation 5 is its immediate consequence.

\[
Q_1, A's are B's
Q_2, (A and B)'s are C's
\]

\[
\geq (Q_1 \otimes Q_2) A's are C's
\]

where the fuzzy number \(\geq (Q_1 \otimes Q_2)\) should be read as "at least \((Q_1 \otimes Q_2)\)." In particular, if the fuzzy quantifiers \(Q_1\) and \(Q_2\) are monotone increasing (for example, when "most\(Q_1\) = \(Q_2\) = most of students are single and male, then

\[
\geq (Q_1 \otimes Q_2) = Q_1 \otimes Q_2
\]

and Equation 6 becomes

\[
Q_1, A's are B's
Q_2, (A and B)'s are C's
\]

(7)

\[
(Q_1 \otimes Q_2) A's are C's
\]

Furthermore, if \(B\) is a subset of \(A\), then \(A\) and \(B = B\), and Equation 7 reduces to the chaining rule

\[
Q_1, A's are B's
Q_2, B's are C's
\]

(8)

\[
(Q_1 \otimes Q_2) A's are C's
\]

For example,

most students are undergraduates

most undergraduates are young

most\(^2\) students are young

where "most\(^2\)" represents the product of the fuzzy number "most" with itself (see Figure 3).

What is important to observe is that the chaining rule expressed by Equation 8
Inference with fuzzy probabilities

An example of an important problem to which the reduction to a nonlinear program may be applied is the following. Assume that from a knowledge base $KB = \{p_0, \ldots, p_n\}$ in which the constituent propositions are true with probability one, we can infer a proposition $q$ which, like the premises, is true with probability one. Now suppose that each $p_i$ in $KB$ is replaced with a probability-qualified proposition "$p_i \overset{sl}{=} p_i \lambda_i$" in which $\lambda_i$ is a fuzzy probability. For example

$$p_i \overset{sl}{=} X \text{ is small}$$

and

$$p_i \overset{sl}{=} X \text{ is small is very likely}$$

As a result of the qualification of the $p_i$, the conclusion, $q$, will also be a probability-qualified proposition that may be expressed as

$$q' = q \lambda$$

in which $\lambda$ is a fuzzy probability. The problem is to determine $\lambda$ as a function of the $\lambda_i$, if such a function exists. A special case of this problem, which is of particular relevance to the management of uncertainty in expert systems, is one in which the fuzzy probabilities $\lambda_i$ are close to unity. We shall say that the inference process is compositional if $\lambda$ can be expressed as a function of the $\lambda_i$; it is robust if whenever the $\lambda_i$ are close to unity, so is $\lambda$.

By reducing the determination of $\lambda$ to the solution of a nonlinear program, it can be shown that, in general, the inference process is not compositional if the $\lambda_i$ and $\lambda$ are numerical probabilities. This result calls into question the validity of the rules of combination of evidence in those expert systems in which the certainty factor of the conclusion is expressed as a function of the certainty factors of the premises. However, compositional inferences hold, in general, if the $\lambda_i$ and $\lambda$ are assumed to be fuzzy probabilities, for this allows the probability of $q$ to be interval-valued when the $\lambda_i$ are numerical probabilities, which is consistent with known results in inductive logic.

Another important conclusion relating to the robustness of the inference process is that, in general, robustness does not hold without some restrictive assumptions on the premises. For example, the brittleness of the transitivity of implication is an instance of the lack of robustness when no assumptions are made regarding the fuzzy predicates $A$, $B$, and $C$. On the other hand, if in the inference schema

$$X \text{ is } A$$
$$Y \text{ is } B \text{ if } X \text{ is } A$$
$$Y \text{ is } B$$

the major premise is replaced by "$X \text{ is } A \text{ is probable},"$ where "probable" is a fuzzy probability close to unity, then it can be shown that, under mildly restrictive assumptions on $A$, the resulting conclusion may be expressed as "$Y \text{ is } B \text{ is } \geq \text{ probable},"$ where "$\geq \text{ probable}"$ is a fuzzy probability that, as a fuzzy number, is greater than or equal to the fuzzy number "probable." In this case, then, robustness does hold, for if "probable" is close to unity, so is "$\geq \text{ probable}"".
application of fuzzy logic to the premises in question is both robust and compositional.

A more complex problem is presented by what in Mycin and Prospector corresponds to the conjunctive combination of evidence. Stated in terms of quantified propositions, the inference rule in question may be expressed as

\[ Q \text{ is } A \text{'s are } C\text{'s} \]
\[ Q \text{ is } B\text{'s are } C\text{'s} \]
\[ Q (A \text{ and } B)\text{'s are } C\text{'s} \]

where the value of \( Q \) is to be determined. To place in evidence the symmetry of evidence, we shall refer to the rule in question as the rule of evidence. Stated in terms of quantified variables, the assumption may be written as

\[ \sum \text{Count} (A \cap B / C) = \sum \text{Count} (A / C) \sum \text{Count} (B / C) \]

where \( \cap \) denotes the intersection of fuzzy sets.

To determine the value of \( Q \) in Equation 9 we have to compute the relative sigma-count of \( C \) in \( A \cap B \). It can be verified that, under the assumption (Equation 11), the sigma-count in question is given by

\[ \sum \text{Count} (C / A \cap B) = \frac{\sum \text{Count} (C / A) \sum \text{Count} (C / B) \delta}{\sum \text{Count} (A \cap B) \sum \text{Count} (C)} \]

where the factor \( \delta \) is expressed by

\[ \frac{\sum \text{Count} (A) \sum \text{Count} (B)}{\sum \text{Count} (A \cap B) \sum \text{Count} (C)} \]

Inspection of Equation 12 shows that the assumption expressed by Equation 11 does not ensure the compositional property of \( Q \). However, it can be shown that compositional property can be achieved through the use of the concept of a relative sigma-count, which is defined as

\[ \delta \sum \text{Count} (B / A) = \frac{\sum \text{Count} (B / A)}{\sum \text{Count} (-B / A)} \]

where \(-B\) denotes the negation of \( B \). The use of sigma-counts in place of sigma-sets is analogous to the use of odds instead of probabilities in Prospector, and it serves the same purpose.

**Interpolation**

An important problem that arises in the operation of any rule-based system is the following. Suppose the user supplies a fact that, in its canonical form, may be expressed as a "linear" combination of fuzzy subsets. The problem is: Given an input \( n \)-tuple \( R_1, \ldots, R_m \) with the \( i \)th row of \( R \) \( i = 1, \ldots, m \), by employing \( \wedge \) (min) as the aggregation operator. Thus,

\[ r_i = r_{i1} \land r_{i2} \land \ldots \land r_{in} \]

which implies that \( r_i \) may be interpreted as a conservative measure of agreement between the input \( n \)-tuple \( R_i \), \( i = 1, \ldots, m \). Then, employing \( \gamma \) as a weighting coefficient, the desired expression for \( X_{n+1} \), may be written as a "linear" combination

\[ X_{n+1} = \gamma_1 Z_1 + \ldots + \gamma_m Z_m \]

in which \( \gamma_i \) denotes the weight, and \( \gamma_i \land Z_i \) is a fuzzy set defined by

\[ (\gamma \wedge Z_i) = \gamma_i (R_i) \land Z_i \]

The above approach ceases to be effective, however, when \( R \) is a sparse relation in the sense that no row of \( R \) has a high degree of consistency with the input \( n \)-tuple. For such cases, a more general interpolation technique has to be employed.
Basic rules of inference

One distinguishing characteristic of fuzzy logic is that premises and conclusions in an inference rule are generally expressed in canonical form. This representation places in evidence the fact that each premise is a constraint on a variable and that the conclusion is an induced constraint computed through a process of constraint propagation — a process that, in general, reduces to the solution of a nonlinear program. The following briefly presents — without derivation — some of the basic inference rules in fuzzy logic. Most of these rules can be deduced from the basic inference rule expressed by Equation 4.

The rules of inference in fuzzy logic may be classified in a variety of ways. One basic class is categorical rules, that is, rules that do not contain fuzzy quantifiers. A more general class is dispositional rules, rules in which one or more premises may contain, explicitly or implicitly, the fuzzy quantifier “usually.” For example, the inference rule known as the entailment principle:

\[
\begin{align*}
X & \text{ is } A \\
A \subset B \\
Y & \text{ is } B
\end{align*}
\]

where \(X\) is a variable taking values in a universe of discourse \(U\), and \(A\) and \(B\) are fuzzy subsets of \(U\), is a categorical rule. On the other hand, the dispositional entailment principle is an inference rule of the form

\[
\begin{align*}
X & \text{ is } A \\
A \subset B \\
Y & \text{ is } B
\end{align*}
\] (14)

where \(A \subset B\) is the intersection of \(A\) and \(B\) defined by

\[
\mu_A \cap \mu_B(u) = \mu_A(u) \wedge \mu_B(u), \quad u \in U
\]

Cartesian product.

\[
\begin{align*}
X & \text{ is } A \\
Y & \text{ is } B \\
(X, Y) & \text{ is } A \times B
\end{align*}
\]

where \((X, Y)\) is a binary variable and \(A \times B\) is defined by

\[
\mu_{A \times B}(u, v) = \mu_A(u) \wedge \mu_B(v), \quad u \in U, v \in V
\]

Projection rule.

\[
\begin{align*}
X & \text{ is } A \\
R & \text{ is } \mu_X \cap \mu_R
\end{align*}
\]

where \(\mu_X\), the projection of the binary relation \(R\) on the domain of \(X\), is defined by

\[
\mu_{\mu_X \cap \mu_R}(u, v) = \mu_X(u) \wedge \mu_R(u, v), \quad u \in U, v \in V
\]

where \(\mu_{\mu_X \cap \mu_R}(u, v)\) is the membership function of \(R\) and the supremum is taken over \(v \in V\).

Compositional rule.

\[
\begin{align*}
X & \text{ is } A \\
(X, Y) & \text{ is } R \\
Y & \text{ is } A \circ R
\end{align*}
\]

where \(A \circ R\), the composition of the unary relation \(A\) with the binary relation \(R\), is defined by

\[
\mu_{A \circ R}(v) = \sup_u (\mu_A(u) \wedge \mu_R(u, v))
\]

The compositional rule of inference may be viewed as a combination of the conjunctive and projection rules.

Generalized modus ponens.

\[
\begin{align*}
X & \text{ is } A \\
Y & \text{ is } C \text{ if } X \text{ is } B \\
Y & \text{ is } A \circ (\sim B \circ C)
\end{align*}
\]

where \(\sim B\) denotes the negation of \(B\) and the bounded sum is defined by

\[
\mu_{B \circ C}(u, v) = 1 \wedge (1 - \mu_B(u)) + \mu_C(v)
\]

An important feature of the generalized modus ponens, which is not possessed by the modus ponens in binary logical systems, is that the antecedent “\(X\) is \(B\)” need not be identical with the premise “\(X\) is \(A\)”.

It should be noted that the generalized modus ponens is related to the interpolation rule which was described earlier. An additional point that should be noted is that the generalized modus ponens may be regarded as an instance of the compositional rule of inference.

Dispositional modus ponens. In many applications involving common sense reasoning, the premises in the generalized modus ponens are usually-qualified. In such cases, one may employ a dispositional version of the modus ponens. It may be expressed as

\[
\begin{align*}
X & \text{ is } A \\
Y & \text{ is } B \text{ if } X \text{ is } A \\
Y & \text{ is } B
\end{align*}
\]

where “usually” is the square of “usually” (see Figure 4). For simplicity, it’s assumed that the premise “\(X\) is \(A\)” matches the antecedent in the conditional proposition; also, the conditional proposition is interpreted as the statement, “The

April 1988

89
value of the fuzzy conditional probability of $B$ given $A$ is the fuzzy number USUALLY.

**Extension principle.** The extension principle plays an important role in fuzzy logic by providing a mechanism for computing induced constraints. More specifically, assume that a variable $X$ taking values in a universe of discourse $U$ is constrained by the proposition "$X$ is $A$." Furthermore, assume that $f$ is a mapping from $U$ to $V$ so that $X$ is mapped into $f(X)$. The question is: What is the constraint on $f(X)$ which is induced by the constraint on $X$?

The answer provided by the extension principle may be expressed as the inference rule

$$X \text{ is } A \quad \Rightarrow \quad f(X) \text{ is } f(A)$$

where the membership function of $f(A)$ is defined by

$$\mu_{f(A)}(v) = \sup_{u \in U} \mu_{A}(u)$$

subject to the condition

$$v = f(u), \ u \in U, \ v \in V$$

In particular, if the function $f$ is 1:1, then Equation 15 simplifies to

$$\mu_{f(A)}(v) = \mu_{A}(v^{-1}), \ v \in V$$

where $v^{-1}$ is the inverse of $v$. For example,

$$X \text{ is small} \quad \Rightarrow \quad X^2 \text{ is small}^2$$

and

$$\mu_{\text{SMALL}}(v) = \mu_{\text{SMALL}}(\sqrt{v})$$

As in the case of the entailment rule, the dispositional version of the extension principle has the simple form

$$\text{usually } (X \text{ is } A) \quad \Rightarrow \quad \text{usually } (f(X) \text{ is } f(A))$$

The dispositional extension principle plays an important role in inference from common sense knowledge. In particular, it is one of the inference rules that play an essential role in answering the questions posed in the introduction.

**The linguistic variable and its application to fuzzy control**

A basic concept in fuzzy logic that plays a key role in many of its applications, especially in the realm of fuzzy control and fuzzy expert systems, is a **linguistic variable**.

A linguistic variable, as its name suggests, is a variable whose values are words or sentences in a natural or synthetic language. For example, "Age" is a linguistic variable if its values are "young," "very young," "very old," "not young," "not very young," "old," "not old," "not very old," and so on.

In general, the values of a linguistic variable can be generated from a **primary term** (for example, "young") and its antonym ("old"), a collection of modifiers ("not," "very," "more or less," "quite," "not very," etc.), and the connectives "and" and "or." For example, one value of "Age" may be "not very young and not very old." Such values can be generated by a context-free grammar. Furthermore, each value of a linguistic variable represents a possibility distribution, as shown in Figure 5 for the variable "Age." These possibility distributions may be computed from the given possibility distributions of the primary term and its antonym through the use of attributed grammar techniques.

An interesting application of the linguistic variable is embodied in the fuzzy car conceived and designed by Sugeno of the Tokyo Institute of Technology. The car's fuzzy-logic-based control system lets it move autonomously along a track with rectangular turns and park in a designated space (see Figure 6). An important feature is the car's ability to learn from examples.

The basic idea behind the Sugeno fuzzy car is the following. The controlled variable $Y$, which is the steering angle, is assumed to be a function of the state variables $X_1, X_2, X_3, \ldots, X_n$, which represent the distances of the car from the boundaries of the track at a corner (see Figure 7). These values are treated as linguistic variables, with the primary terms represented as triangular possibility distributions (see Figure 8).
The control policy is represented as a finite collection of rules of the form

\[ R_i: \text{if } (X_1 \text{ is } A_1') \text{ and } \ldots \text{ and } (X_n \text{ is } A_n'), \text{ then } Y' = a_1' X_1 + \ldots + a_n' X_n \]

where \( R_i \) is the \( i \)th rule; \( A_j' \) is a linguistic value of \( X_j \) in \( R_i \); \( Y' \) is the value of the control variable suggested by \( R_i \); and \( a_1', \ldots, a_n' \) are adjustable parameters, which define \( Y' \) as a linear combination of the state variables.

In a given state \( (X_1, \ldots, X_n) \), the truth value of the antecedent of \( R_i \) may be expressed as

\[ W_i = A_1'(X_1) \land \ldots \land A_n'(X_n) \]

where \( A_j'(X_j) \) is the grade of membership of \( X_j \) in \( A_j' \). The aggregated value of the controlled variable \( Y \) is computed as the normalized linear combination

\[ Y = \frac{W_1 Y_1 + \ldots + W_n Y_n}{W_1 + \ldots + W_n} \quad (16) \]

Thus, Equation 16 may be interpreted as the result of a weighted vote in which the value suggested by \( R_i \) is given the weight \( W_i/(W_1 + \ldots + W_n) \).

The values of the coefficients \( a_1', \ldots, a_n' \) are determined through training. Training consists of an operator guiding a model car along the track a few times until an identification algorithm converges on parameter values consistent with the control rules. By its nature, the training process cannot guarantee that the identification algorithm will always converge on the correct values of the coefficients. The justification is pragmatic: the system works in most cases.

Variations on this idea are embodied in most of the fuzzy-logic-based control systems developed so far. Many of these systems have proven to be highly reliable and superior in performance to conventional systems.

Since most rules in expert systems have fuzzy antecedents and consequents, expert systems provide potentially important applications for fuzzy logic. For example:

IF the search "space" is moderately small
THEN exhaustive search is feasible

IF a piece of code is called frequently
THEN it is worth optimizing

April 1988
IF large oil spill or strong acid spill
THEN emergency is strongly suggested

The fuzziness of such rules is a consequence of the fact that a rule is a summary, and summaries, in general, are fuzzy. However, in the context of expert systems and fuzzy logic control, fuzziness has the positive effect of reducing the number of rules needed to approximately characterize a functional dependence between two or more variables.

Fuzzy hardware. Several expert system shells based on fuzzy logic are now commercially available, among them ReveAl and Flops. The seminal work of Togai and Watanabe at Bell Telephone Laboratories, which resulted in the development of a fuzzy logic chip, set the stage for using such chips in fuzzy-logic-based expert systems and, more generally, in rule-based systems not requiring a high degree of precision.

More recently, the fuzzy computer developed by Yamakawa of Kumamoto University has shown great promise as a general-purpose tool for processing linguistic data at high speed and with remarkable robustness. Togai and Watanabe's fuzzy inference chip consists of four major components: a rule set memory, an inference processor, a controller, and I/O circuitry. In a recent implementation, a rule set memory is realized by a random-access memory. In the inference processor, there are 16 data paths; one data path is laid out for each rule. All 16 rules on the chip are executed in parallel. The chip requires 64 clock cycles to produce an output. This translates to an execution speed of approximately 250,000 fuzzy logical inferences per second (FLIPS) at 16 megahertz.
Lofti A. Zadeh joined the Department of Electrical Engineering at the University of California, Berkeley, in 1959, serving as its chairman from 1963 to 1968. Earlier he was a member of the electrical engineering faculty at Columbia University. He has held a number of visiting appointments, among them a visiting scientist appointment at the IBM Research Laboratory, San Jose, and a visiting scholar appointment at the AI Center, SRI International.

Zadeh's work centered on system theory and decision analysis until 1965 when his interests shifted to the theory of fuzzy sets and its applications.

An alumnus of the University of Tehran in Iran, MIT, and Columbia University, Zadeh is a fellow of the IEEE and AAAS and a member of the National Academy of Engineering. He holds a doctorate honoris causa from Paul Sabatier University, France, in recognition of his development of the theory of fuzzy sets.

Zadeh's address is Computer Science Division, Dept. of EECS, University of California, Berkeley, CA 94720.

April 1988