Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Fuzzy Sets and Systems 146 (2004) 63–77
FUZZY sets and systems
www.elsevier.com/locate/fss

Learning maximal structure fuzzy rules with exceptions

P. Carmona[a,*], J.L. Castro[b], J.M. Zurita[b]

[a] *Depto. Informática, E. Ingenierías Industriales, Universidad de Extremadura, Avenida Elvas, s/n, 06017-Badajoz, Spain*
[b] *Depto. Ciencias de la Computación e IA, Universidad de Granada, ETSI Informática, C/Daniel Saucedo Aranda, s/n, 18071-Granada, Spain*

Abstract

This paper proposes a method to solve the conflicts that arise in the framework of fuzzy model identification with maximal rules (Fuzzy Sets and Systems 101 (1999) 331) where rules are selected as general as possible. This resolution is expressed by including exceptions in the rules, that way achieving a higher model interpretability with respect to other techniques and a more accurate model. Besides, several methods are presented to improve the interpretability, based on compacting the rules and exceptions of the model. Furthermore, in order to reduce the number of conflicts that arise from the maximal rules, a heuristic strategy is proposed to generate those maximal rules. Finally, the method is applied to an example and the results are compared with other identification methods.

Keywords: Fuzzy model identification; Interpretability; Maximal rules; Rule simplification; Conflicting rules

1. Introduction

Fuzzy model identification [2,7,9] represents the model of a system from a set of examples by means of fuzzy rules. This model, which is a universal approximator [3,8], allows to describe linguistically the relation between the input and the output of the system, thus taking care of the interpretability of the result.

In order to achieve a high interpretability, we must try to identify rules as general as possible, so that each rule covers the highest number of examples and, this way, the size of the rule base diminishes. Nevertheless, obtaining those general rules may provoke the appearance of conflicting zones where rules with different consequents coexist, thus affecting negatively to the aforementioned interpretability.

* Corresponding author. Tel.: +34-9-2428-9300; fax: +34-9-2428-9601.
E-mail addresses: pablo@unex.es (P. Carmona), castro@decsai.ugr.es (J.L. Castro), zurita@decsai.ugr.es (J.M. Zurita).

0165-0114/$ - see front matter
doi:10.1016/j.fss.2003.11.006

In this paper, a strategy is proposed in order to solve these conflicts, making use of the information available from the examples in the conflicting zones. The solution to each conflict takes the form of exceptions in the rules, which will allow to diminish the number of rules in the model and increase its interpretability.

Section 2 introduces the original identification algorithm to obtain maximal rules. Section 3 presents the approach for solving the conflicts existing among the maximal rules. Section 4 proposes several methods for improving the interpretability along and after the conflict resolution, based on compacting rules and exceptions. Section 5 describes a heuristic strategy, which selects the best set of maximal rules, based on considering several candidate maximal rules for each training example. Finally, in Section 6 the results obtained when applying the methods to several systems are analysed and compared with other identification methods.

2. Learning maximal structure fuzzy rules

In [4], Castro et al. present a strategy to learn Multiple Input Single Output (MISO) systems ($\Omega : \mathscr{X}^n \rightarrow \mathscr{Y}$) from a set of examples $\theta = \{e^1, \dots, e^m\}$. Each example has the form $e^i = ([x_1^i, \dots, x_n^i], y^i)$, where x_j^i is the value of the jth input variable and y^i is the output value of the system. The identified model will be represented by means of maximal rules with the form:

$$R^i : \text{if } X_1 \text{ is } SX_1^i \text{ and} \dots \text{and } X_n \text{ is } SX_n^i \text{ then } Y \text{ is } LY^i, \tag{1}$$

where each SX_j^i is a set of labels associated disjunctively with the jth input variable and taken from their respective fuzzy domain $\widetilde{\mathscr{X}}_j = \{LX_{j,1}, \dots, LX_{j,t_j}\}$, and LY^i is the label associated with the output variable and taken from its fuzzy domain $\widetilde{\mathscr{Y}} = \{LY_1, \dots, LY_{t_{n+1}}\}$.

The learning algorithm proposed in [1] is:

1. Transform the examples into initial rules.
2. For each initial rule do:
   - 2.1. If the rule does not subsume into any definitive rule:
     - 2.1.1. For each label in each input variable do:
       - 2.1.1.1. If the amplification of the rule is possible, amplify it.
     - 2.1.2. Store the amplified rule in the set of definitive rules.

The translation from examples into initial rules consists in associating each value x_j^i and y^i with the label that presents the highest membership degree out of all contained in its respective fuzzy domain. Amplifying a rule consists in adding a label to one of its premises. An amplification from R^i to $R^{i'}$ is possible if $R^{i'}$ does not conflict with any initial rule, that is, if there is no initial rule R^j such that $SX_k^j \in SX_k^{i'}$ (for all k) and $LY^j \neq LY^{i'}$.

3. Adding exceptions to fuzzy rules

In the above algorithm, the search of maximal rules may provoke different consequents to coexist in some input fuzzy regions. Next, a strategy is proposed to solve these conflicts.

Table 1
A two-inputs/one-output fuzzy model

		X_1		
		N	**Z**	**P**
X_2	**N**	Z	Z	Z
	Z	Z	Z	P
	P	N	P	P

During the learning process, the information contained in the examples is used only for the extraction of the initial rules. From that moment, the process of amplification of a rule to a certain input fuzzy region only verifies whether this region is occupied with an initial rule. Therefore, this process ignores the information that could be contained in the training set about that region. The basis that will support the approach proposed here to solve the conflicts consists in taking advantage of this information.

A compound rule of the form presented in (1) is equivalent to a conjunction of simple rules with one label associated to each input variable. Therefore, the set of simple rules involved in a conflict can be isolated in order to select one of them based on a certain criterion. With that aim, a certainty degree for each simple rule involved in the conflict will be calculated from the number of positive and negative examples that each rule presents in the training set. An example of this type of measure is proposed in [6], where the concepts *positive example* and *negative example* are defined by means of fuzzy sets and where the certainty degree ranges from 0 to 1.

However, it must be noted that the main goal of the amplification is for the amplified rules to be as general as possible. Because of that, the finally obtained consequents in an input subspace does not assure that these consequents are the best, since they proceed from initial rules that can be far away from the subspace under consideration.

Therefore, when solving a conflict, although it must be tried to restrict the selection of the best consequent among those involved in the conflicting rules in order to obtain maximal rules, it seems desirable to extend the space of selection if none of those conflicting rules have a sufficient degree of certainty. For that reason, a threshold μ will be established over the certainty degree in order to decide when the search of the best consequent in the conflicting region should be extended to all the possible consequents.

Once the best rule is selected, it is necessary to modify the rest of compound rules involved in the conflict. In this respect, when several conflicting rules take the highest certainty degree, the consequent with the highest number of occurrences will be selected (notice that it can exist more than one rule with the same consequent among the rules in conflict). This strategy tries to reduce the number of compound rules to be modified as much as possible.

The procedure to modify compound rules consists in adding exceptions to them. An exception is an n-tuple of labels $[LX_{1,i_1},\dots,LX_{n,i_n}]$ that defines the fuzzy region of the input subspace where the compound rule is not applied.

The use of exceptions entails an improvement in the model expressiveness with respect to the traditional description methods. This fact can be observed in the example in Table 1, where a fuzzy model is shown and where the fuzzy domain of every variable is $\{N,Z,P\}$. The number of simple rules describing the model is $3\times 3=9$ rules. A description using the usual technique that associates an input subspace with the same output (consequent) to the antecedent of each rule gives the following five fuzzy rules:

R^1: if X_2 is $\{N\}$ then Y is Z;
R^2: if X_1 is $\{N,Z\}$ and X_2 is $\{Z\}$ then Y is Z;
R^3: if X_1 is $\{P\}$ and X_2 is $\{Z\}$ then Y is P;
R^4: if X_1 is $\{Z,P\}$ and X_2 is $\{P\}$ then Y is P;
R^5: if X_1 is $\{N\}$ and X_2 is $\{P\}$ then Y is N.

However, the same model can be described with only four rules using exceptions:

R^1: if X_2 is $\{N,Z\}$ then Y is Z,
except if X_1 is P and X_2 is Z;
R^2: if X_1 is $\{P\}$ and X_2 is $\{Z\}$ then Y is P;
R^3: if X_1 is $\{Z,P\}$ and X_2 is $\{P\}$ then Y is P;
R^4: if X_1 is $\{N\}$ and X_2 is $\{P\}$ then Y is N

which is equivalent to

R^1: if X_2 is $\{N,Z\}$ then Y is Z;
except if X_1 is P and X_2 is Z then Y is P;
R^2: if X_1 is $\{Z,P\}$ and X_2 is $\{P\}$ then Y is P;
R^3: if X_1 is $\{N\}$ and X_2 is $\{P\}$ then Y is N.

Therefore, the proposed method to solve conflicts is finally described with the following algorithm:

1. For each fuzzy region of the input space where two or more consequents coexist do:
   - 1.1. Work out the certainty degrees of the simple rules involved in the conflict and select the highest, w_{max}.
   - 1.2. If w_{max} does not reach a threshold μ and there exists a rule among the rest of possible rules with a certainty degree higher than w_{max}, select it as the best rule (adding a new compound rule).
   - 1.3. Otherwise, select a rule among the conflicting rules as follows:
     - 1.3.1. If there are more than one different rule with the highest certainty degree, w_{max}, among the conflicting rules, select the one appearing more times in the conflicting region. If all appear the same times, select one of them randomly.
     - 1.3.2. Otherwise, select the rule that has the highest certainty degree, w_{max}.
   - 1.4. Delete each simple rule different from the selected one.
   - 1.5. For each deleted simple rule, form the appropriate exception and add it to its respective compound rule.

It must be noted that the conflict resolution could cause that a compound rule that overlaps in a great extent with another rule with the same consequent ends up covering mainly the redundant part of the latter (when exceptions are added to the remaining regions). In order to avoid this problem, which will provide less interpretable rules, a new restriction is added to the amplification condition: an amplification will be possible only if the addition of the correspondent label to the antecedent of the rule allows to cover some region still not covered by any definitive rule with the same consequent. In this way, the overlapping among rules is restricted.

4. Improving the interpretability

The model generated with the algorithm described in the previous section can still improve its interpretability in different ways. Next, several strategies are described to achieve that goal.

4.1. Merging fuzzy rules

In the algorithm presented in Section 3, a rule is added to the set of definitive rules when the selected rule is not one of the conflicting rules (Step 1.2). This can lead to a considerable increase in the number of rules with respect to the one obtained by the identification algorithm. In order to minimize this increase, after the addition of a new rule it should be tried to merge that rule with any of the existing compound rules.

Proposition 1. *A rule* $R^i : SX_1^i, \dots, SX_n^i \to LY^i$ *with exceptions* $E^i = \{E_1^i, \dots, E_{p_i}^i\}$ *could be merged with another rule* $R^j : SX_1^j, \dots, SX_n^j \to LY^j$ *with exceptions* $E^j = \{E_1^j, \dots, E_{p_j}^j\}$ *if the following is fulfilled*:

1. $LY^i = LY^j$.
2. *There exists an* r *so that* $SX_r^i \neq SX_r^j$.
3. $SX_s^i = SX_s^j$, *for all* $s \neq r$.

resulting a rule $R^* : SX_1^i, \dots, (SX_r^i \cup SX_r^j), \dots, SX_n^i \to LY^i$ *with exceptions* $E^* = E^i \cup E^j$.

The following algorithm describes the method for merging rules and will be called after every addition of a new rule during the conflict resolution. It is a recursive method, since the merged rule could satisfy the merging condition with respect to some other rule existing in the rule base:

1. Given the compound rule trying to be merged $R^i : SX_1^i, \dots, SX_n^i \to LY^i$ with exceptions $E^i = \{E_1^i, \dots, E_{p_i}^i\}$.
2. If there is another rule $R^j : SX_1^j, \dots, SX_n^j \to LY^j$ with exceptions $E^j = \{E_1^j, \dots, E_{p_j}^j\}$ in the set of definitive rules, which can be merged with R^i:
   - 2.1. Replace the rules R^i and R^j by the rule $R^* : SX_1^i, \dots, (SX_r^i \cup SX_r^j), \dots, SX_n^i \to LY^i$ with exceptions $E^* = E^i \cup E^j$.
   - 2.2. Try to merge R^*.

4.2. Reducing fuzzy rules

The interpretability of the rules can increase if the exceptions of a rule are reduced by deleting labels from its antecedent. For example, the rule

if X_1 is $\{N, P\}$ and X_2 is $\{N, Z, P\}$ then Y is Z
except if X_1 is N and X_2 is Z or if X_1 is P and X_2 is Z,

could be reduced to the rule

$$\text{if } X_1 \text{ is } \{N,P\} \text{ and } X_2 \text{ is } \{N,P\} \text{ then } Y \text{ is } Z.$$

The following algorithm determines if a reduction can be carried out after the addition of a new exception and, if that is the case, accomplishes it.

1. Given the rule $R^i : SX_1^i, \ldots, SX_n^i \rightarrow LY^i$ with exceptions $E^i = \{E_1^i, \ldots, E_p^i\}$, where the new exception added to that rule is $E_p^i = [LX_{1,p_1}, \ldots, LX_{n,p_n}]$.
2. For each d from 1 to n do:
   2.1. Set up a set of exceptions E^* taking LX_{d,p_d} in the dth element of every exception and taking the different combinations of the labels from $SX_1^i, \ldots, SX_{d-1}^i, SX_{d+1}^i, \ldots,$ SX_n^i in the rest of elements. That is, $E^* = SX_1^i \times \cdots \times SX_{d-1}^i \times LX_{d,p_d} \times SX_{d+1}^i \times \cdots \times SX_n^i$.
   2.2. If $E^* \subseteq E^i$ then set E^i to $E^i - E^*$ and SX_d^i to $SX_d^i - \{LX_{d,p_d}\}$.
3. If the reduced rule subsumes into some other compound rule, delete it.
4. Otherwise, try to merge it.

It must be noted that the merging of a rule could be presented after its reduction, since the merging condition could be satisfied when the rule loses a label in its antecedent. Because of that, this merging will be tried in the Step 4 of the above algorithm, once it has been verified that the rule does not subsume into any other rule.

4.3. Merging exceptions

Until now, exceptions have been described as n-tuples of labels that define fuzzy regions in the input space similar to the ones defined by the antecedents of the simple rules. Therefore, the exceptions expressed in that way can be considered *simple exceptions*.

Trying to increase the model interpretability, the concept of compound rule can be translated to the representation of exceptions, giving rise to *compound exceptions*. Thus, a compound exception can be defined as an n-tuple $E_i = (SE_{i,1}, \ldots, SE_{i,n})$, where $SE_{i,k} \subseteq \widetilde{\mathscr{X}}_k$.

In order to obtain a description as compact as possible by means of exceptions, it is necessary to state a mechanism for merging exceptions similar to that one used for merging rules explained in Section 4.1. The following proposition establishes the conditions that must be satisfied by two compound exceptions in order to be merged.

Proposition 2. *A compound exception $E_i = (SE_{i,1}, \ldots, SE_{i,n})$ could merge with another one $E_j = (SE_{j,1}, \ldots, SE_{j,n})$ if the following is fulfilled*:

1. *There exists an r so that $SE_{i,r} \neq SE_{j,r}$.*
2. *$SE_{i,s} = SE_{j,s}$, for all $s \neq r$.*

The result of the merging will consist of a new exception with the form $E^* = (SE_{i,1}, \ldots, (SE_{i,r} \cup SE_{j,r}), \ldots, SE_{i,n})$.

The following algorithm describes the method for merging exceptions. It must be noted that, whereas rules are merged on-line (i.e., during the process of solving conflicts), exceptions will be merged off-line (i.e., once the final exceptions of every rule have been obtained). This is due to the use of simple exceptions in the reduction of rules.

1. Given the set of exceptions $E=\{E_1,\dots,E_p\}$ and the exception trying to be merged $E_i=(SE_{i,1},\dots,SE_{i,n})$.
2. If there exists a $j\neq i$, so that it is possible to merge E_i and E_j:
   2.1. Replace the exceptions E_i and E_j by the exception $E^*=(SE_{i,1},\dots,(SE_{i,r}\cup SE_{j,r}),\dots,SE_{i,n})$.
   2.2. Try to merge E^*.

This recursive algorithm will be called iteratively for every rule, while exception merging is possible.

5. Reducing the number of rules and conflicts

The identification algorithm proposed in Section 2 tries to obtain rules as general as possible by amplifying the initial rules. However, each initial rule (i.e., each example) can be amplified in different ways depending on the order considered for taking the input variables during the amplification process.

The original identification algorithm does not consider any criterion for selecting the specific compound rule built up from each initial rule and then, there is no guarantee that this compound rule is the best or even a good amplification. This fact could lead to an unnecessary increase in the amount of rules used to describe the system and, related with it, in the number of conflicts to be solved.

In the present section a heuristic-based solution is presented to avoid the aforementioned problem. In a first stage, the proposed approach generates a set of candidate compound rules for each initial rule, instead of generating just one rule. In a second stage, a greedy algorithm will select iteratively the definitive rules from the global candidate rule set (CRS).

5.1. Generating candidate compound rules

As mentioned above, the candidate rule resulting from the amplification of an initial rule depends on the order the input variables are taken. For example, in a two-input system, an initial rule can be amplified in two ways: amplifying the first input variable and then the second one, or inversely. This could lead to two different amplified rules, as can be seen in the example shown in Table 2, where the initial rule $Z,Z\to P$ can be amplified as $\{N,Z,P\},\{Z\}\to P$ or as $\{Z\},\{N,Z,P\}\to P$. In fact, the resulting model identified from the initial rules in Table 2(a) (the superscripts read the order in which rules are amplified) will be described with the rule base (Table 2(b)):

R^1: if X_1 is $\{N,Z\}$ and X_2 is $\{N,P\}$ then Y is Z;
R^3: if X_2 is $\{Z\}$ then Y is P;
R^4: if X_1 is $\{Z,P\}$ and X_2 is $\{N\}$ then Y is P;
R^5: if X_1 is $\{Z,P\}$ and X_2 is $\{P\}$ then Y is N;

Table 2
Amplification alternatives: (a) initial rules; (b) amplification with input selection order $\{X_1,X_2\}$; (c) amplification with input selection order $\{X_2,X_1\}$; (d) amplification in several orders

		N	Z	P
	N	Z^1		P^4
X_2	**Z**		P^3	
	P	Z^2		N^5

(a)

		X_1		
		N	**Z**	**P**
	N	Z^1	Z^1/P^4	P^4
X_2	**Z**	P^3	P^3	P^3
	P	Z^1	Z^1/N^5	N^5

(b)

		X_1		
		N	**Z**	**P**
	N	Z^1	P^3	P^4
X_2	**Z**	Z^1	P^3	P^4/N^5
	P	Z^1	P^3	N^5

(c)

		X_1		
		N	**Z**	**P**
	N	Z^1	$P^{3'}$	$P^{3'}$
X_2	**Z**	Z^1	$P^{3'}$	$P^{3'}$
	P	Z^1	N^5	N^5

(d)

or with the rule base (Table 2(c)):

R^1 : if X_1 is $\{N\}$ then Y is Z;
R^3 : if X_1 is $\{Z\}$ then Y is P;
R^4 : if X_1 is $\{P\}$ and X_2 is $\{N,Z\}$ then Y is P;
R^5 : if X_1 is $\{P\}$ and X_2 is $\{Z,P\}$ then Y is N;

depending on the order in which the input variables are taken for the amplification. The first case corresponds with the order $\{X_1,X_2\}$, giving four rules—one of them with only a premise in the antecedent—and two conflicts. The second case corresponds with the order $\{X_2,X_1\}$, giving four rules—two of them with only one premise in the antecedent—and only one conflict.

Moreover, it is possible to obtain a more interpretable model if combining both input selection orders and allowing to select properly the order in which the initial rules are amplified. It can be observed in the rule base shown in Table 2(d) and described as:

R^1 : if X_1 is $\{N\}$ then Y is Z;
$R^{3'}$: if X_1 is $\{Z,P\}$ and X_2 is $\{N,Z\}$ then Y is P;
R^5 : if X_1 is $\{Z,P\}$ and X_2 is $\{P\}$ then Y is N

which employs only three rules without conflicts if the order $\{X_2,X_1\}$ is used for the rule $N,N \to Z$, the order $\{X_1,X_2\}$ is used for the rule $P,P \to N$, and the rule $P,N \to P$ is amplified before the rule $Z,Z \to P$.

The number of candidate rules associated with an initial rule depends on the number of input variables. In our approach we will generate all possible candidate rules for each initial rule, setting up a CRS out of which the best rules will be selected during the second stage.

With this aim, the following rule amplification algorithm was developed:

1. Let $\{X_{i_1},\dots,X_{i_n}\}$ the set of input variables to be considered in the amplification of a rule R.

2. For each input variable X_j in $\{X_{i_1},\dots,X_{i_n}\}$ do:
   2.1. For each label in the fuzzy domain of X_j do:
      2.1.1. If the amplification of the rule is possible, amplify it.
   2.2. If $n=1$ then add the compound rule to CRS.
   2.3. Otherwise, call recursively to the algorithm in order to amplify again R, now over the set of input variables $\{X_{i_1},\dots,X_{i_n}\}-X_j$.

The recursive design of the algorithm tries to reduce its computational cost, since each partial amplification of a rule is preserved for all the amplifications stemming from it. This algorithm will be called for each initial rule, using all the input variables as the set of variables to be amplified (i.e., $\mathbf{X}=\{X_1,\dots,X_n\}$). It will substitute from the Steps 2.1 to 2.1.2 in the original algorithm presented in Section 2, resulting the following new *partial* identification algorithm:

1. Transform the examples into initial rules.
2. For each initial rule do:
   2.1. Call to the rule amplification algorithm for that rule over the set of input variables $\{X_1,\dots,X_n\}$.

This algorithm generates the CRS and must be completed with the second stage that selects the definitive rule set (DRS) from the CRS.

However, the restrictive amplification introduced in Section 3 to avoid an excessive overlap among rules will be now postponed to the second stage. This is because the restriction is based on the comparison of the amplified rule with the definitive ones and, while generating the CRS, no rule is still definitive. Therefore, the CRS will be obtained considering the original definition of possible amplification (Section 2) in the Step 2.1.1 of the above amplification algorithm.

5.2. Selecting definitive rules

Once the CRS is generated, a greedy strategy is proposed in order to select the definitive rules. The approach lies in the iterative selection of the best rule among the remaining candidate rules based on some goodness measure. After the selection of each definitive rule, this rule and all the ones generated from initial rules covered by it will be deleted from the CRS. Next, it will be identified all the initial rules that generated compound rules whose amplification would have been restricted in view of the new definitive rule (accordingly with the definition of restrictive amplification stated in Section 3), generating again the candidate rules from these initial rules but now considering the restrictive amplification. Finally, the goodness of each remaining candidate rule in the CRS will be re-evaluated in order to select the next definitive rule. This process will be repeated until the CRS becomes empty.

Three criteria will be used to evaluate the goodness of a candidate rule:

1. *Maximal number of covered initial rules*: This criterion tends to reduce the number of definitive rules, trying to select rules justifying as many examples as possible, that is, covering as many initial rules as possible.
2. *Minimal number of conflicts with definitive rules*: This criterion tends to reduce the number of conflicts that the selected candidate rule generates over the already selected definitive rules, and is defined as the sum of common input regions between the candidate rule and each definitive rule (initially, this value will be zero for all the candidate rules).

3. *Minimal number of labels in the antecedent of the rule*: It rewards the readability of the rules, searching for antecedents as simple as possible. This criterion will consider a premise as "void" if the input variable is associated with all its fuzzy domain, since that premise can be removed from the rule. Concretely, given a rule R^i, the equation will be

$$\sum_{j=1}^{n} remainder\left(||SX_j^i||/p_j\right),$$

where $||SX_j^i||$ is the number of labels in the premise of the variable X_j and p_j is the number of labels in the fuzzy domain $\widetilde{\mathscr{X}}_j$.

The importance of each criterion will be considered sequentially and coincide with the order shown above. That is, firstly the candidate rules covering the highest number of initial rules are selected; secondly, the rules that generate the lowest number of conflicts will be selected among them; and thirdly, a rule will be selected from this group having the minimal number of labels in its antecedent. If more than one of such rules exist, one of them is selected randomly. Finally, this selected rule will be added to the DRS.[1]

The complete algorithm to build the DRS from the CRS states as follows:

1. While *CRS* is not empty do:
   1.1. For each rule in *CRS* evaluate criterion 1 and select the group CRS_1 with rules having a maximum value.
   1.2. For each rule in CRS_1 evaluate criterion 2 and select the group CRS_2 with rules having a maximum value.
   1.3. For each rule in CRS_2 evaluate criterion 3 and select the group CRS_3 with rules having a maximum value.
   1.4. If CRS_3 contains more than one rule, select one of them R randomly, otherwise, let R be the rule in CRS_3.
   1.5. Add the rule R to *DRS*.
   1.6. Delete from *CRS* the rule R and the rules stemming from initial rules covered by R.
   1.7. Regenerate the candidate rules obtained from initial rules affected by the restrictive amplification due to the new definitive rule R.

This algorithm will be run after obtaining the CRS proposed in the previous subsection and therefore must be included as Step 3 in the partial identification algorithm presented in that subsection. Once the DRS is obtained, the algorithm for conflict resolution presented in Section 3 will be applied.

6. Experimental results

The proposed methods were applied to the approximation of several functions with different complexity (Fig. 1):

$$f_1 : [-1,1] \times [-1,1] \to [-1,1],$$

[1] Although the third criterion seems to be contrary to the maximality of the rules, due to the order each criterion is considered, it will be effective only when several rules cover the same number of initial rules and, in this case, the simplicity of the antecedents must be favoured more than an unnecessarily wider covering of the input space.

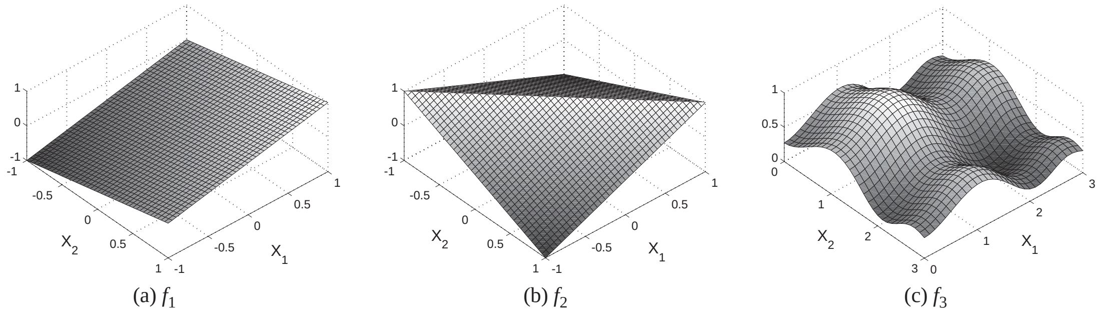


Fig. 1. Output surfaces for functions f_1, f_2, and f_3.

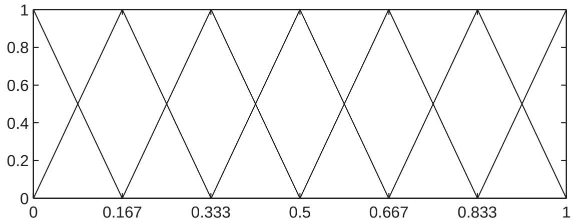


Fig. 2. Normalized fuzzy domain for all the variables in every function.

$$f_1(x_1,x_2) = \frac{x_1 + x_2}{2},$$

$$f_2 : [-1,1] \times [-1,1] \to [-1,1],$$
$$f_2(x_1,x_2) = 1 - |x_1 - x_2|,$$

$$f_3 : [0,3] \times [0,3] \to [0,1],$$
$$f_3(x_1,x_2) = (\sin(x_1^2)\mathrm{e}^{-x_1} + \sin(x_2^2)\mathrm{e}^{-x_2} + c_1)/c_2,$$

where $c_1 = 0.2338$ and $c_2 = 0.8567$ restricts the values of f_3 to the interval $[0,1]$.

The normalized fuzzy domains of the input and output variables were defined as shown in Fig. 2 for all the functions, and appropriate scale factors were used in each case in order to translate that normalized domain into the real domain.

In order to analyse the benefits of the methods proposed here, three different versions of the identification method for learning maximal structural fuzzy rules were considered:

- *LMSFR*: The algorithm presented in Section 2 for learning maximal structure fuzzy rules with conflicts, that is, the original algorithm proposed in [1].
- *LMSFRWE*: The extension of LMSFR proposed in Sections 3 and 4 for solving conflicts and improving the interpretability.
- *LMSFRWE+*: The extension of LMSFRWE proposed in Section 5 for reducing the number of rules and conflicts.

Table 3
Interpretability results

Function	Method	Training set size								
		20			50			100		
f_1	LMSFR	11.1	(–)	[6.9]	18.1	(–)	[5.5]	21.0	(–)	[4.4]
	LMSFRWE	12.4	(9.2)	[4.6]	17.8	(4.5)	[3.6]	19.9	(1.6)	[3.3]
	LMSFRWE+	**11.0**	(8.7)	[4.8]	**16.6**	(5.4)	[3.7]	**19.1**	(1.9)	[3.3]
	W&M	16.2	(–)	[2.0]	30.2	(–)	[2.0]	40.4	(–)	[2.0]
f_2	LMSFR	**12.3**	(–)	[6.9]	21.9	(–)	[5.2]	28.3	(–)	[3.9]
	LMSFRWE	14.3	(9.2)	[4.3]	23.0	(6.2)	[3.2]	28.3	(2.1)	[2.7]
	LMSFRWE+	13.0	(9.4)	[4.5]	**21.7**	(7.8)	[3.3]	**27.5**	(3.3)	[2.8]
	W&M	16.0	(–)	[2.0]	30.3	(–)	[2.0]	40.9	(–)	[2.0]
f_3	LMSFR	11.2	(–)	[7.1]	18.6	(–)	[5.5]	21.3	(–)	[4.5]
	LMSFRWE	12.7	(8.7)	[4.5]	18.5	(5.7)	[3.6]	20.0	(2.0)	[3.3]
	LMSFRWE+	**11.1**	(9.0)	[4.8]	**16.3**	(6.1)	[3.8]	**18.6**	(2.5)	[3.4]
	W&M	16.0	(–)	[2.0]	30.4	(–)	[2.0]	40.3	(–)	[2.0]

Besides, in order to analyse the identification performances with respect to other identification algorithms, we considered the well-known method proposed by Wang and Mendel in [10], which is widely used in the literature for comparison purposes (e.g., [1,4,5]).

The identification processes were carried out with three different training set sizes: 20, 50, and 100 randomly generated examples. For each algorithm and each training set size, 100 runs were performed and the averaged results were obtained.

In order to analyse both the interpretability and the accuracy of the resulting fuzzy models, two different measures were used. On the one hand, the number of rules and exceptions describing the model together with the averaged number of labels in each antecedent were considered for evaluating its interpretability. On the other hand, the mean square error (MSE) between the model and the system outputs was obtained for accuracy evaluation, using the equation

$$MSE = \frac{\sum_{i=1}^{N} [y - \hat{y}]^2}{N}$$

for $N = 2500$ test examples randomly generated.

Table 3 summarizes the interpretability results, which are averaged values over the 100 runs. They are depicted as the averaged number of rules describing the model, the averaged number of exceptions in parenthesis, and the averaged number of labels in the antecedent of the rules in brackets. An intermediate certainty degree threshold μ equal to 0.5 was considered for conflict resolutions.

Firstly, it can be observed that, in some cases, the LMSFRWE increases slightly the averaged number of rules of the model with respect to the original LMSFR method. Nevertheless, although this increase is due to the addition of rules during the conflict resolution, it is not very significant because of the rule merging algorithm presented in Section 4.2, which allows some of these rules to be merged with other rules. In fact, this merging algorithm even allows to reduce the number

Table 4
Accuracy results

Function	Method	Training set size		
		20	50	100
f_1	LMSFR	0.0285	0.0160	0.0076
	LMSFRWE	**0.0153**	**0.0044**	**0.0021**
	LMSFRWE+	0.0154	0.0049	0.0023
	W&M	0.0247	0.0051	0.0022
f_2	LMSFR	0.0561	0.0360	0.0203
	LMSFRWE	**0.0329**	0.0104	0.0038
	LMSFRWE+	0.0352	0.0111	0.0039
	W&M	0.0389	**0.0093**	**0.0032**
f_3	LMSFR	0.0315	0.0183	0.0102
	LMSFRWE	**0.0255**	**0.0116**	0.0080
	LMSFRWE+	0.0266	0.0117	0.0079
	W&M	0.0389	0.0138	**0.0078**

of rules obtained with LMSFR method sometimes. Moreover, the averaged number of labels in the antecedent of the rules is also significantly decreased with the conflict resolution, due to the rule reduction algorithm.

Regarding exceptions added to the model by LMSFRWE, at first sight they could look like a loss of interpretability in comparison with a model with the same number of rules but without exceptions, due to the higher number of input regions that must be considered. However, they indeed improve the interpretability of the model in two ways: on one hand, they remove the contradictions contained in the models, and, on the other hand, they can replace other rules of the model (as was shown in the example in Section 3) and, in such cases, they will not increase the number of input regions to be considered.

Secondly, it can be observed the reduction in the number of rules achieved by LMSFRWE+ method when compared with the LMSFRWE method, while keeping the simplicity of the antecedents. This reduction even allows to outperform the results obtained by the LMSFR method in 11 out of 12 cases, achieving a more compact description of the model. However, a slight increase in the averaged number of exceptions can also be observed.

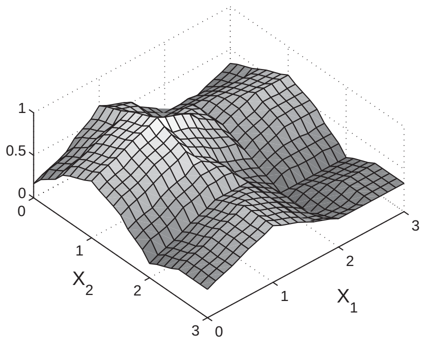
Finally, both identification methods proposed in this paper outperform the number of rules generated by Wang and Mendel's algorithm, especially when large training set sizes are used. Obviously, the number of labels in the rules generated by Wang and Mendel's algorithm is always the same, since those rules are simple rules. Nevertheless, it must be noted that, most of the cases, Wang and Mendel's algorithm provides an incomplete rule base (the averaged number of rules is significantly below $7 \times 7 = 49$ rules), which does the model correctness to be dependent on the strategy for dealing with the holes in the rule base (see below).

In Table 4 the accuracy results are shown. The superiority of the LMSFRWE and LMSFRWE+ methods over the original LMSFR method is clear, confirming the reduction in the model error achieved by the resolution of conflicts. Concerning Wang and Mendel's method, although it reaches similar accuracy results to the ones achieved by LMSFRWE and LMSFRWE+ methods when using

X_1

	XXS	**XS**	**S**	**M**	**L**	**XL**	**XXL**
XXS	XS	S	L	M	XS	XS	S
XS	M	L	XL	M	M	M	M
S	L	XL	XXL	XL	M	L	L
M	M	L	XL	M	S	M	M
L	XS	S	M	M	XS	XS	XS
XL	S	M	L	M	XS	S	S
XXL	S	M	L	M	S	S	S

(a)



(b)

Fig. 3. An example of a final model without conflicts.

large training set sizes, these are precisely the cases where Wang and Mendel's algorithm presents the worst interpretability results.[2]

An example of the fuzzy models obtained by the method LMSFRWE+ for a training set with 50 examples is:

$R^1: \{XXS, L, XL, XXL\}, \{XXS, L\} \rightarrow XS$ excepting $\{XXL\}, \{XXS\}$;
$R^2: \{L\}, \{XL\} \rightarrow XS$;
$R^3: \{XS\}, \{XXS, L\} \rightarrow S$;
$R^4: \{XXS, XL, XXL\}, \{XL, XXL\} \rightarrow S$;
$R^5: \{L\}, \{M, XXL\} \rightarrow S$;
$R^6: \{XXL\}, \{XXS\} \rightarrow S$;
$R^7: \{L\}, \{XS, S\} \rightarrow M$;
$R^8: \{XS, M\}, \{XL, XXL\} \rightarrow M$;
$R^9: \{S, M\}, \{XXS, L\} \rightarrow M$ excepting $\{S\}, \{XXS\}$;
$R^{10}: \{XXS, M, XL, XXL\}, \{XS, M\} \rightarrow M$;
$R^{11}: \{XS\}, \{XS, M\} \rightarrow L$;
$R^{12}: \{XXS, XL, XXL\}, \{S\} \rightarrow L$;
$R^{13}: \{S\}, \{XXS, XL, XXL\} \rightarrow L$;
$R^{14}: \{S\}, \{XS, M\} \rightarrow XL$;
$R^{15}: \{XS, M\}, \{S\} \rightarrow XL$;
$R^{16}: \{S\}, \{S\} \rightarrow XXL$;

where linguistic labels are $\{XXS, XS, S, M, L, XL, XXL\}$ for all the fuzzy domains. The tabular form of such a model is shown in Fig. 3(a) and its output surface is shown in Fig. 3(b), providing an

[2] Besides, it must be noted that the incompleteness of the rule base provided by Wang and Mendel's method could affect seriously the model accuracy depending on the strategy for dealing with holes and the system to be identified. In this paper, a random output is provided only when no rule is fired. Whereas this strategy will favour the model accuracy for smooth systems (the surrounding rules will be fired when an input lies in a hole), such strategy—and any other—could be inappropriate for any other system.

MSE equal to 0.0052. This model uses only 16 rules with two exceptions, whereas it will need 49 simple rules. Moreover, since the antecedent of rule R^6 coincides with the exception in R^1, this rule could be included as an extension of the exception as follows:

$$R^1 : \{XXS, L, XL, XXL\}, \{XXS, L\} \rightarrow XS$$
$$\text{excepting}\{XXL\}, \{XXS\} \rightarrow S$$

which increases the compactness of the model description further.

7. Conclusions

The present paper proposes a method for solving conflicts in the framework of fuzzy model identification with maximal rules. The resolution of conflicts uses the information contained in the training set by defining a certainty degree for every conflicting rule.

Besides, the inclusion of exceptions is proposed as the method for representing the resolution of conflicts, resulting in a more compact description of the model. Furthermore, several strategies are proposed that increase the model interpretability, such as reduction of rules, merger of exceptions and merger of rules.

Finally, a heuristic is proposed in order to reduce the number of rules and conflicts generated when obtaining the maximal rules, with the aim to prevent the selection of an inappropriate set of general rules.

Acknowledgements

This work has been supported by Research Project PB98-1379-C02-01.

References

[1] R. Alcalá, J. Casillas, O. Cordón, F. Herrera, Improvement to the cooperative rules methodology by using the ant colony system algorithm, Mathware Soft. Comput. 8 (3) (2001) 321–335.
[2] R. Babuška, Fuzzy Modeling for Control, Kluwer Academic Publishers, Boston, MA, USA, 1998.
[3] J. Castro, Fuzzy logic controllers are universal approximators, IEEE Trans. Systems, Man Cybernet. 25 (4) (1995) 629–635.
[4] J. Castro, J. Castro-Schez, J. Zurita, Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems, Fuzzy Sets and Systems 101 (1999) 331–342.
[5] I. Chung, C. Lin, C. Lin, A GA-based fuzzy adaptive learning control network, Fuzzy Sets and Systems 112 (2000) 65–84.
[6] A. González, R. Pérez, SLAVE: A genetic learning system based on the iterative approach, IEEE Trans. Fuzzy Systems 7 (2) (1999) 176–191.
[7] H. Hellendorn, D. Driankov (Eds.), Fuzzy Model Identification, Springer, Berlin, Germany, 1997.
[8] B. Kosko, Fuzzy systems as universal approximators, in: Proc. 1st IEEE Internat. Conf. Fuzzy Systems, 1992, pp. 1153–1162.
[9] H. Nguyen, M. Sugeno, Fuzzy Modeling and Control, Kluwer Academic Publishers, Boston, MA, USA, 1998.
[10] L. Wang, J. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Systems, Man Cybernet. 22 (6) (1992) 1415–1427.