Artificial Intelligence: a concise introduction

Vicenç Torra

School of Informatics
University of Skövde
54128 Skövde, Sweden
e-mail: vtorra@his.se

Chapter 3. Knowledge representation

We have seen in the first chapter that initial commercial success within AI was due to expert systems. Expert systems are systems that make diagnosis, questioning a user, using background knowledge on the application domain. Knowledge-based systems is the usual current term for this type of systems. They use intensively knowledge on the application domain.

Although there exist several ways to express knowledge, knowledge-based systems are traditionally seen according to the structure in Figure 1. The main components are the following ones.

- Knowledge base. Background knowledge on the application domain.
- Inference engine. It decides how to apply the knowlege in order to obtain new knowledge.
- User interface. It is in charge of communicating conclusions to the user, and ask the user the information needed by the inference engine.

In addition, we have a few other components that can be used either to create the system (e.g., to incorporate knowledge) or to help the user when describing the solutions. They are the following components.

- Explanatory subsystem. It builds explanations about why and how conclusions are obtained. These explanations are given to the user.
- Knowledge acquisition module. This is used to build the knowledge base. It interacts with the expert interface to create new knowledge that will be added into the knowledge base.
- Expert interface. This component interacts with the experts to help them in the knowledge elicitation process.
- Machine learning. Knowledge bases can be either built using knowledge elicitated from experts or using knowledge extracted from data. Machine learning is used for this latter process. This component is in charge of this process. In addition, machine learning can also be used to improve the quality of the knowledge base taking into account the experience of the system.

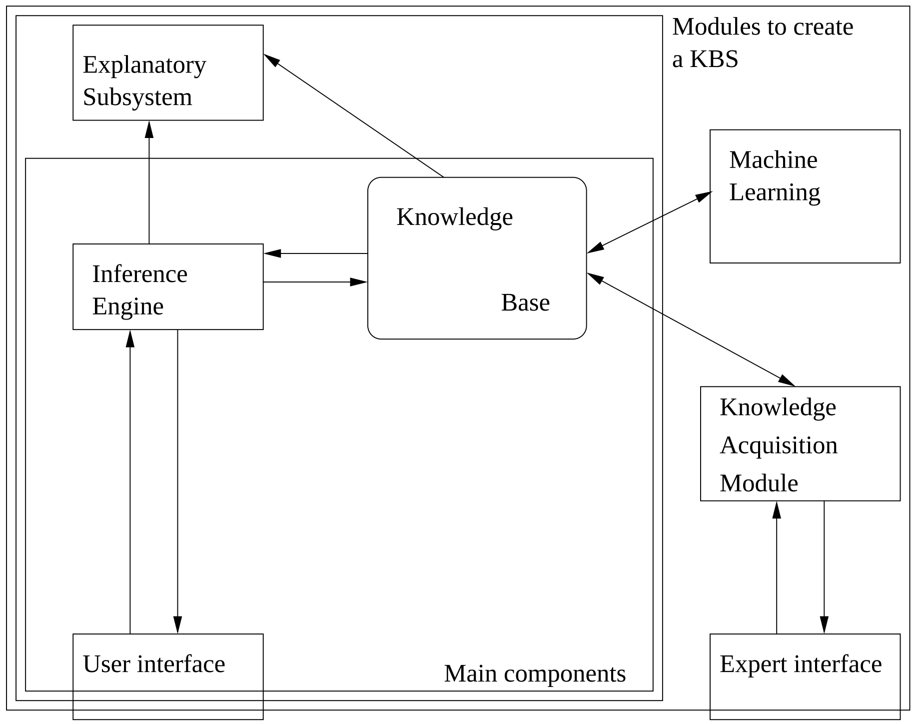

Fig. 1. Classical architecture of a knowledge-based system.

1 Building knowledge-based systems

1.1 Knowledge engineering

Knowledge engineering is the area of AI that focus on the construction of knowledge-based systems. Building a knowledge-based system is about the process of building a model.

Knowledge acquisition is the proces of explicitly stating the knowledge needed in the system. That is, the construction of the knowledge base. In this process, a knowledge engineer works with an expert on the application field. Some methodologies have been defined to help on this process. For example, KADS.

1.2 Machine learning

Machine learning algorithms have been developed to extract knowledge from data. These algorithms depend on the type of knowledge representation formalism. For each of the knowledge representation formalisms discussed in Section 2 we may consider a way to build knowledge from data. See e.g., algorithms for learning rules.

2 Knowledge-representation formalisms

– Logics and languages based on logics.
– Rules.
– Semantic networks and frames.
– Subsymbolic representation. Neural networks.

3 Logics

Propositional logic (also known as zeroth-order logic) and predicate logic (also known as first-order logic) are two examples of logics.

Deductive systems (proof system) permit to infer that one formula is a logical consequence of another. From a knowledge representation point of view, this corresponds to the inference engine.

In propositional logic, natural deduction is one of the deductive systems. It uses a few number of rules of inferences for deriving consequences (one or more propositions) from premises (a set of propositions). The modus ponens is one of these rules. Resolution is another deductive system.

In predicate logic, resolution is one of the deductive systems used. Another is the sequent calculus.

A deductive system is sound when the formulas obtained are logically valid. A deductive system is complete when all logically valid formulas can be obtained.

Recall that a sentence in predicate logic is a well-formed formula with no free variables. In propositional logic, we usually use sentences, formulas and propositions are mainly synonym terms as no variables are included. But in this case, we may require that the proposition is interpretable and thus being able to be either true or false. A formula is logically valid when it is true for all interpretations. We write that ϕ is logically valid as $\models \phi$. A formula ϕ_2 is a logical consequence of another ϕ_1 when any interpretation that makes true ϕ_1 makes also true ϕ_2.

Modal logics and description logics are other types of logics used in artificial intelligence. Modal logics extends propositional and predicate logic with the two operators "necessarily" and "possibly". Description logics can be said to be a logic with an expressive power between propositional and predicate logics in a way that is decidable.

4 Rule-based systems

This type of systems have the knowledge represented in terms of if-then rules. Conclusions are obtained applying the rules. Let us review the main components for this type of system.

– Knowledge base. Knowledge is represented in terms of rules. They are expressions of the following type.

```
If <condition> then <conclusion>
```

For example, [2] (p. 370) includes the following rule:

```
If: (1) the stain of the organism is gram negative, and
    (2) the morphology of the organism is rod, and
    (3) (a) the aerobicity of the organism is aerobic, or
        (b) the aerobicity of the organism is unknown
Then: there is suggestive evidence (0.6) that
         the class of the organism is enterobacteriaceae
```

- Working memory. This is the place in the rule-based system that includes a memory with all facts that are known to be true from the beginning or that have been deduced from other facts using the rules in the knowledge base. Properly speaking, this working memory can be seen as a part of the knowledge base.
- Inference engine. It describes how to infer new facts from the ones already known and the rules in the knowledge base. It has to solve the control problem (which rule to apply).
  The inference engine is usually defined as a loop with the following steps.
  Step 1. Retrieval. The set of all applicable rules are obtained. This corresponds to the conflict set.
  Step 2. Refinement. A rule is selected from the conflict set.
  Step 3. Execution. The selected rule is applied.

5 Approximate reasoning: reasoning under uncertainty

We have already seen that there are different knowledge representation formalisms. Each of them has been developed focusing on some aspects of the knowledge that they want to represent in an intuitive way. For example, semantic networks and frames focus on the relationships between concepts. Neural networks avoid the symbolic representation for the sake of a better approximation (less error).

Approximate reasoning techniques encompass all those approaches for knowledge representation and reasoning when the information available is not of enough quality. The goal is to be able to obtain correct inferences when the available information has for example uncertainty, imprecision, and inconsistency. This type of information is usual in real world applications, and, thus, we need systems to be able to deal with it.

Probability theory is the most well known approach in approximate reasoning. Bayesian decision theory has been developed for decision in case of uncertainty. Nevertheless, there are other approaches and tools for approximate reasoning. We begin defining a few concepts that we need later in order to classify the existing tools. We start with incompleteness and uncertainty.

- **Incompleteness.** Knowledge is incomplete in an application domain when it does not cover all the needs the system has.

Given a (discrete) reference set Ω, a probability measure on Ω is a set function (i.e., $P(A)$ for $A \subset \Omega$) in [0,1] that satisfies the following axioms:

1. $P(\emptyset) = 0$ (boundary condition)
2. $P(\Omega) = 1$ (boundary condition)
3. $P(A \cup B) = P(A) + P(B)$ if $A \cap B = \emptyset$ (additivity axiom)

Fig. 2. Axioms of probability measures.

- **Uncertainty.** Even when the knowledge covers the whole domain, decisions may not have always the same outcome (because of the inherent complexity of the world, because of randomness) or because the information is not of enough quality to answer with certainty (because of imprecision in our knowledge).

Decision theory usually distinguishes between risk and uncertainty. Risk corresponds to the case when each decision leads to a certain state, however we do not know into which state, but we know the probabilities of being in these states. In contrast, we have uncertainty when the probabilities are unknown. We do not distinguish here these two concepts and put them both into the uncertainty class.

As stated above, uncertainty is usually represented by means of probability theory, and we use probability measures to quantify our uncertainty. Recall that a probability measure is a measure that satisfies the axioms in Figure 2. Probability theory have developed rules for reasoning as Bayes' rule and rules for conditioning.

There exist different types of uncertainty, and as we will see below, they are different approaches to deal with them. Then, we will see that probability measures are just one type of uncertainty measures specially suited for one type of uncertainty. Let us consider the basic concepts, and link them with ways of dealing with them.

- **Randomness.** We have randomness when the outcome of an experiment is different when we repeat it under the same conditions. Throwing dice and coins are typical examples of random experiments. Randomness is usually modeled using probability measures. E.g., for a fair die we assign $\Omega = \{1, 2, 3, 4, 5, 6\}$ and $P(\{i\}) = 1/6$ for all $i \in \Omega$. We can also represent unfair (biased) dice with appropriate probability measures.
- **Ignorance.** We have ignorance when we have lack of knowledge. It is important to distinguish ignorance of randomness. In probability theory we usually represent ignorance with uniform distributions. Nevertheless, it is not the same to know that a die is fair that not having any information on the die. Dempster-Shafer theory (evidence theory) can be used to represent ignorance. We give an example below in Section 5.1 and we introduce belief measures. This type of measures is another type of uncertainty measure.

- **Imprecision.** We have a precise statement when only a single value makes it true. We have an imprecise statement when there is more than one value that makes it true. For example, the statement

  Temperature is 24 degrees

  is precise because only when the temperature is exactly 24 the statement is true, in all the other cases it is false. On the contrary, the statement

  Temperature is larger than 15

  is imprecise because any value larger than 15 makes the statement true. This type of situations can be modeled by means of possibility theory. We are able to compute the certainty of a statement (a proposition) given some information. Both the information and the statement can be precise or imprecise. Two measures are considered: possibility and necessity. We will consider them briefly in Section 5.2.
- **Vagueness.** The truth of a statement is usually either asserted or refuted. That is, for a given statement we can only certify its certainty (it is true) or deny it (it is false). Probability theory and randomness do not change this perspective, we may not know in which state we are or will be, or if a statement is or will be true or not, but only these two cases (truth and falsity) are possible. We have vagueness when truth is a matter of degree, and thus truth is no longer a Boolean concept.

  > "Truth is generally understood as the conformity between a statement and the actual state of facts to which it supposedly refers. Here, however, a degree of truth is rather a measure of agreement between the representation of the meaning of a statement and the representation of *what is actually known* about reality. This view is supported by Zadeh (1981), who defines procedures for the computation of meaning. What is known of reality is supposedly stored in a database B, in the form of statements. What can be said of the truth of a statement of a query statement S depends upon our state of knowledge (the information in B), and derives from a matching procedure between the meaning of S and of the contents of B. According to the respective precision of S and the information in B, the truth of S is asserted, refuted, but may also be only partially known (pervaded with uncertainty), or may be a matter of degree (S is vague)." (Dubois and Prade [5], p.288).

  Vagueness is usually represented using fuzzy sets. We will see them in Section 6, and then we will focus turn to uncertainty measures for vagueness in Section 6.2.

5.1 Uncertainty measures for ignorance: belief functions

In this section we give an example and the definition of a belief function.

Given a (discrete) reference set Ω, a Belief function on Ω is a set function (i.e., $Bel(A)$ for $A \subset \Omega$) in [0,1] that satisfies the following axioms:

1. $Bel(\emptyset) = 0$
2. $Bel(\Omega) = 1$
3. $Bel(A) \leq Bel(B)$ if $A \subseteq B \subseteq \Omega$
4. $Bel(A \cup B) \geq Bel(A) + Bel(B)$ if $A \cap B = \emptyset$

Fig. 3. Axioms of Belief functions.

Example 1. This weekend there was a bank robbery in Mariestad. Surveillance cameras of the bank have only registered Alice, Bob, and Claudia around the bank from friday to early monday. No other information is available.

This situation can be modeled with a belief function on the set $\Omega = \{A, B, C\}$ where A represents Alice, B represents Bob, and C represents Claudia that satisfies the following assignment:

- $Bel(\{A\}) = 0$
- $Bel(\{B\}) = 0$
- $Bel(\{C\}) = 0$
- $Bel(\{A, B, C\}) = 1$

First, note that we use Bel as the name of our belief function. Second, note that as we have no evidence that it was only Alice that rob the bank we assign $Bel(\{A\}) = 0$. The same for Bob and Claudia. In contrast, as we are absolutely sure that it was either Alice, Bob or Claudia who rob the bank, we are absolutely sure that the robber is in the set $\{A, B, C\}$ and because of that we assign $Bel(\{A, B, C\}) = 1$.

It is important to see that from a mathematical point of view this function Bel is not a probability because the measure on the set $\{A, B, C\}$ is one, but this is not the addition of the measures for the three suspects. That is, the equality (the additivity axiom for probability measures, see Figure 2) does not hold in the following equation:

$$Bel(\{A, B, C\}) = 1 \neq Bel(\{A\}) + Bel(\{B\}) + Bel(\{C\}) = 0 + 0 + 0 = 0$$

Belief measures are used in Dempster-Shafer theory to represent uncertainty. Belief measures are defined axiomatically as probability measures. Axioms are given in Figure 3. For each belief function there is a dual measure called possibility measure.

In a way similar to what we have in probability theory, there are operations and methods to operate with belief functions. For example, we have the Dempster rule of combination to combine belief functions.

5.2 Uncertainty measures for imprecision: Necessity and possibility

In classical logic we consider the interpretation of atomic formulas (predicates and equalities of terms) which assign a truth value to each of them, and from them we can find the truth value of an arbitrary sentence. So, if s is a sentence we find a truth value for s through a valuation function w (thus, expressed $w(s)$) which is either true or false.

When the information available is precise and the statement is also precise, we have that we can assign a truth value $w(s) \in \{0, 1\}$ where 0 denotes that s is false and 1 that s is true. In fact, we can compute this also in the case that the statement is imprecise but the information is still precise.

When the statement (or proposition) and the information are both imprecise, we can define measures of uncertainty to represent our knowledge on the statement. We use two measures of uncertainty that we call necessity and possibility.

Given a statement s (a proposition), and some information, we say that the possibility[1] of s is true if there are some values in accordance to the information that make the statement true. For example, if we have that our information is that temperature is over or equal to 20, it is clear that it is possible that the statement "temperature equals 25" is true. So, *Pos(s)=true* or using 1 to represent true and 0 to represent false, *Pos(s)=1*. Nevertheless, it is not absolutely necessary that the statement is true, because it may be the case that the temperature is, in fact, 30. So, we have that the necessity of this statement to be true is zero. That is, *Nec(s)=0*. In this case, we have uncertainty on s as it may be either true or false. This case is represented in Table 1 (in blue).

Necessity is true (or equal to one) means that in all situations that are consistent with our information, we have that the statement is true. This was not the case in the previous example (e.g. the case of temperature equal to 30), so that is why necessity is false (or equal to zero).

We can observe that the possibility of a statement is always larger than its necessity. Note also that the necessity of a statement s can be computed from the possibility of $\neg s$ (i.e., the possibility of the complement of the statement). In particular,

$$Nec(s) = 1 - Pos(\neg s). \tag{1}$$

For example, the possibility of "temperature $\neq 25$" is 1 because we know that the temperature is larger or equal to 25. Then,

$$Nec(Temp = 25) = 1 - Pos(\neg(Temp = 25)) = 1 - Pos(Temp \neq 25) = 1 - 1 = 0.$$

Because of Equation 1 we say that Nec and Pos are dual uncertainty measures, and, in fact, one of them imply the other.

In the example above we have uncertainty, but imprecise information and imprecise statements do not always imply uncertain information. We may have certainty. This is the case of the following example.

[1] Dubois and Prade use in [5] the possibility of the meaning of s, that is $M(s)$ instead of just s. Then, $M(s)$ is a subset of values in a given reference set as e.g. the set of possible temperatures.

When our information is that temperature is over or equal to 20, and our statement is that temperature is over or equal to 15, it is clear that *Pos(s)=1* and *Nec(s)=1*. So in this case, we have certainty on s. This is represented in Table 1 (in red).

We include in Table 1 examples with precise and imprecise information and the same for statements. For each of them the uncertainty measures are computed. The cases of precise information discussed above with a valuation function w, as there is no uncertainty, can be seen as with $w(s) = Pos(s) = Nec(s)$.

	Information	
	Precise Temperature = 24	Imprecise Temperature $\geq$ 20
Precise statements:	Truth value $w(s) \in \{0, 1\}$	Possibility measures
s="Temp = 25" ?	w(s)=Pos(s)=Nec(s)=0 (certainty)	Pos(s)=1, Nec(s)=0 (uncertainty)
s="Temp = 10" ?	w(s)=Pos(s)=Nec(s)=0 (certainty)	Pos(s)=0, Nec(s)=0 (certainty)
Imprecise statements:	Truth value $w(s) \in \{0, 1\}$	Possibility measures
s="Temp > 15 ?	w(s)=Pos(s)=Nec(s)=1 (certainty)	Pos(s)=1, Nec(s)=1 (certainty)
s="Temp > 30 ?	w(s)=Pos(s)=Nec(s)=0 (certainty)	Pos(s)=1, Nec(s)=0 (uncertainty)

Table 1. Uncertainty measures for imprecise information and statements.

5.3 Summary of tools for approximate reasoning

We have seen in this section a few tools for apprimate reasoning. We close this section listing them and including a few others that have not yet been mentioned. They are the following ones.

- Nonmonotonic logic
- Probability theory, Bayesian networks
- Dempster-Shafer theory
- Certainty factors
- Possibility theory
- Fuzzy sets and fuzzy inference

A good reference book on approximate reasoning is [4]. See [3] for a description of belief functions (evidence theory) and [5] on possibilistic and fuzzy logic (this is not on fuzzy sets and fuzzy systems).

6 Fuzzy sets and fuzzy systems

One of the most successful applications of fuzzy sets is fuzzy rule-based systems. They are defined in terms of rules defined in terms of fuzzy sets.

Fuzzy systems are rule based systems in which the rules (antecedents and conclusions) are defined in terms of fuzzy sets. These systems are mainly used in control and modeling tasks. In the first case, the goal is to control the behavior of a device. That is, we build a system so that the device follows a predefined schema. For example, we can control the temperature of an oven, the temperature of a room, the speed of a train, or focus a camera. In the second case, the system has to predict a given variable. The prediction of the stock exchange market is an example of this type of system. In all these cases, we use fuzzy systems to represent a functional relationship between a set of inputs and an output (or a set of them).

Let us consider the example of a fuzzy rule based system to control the temperature of a device.

Example 2. The goal of the control system is that the temperature of the device is equal to a given temperature T. In order to reach this goal, we consider that the system has access to the goal temperature (T), the current temperature (we denote by t_i the temperature at time i), and the temperature at the previous moment (t_{i-1}). The system controls the temperature by means of assigning values to a control variable (with positive values, the device increases its temperature, and with negative values the device has to decrease the temperature).

Then, we can control the device by means of rules that assign values to a control variable according to the temperature of the system. We give an example of a rule. The rule uses the error (variable ϵ) between the current temperature and the desired one, and an increment of error (variable $\Delta\epsilon$) that corresponds to the variation of the error variable. If we assume that the desired temperature does not change, that is $\Delta\epsilon = (t_i - T) - (t_{i-1} - T) = t_i - t_{i-1}$.

If ϵ is large-positive and $\Delta\epsilon$ is small-positive
then the control variable is small-negative

In order that this rule is fully defined and that we can apply it when appropriate we need to define the terms that appear in the rule. That is, we need to give a meaning to *large-positive*, *small-positive*, and *small-negative*. Fuzzy sets permit us to define these terms, and then permit us to compute the output of the system (i.e., assign a value to the control variable) using all the applicable rules.

In fact, fuzzy sets permit partial membership of elements to sets, and this implies that rules are partially satisfied. The conclusion of the system will be computed taking into account these partial satisfaction of the rules.

In the remaining part of this section, we will give the basic elements that we need to build a fuzzy system. We start with the definition of fuzzy sets, their operations, and then describing how rules are applied.

6.1 Fuzzy sets: definition and operations

Fuzzy sets generalize classic sets (in order to differentiate both types of sets we call classic sets by *crisp sets*). That is, crisp sets are a particular case of fuzzy sets. Because of that when we define operations on fuzzy sets that exist for crisp sets it is usual that these operations work on crisp sets as usual. For example, if we define a union of fuzzy sets should be defined in a way that when applied to crisp sets reproduces the results of the usual union. When this applies, we say that the union for fuzzy sets generalizes the union for crisp sets.

There exist different ways to represent crisp sets (extensional representation, characteristic function, and elements that satisfy a property). Among them, the most appropriate in our case is the one based on characteristic functions. A characteristic function declares for each element in a reference set whether the element belongs or not to the set. That is, a characteristic functions defines a Boolean function that when applied to the elements of the domain returns their membership (or non-membership) to the set.

In order to move smoothly from crisp sets to fuzzy sets we use a characteristic function that returns either a zero or one. It returns zero when the element does not belong to the set, and one if it belongs. In this way, we formally have that if D is the reference domain where we define the set A using a characteristic function $\chi_A = A$, then we have that χ_A is a function from D into $\{0,1\}$ (that is, $\chi_A : D \to \{0,1\}$). We call D the domain of A or the universe of discourse of A.

Example 3. Let us consider the set of points in the interval [-1,1]. Then, the characteristic function of this set, that we call Z from zero, is

$$\chi_Z(x) = \begin{cases} 1 \text{ if } & x \in [-1,1] \\ 0 \text{ if } & x \notin [-1,1] \end{cases} \tag{2}$$

Given an element of the domain (a real r) we can determine if the element belongs or not to the set Z applying the function χ_Z. In this way, when $\chi_Z(r) = 1$ it means that the element belongs to the set and that when $\chi_Z(r) = 0$ it means that it does not. So, it is clear that $\chi_Z(0.5) = 1$ and thus belongs to the set while 1.5 does not. Figure 4 gives a graphical representation of this set.

In fuzzy sets, membership is no longer Boolean. Instead, we have graded membership. The following example tries to illustrate the difference between probability and fuzziness.

Example 4. (Example 1 in [1]) Let the set of all liquids be the universe of objects, and let fuzzy subset $L = \{$all potable (="suitable for drinking") liquids$\}$. Suppose you had been in the desert for a week without drink and you came upon two bottles, A and B, marked as membership $(A \in L) = 0.91$ and probability $(B \in L)$=0.91.

Confronted with this pair of bottles and given that you must drink from the one that you choose, which would you choose to drink from first? Most

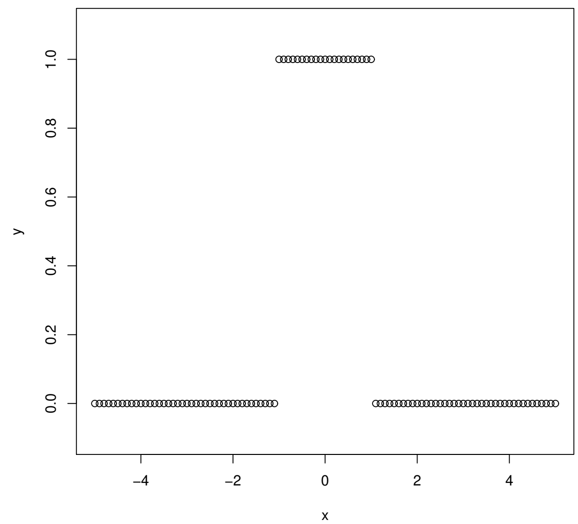

Fig. 4. Graphical representation of the characteristic function χ_Z in Equation 2.

readers familiar with the basic ideas of fuzzy sets, when presented with this experiment, immediately see that while A could contain, say, swamp water, it would not (discounting the possibility of a Machiavellian fuzzy modeler) contain liquids such as hydrochloric acid. That is, a membership of 0.91 means that the contents of A are "fairly similar" to perfectly potable liquids (pure water). On the other hand, the probability that B is potable = 0.91 means that over a long run of experiments, the contents of B are expected to be potable about 91% of the trials; and the other 9%? In these cases the contents will be unsavory (indeed, possibly deadly)-about one chance in ten. Thus most subjects will opt for a chance to drink swamp water, and will choose bottle A.

Operating with fuzzy sets In the same way that we can operate with crisp sets by means of operations as union, intersection and complement, we can operate with fuzzy sets by means of analogous operations. That is, there are ways to compute the union and the intersection of fuzzy sets. Similarly, we can compute the complement of a fuzzy set.

Operations for fuzzy sets are based on the operations for crisp sets. They are generalizations of them. Let us focus on the case of the union. In this case, a generalization means that when two fuzzy sets are in fact crisp (i.e., the membership degrees are restricted to $\{0, 1\}$) then their union correspond to the standard union. Nevertheless, while with crisp sets there is a single way to compute their union, this is no longer the case for fuzzy sets. We will see that there are different operations to compute the union. This is due to the fact, that there are different ways to generalize the union. The same applies to the other usual operations. We will discuss them below.

Logic and set theory are closely related. Some logical operations can be defined in terms of set theory and the other way round. For example, the intersection of two sets can be defined in terms of the logical operator `and` (i.e., a conjunction). For example,

$$A \cap B = \{x | x \in A \texttt{ and } x \in B\}$$

Table 2 gives the correspondences between the logical operators and the operations on sets.

set theory	logic
intersection ($\cap$)	and ($\wedge$) (conjunction)
union ($\cup$)	or ($\vee$) (disjunction)
complement (^{c})	negation ($\neg$)

Table 2.

These operations of union, intersection and complement on sets are closely related with disjunction, conjunction and negation of predicates in logic.

Complement The complement of a (crisp) set A on a reference set X is the set A^c that are not in A. If we consider the values near zero as above (represented by the characteristic function χ_Z), the complement will be the values that are not near zero. So, if near zero corresponds to:

$$\chi_Z(x) = \begin{cases} 1 \text{ if } & x \in [-1, 1] \\ 0 \text{ if } & x \notin [-1, 1] \end{cases} \tag{3}$$

then, not near zero (the complement of χ_Z) is

$$\chi_{Not-Z}(x) = \begin{cases} 1 \text{ if } & x \notin [-1, 1] \\ 0 \text{ if } & x \in [-1, 1]. \end{cases} \tag{4}$$

We can see the computation of the complement in terms of a function N that transforms the membership degree. That is, $\chi_{Not-Z}(x) = N(\chi_Z(x))$. Note that if $\chi_Z(x) = 0$ (i.e., the values is not zero) then the function N returns 1 (i.e., the value is in the complement) and if $\chi_Z(x) = 1$ (i.e., the value is near zero) then N returns 0 (i.e., the value is not in the complement).

When we consider a function N in this way, we can easily consider a generalization for fuzzy sets. In this case, instead of a characteristic function χ_Z we consider membership functions μ_A, and N will take values in [0,1] instead of only considering the values 0 and 1.

Definition 1. *A function $N : [0,1] \to [0,1]$ is a negation function if*

1. *$N(0) = 1$ and $N(1) = 0$ (boundary conditions)*
2. *$N(a) \geq N(b)$ when $a < b$ for all a, b in $[0,1]$ (monotonicity).*

Negation functions are also known as fuzzy complements.

The first condition follows directly from the crisp case. The second one is a monotonicity condition, that means that if in the original set we have two elements x and y such that $a = \mu(x) < \mu(y) = b$ then, when we consider the complement, the relationship is reversed: $N(a) = N(\mu(x)) \geq N(\mu(y)) = N(b)$.

The typical negation is $N(x) = 1 - x$. Nevertheless, there are other negation functions used in the literature. Two families of complements are given below. They are the Yager family, which depends on a parameter w, and the Sugeno family, which depends on the parameter λ.

1. $N_w(a) = (1 - a^w)^{1/w}$ (Yager family)
2. $N_\lambda(a) = (1 - a)/(1 + \lambda a)$ (Sugeno family)

It is easy to see that these functions satisfy the boundary conditions. We can aso prove that they satisfy the monotonicity condition.

In addition to the two conditions given above, it is often required that negation functions satisfy the following two conditions.

1. N is a continuous function
2. $N(N(a)) = a$ for all $a \in [0,1]$ (involutiveness)

The last condition states that a double negation is equivalent to do nothing to the original truth value. $N(x) = 1 - x$ satisfies both conditions.

Intersection Let us consider the following definition for intersection of crisp sets

$$A \cap B = \{x | x \in A \text{ and } x \in B\}$$

or, equivalently, using characteristic functions

$$\chi_{A \cap B}(x) = 1 \text{ if and only if } \chi_A(x) = 1 \text{ and } \chi_B(x) = 1.$$

If we consider the operation for logical conjunction $T : \{0,1\} \to \{0,1\}$ with standard definition

$$T(1,1) = 1$$

and

$$T(1,0) = T(0,1) = T(0,0)$$

we can rewrite the definition as

$$\chi_{A \cap B}(x) = T(\chi_A(x), \chi_B(x)).$$

Fuzzy rule-based systems

`Rule1: if` ϵ `is positive and` $\Delta\epsilon$ `is positive then control-variable is small-negative`

`Rule2: if` ϵ `is zero and` $\Delta\epsilon$ `is zero then control-variable is zero`

In general, we have a structure as follows:

`if` X_1 `is` t_{1a} `and` X_2 `is` t_{2b} `and` $\cdots$ X_n `is` t_{nz} `then` Y `is` t_{yo}

Step 1. Degree of satisfaction of antecedent

$$\alpha = T(\mu_{1a}(x_1), \mu_{2b}(x_2), \ldots, \mu_n(x_{nz}))$$

Step 2. Application of the rule and computation of the consequent

$$\mu(x) = \min(\alpha, \mu_{yo})$$

Step 3. Application of a set of rules and computation of the corresponding consequent. In terms of the membership function, it is the following:

$$\tilde{\mu} = \cup_{R \in KB} \mu_R$$

If we consider memberships at given y in the space of the output Y, this is

$$\tilde{\mu}(y) = \cup_{R \in KB} \mu_R(y)$$

for all $y \in Y$.

Step 4. Defuzzification

$$\frac{\sum_y \tilde{\mu}(y) \cdot y}{\sum_y \tilde{\mu}(y)}$$

This corresponds to a fuzzy inference where rules are considered in a disjunctive way. That is, either rule R_1 or R_2 or $\ldots$ or R_n is applied. Then, we combine the information on the state for each rule by means of a union (disjjunction) of the conclusions.

6.2 Vague knowledge

When information and statements can be vague, things get more complex but follow a pattern similar to the case of imprecise information we have already discussed in Section 5.2.

Case p-p. (Precise statement and precise information) Both statement and information is precise, so, we can only have that the statement is true or false.. This case, correspond to the valuation function $w(s)$.

Case i-p. (Imprecise statement and precise information) As in the previous case, we have only a valuation function $w(s)$ which can be true or false. We know that the information is of the form "x *is* u_0" and we want to know if the statement x *is* A where A is a crisp set holds. Naturally, $w(s)$ is true if

	Information B		
	Precise Temperature = 24 $T = b$	Imprecise Temperature ≥ 20 $T = B$	Vague Temp betwen 10 and 20 (vague interval) $T = \mu_B$
Precise statement: s="$Temp = u_0$"	(case p-p) $w(s)$ (certainty)	(case p-i) $Pos(s) = 1$ iff $u_0 \in B$ $Pos(\neg s) = 1$ iff $u_0 \in B^c$ (uncertainty)	(case p-v) $Pos(s) = \mu_B(u_0)$ $Pos(\neg s) = \mu_{B^c}(u_0)$ (uncertainty)
Imprecise statement: s="x is A"	(Case i-p) $w(s) = \chi_A(b)$	(Case i-i) $Pos(s) = 1$ iff $B \cap A \neq \emptyset$ $Pos(\neg s) = 1$ iff $B \cap A^c \neq \emptyset$	(Case i-v) $Pos(s) = sup\{\mu_B(u) \mid u \in A\}$ $Pos(\neg s) = sup\{\mu_B(u) \mid u \notin A\}$
vague statement: s="x is $\tilde{A}$"	(case v-p) $Pos(v) = 1$ iff $v = m_b$	(case v-i) $Pos(v) = 1$ iff $B \cap \{u \mid \mu_A(u) = v\} \neq \emptyset$	(case v-v) $Pos(v) = sup_u\{\mu_B(u) \mid \mu_A(u) = v\}$

Table 3. Uncertainty measures for imprecise information and statements.

and only if $u_0 \in A$. If χ_A is the characteristic function of A, this is equivalent to state

$$w(s) = t(s|B) = \chi_A(u_0).$$

Note that here we are using $t(s|B)$ to denote the truth value of the statement s when we know B.

Case p-i. (Precise statement and imprecise information) We have already seen that we can use necessity and possibility measures to deal with this situation. We have that $Pos(s)$ is true if $u_0 \in B$ and false otherwise. That is, $Pos(s) = 1$ if and only if $u_0 \in B$. Using the duality condition of necessity, we consider $Pos(\neg s)$ as true if $u_0 \in B^c$ (where B^c is the complement of B).

Case i-i. (Imprecise statement and imprecise information) This case is similar to case p-i. This situation is also represented using necessity and possibility measures. Now, s is of the form x is A. It may happen that the set B has empty intersection with A. In this case it is clear that there is no possibility of s to be true. On the contrary, if the intersection is not null there are values that according to the information (B) can make true the statement. Therefore, we can define $Pos(s) = 1$ if and only if $B \cap A \neq \emptyset$. Similarly, we can define $Pos(\neg s)$.

Case p-v. (Precise statement and vague information) The information is expressed in terms of a fuzzy set μ_B and all values in the domain of B are possible, and the degree of possibility of each value in the domain is $\mu_B(u)$.

Thus, as our statement is about u_0, we have that the possibility of s is the degree of membership of u_0 to B. That is, $\mu_B(u_0)$. Therefore,

$$Pos(s) = \mu_B(u_0).$$

Similarly we can compute $Pos(\neg s)$ by means of the complement of B. I.e.,

$$Pos(\neg s) = \mu_{B^c}(u_0).$$

Case i-v. (Imprecise statement and vague information) Now the statement is imprecise and any element in A will make the statement true. As the possibility of any element u of A is $\mu_B(u)$ (as in the previous case), the possibility for all elements in A is the maximum (or the supreme if we are working in the continuum) of $\mu_B(u)$ for all u in A. That is,

$$Pos(s) = \mu_{t(S|B)}(1) = sup\{\mu_B(u)|u \in A\}$$

A similar argument applies for $Pos(\neg s)$ which can be used to compute the $Nec(s)$ using $Nec(s) = 1 - Pos(\neg s)$. Therefore,

$$Pos(\neg s) = \mu_{t(S|B)}(0) = sup\{\mu_B(u)|u \notin A\}$$

We use the maximum because among all possibilities (possibilities for all $u \in A$) we take the largest one.

So far we have considered possibility measures Pos as functions that given a statement returns a value. In fact, we have considered *Pos(s)* and *Pos(¬ s)*, and we have seen that when s is expressed in terms of a set A then *Pos(¬ s)* can be computed in terms of the complement of A. In all these expressions s was a precise or imprecise statement. We can revisit these functions taking into account that A can be a vague set.

If A is a crisp set, we can only consider the possibility of being true or false. When A is represented using a fuzzy set, we may consider the possibility of any value in the interval [0,1]. In this way, we can consider a function $Pos(v)$ for any v in [0,1] that evaluates the possibility of the truth value of s being v. In other words, $Pos(v)$ corresponds to the possibility that A is certain with degree v. Note that $Pos(s)$ can be understood as $Pos(1)$ as this is the possibility that s is true (with degree 1), and $Pos(\neg s)$ can be understood as $Pos(0)$ as this is the possibility that s is false (or that $\neg s$ is true with degree 1).

Then, for any v, we have that $Pos(v)$ needs to consider the elements of u such that $\mu_A(u) = v$. Note that the elements of this set $\{u|\mu_A(u)\}$ are the only ones that make s to have certainty v. Then, the possibility for this set when the information is μ_B is naturally the maximum (or the suppremum) of μ_B for these values. That is, $\sup_u mu_B(u)$ for u in the set $\{u|\mu_A(u)\}$. That is,

$$Pos(v) = sup_u\{\mu_B(u)|\mu_A(u) = v\}. \tag{5}$$

This expression corresponds to the most general case. We review below the three cases corresponding to whether the information is precise, imprecise or vague.

Case v-p. (Vague statement and precise information) The information is of the type "*x is b*" and the statement is *x is* $\tilde{A}$ where $\tilde{A}$ is represented with a fuzzy set with membership μ_A. If $m_b = \mu_A(b)$, we have that $Pos(v) = 1$ if and only if $v = m_b$, otherwise is zero. Naturally, with this definition we have that $Pos(v)$ is either 0 or 1.

Case v-i. (Vague statement and imprecise information) The information is of the type "*x is B*" and the statement is *x is* $\tilde{A}$ where $\tilde{A}$ is represented with a fuzzy set with membership μ_A. In this case Equation 5 can be translated into:

$$Pos(v) = 1 \text{ if and only if } B \cap \{u | \mu_A(u) = v\} \neq \emptyset$$

Note that this is similar to the case i-i. Note also that $Pos(v)$ is either 0 or 1 in this case.

Case v-v. (Vague statement and vague information) This is the most general case described in Equation 5. Note that now $Pos(v)$ can take any value in [0,1].

References

1. Bezdek, J. C. (1993) Fuzzy models – What are they, and why? IEEE Trans. on Fuzzy Systems 1 1-1993.
2. Shortliffe, E.H., Buchanan, B.G. (1975) A model of inexact reasoning in medicine, Mathematical Biosciences 23 (3-4) 351379.
3. Smets, P. (1988) Belief functions, in P. Smets, E. H. Mamdani, D. Dubois, H. Prade (eds.) Non-Standard Logics for Automated Reasoning, Academic Press, 253-277.
4. Smets, P., Mamdani, E. H., Dubois, D., Prade, H. (eds.) (1988) Non-Standard Logics for Automated Reasoning, Academic Press.
5. Dubois, D., Prade, H. (1988) Possibilistic and fuzzy logics, in P. Smets, E. H. Mamdani, D. Dubois, H. Prade (eds.) Non-Standard Logics for Automated Reasoning, Academic Press, 287-315.