A Simplification Process of Linguistic Knowledge Bases

José M. Alonso, Luis Magdalena
ETSI Telecomunicación
Universidad Politécnica de Madrid (Spain)
{jmam,llayos}@mat.upm.es

Serge Guillaume
Cemagref Montpellier (France)
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 process, linguistic knowledge bases, trade-off, interpretability, accuracy, 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 focused on the ability of fuzzy logic to model natural language [1]. These FIS contain fuzzy rules built from expert knowledge, they can be seen as a fuzzy extension of expert systems. Sugeno [2] 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 [3].

As expert knowledge and knowledge hidden in 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 [3]: interpretable fuzzy partitions, a small number of rules, incomplete rules for large systems.

The cooperation framework has been proposed in [4]. The process consists of three different steps: defining a common universe for each of the variables according to both expert knowledge and data distribution, then inducing rules from data, and finally integrating the induced rules into the expert knowledge base.

During this last step, the fundamental properties of a rule base have to be guaranteed. This communication deals with the simplification process of the rule base. The goal is to design incomplete, more general, rules while checking coherency and avoiding redundancy in the final rule base. Thanks to the common universe rule comparison can be done at the linguistic level.

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 [5] but to improve both at the same time.

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

2 Knowledge Base Simplification Process

Fuzzy knowledge bases used 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 [6] 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.

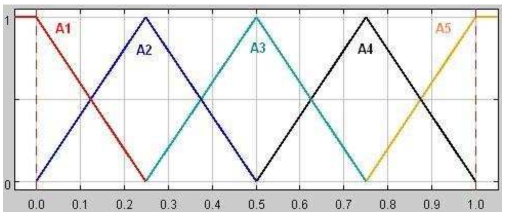


Figure 1: A Strong Fuzzy Partition

Figure 1 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:

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

$$\forall A_i \; \exists x, \; \mu_{A_i}(x) = 1 \tag{2}$$

where *U* is the range, *E* is the number of terms and $\mu_{A_i}(x)$ 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 behavior of the system in an understandable way. Fuzzy linguistic rules are of form “If condition Then conclusion” where both, the premise and conclusion use linguistic terms.

The main goal of the simplification process is to achieve a more compact knowledge base, with a smaller size in order to increase interpretability [7], but without getting worse accuracy of the original knowledge base. Figure 2 shows a schematic diagram of the knowledge base simplification process. It is a cyclical process. Depending on the original knowledge base, the whole process could involve several iterations as data base reduction affects to rule base simplification and vice versa.

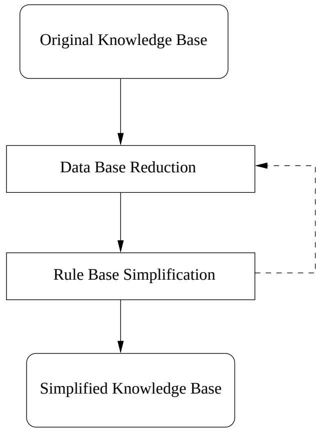


Figure 2: Knowledge Base Simplification

2.1 Knowledge Base Quality

This subsection explains the criteria for evaluating knowledge base quality. It is measured according to data through considering three indexes for each output:

- **Performance**: For regression cases, it is defined as the root mean of sum of squared errors (eq.3). And for classification cases, it is defined as the number of misclassified items (eq.4).

$$Perf = \frac{1}{N}\sqrt{\sum_{i=1}^{N} ||\hat{y}_i - y_i||^2} \tag{3}$$

$$Perf = \sum_{i=1}^{N} \delta_i, \begin{cases} \delta_i = 1 & if \;\; \hat{y}_i \neq y_i \\ \delta_i = 0 & otherwise \end{cases} \tag{4}$$

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

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

These indexes are expected to convey complementary information. A good knowledge base should maximize the coverage and minimize the error indexes.

Also, in classification cases, two more parameters are taken into account:

- **Error cases**: Number of covered cases from data set that produces error, i.e. observed and inferred values are different, in inference.
- **Ambiguity cases**: Number of covered cases from data set that produces ambiguity, i.e. difference between two different classes is smaller than an established threshold, in inference.

If there are data available and the simplification process involves a modification in the knowledge base, its quality is evaluated in order to confirm or discard the change. Modifications will be saved, only if quality doesn't get worse, i.e. on one hand coverage doesn't decrease, and on the other hand performance, max error, error and ambiguity cases don't increase.

If there are no data, then the simplification process is made only at linguistic level, without checking the quality of knowledge base.

3 Data Base Reduction

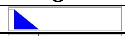
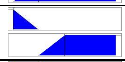
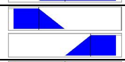
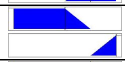
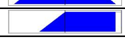
Data base comprises variable definitions, i.e. qualitative and quantitative information about variable behavior. It includes partition definition for each input or output, as well as semantic meaning of each linguistic term related to each fuzzy set.

The fuzzy partition includes only basic labels A_i which correspond with elementary fuzzy sets. Nevertheless in the rule description they can be combined using OR and NOT for building composite linguistic terms. The implementation of OR and NOT 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 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 partition with 5 labels (figure 1), 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.

Table 1: Linguistic terms.

Number	Name	Figure
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 A2 OR Bigger than A2	
8	NOT(A_3) = Smaller than A3 OR Bigger than	
9	NOT(A_4) = Smaller than A4 OR Bigger than A4	
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	

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.

Removing or merging labels change elementary fuzzy sets of given partitions as explained below.

3.1 Remove Labels

If one label is used by none of the rules, 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. For instance, figure 3 shows how the partition in figure 1 is modified after deleting the label A_2.

Deleting a label which is used by none of the rules comes to expand adjacent fuzzy sets. This makes the control surface of the fuzzy inference system smoother.

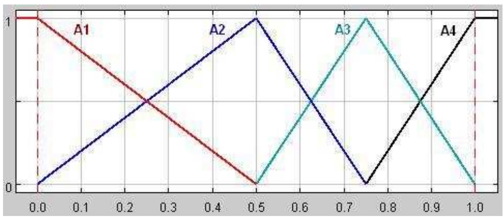


Figure 3: Partition once removing label A_2 in figure 1

3.2 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 a composite label is used but its component labels are not, neither alone nor in another 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. Figure 4 shows an example about how partitions change through merging labels. Partition in figure 3 is modified after merging basic labels A_2 and A_3. Note that final partition is a strong fuzzy partition too.

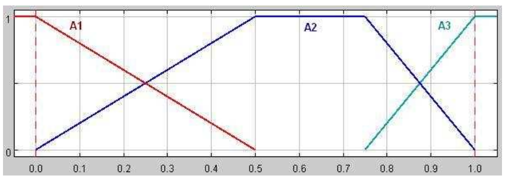


Figure 4: Partition once merging labels A_2 and A_3 in figure 3

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. On the other hand, rule conclusions are made up of couples output variable - linguistic term, where the absence of an output variable means that this rule doesn't concern that output variable. For each input variable, the user can choose between basic labels or composite labels (by using OR/NOT operators).

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 [9] of the knowledge base is made in order to detect redundant rules to remove:

- Rules with the same premise and the same conclusion. 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 by the merger rule in the temporal knowledge base.
5. Evaluate quality of temporal knowledge base.
6. If (new quality isn't worse than old quality) Then (changes in knowledge base are saved) Else (changes are discarded).
7. Go to 2.

In order to merge two rules, it is needed to analyse the premise part of the rules. There is no problem with the conclusion because it is the same in the merger rule and in the original rules. However, for each input the premises of original rules have to be compared. 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:

- 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, i.e. 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 doesn't 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. To modify the fuzzy partition, the user has to run again the Data Base reduction process as shown in figure 2 by the dashed line.

5 Application results

The simplification process was tested with its application to a well-known problem of wine classification. The data set contains 178 instances which have been randomly divided into 2 different subsets at 50%-50%, one for learning and the other for testing. 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 found in each of the three types of wines. Therefore, the knowledge base includes 13 input variables and 1 output.

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

Two methods were used in order to induce rules: Wang and Mendel (WM) [10] and Fuzzy Decision Trees (FDT) [11]. Let us underline that our contribution doesn't rely on such algorithm development but in showing simplification results. Please refer to the cited literature for a complete description.

Table 2: Simplification results over test data set.

Method	**Rules**	**Labels**	**Error Cases**	**Ambiguity Cases**	**Coverage (%)**
WM	89	65	1	1	47
WM + S	6	59	5	2	96
FDT	99	65	3	4	98
FDT + S	28	47	1	8	100
FDT + P	57	65	7	4	97
FDT + P + S	19	34	4	3	100

Table 2 gives the main results. Three cases were studied: induced rules with WM and the simplification (WM + S), induced rules with FDT and the simplification (FDT + S), induced rules with pruned FDT (FDT + P) and the simplification (FDT + P + S). In all cases, the final knowledge base is more compact, with a smaller number of rules which are more general, and a smaller number of labels. As a result, the final knowledge base is more interpretable, but also more accurate than initial one, with a larger coverage and a smaller number of error cases. Note that error and ambiguity cases are measured in relation to the covered examples.

6 Conclusions

Previous work [9] presented an approach to build fuzzy inference systems 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, where the final knowledge base is more compact and transparent, but also more accurate than initial one. The whole process is implemented in KBCT[1], an open source software for generating or refining fuzzy knowledge bases.

Acknowledgments

Data for the wine classification problem were collected from KEEL[2] 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[3], 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] E. H. Mamdani and S. Assilian. “An experiment in linguistic synthesis with a fuzzy logic controller”. *International journal of Man-Machine Studies*. Vol. 7, pp. 1-13, 1975.

[2] T. Takagi and M. Sugeno. “Fuzzy identification of systems and its applications to modeling and control”. *IEEE Transactions on System Man and Cybernetics*. Vol. 15, pp. 116-132, 1985.

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

[4] S. Guillaume and L. Magdalena. “Expert guided integration of induced knowledge into a fuzzy knowledge base”. *Soft Computing*. In Press, 2005.

[5] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena. “Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview”. In [7].

[6] E. H. Ruspini. “A new approach to clustering”. *Information and Control*. Vol.15, pp. 22-32. 1969.

[7] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, editors. “Interpretability issues in fuzzy modeling”. *Studies in Fuzziness and Soft Computing*, Vol. 128, Springer-Verlag, Heidelberg, 2003.

[8] S. Guillaume and L. Magdalena. “An or and not implementation that improves linguistic rule interpretability”. In *IFSA World Congress*. In Press, Beijing, China, July 2005.

[9] José M. Alonso, L. Magdalena and S. Guillaume. “KBCT: a knowledge extraction and representation tool for fuzzy logic based systems”. *IEEE International Conference on Fuzzy Systems*. July, 25-29, 2004, Budapest, Hungary. Proceedings. Vol. 2, pp. 989-994.

[10] L.-X. Wang and J. M. Mendel. “Generating fuzzy rules by learning from examples”. In *IEEE Transactions on Systems, Man and Cybernetics*. Vol. 22 (6), pp. 1414-1427. 1992.

[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*. Vol. 81, pp. 157-167. 1996.

[1] http://www.mat.upm.es/projects/advocate/kbct.htm
[2] http://sci2s.ugr.es/keel-dataset
[3] http://advocate2.e-motive.com