

# Uncertainty in AI - Draft

Davide Sottara

July 3, 2007

# Chapter 1

## Measures of Uncertainty

### 1.1 Introduction

People use predicates, more commonly called sentences, to describe the world. Sentences state the existence of a relation between a set of objects, the arguments, which linguists divide more specifically in subject, object and complements depending on their role. For example, “John drinks water in the kitchen” describes a relation (Drinks) between an Individual, a Liquid and a Place.

In classical boolean logic, a statement is either true or false, i.e. the relation for a given set of arguments holds *xor* not. Unfortunately, in real world situations the exact evaluation of a statement might be difficult due to various reasons - noise, imprecision, lack of information, . . . - which globally cause *Uncertainty*.

In particular, we may have to deal with:

- I Uncertainty on the arguments: sometimes we speak about an object X without knowing which or what X actually is. Unless the relation holds for any X, the statement becomes uncertain until the actual value x is discerned. In particular, the likelihood of x turning out to be an object for which the statement holds may affect the degree of uncertainty.
- II Uncertainty on the truth value - 1: sometimes, given an argument x, relations hold only partially, i.e. they are neither completely true nor completely false. Rather than considering a relation both true and false in some degree, it is convenient to use different, intermediary, truth values.
- III Uncertainty on the truth value (2): sometimes, even with gradual truth values, we are unable to assess the exact degree in which a proposition holds. In extreme cases of total ignorance, all possible truth values are equally likely.
- IV Uncertainty on the relation: sometimes, even when the arguments are well defined and a precise evaluation could be possible, one does not know

exactly what to evaluate. In cases when the relation itself is ill-defined, one should not be confident in the statement.

Uncertainty has been the subject of much research. In the rest of this chapter we will provide a brief introduction and the definitions of the key concepts of the main theories that deal with uncertainty. Chapter 2 will deal with predicate logic and expert systems Chapter 3 will define Chapter ?? will revisit predicate logic.

## 1.2 Measures

Uncertainty can be considered a special case of partiality, namely partial knowledge. Thus, A set  $X$  of evidence and a function  $m$  measuring how much evidence is available is a good general model of uncertainty. Depending on the meaning given to  $X$  and  $m$  we have different theories: Dubois and Prade define the class of "fuzzy measures":

$$m : \wp(X) \rightarrow [0, 1] \tag{1.1}$$

satisfying:

$m(\emptyset) = 0$	null measure
$m(X) = 1$	full measure
$\forall A, B \in \wp(X) : A \subseteq B \Rightarrow m(A) \leq m(B)$	monotonicity
$\forall j \in N : A_j \in \wp(X), (A_j \subset A_{j+1}) \vee (A_j \supset A_{j+1}) \Rightarrow$ $\lim_{j \rightarrow \infty} m(A_j) = m(\lim_{j \rightarrow \infty} A_j)$	continuity

### 1.2.1 Probability

Probability theory focuses on events  $x$  whose outcome is not (yet) known, except that it belongs to an universe of alternatives  $X$ . A probability function  $p : X \rightarrow [0, 1]$  maps every set of alternatives  $A \subseteq X$  to the likelihood that, once observed,  $x$  will turn out to belong to  $A$ . Hence, the additional axioms:

$$\begin{aligned} \forall A \in \wp(X) : p(A) \in [0, 1] \\ \forall j : A_j \in \wp(X), A_i \cap A_j = \emptyset \text{ iff } i \neq j \Rightarrow p\left(\bigcup_j A_j\right) = \sum_j p(A_j) \end{aligned}$$

#### Approaches to probability

Probability has two main interpretations, frequentist and subjective. It is noteworthy that frequentist probability may induce a subjective belief, while the opposite is not possible.

**Frequentist approach** Given  $N$  events  $e_{j:1..N} \in X = \{x_1, \dots, x_K\}$ , whose outcome is known, frequentists define

$$p(x_k) = \lim_{N \rightarrow \infty} \frac{|\{e_j : e_j = x_k\}|}{N}$$

**Subjective (Bayesian) approach** Given  $N$  events  $e_{j:1..N} \in X = \{x_1, \dots, x_K\}$ , whose outcome is **not** known,  $N\hat{p}(x_k)$  is the expected number of times event  $e_j$  will turn out to be  $x_k$ . In other words,  $\frac{1}{p(x_k)}$  is the amount of money one would expect for winning a bet on  $x_k$

### 1.2.2 Belief / Plausibility

Belief measures model concepts such as credit assignment and relevance. Belief measures are also used in fault isolation contexts, whenever the culprit is not known exactly, but is known to be member of a limited group, with a certain degree of confidence.

#### Basic Mass Assignment Functions

A mass assignment is a function  $m : \wp(X) \rightarrow [0, 1]$  satisfying  $\sum_{A \in \wp(X)} m(A) = 1$ . Sets  $A$  for which  $m(A) > 0$  are called focal elements. It can be interpreted as a (subjective) probability distribution on  $\wp(X)$ .

**Significative bma** According to [?] BMAs may have different properties:

**Bayesian** Bayesian mass assignments are focalized on singletons and thus are probability distributions:

$$\forall A \in \wp(X) : m(A) > 0 \Rightarrow |A| = 1$$

**Categorical** A categorical bma assigns all the credit to a single element:

$$\exists ! A^* \in \wp(X) : m(A^*) = 1$$

**Vacuous** A vacuous bma trivially assigns all credit to the universe set:

$$m(X) = 1$$

**Contradictory** A contradictory bma states the impossible:

$$m(\emptyset) = 1$$

**Dogmatic** A dogmatic bma is informative in some degree:

$$m(X) = 0$$

**Normalized** A normalized bma has resolved all of its internal conflict:

$$m(\emptyset) = 0$$

**Bayesian projection techniques** In many cases, Bayesian bmas are particularly appealing: first, they give credit to individuals and so are more precise. Then, from a computational point of view, they grow linearly with  $|X|$  rather than exponentially. So, there are several algorithms which transform a generic bma into a bayesian one. None actually deals with the conflictual mass  $m(\emptyset)$ , which must be handled independently before applying the algorithms. Typical approaches include normalization (Dempster-Shafer), i.e. dividing all masses by  $1 - m(\emptyset)$ , or redistribution (Yager), i.e. transferring  $m(\emptyset)$  to  $m(X)$ .

**Intersection** According to this technique, each singleton receives a share of the global amount of non-bayesianity proportionally to its membership in sets which contribute to the non-bayesianity:

$$m_B(x) = m(x) + \frac{\sum_{|A|>1} m(A)}{\sum_{|A|>1} m(A)|A|} \sum_{A \ni x, |A|>1} m(A)$$

### Orthogonal Projection

**Pignistic** The pignistic transformations shares the mass of each non-singleton focal element between its members:

$$m_B(x) = \sum_{A \ni x} \frac{m(A)}{|A|}$$

### Belief functions

A Belief function is a measure with the additional properties:

$$\begin{aligned} \forall A \in \wp(X) : bel(A) &\in [0, 1] \\ bel(\cup_j A_j) &\geq \sum bel(A_j) - \sum_{i<j} bel(A_i \cap A_j) + \dots \end{aligned}$$

A belief function can be obtained from a bma by setting:

$$bel(A) = \sum_{B \subseteq A} m(B)$$

Belief functions are conservative, as they give credit to sets but not necessarily to the single members. In fact, it is possible to have  $bel(A) + bel(\bar{A}) < 1$  if some mass is assigned to the universe as a whole, i.e. ( $m(X) > 0$ ).

### Plausibility functions

A Plausibility function is a measure with the additional properties:

$$\begin{aligned} \forall A \in \wp(X) : pl(A) &\in [0, 1] \\ pl(\cap_j A_j) &\leq \sum bel(A_j) - \sum_{i<j} bel(A_i \cap A_j) + \dots \end{aligned}$$

Plausibility is computed from a bma:

$$bel(A) = \sum_{B \cap A \neq \emptyset} m(B)$$

Belief and Plausibility are strongly correlated, since  $Bel(A) = 1 - Pl(\bar{A})$ .

### 1.2.3 Lambda-Measures

More generally, probability, belief and plausibility use different approaches in solving the problem:

$$\begin{aligned} A, B \in \wp(X), A \cap B = \emptyset \\ m(A \cup B) = f(m(A), m(B)) \end{aligned}$$

$\lambda$ -measures are defined by choosing  $f(m(A), m(B)) = m(A) + m(B) + \lambda m(A)m(B)$ , leading to the union formula for general non-disjoint sets:

$$m(A \cup B) = \frac{m(A) + m(B) - m(A \cap B) + \lambda m(A)m(B)}{1 + \lambda m(A \cap B)}$$

It can be shown that for different values of  $\lambda$  the measure behaves differently:

$$\begin{aligned} \lambda \leq -1 & \quad \text{Not Defined} \\ -1 < \lambda < 0 & \quad m_\lambda \text{ is a Plausibility measure} \\ \lambda = 0 & \quad m_\lambda \text{ is a Probability measure} \\ \lambda > 0 & \quad m_\lambda \text{ is a Belief measure} \end{aligned}$$

And, like in the general case,  $Bel_\lambda \leq Prob_\lambda \leq Pl_\lambda$ .

### 1.2.4 Possibility / Necessity

The concepts of Necessity and Possibility are similar, but not strictly related, to those of Belief and Plausibility. They are generated by a possibility distribution:

$$\pi : X \rightarrow [0, 1]$$

$\pi(x)$  describes the possibility that a singleton element  $x$  satisfies some criterion or, equivalently, the possibility of  $x$  being the cause of some given effect. Differing from probability distributions, possibility distributions evaluate single elements in an independent way: for example, head and tails are both possible outcomes of a coin toss ( $\pi(h) = \pi(t) = 1$ ), but, being also equiprobable, the probability is split evenly ( $p(h) = p(t) = 0.5$ ).

In fact, in order to be a possibility measure, a function  $\diamond$  must also satisfy:

$$\begin{aligned} \diamond(\emptyset) &= 0 \\ \diamond(X) &= 1 \\ \diamond(\cup_j A_j \subseteq X) &= \sup_j (\pi(A_j)) \end{aligned}$$

The definition highlights the non-interactive nature of the elements in the function evaluation.

Necessity, on the other hand, is defined as the negated possibility of the complement of an event:

$$\diamond(A) = 1 - \diamond(\bar{A})$$

### 1.2.5 Fuzzy Sets

The definition of fuzzy sets, given by Zadeh, is based on an extension of the normal characteristic function. A crisp set  $S$  over an universe  $X$ , in fact, is defined by the function

$$\mu_C(x) = \begin{cases} 1 & x \in S \\ 0 & x \notin S \end{cases}$$

A fuzzy set, instead, allows partial membership, so the characteristic function becomes a membership function  $\mu_F : X \rightarrow [0, 1]$ . Crisp sets can, however, be obtained from fuzzy sets by  $\alpha$ -cuts, taking the set of elements which have a degree of membership greater or equal than  $\alpha$ :

$$\alpha(\mu_F(x)) = \{x | \mu_F(x) \geq \alpha\}$$

Membership functions are plausibility distributions for an object to have the property  $F$  defines. In fact, when looking for an object having the property  $F$  in some degree  $\varepsilon$ , all  $x \in \mu^{-1}(\varepsilon)$  are possible candidates.

## Chapter 2

# Boolean Predicate Logic

### 2.1 Introduction

FOL is a symbolic language modelling relations between individuals (elements) of an universe enriched with combination rules, which allow to extract information contained implicitly in a given set of predicates or theory, and an evaluation function, which allows to establish whether a sentence is true or not. This short chapter is not meant to be an essay on predicate logic (see [], [], and many others), but will highlight the entry points which have and will be used to extend it with uncertainty.

### 2.2 Predicate Logic

#### 2.2.1 Predicates and Arguments

A predicate  $P(\mathbf{X})$  defines a relation over a number  $n$ , called *arity*, of objects, its *arguments*. Arguments are *terms*, i.e. *constants*, *variables* or *functions*. Functions, again, are relations in which the  $n + 1^{th}$  argument is univocally determined by the other  $n$  and thus is omitted.

So, given  $\mathbf{x} \in \mathbf{X}$ , we might be interested in knowing whether  $P(\mathbf{x})$  holds in that world we are trying to describe and reason on.

Formally speaking ([?]), a predicate language is a mathematical structure composed by

- An universe set  $\mathbf{U} \supset \Omega_{bool} = \{true, false\}$ .
- A set of constant symbols  $K$  such that each  $c_j \in K$  is an alias for one and only one individual  $x_j \in U$ . Usually, 0,F,'false' are constants reserved for the concept false, while 1,T and 'true' are used for true.
- A set of relations  $\{P\}$  :

$$P : \mathbf{X} = U^{n_P} \rightarrow \Omega_{bool}$$

which establish the truth (or lack thereof) of a relation  $P$  between the objects in a fixed tuple of arguments. Notice also that an argument  $x_j$  will be more properly a member of a set  $C_j \subset U$ , so  $P$  could be defined over the cartesian product set  $\prod_j C_j$ .

By definition, variable arguments do not reference any particular object, although variables with the same name must refer to the same entity.

## 2.2.2 Formulas and Evaluations

Formulas are complex statements about predicates. In fact:

$$\begin{aligned} \langle \textit{Formula} \rangle & ::= \langle \textit{Predicate} \rangle \\ \langle \textit{Formula} \rangle & ::= \langle \textit{Op} \rangle (\langle \textit{Formula} \rangle, \langle \textit{Formula} \rangle, \dots, \langle \textit{Formula} \rangle) \\ \langle \textit{Formula} \rangle & ::= \langle \textit{Quantifier} \rangle \langle \textit{Formula} \rangle \end{aligned}$$

Predicates are indeed atomic Formulas. Formulas can then be combined using logic connectives, which model the concept of negation ( $\neg$ ), conjunction ( $\wedge$ ), disjunction ( $\vee$ ), implication ( $\rightarrow$ ) and many others. Truth-functional Connectives are actually  $n$ -ary relations themselves over  $\Omega_{bool}^n$ .

Finally, Formulas with variables can be quantified, specifying that a variable could or should reference a particular, (some) or any individual in the universe. Classic logic uses two quantifiers, the universal one  $\forall$  and the existential one  $\exists$ , which is a shortcut for  $\neg\forall\neg$ .

An evaluation  $\|\cdot\|$  assigns an element  $u \in U$  to every variable in a formula, then computes the resulting truth value according to the rules:

- $\|P(\mathbf{X})\| = P(\|\mathbf{X}\|)$
- $\|Op(\mathbf{X})\| = Op(\|\mathbf{X}\|)$
- $\|\forall P(\mathbf{X})\| = \min_{X \setminus x} \|\forall P(x)\|$
- $\|x\| \in K = x \in U$

In practice, variables are bound and then constants are resolved into the objects they refer, so that the predicative relations can be evaluated according to their definitions. This basic evaluation returns a truth value for every predicate, so operators can be evaluated over their domain in order to get the truth value of a formula. Notice that, in general, the universal quantifier requires a predicate to be tested for every possible assignment of the quantified variables.

## 2.3 Inference and Resolution

Inference is a logical process aimed at extracting the consequences of a set of premises. In predicate logic, both premises and consequences are formulas. Many criteria exist, not all of which are safe and sound.

### Deduction (Modus Ponens)

$$\frac{P, P \rightarrow C}{C}$$

The safe rule of deduction combines a (actual) premise with a (general) implication to get a (actual) conclusion.

### Abduction (Modus Tollens)

$$\frac{C, P \rightarrow C}{P}$$

The unsafe rule of abduction combines a (actual) conclusion with a (general) implication to assume a (actual) premise.

### Induction

$$\frac{P(x)}{P(X)}$$

The unsafe rule of induction uses examples to establish the general validity of a formula. Notice that the dual rule,  $\frac{P(X)}{P(x)}$  is safe, instead.

The application of an inference rule produces new information which in turn can be used to apply the rules anew. This is generally called *chaining*, which comes in different variants:

- Forward Chaining : Premises are used to derive Conclusions, which in turn become Premises for new rules.
- Backward Chaining : Given a Goal Conclusion, a rule is sought that produces that Conclusion, so its Premises become new Goals
- Hybrid Chaining : Forward and Backward Chaining are used together. For example, given 1 :  $P$ , 2 :  $P \rightarrow C$ , 3 :  $C \wedge G \rightarrow E$ , P is forwarded from 1 into 3, which requires G, so a backward process is started to infer it.

The resolution algorithm, used in Prolog and many other systems, relies on modus ponens in backward chaining and imposes some constraints on formulas. The details of the algorithm can be seen in [?], so here only the underlying principles will be sketched out.

Complex Formulas must be in the form  $P \rightarrow C$  (Clauses)

$P$  is either 'true' or  $\bigwedge_j P_j$ ,  $P_j$  Atomic

$C$  is atomic

Atoms may be positive or negative

Clauses are rewritten in the form  $\neg P \vee C = \bigvee_j \neg P_j \vee C$

The resolution principle then states:

$$\frac{A_1 \vee \dots \vee A_{j-1} \vee A_j \vee A_{j+1} \vee \dots \vee A_n, B_1 \vee \dots \vee B_{k-1} \vee B_k \vee B_{k+1} \vee \dots, B_n}{A_1 \vee \dots \vee A_{j-1} \vee A_{j+1} \vee \dots \vee A_n \vee B_1 \vee \dots \vee B_{k-1} \vee B_{k+1} \vee \dots, B_n}$$

and may be applied for any two  $A_j, B_k$  that are complementary, i.e. if and only if  $A_j$  and  $\neg B_k$  unify.

Unification is well described in literature, for example in [?]. In a nutshell, two predicates  $P(\mathbf{X})$  and  $Q(\mathbf{Y})$  unify if

- $P = Q$ , i.e. the predicate is the same
- $X \setminus Y$ , i.e. the arguments are recursively compatible

Tuples in which some or all the values are assigned to constants or relations are effectively constrained by value (e.g.  $P(a, X)$  has  $\#_1 = 'a'$ ) or by reference (e.g.  $P(X, X)$  has  $\#_1 = \#_2$ ). The unification of two tuples is effectively the union of the two relative sets of constraints.

**Resolution tree** Resolution is backward and thus goal-driven. Starting from the negated existential goal, the resolution rule is applied choosing a second clause with a compatible atom, until the empty disjunction is obtained by combining  $P$  with  $\neg P$  for some atomic  $P$ . Due to the form of the rule, however, it is clear that at any moment there may be more than one eligible clause with one or more compatible atoms. So, a branching is produced which can be mapped into a tree. Typically, this tree is explored according to a depth-first strategy, which makes the resolution algorithm quite efficient (but also incomplete since it can get trapped in loopy infinite-depth branches). However, as soon as the first success branch is met, the unification path is returned with success.

## 2.4 Logic Networks

The application of forward-chaining strategies, instead, leads to different topological approaches.

### 2.4.1 Boolean Logic Networks

A theory made of ground predicates (with no variable arguments) is equivalent to a theory in propositional logic. The equivalence between propositional logic programs and networks, combinatorial (stateless) or stateful depending on the presence or lack of feedback, is, among other things, the basis of the design of electronic circuits. The interesting point is the chain of equivalences

$$\text{logic program} \equiv \text{flow graph} \equiv \text{generic function}$$

In fact, a logic program - a set of Formulas - is isomorphic to a graph in which nodes represent propositions and connectives while edges model the relation

argument-of. Choosing some nodes as inputs and observing others as outputs allows to implement any computable function  $\{0,1\}^n \rightarrow \{0,1\}^m$  operating on an appropriate binary encoding of its domain and range. On the other side, a function can be described formally by a declarative logic program.

These results do not hold for propositional calculus only: in fact, they are more general.

## 2.4.2 RETE Networks

The RETE algorithm proposed by C.L.Forgy compiles a logic program (set of clauses) into a graph. Like in Boolean networks, computation is performed by the flow truth values along the edges and by the computation at the nodes. In RETEs, however, arguments must be passed along as well. Inputs, in fact, are not simple bits, but true predicates usually called *facts*.

RETE has been designed for rules in the form:

$$\begin{array}{l} \textit{When} \\ \quad P_1 \wedge \dots \wedge P_k \\ \textit{Then} \\ \quad Q, R, \dots \end{array}$$

The network is composed of two parts, called  $\alpha$  and  $\beta$ .

**$\alpha$ -network** The  $\alpha$ -net performs syntactical pattern matching on each predicate  $P_j$ 's arguments. For each constraint on  $P[k]$  (where the vectorial notation selects the  $k^{th}$  arg), an  $\alpha$ -node is created to test it. For each pattern  $P_j$ , the appropriate  $\alpha$ -nodes are chained sequentially. Chains of nodes are reused if two patterns share some constraints.

For example, the presence of two patterns  $P(a, b)$  and  $P(a, X)$  in any two rules produces the type node  $T_P$  and the two select nodes  $\#_{P1} : P[1] = a$  and  $\#_{P2} : P[2] = b$ . Afterwards, an  $\alpha$ -net is created by adding the flow edges  $\{< T_P, \#_{P1} >, < \#_{P1}, \#_{P2} >\}$ . Facts not filtered out by  $\#_{P1}$  are good matches for  $P(a, X)$ , while facts reaching the output of  $\#_{P2}$  as well are matches for  $P(a, b)$  since both constraints have been satisfied.

**$\beta$ -network** The  $\beta$ -net performs between-predicate pattern matching. Each rule premise is a relation over the cartesian product  $\prod_k P_k$ . The individual set of admissible facts  $P_k$  are created by the  $\alpha$ -nets, but these sets must be joined in a relational sense. The join is performed incrementally, from left to right, by  $\beta$ -nodes. If two rules share a part of their premise,  $\beta$ -nodes are shared as well.

**Conflict and Fire** The output of the join, if not empty, *activates* the rule. At a given time, there may be a set of more than one active rule, called conflict set. Unless parallel computation is supported, rules must be fired sequentially, the choice being ruled out by some conflict resolution criterion. As a consequence of the firing of a rule, new facts may be asserted which, in turn, may activate

other parts of the network. This type of interaction between nodes causes rules to be chained in a feed-forward way.

There exist various implementation of the algorithm, most of which optimized to improve performance and functionality. An extension of predicate networks capable of handling uncertainty will be further discussed in later chapters.

## Chapter 3

# Interval-Based Truth Belief System

### 3.1 Truth values representations

In classical logic, only two symbols are required to reference the concepts of "true" and "false". The sets  $\{0,1\}$ ,  $\{T,F\}$  and variants are commonly used. However, the necessity to formalize uncertainty leads to more complex representations. Truth, in fact, can be present in a Degree which is partial and not necessarily known with absolute precision.

#### 3.1.1 Real values

In Multi-valued logic, partial truth degrees are mapped into the continuous interval  $[0,1]$ . Hence, a single real value is sufficient to annotate the predicates.

$$\{predName, [Args], t \in [0, 1]\}$$

#### 3.1.2 Intervals

A simple real number is not sufficient as soon as the truth values become imprecise. The immediate extension makes use of a real interval:

$$[\tau, 1 - \varphi] \subseteq [0, 1]$$

The values of  $\tau$  and  $\varphi$  define the upper and lower bound of the truth value, respectively.  $\tau$ , in fact, measures the necessity of the *truthof* the predicate, while  $\varphi$  measures its possibility.

From these two values, it is immediate to define a trivial "certainty" parameter

$$\chi(P) = \tau + \varphi$$

$\chi(P) \in [0, 1]$  itself measures the truth value of the meta-predicate *certain(P)* = "The truth of P is known with absolute precision "

### 3.1.3 Fuzzy numbers

The interval representation becomes non-informative as the intervals grow broader since it only gives information about the values which are not the actual truth value.

A different approach uses fuzzy numbers, i.e. fuzzy sets on a numerical domain:

$$\mu_N : [0, 1] \rightarrow [0, 1]$$

with the properties:

- Normalized :  $\exists!x : \mu_N(x) = 1$
- Piecewise continuous
- Convex :  $\forall x < N : \mu'_N(x) > 0, \forall x > N : \mu'_N(x) < 0$   
(consider left and right first derivatives in discontinuity points)
- Bounded :  $\exists\tau, \varphi : \forall x \notin [\tau, 1 - \varphi] : \mu_N(x) = 0$

#### Gaussian distributions

Gaussians provide a convenient representation of fuzzy numbers since they are unimodal functions that can be represented by just a pair of values  $\{\mu, \sigma\}$ . Gaussians model the concept of “( $\sigma$ -dependent) *more-or-less*  $\mu$ ”. If a Gaussian is used, one can choose  $\tau = \mu - k\sigma$  and  $1 - \varphi = \mu + k\sigma$ , with  $k$  being typically more-or-less 3.

### 3.1.4 Imprecise Bayesian Belief Mass Distributions

Since fuzzy numbers are actually used to describe the truth value of a fuzzy concept, they are more properly called type-II fuzzy sets. In fact, a membership function  $\mu_{II}$  over  $[0,1]$  assigns to every possible truth value a degree which has different interpretations, namely the usual:

- Probability : the likelihood (subjective or objective) that  $x$  is the real truth value
- Possibility : how acceptable it would be to use  $x$  as the real truth value
- Similarity : how much  $x$  resembles the real truth value

If  $\mu_{II}$  is unimodal (i.e. it has a single maximum), neither is more correct than the other and often the choice is dictated more by the way  $\mu_{II}$  is defined and computed. In a sense, all of them state how suited is a chosen precise value to be used as truth value.

The requisite of unimodality, however, is not strictly necessary. Whenever  $\mu_{II}$  is multimodal, it provides more than one “natural” choice and so its probabilistic side can not be ignored (but neither the others).

Thus, the following step in generalization can be taken by defining a belief structure on  $[0,1]$ , which is the most natural way to describe the observer's credo in the truth of a proposition. The membership function becomes a mass measure function:

$$\mu_{II,bel} : \wp([0,1]) \rightarrow [0,1]$$

Unfortunately,  $\wp(X)$  is a complex domain to be used in practice due to its infinite cardinality. In order to make it tractable, the interval  $[0,1]$  is made discrete by dividing it into  $N$  intervals. (A typical choice for  $N$  is 10).

$$\mu_{II,belD} : \wp \left\{ \left( \frac{j}{N}, \frac{j+1}{N} \right]_{j:0..N-1} \right\} \rightarrow [0,1]$$

For a scalable implementation,  $2^N$  is usually still too large, so one should consider the probabilistic distribution assigning non-null masses only to singletons:

$$\mu_{II,probD} : \left\{ \varepsilon_j = \left( \frac{j}{N}, \frac{j+1}{N} \right]_{j:0..N-1} \right\} \rightarrow [0,1]$$

As before, however, problems may arise in evaluating the exact value of  $\mu_{II}$ . So, an imprecise probability model is adopted to finally define the approach which will be used from now on:

$$\mu_{II,probD} : \left\{ \varepsilon_j = \left( \frac{j}{N}, \frac{j+1}{N} \right]_{j:0..N-1} \right\} \rightarrow \{p_L \leq p \leq p_U\} \in [0,1] \times [0,1] \times [0,1]$$

The triple  $\{p_L, p, p_U\}$  implicitly defines a triangular fuzzy number.

## 3.2 Belief construction

The truth of a proposition  $P$  about some arguments  $\mathbf{X}$  is the actual evaluation of the function  $P(X)$ . Direct observation of  $\mathbf{X}$  and its interpretation in terms of  $P$  may be considered a generalized experiment, a trial subject to errors.

$P(X)$  can be considered a random variable, whose args may be random themselves or not, on the finite domain  $\{\varepsilon_j\}$ , so the "sampling" activity has the aim of evaluating the associated probabilities  $\{p_L^j, p^j, p_U^j\}$ . These probabilities can either be frequentist or subjective, whether the sampling is actual or ideal.

### 3.2.1 Multinomial model

A full bayesian approach requires an adequate prior probability distribution  $\{d_j\}_{j:0..N-1}$ , reflecting prior knowledge, which is then updated by the posterior observations. Since the intervals are finite and mutually exclusive, the obvious

choice is an N-variate Dirichlet distribution:

$$Dirichlet_N(\mathbf{p}|\beta) = \frac{\Gamma\left(\sum_{j=0}^{N-1} \beta_j + N\right)}{\prod_{j=0}^{N-1} \Gamma(\beta_j + 1)} \prod_{j=0}^{N-1} p_j^{\beta_j}$$

The parameters  $\beta_j$  may be considered the actual relative number - or, better, relative weight - of observations of the  $j^{th}$  event, over a total amount of observations defined by:

$$B_{\text{tot}} = \sum_{j=0}^{N-1} \beta_j$$

In fact, the posterior distribution conditioned by observations  $\beta^*$  has parameters  $\beta + \beta^*$ .

The distribution has a maximum, which gives the max-likelihood probability, at

$$\mathbf{p}^{ml} = \frac{\beta}{B_{\text{tot}}}$$

The variance, which when low makes the choice of  $p^{ml}$  quite convenient given its easy computability, is:

$$\sigma_j^2 = \frac{(\beta_j + 1)(B_{\text{tot}} - \beta_j + N - 1)}{(B + N)^2(B + N + 1)}$$

### 3.2.2 Confidence

The variance is maximized by the uniform distribution  $p = 1/N$ :

$$\sigma_{j,max}^2 = \frac{(N - 1)}{N^2(B_{\text{tot}} + N + 1)}$$

So, the total variance is bounded:

$$\sum_{j=0}^{N-1} \sigma_j^2 \leq \frac{N(N - 1)}{N^2(B_{\text{tot}} + N + 1)}$$

Thus, it is possible to define a threshold:  $\exists B_{\text{ref}} : \sum_{j=0}^{N-1} \sigma_j^2 \leq \sigma_0^2$ . Considering that usually  $B_{\text{tot}} \gg N$ , the following approximations can be made:

$$B_{\text{ref}} \approx \frac{N(N - 1)}{\sigma_0 N^2} - (N + 1) \quad (3.1)$$

$$\sum_{j=0}^{N-1} \sigma_j^2 \leq \sigma_0^2 \Leftrightarrow B_{\text{tot}} \geq B_{\text{ref}} \quad (3.2)$$

From the last equation, it is possible to define the confidence in the probability estimate  $p^{ml}$ :

$$\text{Confidence}(p^{ml}|\beta) = \frac{B_{\text{tot}}}{B_{\text{ref}}}$$

### 3.2.3 Priors and posteriors - “dogmatism ”

The empirical observations, up to target amount  $B_{\text{ref}}$ , update the prior distribution  $\{d_j\}$  according to the Dirichlet composition rule. The relative strenght of the priors is defined by a parameter  $\delta$  as follows:

$$\begin{aligned} B_{\text{ref}} &= B_D + B_O \\ B_D &= \delta \cdot B_{\text{ref}} \\ B_O &= (1 - \delta) \cdot B_{\text{ref}} \end{aligned}$$

$B_D$  is prior and given, while  $B_O$  is the actual empirical target. In fact, the parameter vector is so initialized:

$$\beta_j^0 = d_j \cdot B_D$$

As observations are made, the parameters are updated:

$$\beta_j^t = \beta_j^{(t-1)} + w_j^t \cdot B_O$$

Or, equivalently:

$$\beta_j^T = d_j \cdot B_D + \sum_{t=0}^T w_j^t \cdot B_O$$

Each observation increases the belief by assigning a mass  $W^t$  to a set  $J$  of intervals. This mass can be transformed into a bayesian distribution  $\{w_j\}$  according to one of the techniques seen in 1.2.2. Obviously, to have the sum converge to the target value,  $\sum_{t=0}^{\infty} W^t = 1$ , so the weights should depend on the cardinality of the set of observables. Called  $|X|$  the maximum number of observations that can be made, if  $|X|$  is finite and the observations are equally relevant, one can choose

$$W^t = \frac{B_O}{|X|}$$

In case of non-uniform relevance, a measure  $\rho$  on  $X$  is needed to give more weight to more important events:

$$W^t = B_O \cdot \rho^t$$

Notice that the more general case covers the former by putting  $\rho^t = \frac{1}{|X|}$

If  $|X| \rightarrow \infty$ , a different incremental approach must be used. Moreover, it may not be clear whether a new observation is independent of the former ones or not. In general, put  $\beta_o^t = w_o^t \cdot B_O$ :

$$\beta^{(t+1)} = \beta^t + \beta_o^t - \alpha \cdot \beta^t \cdot \beta_o^t$$

The choice of  $\alpha$  is arbitrary, but for particular values:

$$\begin{aligned} \alpha = 0 &: \text{Independent observation} \\ \alpha = 1 &: \text{Cumulative observation} \\ \alpha = \frac{1}{\beta^t} &: \text{Completely dependent observation} \end{aligned}$$

The choice of  $\alpha = 1$  is possibly the most appropriate when the degree of dependence between observation is not known. This, however, leaves the open problem of choosing the weights  $W^t$ . If the weights are constant and equal to  $w$ , however, an appropriate choice is given by the solution of the problem:

$$\begin{cases} W^{T+1} = W^T + w - w \cdot W^T \\ W^0 = 0 \\ \forall T: W^T \leq 1 \\ w \leq 1 \end{cases}$$

Where  $W^T = \sum_{t=0}^T W^t$ . Solving the difference equation yields:

$$W^T = 1 - (1 - w)^T$$

$w$  and  $T$  must be chosen so that  $\lim_{T \rightarrow \infty} W^T = 1$  or, equivalently,  $(1 - w)^T \leq \epsilon$ .

In many cases, especially if  $T$  models a time-equivalent variable, it is reasonable to fix  $w$  so to reach full confidence within a fixed, large number of observations:  $w \geq 1 - e^{\frac{1}{T} \ln \epsilon}$ . Sometimes, however,  $w$  may be fixed: for example, whenever the observation set is finite but sampled randomly, so that independence can't be guaranteed. In such cases, almost-full confidence is achieved after  $T \geq \frac{\ln \epsilon}{\ln 1 - w}$  observations, so  $w$  can be set to be proportional to  $|X|$  accordingly, as needed.

### 3.2.4 Interval definition

**Belief structure** So far, only the contribution to the total confidence has been considered. In the general case each observation adds its weight to a subset  $\varepsilon_J$  ( $J = \{j_1..j_M\}$  is the characteristic set containing the indexes of the singleton members  $\varepsilon_{j_m}$ ). So, for each subset  $\varepsilon_J$ , it is possible to define a basic empirical mass assignment by taking the ratio of the observations for each subset  $\varepsilon_J$  over the total observations for all subsets :

$$m_{\varepsilon_J}^T = \frac{W_{\varepsilon_J}^T}{W^T}$$

From the mass assignment, it is easy to build the empirical belief measure:

$$Bel_{\varepsilon_J}^T = \sum_{\varepsilon_K \subseteq \varepsilon_J} m_{\varepsilon_K}^T = \frac{1}{W^T} \sum_{\varepsilon_K \subseteq \varepsilon_J} W_{\varepsilon_K}^T$$

In general, however, there is a prior mass assignment  $\{d_{\varepsilon_J}\}$  which must be taken into account. This allows to build a global belief and plausibility measures

to bind the probability of a given subset:

$$\frac{\sum_{\varepsilon_K \subseteq \varepsilon_J} (W_{\varepsilon_K} B_O + d_{\varepsilon_K} B_D)}{W B_O + B_D} \leq p_{\varepsilon_J} \leq \frac{\sum_{\varepsilon_K \cap \varepsilon_J \neq \emptyset} (W_{\varepsilon_K} B_O + d_{\varepsilon_K} B_D)}{W B_O + B_D}$$

The denominator is equal to  $B_{\text{tot}}$  as it is the sum of dogmatic priors and actual empirical observations. The numerators, instead, lead to different considerations. One could distrust the prior mass assignment while retaining the dogmatic credit and thus build pure empirical bounds as follows:

$$\begin{aligned} \frac{\sum_{\varepsilon_K \subseteq \varepsilon_J} (W_{\varepsilon_K} B_O + d_{\varepsilon_K} B_D)}{W B_O + B_D} &\geq \\ \frac{\sum_{\varepsilon_K \subseteq \varepsilon_J} W_{\varepsilon_K} B_O}{W B_O + B_D} &= \\ \frac{\sum_{\varepsilon_K \subseteq \varepsilon_J} \frac{W_{\varepsilon_K}}{W}}{1 + \frac{B_D}{W B_O}} &\leq Bel_{\varepsilon_J} \end{aligned}$$

As shown by the last relation, the empirical lower bound is looser than the empirical belief since it is discounted by the factor  $\frac{B_D}{W B_O}$ . Similar considerations can be made for the upper bound and plausibility: the dogmatic weight  $B_D$  is tentatively assigned as a whole to each of the subsets, yielding

$$\begin{aligned} \frac{\sum_{\varepsilon_K \cap \varepsilon_J \neq \emptyset} (W_{\varepsilon_K} B_O + d_{\varepsilon_K} B_D)}{W B_O + B_D} &\leq \\ \frac{\sum_{\varepsilon_K \cap \varepsilon_J \neq \emptyset} (W_{\varepsilon_K} B_O) + B_D}{W B_O + B_D} &= \\ \frac{\sum_{\varepsilon_K \cap \varepsilon_J \neq \emptyset} \frac{W_{\varepsilon_K}}{W} + \frac{B_D}{W B_O}}{1 + \frac{B_D}{W B_O}} &\geq Pl_{\varepsilon_J} \end{aligned}$$

The penalty becomes smaller as  $\frac{B_D}{W B_O} = \frac{\delta}{(1 - \delta)W} \rightarrow 0$ . This happens either if  $\delta \rightarrow 0$ , i.e. the priors are given very little credit, or  $(1 - \delta)W \gg \delta$ , i.e. enough observations have been collected. Notice also that whenever  $\delta \rightarrow 1$  the empirical bounds become non-informative, but the priors are almost non-influenced by the posteriors.

**Probability structure and bounds** When the mass assignment is bayesian (probabilistic) and each observations targets a singleton (i.e.  $|J| = 1$ ), the belief and plausibility bounds coincide, but the empirical bounds can still be used:

$$\frac{W_{\varepsilon_j} B_O}{W B_O + B_D} \leq \frac{W_{\varepsilon_j} B_O + d_{\varepsilon_j} B_D}{W B_O + B_D} \leq \frac{W_{\varepsilon_j} B_O + B_D}{W B_O + B_D}$$

As noted in 1.2.2 it is possible to convert a non-bayesian mass assignment into a bayesian one. The pignistic projection is a typical choice and will be used here to define new bounds. From the functions

$$\delta_j(J) = \begin{cases} 1 & \text{if } J = \{j\} \\ 0 & \text{else} \end{cases} \quad \kappa_j(J) = \begin{cases} \frac{1}{|J|} & \text{if } j \in J \\ 0 & \text{else} \end{cases} \quad \lambda_j(J) = \begin{cases} 1 & \text{if } j \in J \\ 0 & \text{else} \end{cases}$$

Given a prior bayesian distribution  $\{d_j\}$ , the following relations hold:

$$p_j = \frac{B_O \sum_{t=0}^T \kappa_j(J) W_J^t + d_j B_D}{W^T B_O + B_D}$$

$$\frac{B_O \sum_{t=0}^T \delta_j(J) W_J^t + d_j B_D}{W^T B_O + B_D} \leq p_j \leq \frac{B_O \sum_{t=0}^T \lambda_j(J) W_J^t + d_j B_D}{W^T B_O + B_D}$$

$$\frac{B_O \sum_{t=0}^T \kappa_j(J) W_J^t}{W^T B_O + B_D} \leq p_j \leq \frac{B_O \sum_{t=0}^T \kappa_j(J) W_J^t}{W^T B_O + B_D}$$

$$p_L^j = \frac{B_O \sum_{t=0}^T \delta_j(J) W_J^t}{W^T B_O + B_D} \leq p_j \leq \frac{B_O \sum_{t=0}^T \lambda_j(J) W_J^t + B_D}{W^T B_O + B_D} = p_U^j$$

The bounds here defined are technically bias bounds, since they measure the uncertainty about the position of the mode of the Dirichlet distribution. For a fixed set of parameters, however, it is possible to evaluate the variance bounds by analyzing the Dirichlet curve or, more easily, by using the variance bounds defined earlier. Combining bias and variances yields:

$$-k\sqrt{\sigma_j^2} + p_L^j \leq p_j \leq p_U^j + k\sqrt{\sigma_j^2}$$

**Possibility and Necessity bounds** Fixed a “zero” threshold  $\epsilon$ , the features  $\tau$  and  $\varphi$  can be redefined, keeping the same semantics:

$$\tau = \max_j \{k \leq j \Rightarrow p_U^k \leq \epsilon\}$$

$$\varphi = \min_j \{k \geq j \Rightarrow p_U^k \leq \epsilon\}$$

## 3.3 Appendix

### 3.3.1 Default Belief structure

The usage of the Belief Structure defined in this chapter depends on several parameters. Here some reasonable default values are given as an advice and not as a recommendation.

$N = 10$  A compromise between accuracy and complexity. It is also possible to set  $N=2$  to model binary probabilistic logic.

$$\sigma_{j,0}^2 = 10^{-5}$$

$$\sigma_0^2 = 10^{-6}$$

$$B_{ref} \approx 10^5 \quad \text{Computed directly from } N \text{ and } \sigma_0^2.$$

$\delta = 10^{-3}$  From the definition of  $p_j^U$ , it has a bias and a variance component. The latter behaves like  $\sqrt{NB} \approx 10^{-3}$ . The former should behave likewise for large  $B$ , even when  $d_j = 1$ , in order to have a reliable zero-out-of-many-observations  $p_j \leq \epsilon$ .

$\epsilon = 10^{-2}$   $N$  times  $10^{-3}$ , to compensate for multiplicative constants. See above.

### 3.3.2 Noteworthy values

**True**  $\{0, \dots, 0, B_{ref}\}$

**False**  $\{B_{ref}, 0, \dots, 0\}$

**Neither True Nor False**  $\{0, \dots, B_{ref}, \dots, 0\}$

**Not Know**  $\{0, \dots, 0\}$

**Know Not**  $\{\frac{B_{ref}}{N}, \dots, \frac{B_{ref}}{N}, \dots, \frac{B_{ref}}{N}\}$

**Heads or Tails**  $\{\frac{B_{ref}}{2}, 0, \dots, 0, \frac{B_{ref}}{2}\}$

# Chapter 4

## Uncertain Inference

This chapter deals with the issues of uncertain reasoning. In chapter 2, first order predicate logic has been introduced. Here, we start discussing some extension of such logic which take uncertainty into account. Then, we show how the traditional inference processes, namely induction, deduction and abduction, are to be redefined in the new context. Finally, we extend the RETE algorithm to create networks capable of transforming uncertain information.

### 4.1 Fuzzy Logic

Fuzzy logic, as originally proposed by Zadeh, reasons on and with fuzzy sets and relations. As defined in 1.2.5, a fuzzy set over an universe  $\mathbf{X}$  defines a property  $P$  which elements  $\mathbf{x} \in \mathbf{X}$  have in a certain degree set by the membership evaluation function. This property is a predicate, true in the same degree, of its arguments  $\mathbf{x}$ .

$$\mu_P : \mathbf{X} \rightarrow [0, 1]$$

The function  $\mu_P$  defines the concept  $P$ , at the point that  $\mu_P$  and  $P$  are equivalent and thus can be used interchangeably. The notation  $P(\mathbf{X})$ , however, is more suitable to a logic context.

A fuzzy set is usually enumerated by listing the couples element-membership as fractions  $\{\frac{\mathbf{x}}{\mu_P(\mathbf{x})}\}$  but this notation will not be used here. Instead, we will use the annotated logic notation

$$P(\mathbf{x})_\varepsilon$$

with  $\varepsilon = \mu_P(\mathbf{x})$ .

**Linguistic variables** Given the concept of fuzzy set, fuzzy logic defines “linguistic variables”: an enumeration of fuzzy sets on the same domain which, when the domain is ordered, usually correspond to different scales.

For example, consider the domain  $H = \{\text{humans's height}\} = [25..250]$  (values are extreme, arbitrary and not so relevant).

A typical linguistic variable is  $Height = \{\text{VeryShort, Short, Medium, Tall, VeryTall}\}$ , with the five values being fuzzy sets (membership functions) over  $H$ . Various additional conditions can optionally be imposed over a variable  $V$ :

- **Coverage** :  $\forall \mathbf{x} \in \mathbf{X} : \exists V_j \in V : V_j(\mathbf{x}) > 0$
- **Normality** :  $\forall \mathbf{x} \in \mathbf{X} : \sum_j V_j(\mathbf{x}) = 1$
- **Chaining** :  $\forall \mathbf{x} \in \mathbf{X} : \forall i \neq j \neq k : (V_i(\mathbf{x}) > 0 \wedge V_j(\mathbf{x}) > 0) \Rightarrow V_k(\mathbf{x}) = 0$
- **Convexity** :  $\forall j : V_j$  is convex

**The hidden variable issue and Possibility functions** In the example cited above, the memberships  $V_j$  may be defined as functions of their argument  $\mathbf{x}$ . In other cases, the definition is tabular, or declarative. In some circumstances, however, the declarative definition hides a functional one or vice versa. For example, it is common to see the usual linguistic variable,  $Height$ , defined over  $\mathbf{M}$ , the set of all people. Thus,

$$V_j^*(\mathbf{m})_\varepsilon \Leftrightarrow \text{HeightOf}(\mathbf{h}, \mathbf{m})_1 \wedge V_j(\mathbf{h})_\varepsilon$$

So, a person  $m$  is Tall (resp.) if it they have a certain height  $h$  and  $h$  is Tall (resp.) for being a height. This logical equivalence, however, points out one of the limits of fuzzy logic “in a broad sense”([?]): in many cases, the value  $h$  of  $m$ 's height is not known with precision. In that case, the sentence above is better written:

$$V_j^*(\mathbf{m})_\varepsilon \Leftrightarrow (\exists \mathbf{h} : \text{HeightOf}(\mathbf{h}, \mathbf{m})) \wedge V_j(\mathbf{h})_\varepsilon$$

Given the approximation  $h$  that most resembles  $m$ 's height,  $m$  is Tall if  $h$  is Tall and  $h$  is accurate. Or, better:  $m$  is Tall if  $h$  is Tall, the better  $h$  reflects the true value of  $m$ 's height. But what if  $h$  is inaccurate?

As will be shown later, using standard fuzzy logic inference rules leads to the undesirable result that  $m$  is not Tall. This is a form of closed world assumption (what is not known is false) that arises from the assumed lack of coverage and/or normality in the fuzzy set evaluation.

The equivalence, however, can be read from the other side ( $\Rightarrow$ ). Knowing that  $m$  is Tall in a certain degree influences the possibility that a given  $h$  is the real height of  $m$ . For example, a Tall person is very unlikely to be 120cm (4ft.) tall. In general, if we know a full property  $V_j^*(\mathbf{m})_1$ , we obtain a possibility measure on  $H$  by taking  $V_j$  itself. Like before, it is not immediately clear what should happen when  $V_j^*(\mathbf{m})_{\varepsilon < 1}$ : the problem will be dealt with in the following sections.

## 4.2 Belief-based Fuzzy Logic in a Narrow Sense

In order to better understand the implication of the simple example proposed in the last paragraph, it is necessary to give a formal definition of fuzzy logic and its inference mechanisms. We shall mostly refer Hajek's work ([?]), which in turn is based on Pavelka's essays ([?]), which in turn are founded on Lukasiewicz's studies ([?]). In the following section, we will show how the so-called rational logics are particularly suited to elaborate the belief-structured truth values defined in chapter 3.

### 4.2.1 Many-valued Logic

Multi-valued logics are an extension of classical logic and deal with predicates annotated with a truth degree. Even if in chapter 3 various definitions of truth degree have been provided, many-valued logics have been defined for simple real (actually, rational) valued truth degrees, trying to extend classical logic while keeping as many intuitively acceptable properties as possible.

The many-valued extension of a predicate logic (see 2) requires:

- A *T-norm*  $*$  (strong conjunction)
- An ordered set  $L$  of truth values

A proper choice for  $*$  and  $L$  allows the definition of a larger set of operators.  $n$ -ary operators so defined are truth-functional: given the  $n$  truth values  $\varepsilon_{j:1\dots n}$  of the operands, there exists a (membership) function  $Op : L^n \rightarrow L$  which assess the degree of truth of the complex formula.

#### Logic families

**T-Norms** It can be shown (see [?]) that, even if there are very few requirements for a function to be a *T-norm* (namely it must be commutative, associative and non-decreasing in its arguments), there are only three basic T-norms: all other T-norms are isomorphic to one of them. Called  $\varepsilon_A$  and  $\varepsilon_B$  the truth values of the operands:

$$\begin{aligned} \text{Lukasiewicz : } \quad & \varepsilon(A \otimes B) = \max\{0, \varepsilon_A + \varepsilon_B - 1\} \\ \text{Gouguen : } \quad & \varepsilon(A \odot B) = \varepsilon_A \cdot \varepsilon_B \\ \text{Godel : } \quad & \varepsilon(A \wedge B) = \min\{\varepsilon_A, \varepsilon_B\} \end{aligned}$$

A T-Norm is a strong conjunction.

**S-Norms (Residua of T-Norms)** The dual Norm  $\Rightarrow$  models implication. The general definition is:

$$\varepsilon(A \Rightarrow B) = \max_{\varepsilon_{\Rightarrow}} \{\varepsilon_A * \varepsilon_{\Rightarrow} \leq \varepsilon_B\}$$

So, there is a different  $S$  – Norm for each family:

$$\begin{aligned} \text{Lukasiewicz : } & \varepsilon(A \rightarrow_L B) = \min\{1, 1 - \varepsilon_A + \varepsilon_B\} \\ \text{Gouguen : } & \varepsilon(A \rightarrow_\pi B) = \min\{1, \frac{\varepsilon_B}{\varepsilon_A}\} \\ \text{Godel : } & \varepsilon(A \rightarrow_G B) = \varepsilon_A \leq \varepsilon_B ? \varepsilon_B : 1 \end{aligned}$$

Implication is particularly important in the definition of rules, which are tautological implications, and so will be further discusses in the following section 4.2.2.

**Negation** The *Not* operator is defined by the equivalence

$$\neg A = A \Rightarrow 0$$

$$\begin{aligned} \text{Lukasiewicz : } & \neg_L \varepsilon_A = 1 - \varepsilon_A \\ \text{Gouguen : } & \neg_\pi \varepsilon_A = \varepsilon_A > 0 ? 0 : 1 \\ \text{Godel : } & \neg_G \varepsilon_A = \varepsilon_A > 0 ? 0 : 1 \end{aligned}$$

Lukasiewicz’s negation alone preserves graduality. However, it is often seen in Product-like logics.

**T-Conorm** The T-Conorm is a strong disjunctive operator:

$$A + B = \neg A \Rightarrow B$$

It can also be defined using De Morgan’s law:

$$A + B = \neg(\neg A * \neg B)$$

$$\begin{aligned} \text{Lukasiewicz : } & \varepsilon(A \oplus B) = \min\{1, \varepsilon_A + \varepsilon_B\} \\ \text{Godel : } & \varepsilon(A \vee B) = \max\{\varepsilon_A, \varepsilon_B\} \end{aligned}$$

In literature, the commonly used definition for Gouguen’s disjunction is  $A +_\pi B = \neg_L(\neg_L A \odot \neg_L B)$  which is evaluated by the function  $\varepsilon_A + \varepsilon_B - \varepsilon_A \cdot \varepsilon_B$ .

**Inf and Sup** The weak conjunction and weak disjunction operators are defined by the tautologies:

$$A \wedge B = A * (A \Rightarrow B)$$

and

$$A \vee B = ((A \Rightarrow B) \Rightarrow B) \wedge ((B \Rightarrow A) \Rightarrow A)$$

In all logic families,  $\varepsilon(A \vee B) = \max\{\varepsilon_A, \varepsilon_B\}$  and  $\varepsilon(A \wedge B) = \min\{\varepsilon_A, \varepsilon_B\}$ . In Godel’s logic, strong and weak conjunction and disjunction coincide, but this is not true in other logics.  $\vee$  and  $\wedge$  obey De Morgan’s law.

**X-Or and Equivalence** Equivalence is commonly defined by a double implication:

$$A \equiv B = (A \Rightarrow B) * (B \Rightarrow A)$$

X-Or, then, is the negation of equivalence:

$$A \neq B = \neg(A \equiv B)$$

In Lukasiewicz's logic the following property holds

$$\begin{aligned}\varepsilon(A \equiv_L B) &= 1 - |\varepsilon_A - \varepsilon_B| \\ \varepsilon(A \neq_L B) &= |\varepsilon_A - \varepsilon_B|\end{aligned}$$

It shows that in many-valued logic the concept of difference (resp. equality) is strongly connected to the notion of distance.

**Power operator (linguistic edges)** For a given real value  $r \in \mathfrak{R}^+$

$$A^r = A * A * \dots * A \text{ (r times)}$$

The truth value can be computed directly:

$$\begin{aligned}\text{Lukasiewicz : } \varepsilon(A!_r) &= \max\{0, 1 - r(1 - \varepsilon_A)\} \\ \text{Gouguen : } \varepsilon(A^r) &= (\varepsilon_A)^r\end{aligned}$$

The definition is not suitable for Gödel's logic since the T-norm is idempotent. If  $r \gg 1, \varepsilon(A^r) \rightarrow 0$ : the operator model linguistic edges such as very, largely, highly. Instead, when  $r \rightarrow 0, \varepsilon(A^r) \rightarrow 1$ , modeling the dual, broadening concepts.

**Operators for Extended Truth Values** The operators' evaluation functions are defined over the domain  $L^n$  and thus are suitable for real-valued truth degrees. However, they can be generalized to operate on the more complex representation defined in chapter 3.

**Truth Intervals** Truth values are defined by the couple of bounds:  $\varepsilon_P = [\tau_P, 1 - \varphi_P]$ . The couple  $\tau_{Op}, \varphi_{Op}$  may be computed directly from the definition of every operator  $Op$ . For Lukasiewicz's operators, in particular, one gets:

$$\begin{aligned}\varphi_P &\leq \neg P &\leq 1 - \tau_P \\ \max\{0, \sum_j \tau_j - (n - 1)\} &\leq \otimes P_j &\leq 1 - \min\{1, \sum_j \varphi_j\} \\ \min\{1, \sum_j \tau_j\} &\leq \oplus P_j &\leq 1 - \max\{0, \sum_j \varphi_j - (n - 1)\} \\ \min\{\tau_j\} &\leq \wedge P_j &\leq 1 - \max\{\varphi_j\} \\ \max\{\tau_j\} &\leq \vee P_j &\leq 1 - \min\{\varphi_j\} \\ \min\{1, \varphi_P + \tau_C\} &\leq P \rightarrow C &\leq 1 - \max\{0, \tau_P + \varphi_C - 1\} \\ \min\{|\psi|, |\gamma|, \psi \cdot \gamma > 0 ? 1 : 0\} &\leq P \neq Q &\leq 1 - \max\{|\psi|, |\gamma|\} \\ \psi &= 1 - \varphi_P - \tau_Q \\ \gamma &= \tau_P + \varphi_Q - 1\end{aligned}$$

The interval structure allows the definition of two more threshold operators:

$$\begin{aligned} \tau_P \geq \lambda ? 1 : 0 &\leq T_\lambda(P) \leq \varphi_P > 1 - \lambda ? 0 : 1 \\ \varphi_P \geq \mu ? 1 : 0 &\leq \Phi_\mu(P) \leq \tau_P > 1 - \mu ? 0 : 1 \end{aligned}$$

**Truth Distributions** Given  $N$  distributions  $\varepsilon^n : [0, 1] \rightarrow [0, 1]$  and a function  $Op : [0, 1]^n \rightarrow [0, 1]$  the general composition principle states that

$$\varepsilon_{Op} = \int_{\{\varepsilon^n\}:Op(\{\varepsilon^n\})=\varepsilon_{Op}} \prod_n \varepsilon_n$$

When the distributions are finite, the integral becomes a sum. In practice, all possible combinations are computed, then grouped. For example, the binary operator  $\otimes$  with 3-valued distributions returns a cumulation matrix:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \begin{aligned} \varepsilon_{\otimes}[2] &= \varepsilon_A[2] \cdot \varepsilon_B[2] \\ \varepsilon_{\otimes}[1] &= \varepsilon_A[2] \cdot \varepsilon_B[1] + \varepsilon_A[1] \cdot \varepsilon_B[2] \\ \varepsilon_{\otimes}[0] &= \varepsilon_A[0] \cdot \varepsilon_B[0] + \varepsilon_A[0] \cdot \varepsilon_B[1] + \dots \end{aligned}$$

The bounds  $\tau$  and  $1 - \varphi$  can be used for greater efficiency, substituting the whole set  $[0, 1]^N$  with the cartesian product  $\prod_j [\tau_j, 1 - \varphi_j]$

Consider now a general binary operator with N-slotted distributions as operands: the formula may be rewritten as follows.

$$\varepsilon[z] = \frac{1}{B_j \cdot B_k} \sum_{Op(j,k)=z} \beta_j \cdot \beta_k$$

So, the total credit is

$$B_{\text{tot}}^{Op} = B_j \cdot B_k \sum_z \varepsilon[z] = B_j \cdot B_k$$

The confidence then becomes:

$$Conf = \frac{B_{\text{tot}}^{Op}}{B_{\text{ref}}^2} = Conf_j \cdot Conf_k$$

The confidence is multiplicative, as it usually is.

This definition implicitly assumes that the distributions are independent. If it is not the case, conditional distributions must be used instead:

$$\varepsilon[z] = \frac{1}{B_j \cdot B_k} \sum_{Op(j,k)=z} \beta_j \cdot (\beta_k | \beta_j)$$

This, however, would make the operators no longer truth-functional, since their evaluation would not only require the truth distributions of the operands, but also the information on the mutual interactions.

**Imprecise Truth Distributions** The upper and lower bounds of a truth distribution are combined in a slightly more complicated way.

$$\max / \min \quad \varepsilon[z] = \sum_{Op(j,k)=z} \varepsilon^1[j] \cdot \varepsilon^2[k]$$

s.t.

$$\begin{aligned} \sum_{j=0}^{N-1} \varepsilon^1[j] &= 1 \\ \sum_{k=0}^{N-1} \varepsilon^1[k] &= 1 \\ \forall j : \varepsilon_L^1[j] &\leq \varepsilon^1[j] \leq \varepsilon_U^1[j] \\ \forall k : \varepsilon_L^2[k] &\leq \varepsilon^2[k] \leq \varepsilon_U^2[k] \end{aligned}$$

Unfortunately, since both distributions are imprecise there is no direct solution to the problem. If either were precise, however, Dubois and Prade's Theorem (see Appendix) could be applied by setting:

$$\varepsilon[z] = \sum_{j=0}^{N-1} \varepsilon^1[j] \cdot \left( \sum_{Op(j,k)=z} \varepsilon^2[k] \right) = \sum_{j=0}^{N-1} \varepsilon^1[j] \cdot c_j$$

Thus, applying the procedure for each  $z$ , one gets the results  $\{\varepsilon_L[z], \varepsilon_{min}^1\}$  and  $\{\varepsilon_U[z], \varepsilon_{max}^1\}$ . In [?] this computation is used in an iterative procedure which minimizes (resp. maximizes)  $\varepsilon[z]$  by considering  $\varepsilon^1$  and  $\varepsilon^2$  alternately constant and precise: at every step, the resulting minimizing (maximizing) distribution updates the previous value until stability is reached.

## 4.2.2 Inference mechanisms

The operators defined so far would be useful in a propositional logic context since they but aggregate the truth values of their operands. In order to exploit the full expressive capabilities of a full FOL, it is necessary to define some inference rules to project information over new, derived predicates and their arguments.

### Induction (Learning)

Induction is the process of inferring a degree of generality for a property seen in particular cases. The learning-by-observation imprecise model defined for a single predicate is immediately applicable to universe domains where a generality relation  $\preceq$  has been defined.

**Generality** Given a predicate  $P$ , its most general argument tuple is a vector of  $n$  free independent variables. Arguments  $\mathbf{X}$ , then, can be constrained by value or by reference, thus defining a set  $K_{\mathbf{X}}$  of constraints.

For example, given  $P(X, a, X, Y)$  one gets  $K_{\mathbf{X}} = \{P[1] = a, P[0] = P[2]\}$ . Obviously, an unconstrained predicate is more general than a constrained one, so a possible definition for  $\preceq$  is:

$$X \preceq Y \Leftrightarrow K_X \subseteq K_Y$$

Hence,  $P(a, b) \preceq P(a, Y)$ ,  $P(a, X) \preceq P(X, Y)$ ,  $P(a, X) \neg \preceq P(X, X)$ ,  $P(a, a) \preceq P(X, X)$ ,  $\dots$

In case of functional values, the definition is applied recursively.  $\preceq$  is a partial order, since not all tuples are comparable.

**Credit** The belief distribution for a general predicate  $P(\mathbf{X})$  can be built from the distributions assessed for its more specific cases:

$$\int_{\mathbf{X}} P(\mathbf{X}) = \int_{\mathbf{x} \preceq X} P(\mathbf{x}) \cdot w(\mathbf{x}) d\mathbf{x}$$

The mass assignment  $w : \mathbf{X} \rightarrow [0, 1]$  is arbitrary and can be defined as stated in chapter 3. For example, consider the predicate  $P$  defined over  $C \times C$ ,  $C = \{a, b, c\}$ . A possible mass assignment is:  $P(a, X) = 0.3$ ,  $P(b, X) = 0.3$ ,  $P(c, X) = 0.3$ ,  $P(X, X) = 0.1$  provided that the distributions for the four predicates are available (or recursively computed). The total confidence, then, is a weighted linear combination of the individual confidences.

Unless the mass assignment makes one or more of the components trascurable, it can be seen that:

$$\min_{\mathbf{x} \preceq \mathbf{X}} \{\tau_{\mathbf{x}}\} \leq \varepsilon \int_{\mathbf{X}: P(\mathbf{X})} \leq 1 - \min_{\mathbf{x} \preceq \mathbf{X}} \{\varphi_{\mathbf{x}}\}$$

Generality need not be total. The induction quantifier can be used to define a full belief function over subsets  $\mathbf{Y} \subseteq \mathbf{X}$ .

$$\int_{\mathbf{Y}} P(\mathbf{X}) = \int_{\mathbf{Y} \preceq X} P(\mathbf{Y}) \cdot w(\mathbf{Y}) d\mathbf{Y}$$

For limited subsets, e.g.  $\mathbf{Y} = \{a, b\}$  notations like  $P(a||b)$  or  $P(a \cup b)$  could be used, remembering that  $P(a||b) \neq P(a) + P(b)$ . The former, in fact, describes P applied to any one of a,b, no matter which one. The latter, instead, evaluates P for each individual, then aggregates the information depending on the definition of the disjunction +.

**Specialization** The safe dual of induction, specialization allows to define

$$\varepsilon_P(\mathbf{x}) = \varepsilon_P(\mathbf{X}) \Leftrightarrow \mathbf{x} \preceq \mathbf{X}$$

**Other Quantifiers** The inductive quantifier is different from both the classic quantifiers,  $\forall$  and  $\exists$ . These, however, can be easily defined:

$$\forall \mathbf{X} : P(\mathbf{X}) = \bigwedge_{\mathbf{x} \preceq X} P(\mathbf{x})$$

$$\exists \mathbf{X} : P(\mathbf{X}) = \bigvee_{\mathbf{x} \preceq X} P(\mathbf{x})$$

The bounds show the differences:

$$\min_{\mathbf{x} \preceq \mathbf{X}} \{\tau_{\mathbf{x}}\} \leq \varepsilon_{\forall \mathbf{X} : P(\mathbf{X})} \leq 1 - \max_{\mathbf{x} \preceq \mathbf{X}} \{\varphi_{\mathbf{x}}\}$$

$$\max_{\mathbf{x} \preceq \mathbf{X}} \{\tau_{\mathbf{x}}\} \leq \varepsilon_{\exists \mathbf{X} : P(\mathbf{X})} \leq 1 - \min_{\mathbf{x} \preceq \mathbf{X}} \{\varphi_{\mathbf{x}}\}$$

**Learning Implications : Gradual Rules** Generalization by induction is instrumental in learning correlations that can become useful in deduction and other inference mechanisms. Expressing correlation under uncertainty is known in literature as “gradual rules”, a name encompassing different semantics:

- Co-occurrence rules : *When A, also B.*  
This is usually modeled by a logical conjunction and aims at evaluating the conditional probability  $p(B|A)$ . The formula  $A * B$ , however, is more general.
- Graduality Rules : *The more A, the more B.*  
This concept models a truth degree flow, more or less clamped, from A to B. This is the exact semantics of the uncertain implications, so one has to learn  $\int X, Y : \rightarrow (P(X), C(Y))$ .
- Certainty Rules : *The more A, the more certain B.*  
This is a trans-level rule since truth degree (level I) is used to increase belief (level II). This will be the object of further studies.

In learning gradual implications, one must try to avoid the dangerous side-effect known as *Ex Falso Quodlibet*: given an implication  $P(X) \rightarrow C(Y)$ , a false premise makes the operator true, regardless of the conclusion. Hence, the first time the premise becomes true, the conclusion might be assumed to be true as well. This is obviously an undesired behaviour.

In fact, it is common to partition the set of examples according to the fuzzy concept of relevance. So, for each couple  $\langle P, C \rangle$ , membership in the positive, negative and irrelevant sets is given by a trio of functions that should be complementary, i.e.:

$$\forall \langle P, C \rangle : \mu^+(\langle P, C \rangle) + \mu^-(\langle P, C \rangle) + \mu^0(\langle P, C \rangle) = 1$$

In [?] Dubois and Prade define different criteria for fuzzy partitions. The first uses conjunction to define positive samples:

$$S^+ = P * C$$

$$S^- = \neg(P \Rightarrow C)$$

$$S^0 = \neg P$$

$\neg x$  is chosen to be  $1-x$ , so this definition respects the complementarity condition for choices of  $*$  and  $\Rightarrow$  that are not necessarily closed with respect to the logic operator families described earlier. In particular,  $\Rightarrow$  here is defined by  $\neg P + C$ , with valid couples for  $*$  and  $+$  being :  $\wedge_G$  with  $\oplus_L$ ,  $\odot_\pi$  with  $+\pi$  and  $\otimes_L$  with  $\vee_L$ .

An alternative definition of partition given in the same article is based on gradual implications:

$$\begin{aligned} S^+ &= P * (P \Rightarrow C) \\ S^- &= P * (\neg(P \Rightarrow C)) \\ S^0 &= \neg P \end{aligned}$$

In this case, choosing  $*$  =  $\odot_G$  allows complete freedom in the choice of  $\Rightarrow$  due to its distributivity.

This definition leads to an interesting interpretation in our framework. First of all, we get the predicate *Relevant*( $P(X), C(Y)$ ) with its evaluation function  $\varepsilon_R = \varepsilon_P$ . This is a degree that should not be used at the logic level I, but at belief level II since it models the exact concept of weight  $w$  used so far.

$$\int \rightarrow (P(X), C(Y)) = \int \rightarrow (P, C) \cdot \varepsilon_P$$

Consider an example with N=3 slotted distributions, with  $P[0] = p$ ,  $P[1] = \tilde{p}$ ,  $P[2] = \bar{p}$  (the same notation for C). The couples  $p_j c_k$  are mapped according to the definition of  $\rightarrow$ :

$\Rightarrow$	$\tilde{\Rightarrow}$	$\rightarrow$		
$S^-$		$S^+$		$W$
$p\bar{c}$	$p\tilde{c}$	$pc$	$\Rightarrow$	$p$
	$\tilde{p}\bar{c}$	$\tilde{p}(\tilde{c} + c)$	$\Rightarrow$	$\tilde{p}$
		$\bar{p}(\bar{c} + \tilde{c} + c)$	$\Rightarrow$	$\bar{p}$
		$S^0$		$\downarrow$
				$ p $

The map is triangular, with the three vertexes being  $S^+$ ,  $S^0$  and  $S^-$ . For every sample couple  $\langle P, C \rangle$ , the rows of the table are combined according to the weights  $W$  to obtain the sample distribution for  $\rightarrow$ . The sum of all distributions over the sample set becomes the general distribution for the implication  $\int X, Y : P(X) \rightarrow C(Y)$

Furthermore, the total cumulated mass is proportional to  $|P|$ , i.e. the (fuzzy) number of times the Premise is true. The measure

$$\frac{|P \cdot \rightarrow [j]|}{|P|}$$

is proportional to the mass allocated to each one of the N slots and is a fuzzy extension of the concept of confidence defined by Dubois and Prade for binary implications:

$$conf(P \rightarrow C) = \frac{supp(P \rightarrow C)}{supp(P)} = \frac{|S^+|}{|S^+| + |S^-|}$$

### Deduction (Modus Ponens)

In most rule-based systems rules are expressed as general implications. This allows the application of the well-known Modus Ponens which, given a Premise and a compatible Implication, entails a Conclusion. Usually, rules are long-term facts stating a correlation between predicates, while the actual premise is typically a short term fact, coming from a working memory. In many-valued predicate logics, MP has the form:

$$\frac{P(\mathbf{x})_p, P(\mathbf{X}) \rightarrow_i C(\mathbf{Y})}{C(\mathbf{y})_c}$$

Notice that the formula is quite general since operands may be considered predicates, so both P and C may be the roots of full syntactic trees.

In order to be applicable, a predicate must match with the premise of an implication: i.e. their argument patterns must unify under some substitution  $\sigma$  (In many cases  $x \preceq X$ ). The same substitution, applied to the arguments Y of the conclusion, produces the actual output  $y = \sigma(Y)$ . Here, however, we are more interested in the truth degree composition rule

$$c = \rho(p, i)$$

**Composition** In order to obtain information about the conclusion, the implication's conjugate T-Norm must be applied to the premise and implication's truth values. However, given the logic equivalence  $P * (P \Rightarrow C) \equiv P \wedge C$

$$\left. \begin{array}{l} P \otimes_L \rightarrow_L \\ P \odot_\pi \rightarrow_\pi \\ P \wedge_G \rightarrow_G \end{array} \right\} = \min\{p, c\} \leq c$$

Hence, MP does not entail  $\varepsilon_C$  directly, but rather  $\tau_C$  ( $\varphi_C = 0$ ). If imprecise distributions are combined, the result is a mass assignment with the sets  $\{\tau[j], 1\}$  as focal elements.

**Projection** The mass function must be projected to obtain a bayesian distribution. The alternatives proposed in chapter 1 may be applied here.

The pignistic strategy yields:

$$\varepsilon[k] = \sum_{j \leq k} \tau[j] \cdot \frac{1}{N - j}$$

The boundary intervals  $\varepsilon_L[k]$  and  $\varepsilon_U[k]$  can be computed by applying directly Theorem ?? to the imprecise distribution  $\tau$  with the coefficient vector:

$$c[j] = \begin{cases} \frac{1}{N - j} & j \leq k \\ 0 & j > k \end{cases}$$

The vector  $c$  is ordered but for a rotation on index  $k$ .

The intersection projection may be used as well:

$$\varepsilon[k] = \begin{cases} k \neq N-1 : & \frac{\sum_{k=0}^{N-2} \tau_k}{N-2} \sum_{j \leq k} \tau_j \\ k = N-1 : & \sum_{k=0} \tau_k(N-k) \\ & \tau[N-1] + \varepsilon[N-2] \end{cases}$$

Like before, Dubois and Prade's Theorem may be applied. If  $k \neq N-1$  the coefficients in  $c$  are constant up to index  $k$  and 0 afterwards, so a simple rotation makes the vector ordered. When  $k = N-1$  the coefficients are again constant but for  $c[k]$ , which is 1. The others, however, are  $\leq 1$  so the vector is ordered:

$$\frac{\sum_{k=0}^{N-2} \tau_k}{N-2} = \frac{\sum_{k=0}^{N-1} \tau_k - \tau_{N-1}}{N-1} = \frac{\sum_{k=0}^{N-1} \tau_k - \tau_{N-1}}{\sum_{k=0}^{N-1} \tau_k - \tau_{N-1} + \sum_{k=0}^{N-1} \tau_k(N-k-1)}$$

**Combination** In general, many rules may entail the same conclusion  $C$ : moreover, a prior distribution over  $C$  may be available as a fact. So, the distributions must be aggregated in some way. The most natural choice is Dempster-Shafer's combination rule, a specialization of the combination rule with set intersection as operation. Its general formulation is:

$$m_X = \frac{1}{\sum_{A \cap B = \emptyset} m_A^1 \cdot m_B^2} \sum_{A \cap B = X} m_A^1 \cdot m_B^2$$

The mass assignment is bayesianized by discarding  $m(\emptyset)$  and then normalizing the values so that they sum to 1. Working with truth value distributions, the formula becomes:

$$\varepsilon[k] = \frac{1}{\sum_{k=0}^{N-1} \varepsilon^1[k] \cdot \varepsilon^2[k]} \varepsilon^1[k] \cdot \varepsilon^2[k]$$

Since there is no fixed limit to the number of combinations, Confidence should be additive:  $Conf = C^1 + C^2 - C^1 C^2$ .

The computation of the imprecise bounds may be optimized. First one computes the direct bounds  $\lambda[k] = \varepsilon_L^1[k] \cdot \varepsilon_L^2[k]$  and  $\mu[k] = \varepsilon_U^1[k] \cdot \varepsilon_U^2[k]$ . Then, the bounds  $\lambda(\emptyset)$  and  $\mu(\emptyset)$  are computed by applying Dubois and Prade's theorem to maximize and minimize  $\sum_{k=0}^{N-1} \varepsilon^1[k] \cdot \varepsilon^2[k]$ . Finally, the normalized bounds

are computed by

$$\varepsilon_L[k] = \frac{\lambda[k]}{1 - \max\{\lambda(\emptyset), 1 - \lambda[k] - \sum_{j \neq k} \mu[k]\}}$$

$$\varepsilon_U[k] = \frac{\mu[k]}{1 - \min\{\mu(\emptyset), 1 - \mu[k] - \sum_{j \neq k} \lambda[k]\}}$$

Dempster-Shafer combination is intersective. It is easy to see that the truth interval becomes:

$$\max\{\tau_j\} \leq \varepsilon_{DS} \leq 1 - \max\{\varphi_j\}$$

**Discounting** The Dempster-Shafer combination does not take confidence into account when evaluating bounds: a low-confidence distribution should not alter much a higher confidence one. So, it is common practice to discount a distribution before combining it.

Linear discounting by a factor  $\alpha = Conf$  shifts the values towards the uniform  $1/N$ :

$$\varepsilon[k]^d = \alpha\varepsilon[k] + \frac{1 - \alpha}{N}$$

The missing mass,  $1 - \alpha$ , is assigned to the universe set and redistributed pig-nistically. So, the bounds become:

$$\varepsilon_L[k]^d = \alpha\varepsilon_L[k]$$

$$\varepsilon_U[k]^d = \alpha\varepsilon_U[k] + 1 - \alpha$$

Discounting may be executed as a stand-alone procedure. In that case, one can set the new confidence to  $\frac{\alpha B_{tot} + N(1 - \alpha)}{B_{ref}}$

**Fuzzy Modus Ponens** Fuzzy logic uses a slightly different form of Modus Ponens. Choice of operators apart (typically, Gouguen's operators are used), the applied rule is:

$$\frac{P'(\mathbf{x})_p, P(\mathbf{X}) \rightarrow_i C(\mathbf{Y})}{C(\mathbf{y})_c}$$

where  $P'$  is not necessarily the same predicate as  $P$  but only has to belong to the same linguistic variable. In literature, several combination rules exist.

The simples applies standard MP directly, ignoring the difference in predicates:

$$C'(\mathbf{y}) \equiv \exists \mathbf{x}(\mathbf{y}) : P'(\mathbf{x}) * (P(\mathbf{X}) \rightarrow C(\mathbf{Y}))$$

From following chain of tautologies (unconstrained args are omitted for brevity)

$C \rightarrow (P \rightarrow C)$	Logic Tautology
$(P' * C) \rightarrow (P' * (P \rightarrow C))$	Monotony of *
$\exists x : (P'(x) * C) \rightarrow \exists x : (P'(x) * (P \rightarrow C))$	$\forall \rightarrow \exists$
$\exists x : (P'(x) * C) \rightarrow C'(y)$	Definition of FMP
$(\exists x : P'(x)) * C \rightarrow C'(y)$	Scope
$(\exists x : P'(x)) \rightarrow (C \rightarrow C'(y))$	Logic Tautology

one sees that either the formula  $(\exists x : P'(x))$  is true (i.e. evals to 1), so  $C'(y)$  is a tight evaluation for  $C(y)$ , or  $\rightarrow (C(y), C'(y)) < 1$ , meaning that  $C'(y) \downarrow C(y)$ .

A common fuzzy alternative is the conjunctive rule:

$$C'(\mathbf{y}) \equiv \exists \mathbf{x}(\mathbf{y}) : P'(\mathbf{x}) * P(\mathbf{X}) * C(\mathbf{Y})$$

Here, it is immediate to see that the output  $C'(y) \leq C(y)$  since a T-Norm can't but decrease the truth value of any one of its operands. It is also evident that this rule can be applied only when  $P'$  and  $P$  are not disjunct.

A third option is:

$$C'(\mathbf{y}) \equiv (P'(\mathbf{x}) \rightarrow P(\mathbf{X})) \rightarrow C(\mathbf{Y})$$

The tautology

$$[P'(\mathbf{x}) * (P(\mathbf{X}) \rightarrow C(\mathbf{Y}))] \rightarrow [(P'(\mathbf{x}) \rightarrow P(\mathbf{X})) \rightarrow C(\mathbf{Y})]$$

shows that the bound so obtained is implied by the one computed via basic FMP, which is not less tight. Moreover, this formula is equivalent to:

$$[P'(\mathbf{x}) * (P'(\mathbf{x}) \rightarrow P(\mathbf{X})) * (P(\mathbf{X}) \rightarrow C(\mathbf{Y}))] \rightarrow C(\mathbf{Y})$$

Considering that usually  $(P(\mathbf{X}) \rightarrow C(\mathbf{Y}))$  is a rule and thus is assumed to be true, the antecedent of the formula above evaluates to  $\min\{P'(x), P(x)\}$ . This is exactly the standard FMP with  $* = \wedge$  that is commonly applied in literature. This relation shows that a special implementation of FMP is not necessary as long as the gradual implications  $P' \rightarrow P$  are provided as part of the theory.

**Closed World Assumption** In any case, the output of FMP (any version) should be considered a lower bound for the truth degree of the sought conclusion, unless  $\forall y : \exists x : P(x)_1$ .

If the coverage condition is not guaranteed, the following situation may present itself:

$$\exists y_0 : (\exists x : P(x))_0$$

i.e. the best premise evaluates to 0 (complete ignorance). This yields naturally  $C(y_0)_0$ , so  $C(y_0)$  is considered false, which may not be the case. Considering, instead,  $\tau_{C(y_0)} = 0$  models correctly the state of complete ignorance (Open World Assumption).

### Abduction (Modus Tollens) and other rules

Modus Tollens is the act of entailing a possible premise from a conclusion and an implication.

$$\frac{C(\mathbf{y})_c, P(\mathbf{X}) \rightarrow_i C(\mathbf{Y})}{P(\mathbf{x})_p}$$

The underlying logic tautology is:

$$P(\mathbf{x}) \rightarrow_1 [(P(\mathbf{X}) \rightarrow C(\mathbf{Y})) \rightarrow C(\mathbf{y})] \equiv P \vee C$$

While theoretically unsafe, it can be used following the very algorithm defined for MP, with only two exceptions:

- The composition operator is  $\rightarrow$  instead of  $*$
- Being an upper bound, the composition returns a distribution for  $\varphi$  and not for  $\tau$ .

This formula can be set in a larger framework: (args are omitted since we are interested in truth values only)

$$\begin{array}{ccccccc} \neg C * \neg(P \rightarrow C) & \rightarrow_1 & P & \rightarrow_1 & (P \rightarrow C) \rightarrow C \\ P * (P \rightarrow C) & \rightarrow_1 & C & \rightarrow_1 & C \rightarrow (P \rightarrow C) \end{array}$$

The bounds are consistent, since they evaluate to:

$$\begin{array}{l} p - 2c = \bar{c} * \bar{i} \leq p \leq \bar{i} + c = \max\{p, c\} \\ \min\{p, c\} = p * i \leq c \leq \bar{c} + i = c + 2\bar{p} \end{array}$$

These formulas are interesting because they rely only on premise, implication and conclusion. A different inference strategy involves the availability of more implications.

Modus Ponens Upper Bound.  $\neg(P * (P \rightarrow \neg C))$  vs  $P \rightarrow (P \rightarrow B)$ : The first, if available, is always preferable.

Modus Tollens Lower Bound.  $C * (C \rightarrow P)$  vs  $\neg C * \neg(P \rightarrow C)$ : The first option yields tighter bounds when  $p \leq 3c$

Modus Tollens Upper Bound.  $(P \rightarrow C) \rightarrow C$  vs  $\neg(C * (C \rightarrow \neg P))$ : Opt for the first when  $c \leq 1/2$

In conclusion, direct deduction is almost always better in terms of information gain, but requires the evaluation of more correlations.

### 4.2.3 Appendix

#### Theorems

**Linear bounding** Given the linear problem:

$$\begin{aligned} \max / \min \quad & f = \sum_{t=0}^{N-1} c_t \cdot x_t \\ \text{s.t.} \quad & \sum_{t=0}^{N-1} x_t = 1 \\ & \forall t : a_t \leq x_t \leq b_t \\ & \forall t_1, t_2 : t_1 \leq t_2 \Leftrightarrow c_{t_1} \leq c_{t_2} \end{aligned}$$

The problem has a closed form solution given by:

$$\begin{aligned} \min_f &= \max_{k:0..N-1} \left\{ \sum_{t=0}^{k-1} c_t b_t + \left( 1 - \sum_{t=0}^{k-1} b_t + 1 - \sum_{t=k+1}^{N-1} a_t \right) c_k + \sum_{t=k+1}^{N-1} c_t a_t \right\} \\ \max_f &= \min_{k:0..N-1} \left\{ \sum_{t=0}^{k-1} c_t a_t + \left( 1 - \sum_{t=0}^{k-1} a_t + 1 - \sum_{t=k+1}^{N-1} b_t \right) c_k + \sum_{t=k+1}^{N-1} c_t b_t \right\} \end{aligned}$$

# Chapter 5

## Inferential Engines

In this chapter we describe some possible architectures of an engine capable of elaborating an uncertain logic theory contained in an uncertain knowledge base.

### 5.1 Formal Reasoning

The usual concepts of Theory, inference Rules and Proof must be extended to take uncertainty into account. In uncertain logic, well-formed Formulas are enriched by annotation with a truth degree of some sort, namely a value, an interval or an (imprecise) distribution.

In this context, we have a set of facts  $F$ , which are all well syntactically formed formulas, and a set of truth values  $L$ . A Theory is a map

$$T : F \rightarrow L$$

assigning a truth value to every formula.

Inference is the act of explicating the information contained in  $T$  by applying reasoning rules: this action, normally carried out on a semantical level, is isomorphic to an appropriate set of syntactical transformations. A Syntax  $\sigma$  is defined by a couple of sets  $\langle A, R \rangle$ :

- Set of Axioms  $A$ , facts known a priori.
- Set of Rules  $R$ . A Rule is a couple of functions  $\langle \rho, \lambda \rangle$ :
  - $\rho : F^n \rightarrow F$  operates on syntax (predicates and args)
  - $\lambda : L^n \rightarrow L$  operates on truth values

All the inference rules defined in chapter 4 (Deduction, Abduction, Induction, ...) fall in this category. Rules are used to define the concept of Proof: a formula  $\phi$  is derivable by Theory  $T$  in degree  $\varepsilon$  if and only if either:

- $\phi_\varepsilon \in A \cup T$

- $\exists r \in R : \phi_\varepsilon = r(\phi_{\varepsilon_j}^j)$  and formulas  $\phi^j$  have been proved in degree  $\varepsilon_j$ .

According to this recursive, goal-driven definition, a formula  $\phi$  is derived by derivation of a sequence of intermediate formulas: this sequence is a Proof for  $\phi$ . Syntactical derivability is written:

$$T \vdash_\sigma \phi_\varepsilon$$

Given a Theory  $T$ , there might exist several Proofs for a formula: the syntactic closure of a Theory over a formula:

$$\sigma(\phi) = \sup_\varepsilon \{T \vdash_\sigma \phi_\varepsilon\}$$

assigns to a formula the tightest-bounded truth value that can be derived by proofs within  $T$ .

**Correctness and Completeness** Inference is Correct and Complete if the truth value of a formula is computed by the tightest Proof using sound Rules.

## 5.2 Engine architectures

The necessity of computing the syntactic closure of a theory has a profound impact on the structure of and engine elaborating uncertain predicates. In fact, reasoning under uncertainty gives many degrees of freedom in engine design. The rest of this section will address some of the properties an engine could have in order to gain expressiveness.

**Transparency** First of all, the engine should apply the Rules transforming both the arguments and the truth values natively: for example, the truth value of a predicate obtained by modus ponens should be computed automatically from the truth values of the premise and implication without any explicit, additional instruction.

In particular, the merging of truth values obtained by different Proofs should be transparent to the user.

**Parallelism** The necessity of finding different - namely, all - the possible Proofs of a formula might increase the computation cost dramatically, especially if the candidate Proofs are many. This cost may be levied by introducing parallelism in the engine. Technically, there are two types of parallelism:

- Or-Parallelism ( $\omega_{//}$ ) : Given a single Goal  $G$ ,  $\omega_{//}$  is the act of searching all different Proofs for  $G$  by applying all the rules entailing  $G$ .
- And-Parallelism ( $\alpha_{//}$ ) : Given a set of joint Goals,  $\alpha_{//}$  is the act of proving the individual Goals in parallel.  $\alpha_{//}$  may be limited by interactions between the Goals (e.g. shared variables).

In a backward resolution scheme,  $\omega_{//}$  and  $\alpha_{//}$  are always alternated as each formula is resolved by trying all applicable rules ( $\omega_{//}$ ), whose operands must be joined ( $\alpha_{//}$ ) after having been evaluated individually ( $\omega_{//}$ ).

**Heuristic cuts** Even with multiple computational units, the cost of exploiting full  $\omega$ -parallelism may be excessive. Sometimes, one or more alternatives should be pruned to reduce the computational effort, even if this could hamper correctness (in particular, one would get looser bounds, but not inconsistent ones). Applicable criteria include:

- Certainty-Cut: Stop if  $\chi_G > \chi_0$
- Information-Cut: Stop if  $H_G < H_0$
- Confidence-Cut: Stop if  $Conf_G > Conf_0$

**Queries** In classic logic, one is typically interested in knowing whether a formula is true or not:  $P(x)?$ . Under uncertainty, this translates into asking up to which degree a formula is true. When a non-ground goal is given, such as  $P(X)?$ , different questions could be posed, yielding different answers:

- $?P(X)$  : *Given any  $x \in X$ , does  $P$  hold?*
- $? \forall P(X)$  : *In which degree does  $P$  hold for all  $x$ ?*
- $? \exists P(X)$  : *In which degree does  $P$  hold for one  $x$ ?*
- $P(?X)$  : *For which  $x$   $P$  holds the most?*

All queries should return a truth value: the first three ask for a global property of  $P$  over its domain and the coincide with the aggregation quantifiers defined in chapter 2. The fourth, instead, aims at finding one argument  $x$  making  $P$  true or at least maximizing the degree of  $P(x)$ , in addition to returning the associated truth degree.

In any case, failure by lack of provability can return either False (Closed World Assumption) or Unknown (Open World Assumption). Unless truth degrees are implemented using single values, use of the OWA is encouraged due to the ease of modeling Unknown using intervals or more refined structures.

**Propagation** The two common strategies of information propagation are called forward chaining and backward chaining.

The former is based on a push model, in which predicates obtained by inference rules are asserted and then re-used to activate new rules, until the goal is entailed almost as a side effect.

The latter, instead, cannot be used but in a pull-oriented, two-phase form. In the first phase, goals are propagated backward properly, each rule matching a goal generating new goals as it has operands. When dealing with uncertain

predicates, a second, forward phase is nevertheless necessary to recollect and combine the obtained truth values.

This makes the use of a hybrid strategy very easy to implement since forward chaining must be supported in any case.

**Complex Clauses** Considering operators a special class of predicates allows the definition of complex rules. In particular, the truth value of any operator may be entailed by Proof if the adequate rules are present within the Theory.

Entailing the truth value of an operator has an interesting side effect: it can be propagated normally in a forward manner (if the case, as a consequence of a backward query) - call it Outer Chaining - but it can also be used to update the truth values of its operands - which may be defined Inner Chaining. Inner chaining should never change the truth values of the operands which generated it (stable retroaction).

It should be remembered that implications are operators: even if they are often used to define rules, implications included in a Theory are but facts which true inference Rules typically can exploit. Modifying the truth value of an implication is the basis for the realization of a dynamic logic program.

### 5.2.1 Prolog resolution

Though Prolog has universally been used for predicate logic programming, the standard resolution strategy used in Prolog engines and introduced in chapter 2 is not suitable for use with uncertain logic. The main issues are:

- **Classic logic:** there is no native support for truth degrees.
- **Existential Queries:**  $? - P(X)$  is interpreted as “Does it exist an  $x \preceq X$  making  $P$  true?”
- **Return on Success:** Proof search returns upon finding the first success branch since there is no notion of partial success.
- **Closed World Assumption:** Failure always returns false.

Prolog, however, can and has been used to program an uncertain logic meta-interpreter with limited features. This tool has been used mainly for testing purposes, in order to verify the operators’ definitions and the generality relation. The features of such a toy engine, with respect to the criteria defined in the former section, are:

- **Explicit Truth Intervals:** Truth degrees must be added explicitly as arguments and combined adequately. Given the declarative nature of Prolog and the strict algorithmic nature of the combination rules for complex truth structures, interval representation has been chosen. Even if truth distributions could be used and operated on, imprecise distributions would be hardly tractable.

- **Essential Queries:** The engine answers queries of the  $?P(X)$  type only.
- **$\omega$ -Parallelism:** Given a Goal, all more general facts (according to the  $\preceq$  relation) are collected, evaluated and merged. Moreover, all implications entailing a predicate more general than the goal are evaluated, specialized and merged as well.

No cut is used, so all possible Proofs are checked.

- **Backward Chaining:** Given the underlying resolution strategy, two-phase backward chaining (actually implemented by recursion) is the most natural choice.
- **Complex Clauses:** Operators are defined by key predicates, implication being one of them. Thus, Prolog facts are used to define initial values for predicates and operators. Prolog clauses are used to implement inferential rules explicitly: at the moment, only Modus Ponens is supported in the form

$$\begin{aligned} & \text{ImPLY}(P, C). \\ C : -P, \text{ImPLY}(P, C). \end{aligned}$$

Inner Chaining is not implemented.

- **Open World Assumption:** Upon failure, the Unknown interval  $[0,1]$  is returned instead.

### 5.2.2 RETE-U

The parallel nature of the inferential problem makes distributed algorithms, such as RETE (see chapter 2) more appealing. Here we will define an immediate generalization capable of handling uncertainty, from network construction to information propagation. This architecture has been tested in a prototype implementation using tuple centres.

**Logic Program** A logic program is always a set of truth-annotated Facts  $\{F_j\}$ . Facts may be primitive Predicates or complex Operators: in the latter case, the associated truth degree is mapped on the operator itself for the given argument pattern. We remark here that implications are predicates.

**Abstract Network** The program is compiled into a network according to the following procedure:

- Every Fact  $F_j$  is converted into the Abstract Syntax Tree mapping its structure, with non-predicate arguments as leaves.
- Every Abstract Node of the tree (leaves excluded) represents a Predicate  $P$  with arity  $n_P$ . Abstract Nodes have  $n$  input sites, called  $D_P[j]$  and one output site called  $A_P$ .
- The AST forest is transformed into an Abstract Syntax Graph by finding and merging equivalent subtrees.

**Nodes** Every Node mapping a predicate  $P(\mathbf{X})$  is a RETE subnetwork, composed by a standard  $\alpha$ -net filtering the  $n$  arguments through the inlinks  $D_P[j]$  and a  $\beta$ -net performing the joins. The terminal node of the network  $N_P$ , stores the matching patterns  $\mathbf{x} \preceq \mathbf{X}$  and contains the membership function  $\mu_P(\mathbf{x})$ , which may be either truth-functional (for operators) or domain-evaluating (for basic predicates).  $N_P$  is connected to  $A_P$  directly.

**Inferential Rule Nodes** These Nodes are generated automatically from the existing Nodes. For example, a Modus Ponens Inferential Node is created for every implication node: a MP Node has two input Nodes, namely the Implication and the Premise which are matched within the Output node of the Node.

**Reactive behaviour** Output nodes  $N_P$  are reactive and may be programmed, typically by plugging observers into them. They react to the insertion or the retraction of a tuple from the relation they hold: this behaviour may be used for several purposes:

- Forward Outer Chaining: Forward links between Nodes are used to propagate information from  $A_P$  to  $D_Q[j]$ , where they will traverse the alpha network.
- Inference Rule Execution: Since the output pattern is known a priori, the result can bypass the  $\alpha$ -net of the conclusion predicate and be inserted directly in the appropriate output node(s). For example, Modus Ponens Nodes generate conclusions upon finding a Premise/Implication match.
- Inner Chaining: When a tuple is inserted directly in the output relation of an operator, it may be used to refine the information on the operands. This generates distributions that are directly inserted in the operands' output nodes.
- Merging: Whenever an existing tuple is asserted again (typically by insertion), the corresponding truth degrees are merged according to the appropriate strategy (e.g. intersection or imprecise DS).
- Side Effects: The presence of a certain tuple may trigger non-logic events, such as writing some data on a file.
- Filtering: Filters may be imposed on the relation: for example, a tuple with unknown truth degree or low confidence might be discarded.

**Aggregation Nodes** An Node's output node contains tuples  $\mathbf{x}$  that are more specific than the pattern  $\mathbf{X}$  that generated it. Quantifiers may be implemented by (output) nodes observing a relation.

## Features

The architecture is quite flexible and configurable: in fact, it supports the following features and it might be further extended:

- **Truth Intervals:** Any kind of truth value may be used. Being more general (and more expensive), imprecise distributions are recommended for greater expressiveness, which typically means fewer rules.
- **Queries:** By default, queries are essential. Aggregation queries can be answered by placing an observer node over an output node.
- **Parallelism:** . Actual parallelism is implementation-dependent (at the very limit, one thread is assigned to each Node).
- **Chaining:** Nodes can be configured separately, specifying whether a new available predicate should be pushed or pulled from each output relation node.
- **Complex Clauses:** Full support is provided. Output nodes reactions may be used to implement either Inner or Outer Chaining - or both.
- **Open World Assumption:** OWA is supported by returning the null distribution (or the unitary interval) upon failure.