Mathware & Soft Computing 13 (2006) 203-216

Linguistic Knowledge Base Simplification Regarding Accuracy and Interpretability

J.M. Alonso[1], L. Magdalena[1,2] and S. Guillaume[3]
[1] ETSI Telecomunicación, UPM,
Ciudad Universitaria, 28040 Madrid, Spain
[2] European Centre for Soft Computing. Edif. Científico-Tecnológico,
C/. Gonzalo Gutiérrez Quirós s/n. 33600 Mieres, Asturias, Spain
[3] Cemagref Montpellier, BP 5095, 34033 Montpellier Cedex, France
{*jmam,llayos*}*@mat.upm.es, luis.magdalena@softcomputing.es*
serge.guillaume@montpellier.cemagref.fr

Abstract

This work proposes a new method in order to simplify linguistic knowledge bases. The main goal consists of improving simultaneously accuracy and interpretability when it is possible, or at least ensuring a good trade-off between them, as well as consistency of the final knowledge base. It is used with linguistic rules which can be defined by expert, induced from data, or both of them. The simplification process is applied to the well known wine classification problem. The results are encouraging.

Keywords: Simplification, linguistic knowledge bases, interpretability-accuracy trade-off, expert and induced knowledge.

1 Introduction

The strength of Fuzzy Inference Systems (FIS) relies on their twofold identity: On one hand they are able to handle linguistic concepts; on the other hand they are universal approximators able to perform non linear mappings between inputs and outputs. These two characteristics have been used to design two kinds of FIS. The first kind of FIS to appear was focused on the ability of fuzzy logic to model natural language [12]. These FIS contain fuzzy rules built from expert knowledge, they can be seen as a fuzzy extension of expert systems. Sugeno [15] was one of the first to propose self learning FIS and to open the way to a second kind of FIS, those designed from data. As this field of fuzzy logic has become very popular, a lot of methods are available [6, 10].

As expert knowledge and induced knowledge from data are complementary, combining these two approaches may lead to more accurate systems. To make the cooperation efficient, induced rules must be as interpretable as expert rules are. The three conditions for a rule base to be interpretable have been stated in [6]:

1. Interpretable fuzzy partitions.
2. A small number of rules.
3. Incomplete rules for large systems. A fuzzy rule is incomplete if its premise is defined by a subset of the input variables. The number of variables used by rule is very important regarding interpretability. Understand a rule with only one input variable is very easy. Rule analysis is still easy when two variables are involved because the rule can be described within a table. Nevertheless, when the number of variables increases, the task becomes harder, the growth of complexity is not linear. Large systems have lots of inputs and as a result the use of complete rules yields quite poor interpretability.

The cooperation framework has been proposed in [9]. The process consists of next steps:

1. Defining a common universe for each of the variables according to both expert knowledge and data distribution.
2. Describing the system behavior. The expert is invited to express his/her system knowledge as linguistic rules. Also, rules are induced from data.
3. Integrating both expert and induced rules into the knowledge base. Thanks to the common universe previously defined both types of rules use the same linguistic terms defined by the same fuzzy sets. In consequence rule comparison can be done at the linguistic level. During this last step, the fundamental properties of a rule base have to be guaranteed.

Such an integrated knowledge base would potentially be simplified. This communication deals with the simplification process of the whole knowledge base, both data base and rule base are taken under consideration. The usefulness of this simplification approach is not reduced to such a kind of integrated knowledge bases. In many cases, induced knowledge bases will also reduce its complexity after the simplification process. We assume that the knowledge base is previously built (no matter how it was generated) and this paper goes into detail with the simplification task. Note that the knowledge base generation is not the main topic of our contribution. Please refer to the cited literature for a complete description.

The goal is to design incomplete, more general, rules while checking coherency and avoiding redundancy in the final rule base. Building more general rules, as expert rules usually are, makes the system more robust and more interpretable. The aim is not only to find a balance between accuracy and interpretability [4] but to improve both at the same time.

The structure of the paper is as follows. Section 2 explains how to evaluate the quality of the knowledge base. Section 3 offers a perspective of the overall simplification process. Section 4 explains how to simplify the rule base. Section 5 describes how to reduce the data base. Section 6 shows the application of the simplification process to a well-known problem of wine classification. Finally, section 7 offers some conclusions.

2 Knowledge Base Quality

Quality indices are needed for system evaluation and optimization. The fuzzy system quality can be assessed in terms of accuracy and/or interpretability. Nevertheless, the most difficult task in fuzzy modelling consists of setting a good trade-off between both [4].

In the last few years lots of papers [16] related to interpretability in fuzzy logic based systems have been written. However, interpretability is subjective, it depends on the person who makes the assessment, and its quantification is still an open problem. A first proposal [13] opens this way but a lot of work remains to do in order to find an universal index.

On the other hand, measuring accuracy is relatively easy: The only thing to do is to compare the inferred output, $\hat{y}$, and the observed one, y, in the real system to build an error index. A lot of indices are available in the literature. We use the three following ones:

- **Performance**: For regression cases, it is defined as the root mean of sum of squared errors (eq.1). And for classification cases, it is defined as the number of misclassified items (eq.2).

$$Perf = \frac{1}{N}\sqrt{\sum_{i=1}^{N} ||\hat{y}_i - y_i||^2} \tag{1}$$

$$Perf = \sum_{i=1}^{N} \delta_i, \begin{cases} \delta_i = 1 & if \;\; \hat{y}_i \neq y_i \\ \delta_i = 0 & otherwise \end{cases} \tag{2}$$

- **Max error**: Maximum difference between observed and inferred value.

- **Coverage**: Percentage of examples from data that fires at least one rule with a degree higher than Δ.

As the present paper is focused on classification problems, the next indices are considered:

- **Error cases**: Number of covered cases from data set that produces error, i.e., observed and inferred values are different, in inference. Note that it is a **Performance** index.

- **Ambiguity cases**: Number of covered cases from data set that produces ambiguity, i.e., difference between the two highest confidence levels is smaller than an established threshold.

- **Unclassified cases**: Number of cases from data set that do not fire at least one rule with a degree higher than Δ. Note that this parameter expressed in percentage is the complementary of **Coverage**.

These indices convey complementary information. A good knowledge base should minimize them by offering an accurate (reducing error cases), consistent (reducing ambiguity cases) and complete (reducing unclassified cases) set of rules. Note that **Max error** index is not taken into account. It does not make sense in classification problems disregarding the order of classes, for example a wine classification problem. Of course, it should be used in those problems where the class order is essential. For instance, assessing the risk levels for an insurance policy.

When the simplification process involves a modification in the knowledge base, its accuracy is evaluated in order to confirm or discard the changes. Modifications are saved, only if accuracy does not decrease, i.e., the indices do not increase.

Note that the simplification process can also be applied when no data are available. In this case it is made only at linguistic level, without checking the knowledge base accuracy.

3 Knowledge Base Simplification Process

The main goal of the simplification process is to achieve a more compact knowledge base, with a smaller size in order to increase interpretability, but without getting worse accuracy than the original base. Figure 1 shows a schematic diagram of knowledge base simplification process.

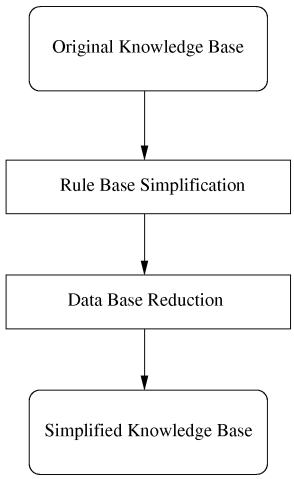

Figure 1: Knowledge Base Simplification

Linguistic knowledge bases are highly interpretable. In order to guarantee this interpretability, two aspects are considered, one for each part of the knowledge base:

1. **Data base**: The use of strong fuzzy partitions [14] satisfies semantic constraints on membership functions in order to respect semantic integrity within

the partitions. As a result, all the fuzzy sets for each variable are interpretable as linguistic terms.

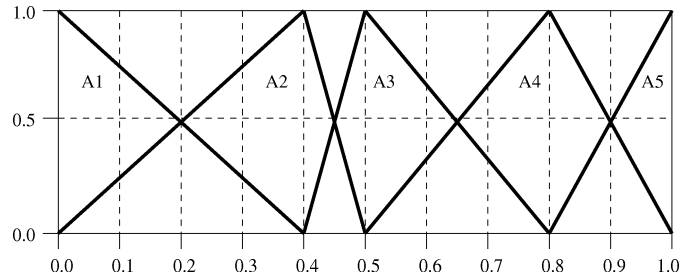

Figure 2: A Strong Fuzzy Partition

Figure 2 shows a strong fuzzy partition with 5 terms. In this kind of partitions, all fuzzy sets are of triangular shape, except at the domain edges where the shape is semi trapezoidal, and they satisfy next conditions:

$$\forall x \in U, \ \sum_{i=1}^{E} \mu_{A_i}(x) = 1 \tag{3}$$

$$\forall A_i \ \exists x, \ \mu_{A_i}(x) = 1 \tag{4}$$

U is the range, E is the number of terms and μ_{A_i} is the membership degree of x to the A_i fuzzy set.

2. **Rule base**: The use of a small number of linguistic rules helps to express the system behavior in an understandable way. Fuzzy linguistic rules are of form **If** *condition* **Then** *conclusion* where both, the premise and conclusion use linguistic terms.

4 Rule Base Simplification

Rule base comprises linguistic rules. Rule premises are made up of couples (input variable, linguistic term), where the absence of an input variable in a rule means that the variable is not considered in the evaluation of the rule. The rule conclusions are also made up of couples (output variable, linguistic term).

The fuzzy partitions include only basic labels, A_i, which correspond with elementary fuzzy sets. Nevertheless in the rule description, for each input variable the user can choose between basic labels (elementary terms) or composite labels (composed by elementary terms linked through disjunction or negation operators). The implementation of OR (disjunction) and NOT (negation) operators is explained in [8]. The OR composite labels are defined as the convex hull of the combined terms. And the NOT composite labels are defined by equation 5. Note that AND is not

Table 1: Linguistic terms.

Number	Name	Fuzzy set
1	A_1	
2	A_2	
3	A_3	
4	A_4	
5	A_5	
6	NOT(A_1)	
7	NOT(A_2) = Smaller than A_2 OR Bigger than A_2	
8	NOT(A_3) = Smaller than A_3 OR Bigger than A_3	
9	NOT(A_4) = Smaller than A_4 OR Bigger than A_4	
10	NOT(A_5)	
11	A_1 OR A_2	
12	A_2 OR A_3	
13	A_3 OR A_4	
14	A_4 OR A_5	
15	A_1 OR A_2 OR A_3	
16	A_2 OR A_3 OR A_4	
17	A_3 OR A_4 OR A_5	

used for building composite labels because of the result would be a subnormal fuzzy set while we always work with strong fuzzy partitions.

$$NOT(A_i) \Rightarrow \begin{cases} A_1 \;\; OR \;\cdots\; OR \;\; A_{i-1} \;\; = \;\; Smaller \;\; than \;\; A_i \\ A_{i+1} \;\; OR \;\cdots\; OR \;\; A_m \;\; = \;\; Bigger \;\; than \;\; A_i \end{cases} \tag{5}$$

For example, for one regular partition with 5 labels, the user can choose between the linguistic terms of table 1 in rule definition. As it can be seen in this table, OR combination is restricted to neighboring terms in order to ensure that the aggregated fuzzy set is convex.

Rules can be defined by expert or induced from data. Rule nature is taken account into the simplification process and in case of conflict expert rules are our priority.

The overall reduction process involves two steps. First the *Simplify RB* procedure is applied in order to remove redundant rules. Second the *Merge RB* procedure is used for building more compact and general rules.

4.1 Simplify RB

A consistency analysis of the knowledge base is made in order to detect redundant rules:

- Rules with the same premise and the same conclusion. The recommended action is remove one of them.
- The input space covered by one rule is included into the one covered by the other, and both rules have the same conclusion. The most specific rule is removed, but only if it is an induced rule, or it is an expert rule and the most general rule is an expert rule too. Expert rules have priority.

4.2 Merge RB

A linguistic analysis of the rule base is made in order to detect rules that can be merged. Two rules can be merged if they are of the same nature (expert or induced rules) and also satisfy next condition: *They have the same conclusion and their premises can be merged, i.e. there must be a composite linguistic term equivalent to the merger of both labels.*

The procedure for merging rules follows next sequence of steps, which are repeated until the whole rule base is analysed:

1. Evaluate quality of knowledge base.
2. Look for two rules which can be merged.
3. Generate a temporal copy of the knowledge base.
4. Substitute both rules for the merger rule in the temporal knowledge base.
5. Evaluate quality of temporal knowledge base.
6. If (new quality is not worse than old quality) Then (changes in knowledge base are saved) Else (changes are discarded).
7. Go to 2.

Note that knowledge base quality is measured in terms of accuracy (see section 2). However, in order to keep or discard changes, the rule base consistency has to be checked too. The simplification process produces a generalization, a simplified rule may cover an input space which was not initially managed, for instance because there are no available data in this zone. It is necessary to check that the simplification does not lead to inconsistency as illustrated in figure 3. R3 can not be merged with R1 and R2 because they have different conclusions. Linguistically R1 and R2 can be merged. However, R3 input space is included into the one covered by the merger rule R12, and they are contradictories rules because they have different conclusion. In consequence the fusion has to be discarded.

In order to merge two rules, there is a need to analyze their premises. There is no problem with the conclusion because it is the same in the merger rule and in the original rules. However, the premises of original rules have to be compared for each input. If both premises are the same, then this is the merger rule premise. But, if premises are different, then a new premise has to be built as merger of both. Considering the two types of labels (basic or composite with NOT/OR) used, there could be six situations:

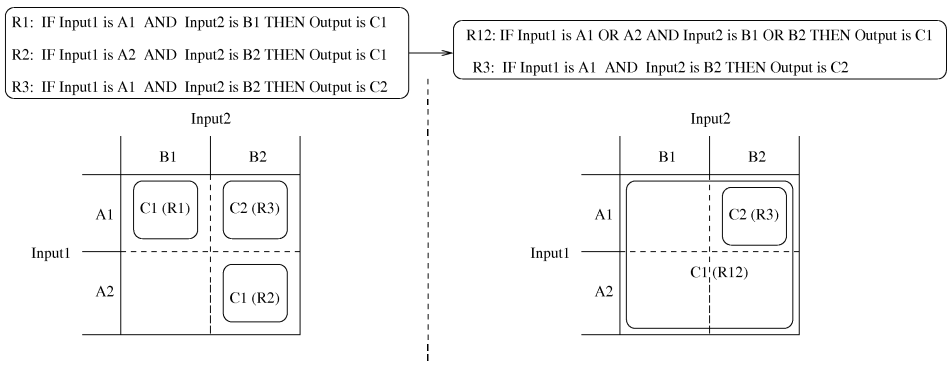

Figure 3: Rule generalization and possible conflicts.

- Two adjacent basic labels.
  - New premise is an OR composite label which includes both basic labels.
- One basic label and its respective NOT composite label.
  - New premise is the universal set. This condition will be always true.
- One basic label and one NOT composite label regarding to a different basic label.
  - If (Number of Labels equals two) Then (New premise is the basic label).
  - Else (New premise is the NOT composite label).
- One basic label and one OR composite label.
  - If (the basic label is included into the OR composite label) Then (New premise is the OR composite label).
  - Else if (both labels are adjacent) Then (New premise is a new OR composite label that includes both labels).
- One NOT composite label and one OR composite label which is included into the NOT composite label.
  - New premise is the NOT composite label.
- Two OR composite labels.
  - If (one label is included into the other one) Then (New premise is the biggest OR composite label).
  - Else if (both labels are adjacent) Then (New premise is a new OR composite label which includes both labels).

Merging rules changes rule base configuration, but it does not modify the fuzzy partitions in data base. Nevertheless, as a result of *Merge RB* process, composite labels could appear in the premise part of the rules. The fuzzy partition modification is made in the Data Base Reduction process.

5 Data Base Reduction

Data base comprises variable definitions, qualitative and quantitative information about variable behavior. It includes partition definition for each input or output, as well as semantic meaning of each linguistic term related to each fuzzy set.

This reduction process includes next steps:

1. Look for variables which are used by none of the rules and propose to remove them.
2. Look for labels which are used by none of the rules and propose to remove them.
3. Look for adjacent labels which are always used together and propose to merge them into a new one.

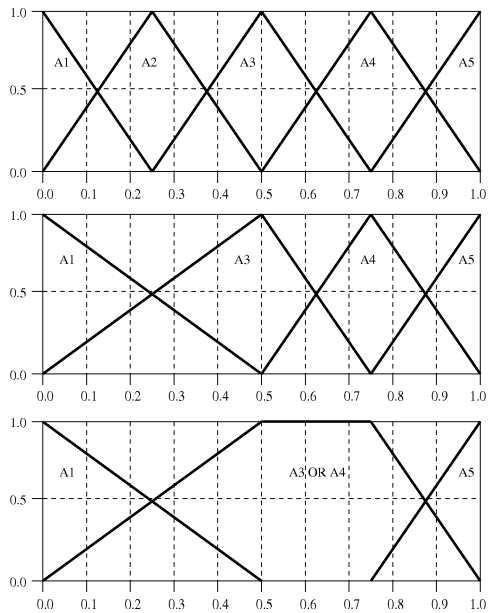

Figure 4: Removing and merging labels within a strong fuzzy partition.

Removing or merging labels change elementary fuzzy sets of given partitions as explained below. For instance, figure 4 illustrates how a regular strong fuzzy partition with five linguistic terms (picture on the top) is altered through two

consecutive modifications: First, in the middle part A_2 has been deleted, then A_3 and A_4 have been merged in the bottom part of the figure. Let us note that final partition is still a strong fuzzy partition.

5.1 Remove Labels

If one label is used by no rule, then it can be removed. However, in order to keep a strong fuzzy partition, adjacent fuzzy sets are expanded. The right boundary of the prior fuzzy set and the left boundary of the subsequent one are moved up to the center of the other. Thus, deleting a label comes to expand adjacent fuzzy sets. This makes the control surface of the fuzzy inference system smoother.

5.2 Merge Labels

Labels which are always used together, can be grouped in only one label. The algorithm makes a linguistic analysis of the rule base in order to find out which labels are used for each variable in at least one rule. When an OR composite label is used but its component labels are not, neither alone nor in another OR composite label, then the merger of component labels is proposed. Merge labels at this level implicates to modify the fuzzy partition definition. The number of labels is decreased and a new label with trapezoidal or semi trapezoidal shape substitutes to the old component labels.

6 Application results

The simplification process was tested with its application to a well-known problem of wine classification. The data set was collected from KEEL[1] data repository. It contains 178 instances which have been divided into two different subsets of equal size for five times. The original class distribution is kept within each subset. As a result, five pairs of training (50% examples) and test (50% examples) sets were generated. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents (Alcohol, Malic acid, Ash, Alcalinity of ash, etc.) found in each of the three types of wines. Therefore, the knowledge base includes 13 input variables and 1 output (type of wine).

For each input variable, a regular strong fuzzy partition with 5 fuzzy sets was built in the variable range. Afterwards, with the previously defined regular partitions and the training data set, several rules were induced from data. Finally, the simplification process described in the paper was applied over the resultant knowledge base.

A lot of rule induction methods are available in the fuzzy literature [6, 10]. We choose two of them[2]: *Wang and Mendel* (WM) [17] and *Fuzzy Decision Trees*

[1] http://sci2s.ugr.es/keel-dataset

[2] Let us underline that our contribution does not rely on such algorithm development but in showing simplification results. Please refer to the cited literature for a complete description.

(FDT) [11]. According to the classification proposed in [3], WM is a prototype oriented strategy because it starts from examples. It generates complete rules, induced rules are defined considering all the available variables, and as a result they are likely to be simplified. On the other hand, FDT builds incomplete, more general, rules that also can be simplified.

Table 2: Simplification results.

Method	**Rules**	**Premises**	**Inputs**	**Labels**
WM	88.8	1154.4	13	65
WM + S	6.2	48.4	8	37
FDT	35.8	107	13	65
FDT + S	9.2	24	6.6	17.8

Table 2 gives the main simplification results, expressed in average value over the five data sets. Two cases were studied: Induced rules with WM and their simplification (WM + S), and induced rules with FDT and their simplification (FDT + S).

Tables 3 and 4 show how the simplification process alters the knowledge base accuracy.

Table 3: Accuracy over training data set (89 cases).

Method	**Error Cases**	**Ambiguity Cases**	**Unclassified Cases**
WM	0	0	0
WM + S	0	0	0
FDT	0	0.6	0
FDT + S	0	0.2	0

Table 4: Accuracy over test data set (89 cases).

Method	**Error Cases**	**Ambiguity Cases**	**Unclassified Cases**
WM	0.8	1.2	33
WM + S	5.4	5	3
FDT	2.8	2.8	0.8
FDT + S	3	5	0.8

The simplification process yields a more compact and interpretable knowledge base. The size of both, data base (number of inputs and labels defined by input) and rule base (number of rules and inputs used by rule, premises), is reduced. More

Notice that we use the methods implemented in Fispro [7]. This WM implementation differs from the original one in the fuzzy partition design step. Interpretable fuzzy partition are built previous to rule induction.

general rules are built and, as a consequence, after simplification the knowledge base covers input space parts where there was not available knowledge.

WM produces complete rules which are very specific ones. As a result, they yield a very good performance with respect to the training data set while the accuracy is not so good in relation with the test data set, there are lots of unclassified cases (the number of uncovered cases rises up to 33 over 89). The simplification process lets us achieve a very reduced number of rules (it decreases from 88.8 to 6.2), while the number of unmanaged samples drops to 3. Most of these rules are complete ones (rule number multiplied by input number is almost equal to premise number, see table 2), but they use OR and NOT composite linguistic terms (the induced rules only use basic labels) and in consequence they are much more general rules than the initial ones. On the other hand, the accuracy is kept over training (see table 3) but it is also improved over test data set (see table 4). Notice that although the accuracy is improved on the test set after the simplification process (in the sense that the number of unclassified cases is reduced), the number of errors increases.

FDT generates incomplete rules which are quite general ones. The initial coverage is high (smaller number of unclassified cases) for both, training and test sets. Nevertheless, in this case there is another problem, it is possible that the merger rules produce contradictions (partially redundant premises with different conclusions) in relation with the other rules. In those cases, the rule fusion is discarded, and as a result the final knowledge base looks less compact: The number of rules is higher than in the WM + S case, 9.2 instead of 6.2. However, most of these rules are incomplete ones (the number of premises is much smaller than in WM + S, 24 instead of 48.4). Therefore, they are more general rules and the number of uncovered samples drops to 0.8 for the test set. Regarding accuracy, results are improved over training (see table 3), but they slightly get worse over test data set (see table 4). This accuracy worsening is negligible in comparison with the achieved interpretability improvement as result of the knowledge base size reduction.

To sum up, in all cases, the final knowledge base is more compact and interpretable, but it is also as accurate as the initial one, or more accurate (in some cases), with a larger coverage (smaller number of unclassified cases) and a smaller number of error and ambiguity cases. Note that error and ambiguity cases are measured in relation to the covered examples. If the number of managed cases is much larger, then the number of error and ambiguity cases would probably be slightly increased.

7 Conclusions

Previous work [2] presented an approach to build FIS through using both, expert and induced knowledge, by focusing in the interpretability.

This paper describes a simplification procedure for linguistic knowledge bases, whose aim is not only to find a balance between accuracy and interpretability but to improve both at the same time. That was illustrated with the well-known wine classification problem. Achieved results show that the final knowledge base is more compact and transparent, but also more accurate than initial one.

Thanks to the fuzzy logic formalism the induced knowledge can be described with the same kind of linguistic rules that those used for expressing expert knowledge. The simplification process deals with linguistic rules which can be of different nature: Defined by expert, induced from data, or both of them.

The whole process is implemented in KBCT [1], an open source software (distributed under the terms of the GNU General Public License) for generating and refining fuzzy knowledge bases.

Acknowledgement

Data for the wine classification problem were collected from KEEL data repository. We wish to thank KEEL team for keeping this repository.

KBCT was developed in the framework of the european research project ADVOCATE II, supported by the European Commission (IST-2001-34508). We would like to acknowledge the partners of this project: Universidad de Alcalá, Getronics Decan, Universidad Politécnica de Madrid, ATL Elektronics, Ifremer, Hugin Expert A/S, Innova S.p.A and E-Motive.

References

[1] J. M. Alonso, S. Guillaume, and L. Magdalena. Kbct: A knowledge management tool for fuzzy inference systems. *http://www.mat.upm.es/projects/advocate/kbct.htm*, 2003.

[2] J. M. Alonso, L. Magdalena, and S. Guillaume. KBCT: A knowledge extraction and representation tool for fuzzy logic based systems. In *FUZZ-IEEE 2004. IEEE International Conference on Fuzzy Systems*, volume 2, pages 989–994, Budapest, Hungary, July, 25-29, 2004.

[3] P. Carmona, J. L. Castro, and J. M. Zurita. Strategies to identify fuzzy rules directly from certainty degrees: A comparison and a proposal. *IEEE Transactions on Fuzzy Systems*, 12 (5):631–640, October 2004.

[4] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena. Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview. *In [5]*, pages 3–22, 2003.

[5] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena. *Interpretability issues in fuzzy modeling*, volume 128. Studies in Fuzziness and Soft Computing, Springer-Verlag, Heidelberg, 2003.

[6] S. Guillaume. Designing fuzzy inference systems from data: an interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, 9(3):426–443, June 2001.

[7] S. Guillaume, B. Charnomordic, and J.-L. Lablée. Fispro: An open source portable software for fuzzy inference systems. *http://www.inra.fr/internet/Departements/MIA/M/fispro/*, 2002.

[8] S. Guillaume and L. Magdalena. An or and not implementation that improves linguistic rule interpretability. In *IFSA World Congress*, volume 2809 of LNCS, pages 302–313, Beijing, China, July 2005.

[9] S. Guillaume and L. Magdalena. Expert guided integration of induced knowledge into a fuzzy knowledge base. *Soft Computing - A Fusion of Foundations, Methodologies and Applications.*, 2006.

[10] E. Hüllermeier. Fuzzy methods in machine learning and data mining: Status and prospects. *Fuzzy Sets and Systems*, 156:387–406, 2005.

[11] H. Ichihashi, T. Shirai, K. Nagasaka, and T. Miyoshi. Neuro-fuzzy ID3: A method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning. *Fuzzy Sets and Systems*, 81:157–167, 1996.

[12] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of Man-Machine Studies*, 7(1):1–13, 1975.

[13] D. D. Nauck. Measuring interpretability in rule-based classification systems. In *XII IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2003)*, volume 1, pages 196–201, St. Louis, Missouri, USA, May, 25-28, 2003.

[14] E. H. Ruspini. A new approach to clustering. *Information and Control*, 15(1):22–32, July 1969.

[15] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics. Part C: applications and reviews.*, 15:116–132, 1985.

[16] P. Villar. Keel: Interpretability issues in fuzzy rule based systems. Technical report, http://sci2s.ugr.es/keel/interpretability.php, 2004.

[17] L.-X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22 (6):1414–1427, 1992.