Fuzzy Sets and Systems 59 (1993) 247–257
North-Holland

Learning rules for a fuzzy inference model*

Luis M. de Campos and Serafín Moral
Departamento de Ciencias de la Computación e I.A., Universidad de Granada, 18071-Granada, Spain

Received September 1992
Revised January 1993

Abstract: The problem of learning rules for a fuzzy inference model directly from empirical observations, without resorting to assessments from experts is considered. We develop a method that builds uncertain rules from a set of examples. These rules match the following pattern: If X is A then Y is B is $[\alpha, \beta]$, where A and B are fuzzy sets representing fuzzy restrictions on the variables X and Y; α and β are real numbers expressing lower and upper degrees of certainty in the truth of the rule. The method is based on the minimization of a distance measure between the real output associated to a given input and the output predicted by the inference model using a parameterized version of the same rule to be learnt. Our approach is computationally efficient in running time as well as in storage requirements. Moreover, it can be used in both training (batch-processing) and adaptation (iterative-processing) modes of learning.

Keywords: Inductive learning; approximate reasoning; fuzzy inference model; upper and lower probabilities.

1. Introduction

Fuzzy rule-based systems have proved to be useful tools to perform reasoning tasks. The usual way of building the rules for these systems is by eliciting opinions from the experts. However, methods for automatic learning of the rules directly from raw data represent an interesting alternative. In this paper one of these methods is proposed (different models of automatic learning are considered in the literature, for example, see [2, 7, 8 and 9]).

The specific problem we are going to study could be stated as follows: Consider two sets of labels $\{A_i, i = 1, \ldots, n\}$ and $\{B_j, j = 1, \ldots, m\}$; these labels represent soft restrictions on two variables X and Y, and they are fuzzy sets defined on two domains U and V. Consider also a set of examples E, whose elements are pairs (x, y), x and y being instances of the two variables X and Y respectively (these examples could be crisp values, sets of values or more generally, normalized fuzzy sets).

We want to obtain a set of rules

If X is A_i then Y is B_j is $[\alpha_{ij}, \beta_{ij}]$

describing the relationships between the A_i's and B_j's displayed by E; the weights α_{ij}, β_{ij} are real numbers that represent degrees of certainty in the truth of the rule. More precisely, we seek to learn the weights α_{ij}, β_{ij} from E (we are not trying to infer the labels in the rule but only the weights). These rules can be used later through an inference model described in [4].

The basic idea will be to use the inference model itself to learn the weights, by means of the following process: first, to compare for each example the real output with that predicted by the rule for the corresponding input, and second, to estimate the weights that minimize these differences.

The paper is arranged in 5 sections. The kind of rules we will consider, together with the inference

Correspondence to: Dr. L.M. de Campos, Universidad de Granada, Dept. de Ciencias de la Comp. e I.A., 18071 Granada, Spain.

* This work has been supported by the European Economic Community under Project DRUMS (Esprit b.r.a. 3085).

model that uses them, and the underlying uncertainty propagation model that constitutes the basis for the inference process, are described in Section 2. In Section 3 we develop the learning method: we start with the simple case in which we have only one example, and next we consider two different extensions to the general case, that we call average and aggregation approaches. Section 4 is devoted to studying the performance of our method in the crisp case, where no fuzziness is present. The crisp case is also useful to show what kind of conditioning underlies the formulas obtained for the certainty degrees in the rules. Finally, Section 5 ends the paper with some comments on the performance and flexibility of our approach, also pointing out some open questions.

2. The rules and the inference model

As we said previously, the rules we are trying to infer fit the following pattern:

If X is A then Y is B is $[\alpha, \beta]$,

where X and Y are variables taking their values on reference sets U and V respectively, A and B are fuzzy sets on U and V, and the weights α and β are real numbers satisfying the inequalities $0 \leq \alpha \leq \beta \leq 1$.

The meaning of this rule is based on the following ideas:

- The rule defines a relation between the sets $U_A = \{A, \neg A\}$ and $V_B = \{B, \neg B\}$ (instead of a relation between U and V, that is, the level of granularity is similar to that of the elements involved), and this relation is interpreted as a conditioning.
- The weights α, β represent the (conditional) lower and upper degrees of certainty for Y being B given that X is A (for the sake of simplicity, in the following we will refer to a generic proposition 'Z (a variable) is C (a fuzzy set)' by writing only 'C').

So, the above rule is translated into the following pairs of conditional lower and upper probability measures $(l(\cdot \mid A), u(\cdot \mid A))$ and $(l(\cdot \mid \neg A), u(\cdot \mid \neg A))$ defined on V_B:

$$
\begin{aligned}
&l(B \mid A) = \alpha, \qquad l(\neg B \mid A) = 1 - \beta,\\
&u(B \mid A) = \beta, \qquad u(\neg B \mid A) = 1 - \alpha,\\
&l(B \mid \neg A) = 0, \qquad l(\neg B \mid \neg A) = 0,\\
&u(B \mid \neg A) = 1, \qquad u(\neg B \mid \neg A) = 1.
\end{aligned}
$$

These measures represent the following pieces of information:

- If A is true, then the degree of certainty on B being also true lies between α and β, and therefore the degree of certainty on B being false lies between $1 - \beta$ and $1 - \alpha$.
- If A is false, ($\neg A$ is true) then we cannot infer anything about the truth of B: the degrees of certainty on B being true and B being false both lie between 0 and 1.

Remark: Note that the rule 'if X is A then Y is B is $[\alpha, \beta]$' is equivalent to the rule 'if X is A then Y is $\neg B$ is $[1 - \beta, 1 - \alpha]$'. So, the rule gives information about B and $\neg B$, when A is true.

This kind of rule can be used as the basic component of a fuzzy inference model working in the following way: A given input A^* is compared with the antecedent of the rule, A, and its negation $\neg A$, thus obtaining two degrees of matching, $m(A^*, A)$ and $m(A^*, \neg A)$, which may be interpreted as the values of an upper probability measure $u_x(\cdot)$ on U_A, $u_x(A) = m(A^*, A)$, $u_x(\neg A) = m(A^*, \neg A)$ (the lower probability would be obtained by duality). Next, using a propagation model, from $u_x(\cdot)$ and the conditional measures representing the rule, we get an upper measure $u_Y(\cdot)$ on V_B. We can stop here, or go further on, to combine the membership functions μ_B and $\mu_{\neg B}$ of B and $\neg B$ with u_Y, obtaining as output a single fuzzy set B^* instead of an uncertainty measure on V_B.

Now we are going to briefly give some details about the inference and the propagation models which will be needed later on (see [4] and [5] for additional details).

The matching between A^* and each value in U_A is calculated as a compatibility degree between two fuzzy sets using the Łukasiewicz t-norm (although any other t-norm could be used too). Therefore the upper measure on U_A induced by an input A^* is (see [4]):

$$u_X(A) = m(A^*, A) = \sup_r \{\max(\mu_{A^*}(r) + \mu_A(r) - 1, 0)\},$$
$$u_X(\neg A) = m(A^*, \neg A) = \sup_r \{\max(\mu_{A^*}(r) - \mu_A(r), 0)\} \tag{1}$$

where μ_A and μ_{A^*} denote the membership functions of A and A^* respectively.

It can easily be proved that the condition

$$m(A^*, A) + m(A^*, \neg A) \geq 1 \tag{2}$$

holds if we only impose the input A^* to be a normalized fuzzy set (that is, $\exists r \in U$ such that $\mu_{A^*}(r) = 1$). We will always suppose that the inputs verify this property in the rest of the paper. So, from (2) we can interpret the matching degrees as the values of an upper probability measure. More precisely, u_X defined in (1) is a plausibility measure (see [10]); the corresponding belief measure is obtained by duality: $l_X(A) = 1 - u_X(\neg A)$, $l_X(\neg A) = 1 - u_X(A)$; the basic probability assignment (b.p.a) associated to this belief-plausibility pair is $m_X(A) = 1 - m(A^*, \neg A)$, $m_X(\neg A) = 1 - m(A^*, A)$, $m_X(U_A) = m(A^*, A) + m(A^*, \neg A) - 1$.

The propagation of this upper measure, using the conditional upper measures obtained from the rule, will produce another upper measure u_Y on V_B. The propagation model is very simple for this case: to obtain $u_Y(C)$, being $C = B$ or $C = \neg B$, we only need to compute the Choquet integral (see [6]) of the function $f(A) = u(C \mid A)$, $f(\neg A) = u(C \mid \neg A)$ with respect to the measure u_X. The result is the following plausibility measure (see [4] and [5]):

$$u_Y(B) = \beta(1 - m(A^*, \neg A)) + m(A^*, \neg A), \qquad u_Y(\neg B) = 1 - \alpha(1 - m(A^*, \neg A)). \tag{3}$$

The corresponding belief measure is

$$l_Y(B) = \alpha(1 - m(A^*, \neg A)), \qquad l_Y(\neg B) = (1 - \beta)(1 - m(A^*, \neg A)),$$

and the associated b.p.a. is

$$m_Y(B) = \alpha(1 - m(A^*, \neg A)), \qquad m_Y(\neg B) = (1 - \beta)(1 - m(A^*, \neg A)),$$
$$m_Y(V_Y) = m(A^*, \neg A) + (\beta - \alpha)(1 - m(A^*, \neg A)).$$

Observe that only the matching degree between A^* and $\neg A$ is needed.

3. Learning the rules

In this section, we will develop a methodology for learning, from a set of examples, the rules necessary to perform the previous inference process.

Let us suppose that we want to estimate the weights α and β of the rule

if X is A then Y is B is $[\alpha, \beta]$. (4)

3.1. A simple case

Let us also suppose that we know only one example (x_1, y_1), which is a particular instantiation of the variables X and Y. If we apply the rule to the input x_1 then we get on V_B an upper measure u_Y^p, as in (3):

$$u_Y^p(B) = \beta(1 - m(x_1, \neg A)) + m(x_1, \neg A), \qquad u_Y^p(\neg B) = 1 - \alpha(1 - m(x_1, \neg A))$$

which depends on the labels A and B, the input x_1 and the weights α and β. This measure is considered

as the output predicted by the rule, given the input x_1. But we also have a real output y_1; starting from y_1 we can build another upper upper measure on V_B (in a similar way as we obtained u_X on U_A from x_1 through a matching process) which is

$$u_Y^r(B) = m(y_1, B), \qquad u_Y^r(\neg B) = m(y_1, \neg B).$$

So, we have on V_B a 'real' and a 'predicted' measure, u_Y^r and u_Y^p respectively. The idea is to select the weights α and β making u_Y^r and u_Y^p as close as possible. We can do that by minimizing some distance measure between u_Y^r and u_Y^p; in [3] a general procedure to define distances between fuzzy measures was proposed. Here we will use an Euclidean-based distance d (see [3]) which produces the following function f_1 when it is applied to u_Y^r and u_Y^p: $f_1(\alpha, \beta) = d(u_Y^r, u_Y^p)$, where

$$\begin{aligned} d(u_Y^r, u_Y^p) = \operatorname{sqrt}((&\beta(1 - m(x_1, \neg A)) + m(x_1, \neg A) - m(y_1, B))^2 \\ &+ (1 - \alpha(1 - m(x_1, \neg A)) - m(y_1, \neg B))^2) \end{aligned} \tag{5}$$

and the symbol sqrt(·) stands for the square root function.

Therefore, to estimate the weights α and β for rule (4), on the basis of only one example, we must solve the following non-linear optimization problem (as the square root is an increasing function for positive numbers, the symbol sqrt could be dropped without affecting the resultant optimum):

$$\begin{aligned} &\text{Find} \quad \alpha_1, \beta_1 \text{ such that } f_1(\alpha_1, \beta_1) = \operatorname{Min} f_1(\alpha, \beta) \\ &\text{s.t.} \quad 0 \leq \alpha \leq \beta \leq 1 \end{aligned} \tag{6}$$

where f_1 is defined as in (5). We obtain the solution in the following proposition.

Proposition 1. *The solution to the above problem* (6) *is*:

Case 1: *if* $m(x_1, \neg A) = 1$
then the values of α *and* β *are arbitrary in* $0 \leq \alpha \leq \beta \leq 1$. *We choose* $\alpha_1 = 0$, $\beta_1 = 1$ *by using the minimum specificity principle.*

Case 2: *if* $m(x_1, \neg A) \neq 1$ *and* $1 + m(x_1, \neg A) < m(y_1, B) + m(y_1, \neg B)$

then $$\alpha_1 = \frac{1 - m(y_1, \neg B)}{1 - m(x_1, \neg A)}, \qquad \beta_1 = \frac{m(y_1, B) - m(x_1, \neg A)}{1 - m(x_1, \neg A)}.$$

Case 3: *if* $m(x_1, \neg A) \neq 1$ *and* $1 + m(y_1, \neg B) < m(x_1, \neg A) + m(y_1, B)$
then $\alpha_1 = \beta_1 = 1$.

Case 4: *if* $m(x_1, \neg A) \neq 1$ *and* $1 + m(y_1, B) < m(x_1, \neg A) + m(y_1, \neg B)$
then $\alpha_1 = \beta_1 = 0$.

Case 5: *if* $m(x_1, \neg A) \neq 1$, $1 + m(x_1, \neg A) \geq m(y_1, B) + m(y_1, \neg B)$,
$1 + m(y_1, \neg B) \geq m(x_1, \neg A) + m(y_1, B)$, *and* $1 + m(y_1, B) \geq m(x_1, \neg A) + m(y_1, \neg B)$

then $$\alpha_1 = \beta_1 = \frac{1 - m(x_1, \neg A) + m(y_1, B) - m(y_1, \neg B)}{2(1 - m(x_1, \neg A))}.$$

Proof. For this problem the Kuhn–Tucker optimality conditions (see [1]) are the following:
There exist real numbers u_1, u_2 and u_3 such that:

$$-2(1 - m(x_1, \neg A))(1 - \alpha(1 - m(x_1, \neg A)) - m(y_1, \neg B)) + u_1 - u_2 = 0,$$
$$2(1 - m(x_1, \neg A))(\beta(1 - m(x_1, \neg A)) + m(x_1, \neg A) - m(y_1, B)) - u_1 + u_3 = 0,$$
$$u_1(\alpha - \beta) = 0,$$
$$u_2\alpha = 0,$$
$$u_3(\beta - 1) = 0,$$
$$u_1, u_2, u_3 \geq 0.$$

The Kuhn–Tucker conditions are, in our case, necessary and sufficient because the function f_1 is convex and the inequality constraints are linear functions.

So, to prove the result it suffices to check that the Kuhn–Tucker conditions are verified for the values of α_1 and β_1 above, using the following Lagrangian multipliers u_i for each case:

Case 1: f_1 is a constant function, it does not depend on α and β. So, any values for α and β are possible. Therefore, we select the least specific values $\alpha = 0$ and $\beta = 1$.

Case 2: $u_1 = u_2 = u_3 = 0$.

Case 3: $u_1 = 2(1 - m(x_1, \neg A))(m(x_1, \neg A) - m(y_1, \neg B))$,
$u_2 = 0$,
$u_3 = 2(1 - m(x_1, \neg A))(m(x_1, \neg A) + m(y_1, B) - m(y_1, \neg B) - 1)$.

Case 4: $u_1 = 2(1 - m(x_1, \neg A))(m(x_1, \neg A) - m(y_1, B))$,
$u_2 = 2(1 - m(x_1, \neg A))(m(x_1, \neg A) + m(y_1, \neg B) - m(y_1, B) - 1)$,
$u_3 = 0$.

Case 5: $u_1 = (1 - m(x_1, \neg A))(1 + m(x_1, \neg A) - m(y_1, B) - m(y_1, \neg B))$,
$u_2 = u_3 = 0$. □

The following simple examples illustrate the performance of the method: Suppose that we have an example (x_1, y_1) such that:

(1) $m(x_1, \neg A) = 1$, that is, x_1 is completely compatible with the negation of the antecedent of the rule. In this case, $\alpha = 0$ and $\beta = 1$, that is, we learn nothing: we cannot infer anything about a rule $A \rightarrow B$ from an example that supports $\neg A$ at maximum degree.

(2) $m(x_1, \neg A) = 0$ (and then $m(x_1, A) = 1$). For that case, $\alpha = 1 - m(y_1, \neg B)$, $\beta = m(y_1, B)$, and the example (x_1, y_1) supports the rule exactly as much as y_1 supports B.

(3) $m(y_1, B) = 1$; in this case y_1 is completely compatible with B. So, there is no evidence against the fact that 'A implies B' and therefore $\beta = 1$. The degree of evidence supporting 'A implies B' depends on the difference between $m(x_1, \neg A)$ and $m(y_1, \neg B)$, and is $\alpha = \min[1, (1 - m(y_1, \neg B))/(1 - m(x_1, \neg A))]$; the greater $m(y_1, \neg B)$ is, the lesser α is; on the other hand, the greater $m(x_1, \neg A)$ is, the greater α is.

(4) $m(y_1, B) = m(y_1, \neg B)$. In that case $\alpha = \frac{1}{2} - \gamma$ and $\beta = \frac{1}{2} + \gamma$, where

$$\gamma = \max[(2m(y_1, B) - m(x_1, \neg A) - 1)/(2(1 - m(x_1, \neg A)), 0].$$

As the example provides the same evidence for B and $\neg B$, the weights α and β are symmetrical with respect to $\frac{1}{2}$.

(5) $m(y_1, \neg B) = 0$ (and then $m(y_1, B) = 1$). We get $\alpha = \beta = 1$ and the example strongly supports the rule.

In general, the results are very intuitive when $m(x_1, \neg A)$ is small. When $m(x_1, \neg A)$ increases (but remains different to 1), the results become unjustifiably more and more conclusive in some cases. For example, if $m(y_1, B) = 1$ and $m(y_1, \neg B) = 0$ then $\alpha = \beta = 1$. This means that we are sure that A implies B; however it is difficult to be so sure when the compatibility of x_1 with $\neg A$ is high (e.g., $m(x_1, \neg A) = 0.9$). Nevertheless, the values $\alpha = \beta = 1$ still give the best adjustment between u_Y^r and u_Y^p for that case. The problem is that even the best adjustment is very poor: this example does not correctly support the rule, A implies B, at any degree $[\alpha, \beta]$ (the example could strongly support a rule $\neg A$ implies B). Although this problem weakens when we have more than one example, we will provide a method to eliminate it in Section 3.3; this method will consist in discounting (in the sense of Shafer [10]) the conditional measure resultant from the optimization process by a factor depending on the degree of adjustment between u_Y^r and u_Y^p.

3.2. The general case

When we do not have only one example but a set $E = \{(x_k, y_k), k \in K\}$ of examples, we may adopt two different approaches:

(1) *Average approach:* Apply the previous process to each of the examples, obtaining a set of weights $\{[\alpha_k, \beta_k], k \in K\}$. These weights produce the minimum distance, $f_k(\alpha_k, \beta_k)$ between the two

upper measures being considered. Then we could choose some kind of average of the intervals $[\alpha_k, \beta_k]$, weighted by their corresponding degrees of fitness t_k as the final result.

The degree of fitness t_k for the weights $[\alpha_k, \beta_k]$ corresponding to the example (x_k, y_k) is a quantity that measures how well the example fits the antecedent of the rule, and it tries to partially avoid the problem that we commented in the previous subsection.

The value t_k could be defined as the difference between the maximum and the minimum possible distances between u_Y^r and u_Y^p, normalized to range from zero to one. As the minimum distance corresponds to the value $f_k(\alpha_k, \beta_k)$, then t_k is

$$t_k = 1 - \frac{f_k(\alpha_k, \beta_k)}{\max_{\alpha,\beta} f_k(\alpha, \beta)}.$$

So, the greater t_k is, the better the degree of fitness between the real and predicted outputs is. The rationale for this definition is the following: if t_k is small, there is little difference between the maximum and the minimum values of $f_k(\alpha, \beta)$ and therefore any possible values we give to the weights α and β do not improve the adjustment very much. In that case the example fits the rule poorly. On the contrary, if t_k is great, a judicious selection of the weights α and β make the difference between u_Y^r and u_Y^p considerably lesser than it could be if we selected different values for α and β. Therefore, the adjustment will be greatly improved.

Remark: Observe that it is very easy to calculate the value of t_k, because the maximum value of $f_k(\alpha, \beta)$ can be obtained by evaluating the function $f_k(\cdot,\cdot)$ only for the three points $(\alpha, \beta) = (0, 0)$, $(0, 1)$ and $(1, 1)$ and then selecting the maximum. The reason is that f_k is a convex function, and therefore the maximum must be achieved at an extreme point of the region $0 \leq \alpha \leq \beta \leq 1$ (see [1]).

Now we need some kind of average for the intervals $[\alpha_k, \beta_k]$. For example, using a classical average, the resultant weights for the rule (4) are

$$[\alpha, \beta] = \left[\frac{\sum_k t_k \alpha_k}{\sum_k t_k}, \frac{\sum_k t_k \beta_k}{\sum_k t_k}\right]. \tag{7}$$

(2) *Aggregation approach:* Use some aggregation function g, to summarize all the distances between the real and predicted output measures into a single value, $F(\alpha, \beta) = g(\{f_k(\alpha, \beta), k \in K\})$. Then find the values of α and β that minimize F, subject to the restrictions $0 \leq \alpha \leq \beta \leq 1$.

For example, by using the sum of squares, $\sum_k f_k^2(\alpha, \beta)$, as aggregation function, we get

$$F(\alpha, \beta) = a\beta^2 + a\alpha^2 + 2b\beta - 2c\alpha + d + e$$

where

$$\begin{aligned}
a &= \sum_k (1 - m(x_k, \neg A))^2, \\
b &= \sum_k (1 - m(x_k, \neg A))(m(x_k, \neg A) - m(y_k, B)), \\
c &= \sum_k (1 - m(x_k, \neg A))(1 - m(y_k, \neg B)), \\
d &= \sum_k (m(x_k, \neg A) - m(y_k, B))^2, \\
e &= \sum_k (1 - m(y_k, \neg B))^2
\end{aligned} \tag{8}$$

and the optimization problem to be solved is

$$\begin{aligned}
&\text{Minimize} \quad F(\alpha, \beta) \\
&\text{s.t.} \qquad 0 \leq \alpha \leq \beta \leq 1
\end{aligned} \tag{9}$$

Proposition 2. *The solution to the above problem* (9) *can be computed as follows:*

$$
\begin{aligned}
&\text{if } a = 0 \text{ then } \alpha = 0,\ \beta = 1\\
&\text{else if } b - c \geqslant 0 \text{ then } \alpha = \beta = 0\\
&\quad \text{else if } c - b - 2a \geqslant 0 \text{ then } \alpha = \beta = 1\\
&\qquad \text{else if } c + b \geqslant 0 \text{ then } \alpha = \beta = (c-b)/2a\\
&\qquad\quad \text{else } \alpha = c/a, \qquad \beta = -b/a
\end{aligned}
\tag{10}
$$

where the numbers a, b and c are defined in (8), *or in a more expanded form:*

$$
\begin{aligned}
&\text{if } \sum_k (1 - m(x_k, \neg A))^2 = 0 \text{ then } \alpha = 0,\ \beta = 1\\
&\text{else if } \sum_k (1 - m(x_k, \neg A))(1 + m(y_k, B)) \leqslant \sum_k (1 - m(x_k, \neg A))(m(x_k, \neg A) + m(y_k, \neg B))\\
&\quad \text{then } \alpha = \beta = 0\\
&\quad \text{else if } \sum_k (1 - m(x_k, \neg A))(1 + m(y_k, \neg B)) \leqslant \sum_k (1 - m(x_k, \neg A))(m(x_k, \neg A) + m(y_k, B))\\
&\qquad \text{then } \alpha = \beta = 1\\
&\qquad \text{else if } \sum_k (1 - m(x_k, \neg A))(1 + m(x_k, \neg A)) \leqslant \sum_k (1 - m(x_k, \neg A))(m(y_k, B) + m(y_k, \neg B))\\
&\qquad\quad \text{then } \alpha = \beta = \frac{\sum_k (1 - m(x_k, \neg A))(1 + m(x_k, \neg A) + m(y_k, B) - m(y_k, \neg B))}{2\sum_k (1 - m(x_k, \neg A))^2}\\
&\qquad\quad \text{else } \alpha = \frac{\sum_k (1 - m(x_k, \neg A))(1 - m(y_k, \neg B))}{\sum_k (1 - m(x_k, \neg A))^2},\\
&\qquad\qquad \beta = \frac{\sum_k (1 - m(x_k, \neg A))(m(y_k, B) - m(x_k, \neg A))}{\sum_k (1 - m(x_k, \neg A))^2}.
\end{aligned}
$$

Proof. For this problem the Kuhn–Tucker optimality conditions are once again necessary and sufficient. The conditions in this case are:

There exist real numbers u_1, u_2 and u_3 such that:

$$
\begin{aligned}
&2a\alpha - 2c - u_1 + u_2 = 0,\\
&2a\beta + 2b - u_2 + u_3 = 0,\\
&u_1\alpha = 0,\\
&u_2(\alpha - \beta) = 0,\\
&u_3(\beta - 1) = 0,\\
&u_1, u_2, u_3 \geqslant 0.
\end{aligned}
$$

Taking into account that $a \geqslant 0$, $c \geqslant 0$ and $a + b \geqslant 0$, it is easy to check that the Kuhn–Tucker conditions are verified by the following values of α, β and u_i:

Case 1: if $a = 0$ then $b = c = 0$ and F is a constant function. So we again select the least specific values $\alpha = 0$, $\beta = 1$.

Case 2: if $b - c \geqslant 0$ then $\alpha = \beta = 0$, $u_1 = 2(b - c)$, $u_2 = 2b$, $u_3 = 0$.

Case 3: if $c - b - 2a \geq 0$ then $\alpha = \beta = 1$, $u_1 = 0$, $u_2 = 2(c - a)$, $u_3 = 2(c - b - 2a)$.
Case 4: if $b + c \geq 0$, $c - b \geq 0$ and $2a + b - c \geq 0$ then $\alpha = \beta = (c - b)/2a$, $u_1 = 0$, $u_2 = c + b$, $u_3 = 0$.
Case 5: if $b + c \leq 0$ and $a - c \geq 0$ then $\alpha = c/a$, $\beta = -b/a$, $u_1 = u_2 = u_3 = 0$.

Now, it only remains to put these questions in order, from Case 1 to Case 5, to obtain the nested if–then–else procedure that gives the optimum. □

Observe that in both approaches the calculations are very easy to perform and they can be done efficiently: the running time will be linear in the size K of the set of examples, for each rule. So, for n and m labels in the antecedent and consequent sets respectively, the running time for studying all the possible rules is of the order $O(nmK)$. Moreover, both approaches can support two modes of learning: training and adaptation. By training we mean the method of operation that creates models by batch-processing of large data bases; this is the mode we have described so far. Adaptation consists in modifying a model through experience. Our methods could perform an iterative learning giving rise to an adaptive process. Next, we are going to describe the adaptation mode for both the average and the aggregation approaches:

(1) In the average approach, it suffices to record the quantity $T = \sum_k t_k$, and the current weights α and β. Once we get another example (x_0, y_0), we calculate its associated weights α_0 and β_0, and the fitness degree t_0. The updated weights α and β are then

$$[\alpha, \beta] = \left[\frac{T\alpha + t_0\alpha_0}{T + t_0}, \frac{T\beta + t_0\beta_0}{T + t_0}\right]$$

and T is updated to $T + t_0$.

This may be interpreted as an average between the old weights of the rule and the weights corresponding to the new example. The number $T/(T + t_0)$ represents the strength of the old weights when compared with the new ones (whose strength is $t_0/(T + t_0)$). Normally $T/(T + t_0)$ will be much greater than $t_0/(T + t_0)$, if the number of examples is large. So, the updating process matches our intuition that one additional example will not greatly modify the previously established conclusions.

(2) In the aggregation approach, it is only necessary to record the parameters a, b and c of (8). After obtaining a new example (x_0, y_0), these parameters are updated as follows:

$$a' = a + (1 - m(x_0, \neg A))^2,$$
$$b' = b + (1 - m(x_0, \neg A))(m(x_0, \neg A) - m(y_0, B)),$$
$$c' = c + (1 - m(x_0, \neg A))(1 - m(y_0, \neg B)).$$

Next, we repeat the simple process given in (10) that determines the optimal weights from a', b' and c'.

Here, the updating process is also gradual: as the difference between a, b, c and a', b', c' will usually be small, a drastic change in the values of α and β is not expected.

3.3. Discounting the rules

As we commented at the end of Section 3.1, in some cases, even the optimal weights associated to a rule by an example (x, y) give rise to a very poor adjustment between the real and the predicted outputs. This situation arises when x matches very well with the negation of the antecedent A of the rule. For example, if for a given example (x_1, y_1), the matching degrees $m(x_1, \neg A)$, $m(y_1, B)$ and $m(y_1, \neg B)$ are 0.9, 1 and 0 respectively, then the weights associated to the rule are $\alpha_1 = \beta_1 = 1$, that is, the rule is completely supported by the example. However, intuition says that such an example should not be very significant for the rule, because of the high degree of matching between x_1 and $\neg A$. This is reflected by the lower value of the fitness degree, $t_1 = 0.104$. Although the problem diminishes when we have more than one example, it does not disappear completely.

For example, let us suppose that we have another example (x_2, y_2), such that $m(x_2, \neg A) = 0.1$, $m(y_2, B) = 0.2$ and $m(y_2, \neg B) = 1$; then the weights associated to the rule by this example are $\alpha_2 = 0$ and $\beta_2 = 0.111$, that is, this example provides strong support to the rule ($A \rightarrow B$ is $[0, 0.111]$, or

equivalently, $A \rightarrow \neg B$ is $[0.889, 1]$), but now the fitness degree is $t_2 = 1$. If we use both examples, according to (7), then in the average approach we get the values $\alpha = 0.095$, $\beta = 0.195$, and we can see that the strange effect caused by (x_1, y_1) has been greatly diminished (the same thing happens in the aggregation approach, where we obtain the values $\alpha = \beta = 0.122$ when we use the two examples). Usually, the more examples we have, the more the problem is reduced.

Now let us suppose that instead of (x_2, y_2), the second example is (x_2', y_2'), such that $m(x_2', \neg A) = 0.8$, $m(y_2', B) = 0.9$ and $m(y_2', \neg B) = 0.2$. In that case, $\alpha_2' = \beta_2' = 1$ ($t_2' = 0.246$). By using (x_1, y_1) and (x_2', y_2') together we again get $\alpha = \beta = 1$. So, in this case, the problem persists.

One way to avoid this problem is to increase the uncertainty of the rule when the examples do not support it (reflecting the idea that we cannot correctly learn a concept from examples that do not match this concept). To do that, we will use the idea of discounting (in the sense of Shafer [10]) the conditional measure resultant from the optimization process, by a factor depending on the degree of adjustment between the examples and the rule.

Remember that the conditional measures (l, u) resultant from the optimization process are

$$l(B \mid A) = \alpha, \qquad l(\neg B \mid A) = 1 - \beta,$$
$$u(B \mid A) = \beta, \qquad u(\neg B \mid A) = 1 - \alpha$$

with α and β obtained from (7) or (10).

If we discount these measures by a factor, say ε, $0 \leqslant \varepsilon \leqslant 1$, we obtain as a result a new pair $(l_\varepsilon, u_\varepsilon)$ given by

$$l(B \mid A) = \alpha(1 - \varepsilon), \qquad l(\neg B \mid A) = (1 - \beta)(1 - \varepsilon),$$
$$u(B \mid A) = \beta(1 - \varepsilon) + \varepsilon, \qquad u(\neg B \mid A) = (1 - \alpha)(1 - \varepsilon) + \varepsilon.$$

Now, we must decide how to select the factor ε; it depends on the approach we are using for learning:

(1) Average approach. We use as the factor the quantity

$$\varepsilon = 1 - \frac{\sum_k t_k}{K},$$

where K is the number of examples. The greater ε is the worse the adjustment is. Therefore, taking into account (7) and the value of ε, the discounted weights α_ε and β_ε for the rule are

$$[\alpha_\varepsilon, \beta_\varepsilon] = \left[\frac{\sum_k t_k \alpha_k}{K}, 1 - \frac{\sum_k t_k (1 - \beta_k)}{K}\right]. \tag{11}$$

(2) Aggregation approach. Now we use as the factor the quantity

$$\varepsilon = 1 - \frac{a}{K} = 1 - \frac{\sum_k (1 - m(x_k, \neg A))^2}{K}.$$

So, taking into account (10), the procedure to obtain the discounted weights α_ε and β_ε is:

$$\begin{aligned}
&\text{if } a = 0 \text{ then } \alpha_\varepsilon = 0, \qquad \beta_\varepsilon = 1 \\
&\text{else if } b - c \geqslant 0 \text{ then } \alpha_\varepsilon = 0, \qquad \beta_\varepsilon = 1 - \frac{a}{K} \\
&\quad \text{else if } c - b - 2a \geqslant 0 \text{ then } \alpha_\varepsilon = \frac{a}{K}, \qquad \beta_\varepsilon = 1 \\
&\qquad \text{else if } c + b \geqslant 0 \text{ then } \alpha_\varepsilon = \frac{c - b}{2K} \qquad \beta_\varepsilon = 1 + \frac{c - b - 2a}{2K} \\
&\qquad\quad \text{else } \alpha = \frac{c}{K} \qquad \beta = 1 - \frac{a + b}{K}.
\end{aligned} \tag{12}$$

For the previous example involving (x_1, y_1) and (x_2', y_2'), now using (11) in the average approach, we get $[\alpha, \beta] = [0.175, 1]$. Using (12) in the aggregation approach we obtain $[\alpha, \beta] = [0.025, 1]$. Both intervals represent a great uncertainty about the rule, reflecting much better than the previous ones the idea that the two examples being used are not very significant for the rule.

4. The crisp case and the underlying concept of conditioning

In order to understand better the learning methods just explained, the study of some simple cases is revealing:

Consider first the case of the antecedent and consequent sets A and B being crisp sets, and all the examples x_k and y_k being crisp values. Denote by $n(A \cap B)$ ($n(A)$ respectively) the number of examples satisfying $x_k \in A$ and $y_k \in B$ ($x_k \in A$ respectively). Then it is very easy to see that

$$\alpha = \beta = \frac{n(A \cap B)}{n(A)}$$

in both the average and the aggregation approaches. This means that we always obtain a probabilistic rule, the weight $\alpha = \beta$ being the relative frequency of examples verifying A and B among those that verify A, that is, the conditional probability of B given A:

$$\alpha = \beta = p(B \mid A).$$

If we now consider the case of the antecedent and consequent sets being crisp sets again, and the examples also being crisp sets (instead of crisp values), we get, again for both the average and aggregation approaches, the following formulas for α and β:

$$\alpha = \frac{n(A \cap B)}{n(A)}, \qquad \beta = \frac{n(A) - n(A \cap \neg B)}{n(A)}$$

where $n(A \cap B)$ is now the number of examples verifying $x_k \subseteq A$ and $y_k \subseteq B$ (and similarly, $n(A)$ is the number of examples satisfying that $x_k \subseteq A$ and $n(A \cap \neg B)$ represents the number of examples in which $x_k \subseteq A$ and $y_k \subseteq \neg B$ is true).

When the examples are crisp sets instead of crisp values we have a lack of precision about the truth of A and B. If for instance $x_k \cap A \neq \emptyset$ and $x_k \cap \neg A \neq \emptyset$, then the example x_k can support both A and $\neg A$; we will only be sure when $x_k \subseteq A$ or $x_k \subseteq \neg A$ (the same thing happens for y_k and B). This gives rise to a greater uncertainty: there are examples that confirm A, examples that confirm $\neg A$, and examples that confirm neither A nor $\neg A$. This situation is modeled by interpreting the quantity $n(C) \mid K$ as the value of a belief measure, bel, expressing our (frequentistic) confidence on C being absolutely true. For example, if we have three examples satisfying that $x_k \subseteq A$, two examples verifying that $x_k \subseteq \neg A$ and the remaining five examples satisfy $x_k \cap A \neq \emptyset$ and $x_k \cap \neg A \neq \emptyset$, then we get a belief–plausibility pair whose b.p.a. is $m(A) = 0.3$, $m(\neg A) = 0.2$ and $m(U_A) = 0.5$.

After these comments it is clear that the values of α and β above can be rewritten as

$$\alpha = \mathrm{bel}(B/A) = \frac{\mathrm{bel}(A \cap B)}{\mathrm{bel}(A)}, \qquad \beta = \mathrm{Pl}((B/A) = \frac{\mathrm{bel}(A) - \mathrm{bel}(A \cap \neg B)}{\mathrm{bel}(A)}$$

and we obtain the strong conditioning proposed for belief–plausibility measures by Shafer [11] and Suppes and Zanotti [12].

In the general case, when we have fuzzy sets instead of crips sets, the resultant optimum weights α and β for the rule (4) can be considered as a generalization of this kind of conditioning.

5. Concluding remarks

We have developed a methodology for learning rules in a fuzzy environment which uses only empirical information. Although the rules considered here are simple, it is easy to extend the model to include rules with conjunctions and disjunctions in premises (see [4]).

The methods described in this paper are efficient in both running time and storage requirements. Moreover, they can be used in training mode (at an initial stage) as well as in adaptation mode (after collecting new examples).

Another interesting property of our methodology is its flexibility: several alternatives could be considered within the model, without affecting its basis; we briefly list some of them:

- We chose a type of matching based on the Łukasiewicz t-norm, but different t-norms could be used instead (for example, the minimum or the product). It would be interesting to study how sensitive the method is with respect to the choice of the t-norm.
- The proposed propagation model is based on the integration of conditional upper measures $(u(\cdot \mid A),\ u(\cdot \mid \neg A))$ with respect to a marginal upper measure $(u_X(\cdot))$. The use of marginal lower measures would possibly entail a change in the underlying concept of conditioning. Moreover we chose the Choquet integral as integral operator, but there are other alternatives, as for instance the Sugeno integral.
- The selected distance measure and either the aggregation function or the kind of average could be changed too.

The exploration of these alternatives and their comparison with different approaches for learning will be the object of further work.

References

[1] M.S. Bazaraa and C.M. Shetty, *Nonlinear Programming. Theory and Algorithms* (Wiley, New York, 1979).

[2] E. Binaghi, Learning of uncertain classification rules in medical diagnosis, in: R. Kruse, P. Siegel, Eds., *Symbolic and Quantitative Approaches to Uncertainty, Lecture Notes in Computer Science* 548 (Springer Verlag, Heidelberg, 1991) 115–119.

[3] L.M. de Campos, M.T. Lamata and S. Moral, Distances between fuzzy measures through associated probabilities: Some applications, *Fuzzy Sets and Systems* **35** (1990) 57–68.

[4] L.M. de Campos and A. González, A fuzzy inference model based on an uncertainty forward propagation approach, in: R. Lowen, M. Roubens, Eds., *Proceedings of the IFSA'91* (1991) 13–16. To appear also in *Int. J. of Appr. Reasoning.*

[5] L.M. de Campos and S. Moral, Propagating uncertain information forward, *Int. J. of Intelligent Systems* **7** (1992) 15–24.

[6] G. Choquet, Theory of capacities, *Ann. Inst. Fourier* **5** (1953) 131–295.

[7] M. Delgado and A. González, An inductive learning procedure to identify fuzzy systems, *Fuzzy Sets and Systems* **55** (1993) 121–132.

[8] H. Nomura, I. Hayashi and N. Wakami, A learning method of fuzzy inference rules by descent method, *Proc. of the IEEE Intern. Conf. on Fuzzy Systems* (1992) 203–210.

[9] W. Pedrycz, An identification algorithm in fuzzy relational systems, *Fuzzy Sets and Systems* **13** (1984) 153–167.

[10] G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, NJ, 1976).

[11] G. Shafer, A theory of statistical evidence, in: Harper, Hooker, Eds., *Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science,* vol. II (D. Reidel Publishing Company, Dordrecht, 1976) 365–436.

[12] P. Suppes and M. Zanotti, On using random relations to generate upper and lower probabilities, *Synthese* **36** (1977) 427–440.