Learning fuzzy systems with similarity relations

Isabela Drummond[1], Lluis Godo[2], and Sandra Sandri[1]

[1] LAC - INPE
12201-970 S.J. Campos, Brazil
{isabela,sandri}@lac.inpe.br

[2] IIIA - CSIC
08193 Bellaterra, Spain
godo@iiia.csic.es

Abstract. We present a method to learn terms in a fuzzy rule based system by means of similarity relations. The method applies to conjunctive systems, such as Mamdani fuzzy controllers, as well as to implicative systems. The approach is implemented in such a way to guarantee both rule base consistency and covering.

1 Introduction

In [6] we defined a fuzzy rule-based system as being complete when it is both consistent and covers all the input domain. Using our terminology, a system covers the input domain when at least one rule addresses each possible input, i.e., when all the values in the input domain are considered to be valid. On the other hand, a system is consistent (at least to some degree) when each valid input generates a meaningful output fuzzy set.

Lack of covering can appear in both conjunctive systems, in which a conjunctive operator (i.e. a t-norm) is used to model the if-then relation [4], as well as in implicative systems, which employ truly implicative operators to model the if-then relation in the rules. Inconsistency, as we consider it, never occurs in conjunctive systems. Indeed, in these systems, the rule base is seen as a disjunction of multi-dimensional pieces of data [5], and thus the fuzzy sets obtained as result of the firing of the rules are agregated using a disjunctive operator, such as max, which never leads to inconsistency. In implicative systems, however, inconsistency can easily occur because in these systems the rule base is seen as a conjunction of implications [5], and thus the fuzzy sets obtained as result of the firing of the rules should be agregated using a conjunctive operator, such as min, which can easily lead to an empty fuzzy set, even with only two rules fired.

In [6] we have proposed the use of similarity relations to solve inconsistency problems in fuzzy systems of gradual rules, a particular kind of implicative systems, in which the rules employ residuated operators as implication function and that are semantically equivalent to statements of the form "The more x is A, the more y is B" [5]. The idea is to transform rules "If x is A then y is B", where A and B are fuzzy sets, into rules 'If x is A then y is $approximately\_B$", where "$approximately\_B$" is the result of the application of a similarity relation to fuzzy set B, and hence wider than B. When suitable similarity relations are used, we can thus eliminate the inconsistencies generated by a set of rules. We also proposed to solve the lack of coverage problem using similarity relations [7]. In this case, we proposed to transform rules "If

x is A then y is B", into rules 'If x is $approximately\_A$ then y is B", thus enlarging the term in the rule premise, in order to increase covering. In the case that a rule-base is both inconsistent and uncovered we propose to first apply a similarity relation to the terms in the premise to overcome the lack of covering and then apply another (possibly different) relation to the terms in the rule consequents to deal with the inconsistencies.

However, in our previous works we have not addressed the problem of determining which are the set of parameters (determining e.g. a parametric similarity relation) that are best suited for a given application, which depends largely on the application itself. A natural procedure along this line of work is thus to consider some kind of learning procedure to determine parameters leading to complete rule bases that yield reasonable, if not optimal, results. Contrary to the previous works in which full consistency was required, here we only require the base not to be inconsistent.

In this work we investigate the use of similarity relations in the process of tuning the terms in a fuzzy system application, in the sense of solving posible covering and/or consistency problems, but possibly also for fitting some data. We first of all present some basic concepts in Section 2. In Section 3 we present some ideas on similarity relations and then in Section 4 we discuss some aspects of learning such relations. We have tested our approach on a simple fuzzy control system using genetic algorithms, but we tried to orient this discussion in such a way to be as much independent as possible of the learning framework chosen to find the parameters needed in a given application.

2 Basic definitions and notations

In this section we recall some basic definitions that are used in the rest of the paper and provide some notation. Most of the definitions and remarks are well-known in the literature.

In the rest of the paper, unless stated otherwise, we shall work with fuzzy subsets of the real line, so the domain U below is assumed to be $\mathbb{R}$. The core (respec. support) of a fuzzy set $A : U \to [0,1]$ is defined as $core(A) = \{x \mid A(x) = 1\}$ (respec. $supp(A) = \{x \mid A(x) > 0\}$). For any $\alpha \in [0,1]$, the α-cut of A is defined as $[A]_\alpha = \{x \in U \mid A(x) \geq \alpha\}$. A is said to be *normalized* when there exists x such that $A(x) = 1$, and *convex* when for all x, y, z, if $x \leq y \leq z$, there exists $\alpha \in [0,1]$ such that $A(y) = \alpha \cdot A(x) + (1-\alpha) \cdot A(z)$. A convex fuzzy set A will be denoted as $< a_1, a_2, a_3, a_4 >$ when $supp(A) = (a_1, a_4)$ and $core(A) = [a_2, a_3]$ and when its shape is not relevant[1]. In such a case, the *opposed* fuzzy set of A is given by $\overline{A} =< -a_4, -a_3, -a_2, -a_1 >$. We shall say that fuzzy set $A =< a_1, a_2, a_3, a_4 >$ *comes before* fuzzy set $B =< b_1, b_2, b_3, b_4 >$, denoted by $A << B$, if $a_i \leq b_i, i = 1,4$. It can be noticed that if $A << B << C$, with A, B, C non empty, then $supp(A) \cap supp(C) \subseteq supp(B)$. Finally, $A =< a_1, a_2, a_3, a_4 >$ is said to be a *fuzzy interval* when the membership function of A in the intervals $[a_1, a_2]$ and $[a_3, a_4]$ are strictly monotonic, if the intervals are not points.

An operator $\otimes : [0,1]^2 \to [0,1]$ is called a *t-norm* if it is commutative, associative, monotonic and has 1 as neutral element. A *residuated implication operator* $\to_\otimes$ is de-

[1] Actually, if A denotes a crisp interval $[a, b]$ (or a point in particular) we can use still the same notation $< a, a, b, b >$ although it is in fact somehow inconsistent.

fined as $a \rightarrow_\otimes b = \sup \{c \in [0,1] \mid a \otimes c \leq b\}$, where $\otimes$ is a left-continuous t-norm ($\rightarrow_\otimes$ is said to be the residuum of $\otimes$). One well-known residuated operator is Gödel implication (residuum of $\otimes = \min$) defined as $a \rightarrow_{\otimes_G} b = 1$ if $a \leq b$ and $a \rightarrow_{\otimes_G} b = b$ otherwise. Another one is Goguen implication, defined as $a \rightarrow_{\otimes_\Pi} b = 1$ if $a \leq b$ and $a \rightarrow_{\otimes_\Pi} b = b/a$ otherwise. The so-called Rescher-Gaines implication function, defined as $a \rightarrow_{\otimes_{RG}} b = 1$ if $a \leq b$ and $a \rightarrow_{\otimes_{RG}} b = 0$ otherwise, is not a residuated operator but it is in fact the point-wise infimum of all residuated implications.

Once fixed a (continuous) t-norm $\otimes$, a *gradual fuzzy rule* "If x is A_i then y is B_i" induces a fuzzy relation between input and output values which is defined as $R_i(x,y) = (A_i \rightarrow_\otimes B_i)(x,y) = A_i(x) \rightarrow_\otimes B_i(y)$. Then, given a gradual rule set $K = \{$"If x is A_i then y is B_i"$\}_{i \in I}$, the induced global fuzzy relation R_K is the intersection of the individual ones, i.e. $R_K(x,y) = \min_{i \in I} R_i(x,y)$. Finally, given an (imprecise) input "x is A_0", the output "y is B_0" produced by K is usually computed as sup-$\otimes$ composition of A_0 with R_K, that is, $B_0(y) = output(K, A_0)(y) = \sup_x A_0(x) \otimes R_K(x,y)$. In particular, if the input is a precise value, $x = x_0$, then one gets as output $output(K, \{x_0\}) = R_K(x_0, y)$. The degree of consistency of a rule set K is given by

$$Con(K) = \inf_x \sup_y output(K, \{x\})(y),$$

i.e. the infimum of the all possible output heights, considering all valid input values. If $Con(K) = \alpha$ then the rule base K is said to be α-consistent. K is considered *fully consistent* if $Con(K) = 1$, that is, if we always get a normalized output for any precise input. Full consistency corresponds to the notion of *coherence* in [5] and *1-consistency* in the context of π-reasoning in [9]. Conditions characterizing full consistency in systems of fuzzy gradual rules have been basically addressed by Dubois, Prade and Ughetto in [5]. In particular, two gradual fuzzy rules $R_i(x,y) = A_i(x) \rightarrow_\otimes B_i(y)$, $i = 1,2$, are fully consistent iff, for any precise input $X = x_0$, the intersection of the cores of the outputs is non-empty, i.e. if $[B_1]_{A_1(x_0)} \cap [B_2]_{A_2(x_0)} \neq \emptyset$ for any x_0 [5]. As it can be seen, the full consistency condition for pairs of gradual rules is actually independent from the particular residuated implication $\rightarrow_\otimes$ used to define the fuzzy relations induced by the rules, since $a \rightarrow_\otimes b = 1$ iff $a \leq b$ for any residuated operator $\rightarrow_\otimes$. Due to this fact, to guarantee that a set of implicative rules is consistent one can simply verify the consistency using Rescher-Gaines implication function (see [3] for more details).

A rule base K is said to be *uncovered* if for a given input there exists no rule in K whose antecedent (partially) matches the input [7]. The lack of covering in a rule base can be either local or global. The local case occurs when there exists "holes" in the domain partition of an input variable, i.e., when for a given input there exists no term of the variable that covers it. To solve this problem only the variable itself has to be considered. The global case occurs when for a given input, there exist at least one set of terms that cover it, but there does not exist any rule in K that addresses the set of terms at the same time (see [3] for a discussion on this matter).

A set of terms $\{D_1, \ldots, D_n\}$ is said to be in *consecutive order* when for all i, if $supp(D_i) \cap supp(D_{i-1}) \neq \emptyset$ and $supp(D_i) \cap supp(D_{i+1}) \neq \emptyset$, then $supp(D_i) \cap supp(D_j) = \emptyset, \forall j \notin \{i-1, i+1\}$. A set of terms associated with a variable in a given application are said to form a *Ruspini fuzzy partition* of the domain of the variable are such that: (i) all terms are fuzzy intervals, (ii) all terms can be put in consecutive order, and (iii) every two consecutive terms D and D' are such that $D(\omega) + D'(\omega) = 1$ for all $\omega \in supp(D) \cap supp(D')$, Ruspini partitions are very often found in fuzzy control applications, as the set of supports (of the fuzzy terms) are completely determined by the set of cores, and vice-versa.

3 Similarity relations in view of applications

A similarity relation S on a domain U is a binary fuzzy relation, i.e. a mapping $S : U \times U \rightarrow [0, 1]$ that satisfies the following properties:

(i) $\forall v \in \Omega, S(v, v) = 1$ (*reflexivity*);
(ii) $\forall v, v' \in \Omega, S(v, v') = S(v', v)$ (*symmetry*);

Some authors require similarity relations to also satisfy the T-norm transitivity property ($S(u, v) \otimes S(v, w) \leq S(u, w)$ for all $u, v, w \in U$ and some t-norm $\otimes$), but we do not take it into consideration here as it does not seem to play a role in our framework.

The application of a similarity relation S on a fuzzy term A, denoted by $S \circ A$, creates a "larger" term $approximately\_A$. Formally, we have

$$(S \circ A)(v) = \sup_{v' \in U} \min(S(v, v'), A(v')).$$

The set of similarity relations on a given domain U forms a lattice (not linearly ordered) with respect to the point-wise ordering (or fuzzy-set inclusion) relationship. The top of the lattice is the similarity $S_\top$ which makes all the elements in the domain maximally similar: $S_\top(v, v') = 1$ for all $v, v' \in U$. The bottom of the lattice $S_\bot$ is the classical equality relation: $S_\bot(v, v') = 1$ if $v = v'$, $S_\bot(v, v') = 0$, otherwise. The higher a similarity is placed in the lattice (i.e. the bigger are their values), the less discriminating it is.

In this paper, for the sake of easier tuning tasks, we are interested in parametric families $\mathcal{S} = \{S_0, S_{+\infty}\} \cup \{S_\beta\}_{\beta \in I \subseteq (0, +\infty)}$ of similarity relations such that:

(i) $S_0 = S^\bot$,
(ii) $S_{+\infty} = S^\top$, and
(iii) $\beta < \beta'$, then $S_\beta \prec S_{\beta'}$.

Here $S \prec S'$ means $S(x, y) \leq S'(x, y)$ for all $x, y \in U$ and $S(x_0, y_0) < S'(x_0, y_0)$ for some $x_0, y_0 \in U$.

Considering their use in applications, we shall also require some further interesting features to similarity functions in parametric families as described above. Namely we are interested in similarity functions which are:

- *compatible with the euclidean distance*: we shall consider only similarity relations which are compatible with the usual distance on the reals, in the sense that if $x < y < z$, then $S(x, z) \leq \min(S(x, y), S(y, z))$. This property is not found in the literature, but it seems interesting to have it since it grasps the meaning of something being similar to something else as opposed of being distant, in the usual (euclidean) sense.

- *convexity-preserving*: since linguistic terms will are represented by convex fuzzy numbers, we want similarities such that if A is a convex fuzzy set, then $S \circ A$ is also convex.
- *order-preserving*: we want similarities that are order-preserving in the sense that if $A << B$, then $S \circ A << S \circ B$. This is an essential characteristic for our purposes, as this will allow us to easily make consistency checks.
- *core-preserving*: in this specific work we are interested in similarity functions that are core-preserving, i.e. $core(S \circ A) = core(A)$. This is achieved by requiring the similarity relations to be *separating*, i.e. satisfying $S(x, y) = 1$ iff $x = y$. This also allows us to obtain efficiency, but it does not mean that non core-preserving similarities cannot useful in many applications. Indeed, the use of similarity families that would also enlarge the core can be very interesting when the terms are learned from stored data, though that option would be harder to justify when the available data would have been furnished by experts.

Easy examples of similarity functions satisfying the above requirements are the following, for respectively $\Omega = \mathbb{R}$ and $\Omega = \mathbb{R}^+$:

$$S_\alpha(x, y) = \max(0, 1 - \alpha^{-1} \cdot |x - y|)$$

$$S^\beta(x, y) = \begin{cases} 1, & \text{if } x = y = 0 \\ 0, & \text{if } \min(x, y)/\max(x, y) < \beta \\ \min(x, y)/\max(x, y), & \text{otherwise} \end{cases}$$

where $\alpha > 0$ and $0 < \beta \leq 1$.

Let $A = < a, b, c, d >$ be a fuzzy set on $\mathbb{R}^+$ (here we are only interested in the effects of similarity relations over positive fuzzy sets). The above parametric similarity relations are such that:

1. for $\alpha \geq 0$, $supp(S_\alpha \circ A) = [a - \alpha, d + \alpha]$;
2. for $\beta \in (0, 1]$, $supp(S^\beta \circ A) = [a \cdot \beta, d/\beta]$;
3. for $\alpha \geq 0$ and $\beta \in (0, 1]$, $supp(S_{\alpha,\beta} \circ A) = [a \cdot \beta - \alpha, d/\beta + \alpha]$, where $S_{\alpha,\beta} = S_\alpha \circ S^\beta$.
4. for $\alpha \geq 0$ and $\beta \in (0, 1]$, $supp(S^{\alpha,\beta} \circ A) = [(a - \alpha) \cdot \beta, (d + \alpha)/\beta]$, where $S^{\alpha,\beta} = S^\beta \circ S_\alpha$.

Families S_α and S^β are particular cases of the 2-parameter family classes $S_{\alpha,\beta}$ and $S^{\alpha,\beta}$, as $S_{\alpha,1} = S^{\alpha,1} = S_\alpha$, and $S_{0,\beta} = S^{0,\beta} = S^\beta$. Figure 1 illustrates the application of $S_{0,\beta}$ and $S_{\alpha,\beta}$, $0 < \alpha, \beta < 1$, to a set of symmetric precise terms[2].

Class S_α is such that the support of A is augmented by 2α, no matter if the term is close to 0 or not. Moreover, both sides of the fuzzy set are also enlarged with the same constant value (if A is symmetric, $A' = S_\alpha \circ A$ will also be symmetric).

Class S^β is such that the support of terms closer to 0 are enlarged less then those in the domain extreme; the sides of the fuzzy sets are enlarged with different constant values (if A is symmetric, $A' = S^\beta \circ A$ will not be symmetric, except when $\beta =$

[2] The similarity relations were applied to the positive terms and the negative terms are their negative counterparts.

1). If all we have from the terms are prototypical values, the learning of this kind of similarity relation will allow the terms to be more precise and crumpled when close to 0, a characteristic that is usually very interesting in some applications, such in fuzzy control. However, if $A_0 =< 0,0,0,0 >$ then it is not possible to enlarge A_0 with any S^{β}. An example of a similarity relation related to this latter class is

$$SS_{m,n}(x,y) = \begin{cases} S(x,y)^n, \text{ if } (S(x,y))^n > m \\ 0, \quad \text{otherwise} \end{cases}$$

with $m \in [0,1], n \geq 1$ and where $S(x,y) = \min(x,y)/\max(x,y)$ or $S(x,y) = 1 - |x-y|/(x+y) = 2 \cdot min(x,y)/(x+y)$.

The general classes of similarity relations $S_{\alpha,\beta}$ and $S^{\alpha,\beta}$, with $(\alpha,\beta) \neq (0,1)$, combine the features of the two previous classes. In particular, the class $S_{\alpha,\beta}$ is specially well suited to deal with implicative systems since the multiplicative constant can be used to determine the support of the fuzzy terms and the additive one to eliminate inconsistencies.

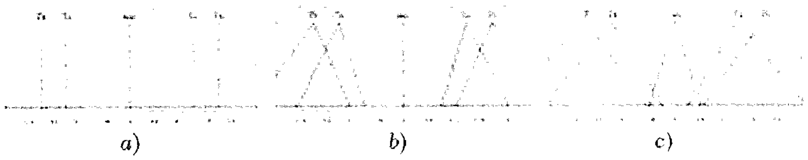

Fig. 1. Fuzzy terms: (a) precise terms A_i, (b) terms $S_{0,\beta} \circ A_i$ and (c) terms $S_{\alpha,\beta} \circ A_i$, $\alpha > 0$.

4 Learning with similarity relations

Here we consider learning fuzzy terms using similarity relations in the context of applications in which at least the rule base structure itself is known. The fuzzy terms are either given, in which case we only need to learn the parameters of a similarity relation that will transform them, or they are created from scratch using a learning tecnique. Our framework is restricted to terms defined by linear by parts convex membership functions (points, crisp intervals or triangular/trapezoidal shaped functions). In this way, we only need to manipulate the core and support of the terms, as we can easily calculate the membership functions associated to them.

Therefore, since we are only interested in linearized terms, we will not bother with the form of the similarity relations used, as we would linearize the terms resulting from its application anyways. In this way, we only need to know the effect on the core and support of a given term A (eventually just a single point), as we apply a similarity relation on it. Thus, in our case, we only need to know what is the core and support of $S_{\alpha,\beta} \circ A$, which have in fact been described by the restrictions in the previous section, and not the relation itself. In the following we list some of the possibilities for learning parameters for systems of gradual rules. The rule base can be in one of the folowing situations: (1) the rule base is consistent and covers the input, (2) the rule base is inconsistent but covers the input, (3) the rule base is consistent but does not cover the input, (4) the rule base is inconsistent, does not cover the input and the terms are known, and (5) only the number of terms per variable are known.

In the first case, one may use similarity relations on the already existing set of terms only to try to improve the system performance. However, the cost of learning the parameters for the relations may not be worth the effort. In the second case we may learn parameters for either $S_{\alpha,1}$, $S_{0,\beta}$ or $S_{\alpha,\beta}$ on the output variables. However, one should note that the support of the fuzzy set $zero = < 0, 0, 0, 0 >$ remains the same when $S_{0,\alpha}$ is used. Case (3) is similar to case (2) but the parameters are learned for the input variables. Case (4) is treated like both cases (2) and (3). In case (5) we can choose to learn precise or imprecise cores. For precise cores an extra parameter (the core itself) has to be learned. For the imprecise core choice, instead of learning 2 extra parameters, a low cost solution consists to learn a precise core per term, plus a single extra parameter for each variable for a relation $S_{0,\beta}$ that would enlarge the precise cores. Then, no matter how the cores are to be determined, in order to determine the supports, we could use Ruspini partitions, with no extra cost, or learn the supports directly using similarity relations, as discussed in the previous cases. In what regards conjunctive systems, the reasoning remains the same but the number of cases decreases, since conjunctive systems are never inconsistent.

The approach proposed here has been applied to a fuzzy control aplication [1] using the *shower system*, available in Matlab version 4 Simulink toolbox. A genetic algorithm has been used to learn fuzzy terms membership functions (points and/or parameters for similarity relations) for an already fixed set of rules, i.e. the rule base structure in itself was not learned (see [8] for an introduction on genetic algorithms for learning fuzzy systems). In all the experiments, a candidate solution was disregarded if it led to inconsistency and/or lack of covering. The experiments considered the use of the Mamdani conjunctive implication (min), and Gödel and Goguen residuated operators to treat situation (5). The results obtained, not reported here due to lack of space, can be found in [2].

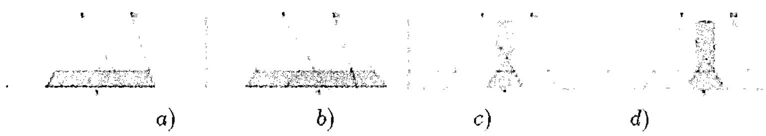

Fig. 2. Result of inference using: a) and b) Mamdani, c) and d) Gödel.

All experiments presented good results, with a slight superiority of those employing Mamdani operator. This superiority is probably partly due to the fact that in some situations the implicative systems may lose significative information, mainly related to the support of the original output sets. To illustrate this problem, let us consider a pair of rules R_1 and R_2, with output terms B_1 and B_2, and for each i let B_i^C (respec. B_i^R) be the result of the inference with each rule using Mamdani (respec. Gödel) operator for a given precise input. Finally let B^C (respec. B^R) be the result of the disjunctive (respec. conjunctive) aggregation of B_1^C and B_2^C (respec. B_1^R and B_2^R). It is easy to check that $supp(B_i^C) = supp(B_i^R) = supp(B_i)$ and hence $supp(B^C) = supp(B_1) \cup supp(B_2)$ and $supp(B^R) = supp(B_1) \cap supp(B_2)$. Figures 2 (a) and (c) show an example of all these fuzzy sets where, without lack of generality, the compatibility α of the input with both rules is assumed to be same. If we choose a final precise value taking the usual centroid defuzzification method [1], it is clear that the predominant dimension in B^C is the length of its support, while in B^R that is not so clear. Namely, a change in the supports of B_1 and B_2 will likely affect the location of the defuzzied output from B^C much more than the ouput from B^R, as can be seen in Figure 2. Moreover, for some kinds of modifications, it can even result in no change at all in the outputs from B^R. For instance, when $\alpha \geq height(B^R)$, in which case the base is not 1-consistent, the problem is that the defuzzified output will always be the same, no matter what is the value of α itself. This lack of sensitivity of Gödel (and other residuated operators) fuzzy systems can play an important role in the performance of systems such as fuzzy controllers, depending on the structure of the rule base. However, the experiments showed that this problem is partly compensated by the genetic algorithm with the creation of output partitions whose terms have approximately the same support length.

5 Conclusion

We have presented an investigation about learning fuzzy terms using similarity relations. It can be employed in both conjunctive and implicative fuzzy rule-based systems. In what regards conjunctive systems, such as the so-called Mamdani controllers, this approach represents an alternative to the use of Ruspini partitions, usually employed in this kind of systems. In relation to implicative systems, the framework proposed here represents a means to obtain implicative systems that are not only consistent but that also present good performance. We are presently investigating the suitability of a set of defuzzification methods to try to further augment the performance of these systems.

Acknowledgments: Isabela Drummond, Lluis Godo and Sandra Sandri acknowledge partial support by the Brazilian agency CAPES, by the the Spanish project LOGFAC (TIC2001-1577-C03-01) and by the Brazilian agency CNPq (proc. 200423/87-8), respectively.

References

1. Driankov, D., Hellendoorn, H., Reinfrank, M., *An Introduction to Fuzzy Control*. Springer-Verlag, 1996.
2. Drummond I., Uso de relações de similaridade para tratamento de consistência e cobertura em sistemas de regras difusas. Master degree thesis, CAP/INPE, S.J.Campos, Brazil, 2003
3. Drummond I., Godo L., Sandri S. Restoring consistency in systems of fuzzy gradual rules using similarity relations. In *LNAI* 2507, 386–396, 2002.
4. Dubois D., Prade H. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems* 84, 169–185, 1996.
5. Dubois D., Prade H., Ughetto L. Checking the coherence and redundancy of fuzzy knowledge bases. In *IEEE Trans. on Fuzzy Systems* 5(3), 398–417, 1997.
6. Godo L., Sandri S. A similarity-based approach to deal with inconsistency in systems of fuzzy gradual rules. In *Proc. of IPMU'02*, Annecy (France), 1655–1662, 2002.
7. Godo L., Sandri S. Dealing with covering problems in fuzzy rule systems by similarity-based extrapolation. In *Proc. of Fuzz-IEEE'02*, Honolulu (USA), 2002.
8. Pedrycz W., Gomide F. *An introduction to Fuzzy sets: Analysis and Design*. MIT Press, 1998.
9. Weisbrod J., Fantana N.L. Detecting local inconsistency and incompleteness in fuzzy rule bases. In *Proc. EUFIT'96*, Aachen (Germany) 656–660, 1996.