Mathware & Soft Computing 5 (1998) 345-353

A Methodology for Developing Knowledge-Based Systems

J.L. Castro[1], J.J. Castro-Schez[2], A. Espin[3], J.M. Zurita[1]
[1]Dpto. Ciencias Comp. e IA. E.T.S.I. Informática. Univ. Granada
Avda. Andalucía, 38, 18071 Granada. *e-mail: castro,zurita@decsai.ugr.es*
[2]Dpto.Informática. E.U. Informática. Univ. Castilla la Mancha
Ronda de Calatrava, 5, 13071 Ciudad Real
[3]Dpto.Ingeniería Civil. E.T.S.I. Caminos. Univ. Granada
18071 Granada *aespin@platon.ugr.es*

Abstract

This paper presents a methodology for developing fuzzy knowledge based systems (KBS), which permits a complete automatization. This methodology will be useful for approaching more complex problems that those in which machine learning from examples are successful.

Keywords: Knowledge based systems, knowledge acquisition, machine learning.

1 Introduction

The National and European scientific policy has one of the priority themes, the development of knowledge based systems. In fact, in the last few years the number of research groups devoted to this topic, mercantile and non-mercantile shells for designing KBS, and techniques and tools for developing KBS has increased significatively. At the present time one of the more suggestive fields is the development of techniques and tools, which automatically design KBS, especially in the task of knowledge acquisition. The aim of these techniques and tools is helping the knowledge engineer to design the KBS, and sometimes to automatically develop KBS with the support of the domain expert.

An essential feature of the KBS is need to manage the vagueness and uncertainty present in the human knowledge. In this direction, Fuzzy Logic has proven be a very effective tool [8] sometimes indispensable, for problem-solving in which information and data have non probabilistic vagueness and uncertainty.

This is due to its:

(1) *Simplicity and naturalness* as a formalism of knowledge representation, because it allows us to translate from expressions into natural language to fuzzy rules [16].

(2) *Effectiveness,* demonstrated in a theoretical [3] and practical [8] form.

(3) *Capacity* for managing vagueness and uncertainty in a simple manner [15].

Thus, the design of fuzzy KBS, that is to say those which make use of Fuzzy Logic [7] [11] [1], is natural. These systems make use of Fuzzy Logic as a formalism of knowledge representation and fuzzy inference for designing an Inference engine. There are also tools for designing Fuzzy KBS that make use of Fuzzy Logic and many papers dedicated to the study of fuzzy knowledge based systems.

Amongst the works dedicated to studying fuzzy KBS, there are works dedicated to machine learning of fuzzy rules from examples [13] [14] [12] [6] [5] [10, *Machine Learning sessions*]. We may also to point out that there are some research projects into this topic, for example TIC96-0778 "*Learning of fuzzy rules based systems making use of Genetic Algorithm and Neural networks*" and TIC95-0453 "*Learning in vagueness and uncertainty environment. Applications*".

Regarding the design of tools that automatically develop fuzzy KBS, an important piece of work is the MILORD shell. This shell was developed at the IIA, CSIC in Barcelona [11]. This shell makes use of a module of knowledge elicitation by means of an Elicit-Analyse-Refine system, and it is based on Kelly's Personal Construct Theory. Elicitation is based on a fuzzy repertory grid where each construct has a linguistic value assigned when it is applied to an element from the domain. This leads to a repertory grid with cross-references between the constructs and the elements from the domain. Then repertory grid is analysed by the system for finding:

a) Elements not too well defined.

b) Elements missing or forgotten.

These problems are solved with the introduction of new constructs and elements.

In a second step the system obtains a set of initial fuzzy rules in the form of an inference net from repertory grid. In the refinement step, the expert analyses the inference net and points out his disagreement. The system makes use of some techniques from the repertory grid for rectifying the set of rules. This analysis ends when no disagreement is found.

A second tool for automatically developing fuzzy KBS is KAFES [9]. The aim of this shell is knowledge acquisition for expert systems. This shell is also based on a fuzzy repertory grid and performs in a similar manner to MILORD. KAFES is designed for making CLIPS expert system. The difference between MILORD and KAFES is that the latter does not carry out elicitation by using Elicitation-Analyse-Refine, the structure of the fuzzy rules is different and it has a module for obtaining the membership functions of fuzzy labels.

Another interesting development about this type of tools is the methodology proposal by Slany [10, *Applications I session*] which makes use of ChekFLIP++ as an Acquisition knowledge tool for developing fuzzy restrictions based systems.

The works and research projects for machine learning from examples mentioned above, are very useful for solving problems that are not too complex and they all obtain very similar results or conclusions. The results are less successful when the problems have a more complex structure. In our opinion, this is so because all the algorithms and techniques developed attempt to learn the system completely with a structure of very simple rules without obtaining benefit from fuzzy logic ability for representing human knowledge and thus extract more knowledge from human expert. Knowledge acquisition systems, MILORD and KAFES, do not make use of machine learning from examples. Moreover, since its aim will be to develop KBS automatically, which can be very complex, it requires expert assistance.

This paper presents a methodology for automatically developing fuzzy KBS, which gets nearer to more complex problems than those in which machine learning from examples are successful. The fact that our main objective will be the complete automatization of the KBS development process restricts the number of problems to which it can be applied. We attempt to design an intermediate methodology among previously mentioned methodologies, and which will combine the main benefits from each one: automatization and the ability for managing complex problems by means of knowledge acquisition from experts.

2 Establishing the problem

Our initial hypothesis is that the proposed methodology should be devoted to designing a KBS for solving a classification or decision problem. This problem involves determining to which class an object belongs (or which decision to take) from an input information about object features (or input information). Input information will be values of input variables $\{X_1, ..., X_n\}$ and output will be a class belonging to a set of previously established classes $y_1, ..., y_d$.

Input variables values can be exact values (crips values) or vagueness (linguistic expressions which are explained as fuzzy sets). Moreover, they can be discrete or continuous, but each input variable X_i in a known range as $[a_i, b_i]$ in the continuous case or $\{V_i^1, ..., V_i^{ni}\}$ in the discrete case.

We shall assume that each variable V_i has a set of possible attributes $\mathfrak{L} = \{L_i / i \in I_i\}$. This set of attributes can change from the fuzzy subset family in the V_i domain, when we do not start out from a previously established set of attributes, to a finite subfamily in this label, when we start out from a previously established set.

New variable creation will be possible when necessary for correctly designing the KBS.

Our idea is to design the KBS as a set of fuzzy rules so if there are conflicts, the conflict-resolving strategy used will select the rule with the highest degree of fit for the input information.

The form of the rules will be the following:

IF X_1 is [set of attributes] and ... and X_n is [set of attributes]
THEN Y is [decision]

This type of rules comprise simpler rules such as:

IF X_3 is High **THEN** Y is y_1

In the case of three input variables: X_1, X_2 and X_3. Since this rule has the following equivalent rule:

IF X_1 is [complete set of attributes] and X_2 is [complete set of attributes] and X_3 is High **THEN** Y is y_1.

The degree of fit will be calculated in the usual manner, i.e., making use of minimum T-norm. Obviously some other T-norms can be used with the same purpose.

3 A methodology for developing KBS with the help of an expert

In this case, the system will take into account two classes of information:
- A set of examples with the correct output for a set of input values.
- A domain expert.

The proposed methodology is the following:

[1.] **Elicitation of Relevant Variables Phase.** In this phase the relevant variables of the problem, its domains and attributes are elicited. To do so, we make use of a set of interviews with the domain expert.

- In a first interview with the domain expert we attempt to make the expert point out how he solves the problem. The response will be transcribed.
- It elicits nouns and adjectives used by the domain expert. The elicited terms will be our candidates for variables and attributes respectively.
- In a second interview we verify whether the elicited terms are really the variables and attributes used in problem-solving method.
- Variables, their types and possible attributes are established.

[2.] **Choosing a Representative Set of Examples Phase.** The set of examples is divided into two groups, the first contains 80% of the instances for knowledge acquisition, and the second contains 20% of the instances for verifying. This will be carried out with the help of the domain expert. The expert verifies that examples in the test set are similar cases to those belonging to the learning set.

[3.] **Knowledge Acquisition Phase.** For each test example, we ask the domain expert to explain the classification. This explanation is converted into a rule, which solves that example and similar ones.

When we study a new example, we check whether it can be solved with the knowledge acquired previously and we also check whether the expert's classification and/or explanation are the same as the one system presents. Then the following events may occur:

- The classification and explanation yielded by the expert and the system are the same. In this case we do not do anything, and it passes on to the following example in the test set.

- The classification yielded by expert and the system is the same but the explanation is different. In this case, the new explanation is converted into a new rule, because this new rule can allow new cases to be solved.
- If the classification is wrong or the explanation is inadequate, the system interacts with the expert to acquire more knowledge. It is possible that there are two (or more) rules that yield inconsistency. The conflict-solving methods can be the following:
  1. Explain in a more precise manner some or all the rules that participate in the conflict. This can be done with the introduction of new antecedents or explain in a more precise manner the attributes of the variables which appear in the antecedent part of the rules.
  2. All rules are valid, but when two or more rules can be used, we shall have to apply some strategy (non-monotonic reasoning) for choosing between these rules. In this case, the system solves this problem by creating a new module, which is performed only when these rules yield the conflict and its aim is to resolve this particular inconsistency. The inconsistency-solving method goes away from weighing both rules and chooses those with the greatest weight to introduce new rules which solve these particular cases.

In some of the cases mentioned above, it may be necessary to introduce a new variable, which has not been considered yet.

When this process ends, we have a rules base, which explains all the examples belonging to the acquisition knowledge set as the expert does. We don't guarantee that this base of rules will be complete or solid. That is because there are inputs for which it does not find examples in the knowledge acquisition set, or its output is not known or it leads to inconsistency.

For searching a complete and solid rules base we propose the following phase:

[4.] **Validation Phase.** We obtain inputs where the system is not able to classify them or yields inconsistency. This will be carried out making use of the algorithm [2]. The inputs found will be shown to the domain expert. Then two events may occur:

- This input doesn't occur in real life. In this case it does nothing.
- This input can occur in real life. In this case, it performs as it does with the examples of knowledge acquisition.

When this process ends, we have a rules base that is complete and solid, which explains all the examples belonging to the acquisition knowledge set as the expert does.

[5.] **Verification Phase.** The system is performed for each example belonging to the test set. The aim of this phase is to check whether the classification and explanation are the same as expert's. When both classifications and explanations are equal, the system does not work. If these classifications and explanations are different, the system performs in the same manner as it performs when a conflict is found in the knowledge acquisition phase.

4 A methodology for machine learning of KBS from examples

In this case, we only have a set of training examples. These examples have input information and one output. We know the input variables and the expert's output.

Our idea is to make use of the same methodology studied earlier. To do so, we propose replacing the expert interaction with a machine learning method. This method will learn a set of fuzzy rules.

The proposed methodology is the following:

[1.] **Design Labels Phase.** We choose the set of labels, which represent fuzzy attributes to be considered for each variable.

[2.] **Choose a Set of Examples Phase.** The set of examples is divided into two groups, the first contains 70% of the instances for learning, and the second contains 30% of the test instances.

[3.] **Learning Phase.** For each example belonging to the set of learning, the system learns more general (simpler) fuzzy rule, which explains to that example. To do so, we propose making use of the algorithm [10] [5]. This algorithm obtains a rule iff the number of contradictory examples have a degree of convenience $< \epsilon$ (valid rules).

When we take a new example, we check whether this example can be explained with a fuzzy rule learnt previously, which has a degree of convenience $\geq \epsilon$. If this happens, the system answers with that rule and it passes on to the following example. If this does not happen, the system learns the rule that explains this new example and it compares both rules. When we carry out this comparison the following events can occur:

- The answers are equal. In this case, the new rule is included in the set of fuzzy rules. This new rule may be necessary for explaining different cases to the rule learnt previously.
- The answers are different. In this case, we have two or more rules that yield inconsistency. This conflict is solved by analysing and learning new rules. These rules will be applied only when there is conflict. The structure of these rules will be determined by the examples, which yield that conflict.

[4.] **Validation Phase.** We obtain inputs where the system is not able to classify them or yields inconsistency. This will be carried out making use of the algorithm [2]. When there are examples, which correspond with those inputs found, we accept as a solution the answer that holds the largest number of examples. If this does not happen, the system does not do anything.

[5.] Verification Phase. The system is performed for each example belonging to the test set. The aim of this phase is to check whether classification and explanation are the same as the expert's. Thus, we obtain a measure about the goodness of the system learnt.

5 Example

We have carried out a very simple test for checking the goodness of the proposed methodology. The test consists of developing a KBS for classifying the Iris Plant. R.A. Fisher created the Iris Plants Database. This database has been used in many publications. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The number of instances in the Iris Plants Database is 150 (50 in each one of three classes), and the number of attributes is 4 numeric-predictive and the class. The first attribute indicates sepal length, the second sepal width, the third petal length and the final one indicates petal width. All the attributes are measured in centimetres. The classes may be Iris-setosa, Iris-versicolor or Iris-virginica. In the test, we use 105 instances (70%) for learning and 45 instances (30%) for proving the rules that have been learnt.

We have divided the domain intervals of each input variable into three *linguistic notions*, which are Small, Medium and High. These three fuzzy sets can be given by dividing the respective domains of each input variable into three equal parts and modelling each notion as trapezoidal fuzzy labels.

The algorithm for learning fuzzy rules (with $\epsilon = 0.3$) obtained the following three rules:

1.1 **IF** Petal Width is Small **THEN** the Plant is Iris-Setosa.
1.2 **IF** Petal Width is High **THEN** the Plant is Iris-Virginica.
1.3 **IF** Petal Length is Medium **THEN** the Plant is Iris-Versicolor.

In a validation phase, we find two conflicts (for which we have examples). The conflicts are the following:

· If Width Petal is Medium and Length Petal is High then we can not answer.
· If Length Petal is Medium and Width Petal is High then it yields an inconsistency.

The analysis of the first conflict leads to the introduction of a new rule:

1.4 **IF** Petal Length is Medium and Petal Width is High **THEN** the Plant is Iris-Virginica.

The analysis of the second conflict leads to the introduction of a new module. The system performs this module only when inconsistency is yielded by rules 1.2 and 1.3 with similar values (difference$\leq$0.2). This module holds the following rules:

2.1. **IF** Width Sepal is Medium **THEN** the Plant is Iris-Versicolor.
2.2. **IF** Width Sepal is Small **THEN** the Plant is Iris-Virginica.

This simple KBS obtained highly satisfactory results. The system correctly classifies all the examples belonging to the set of test examples, and it fails only with one example in the training examples set. Finally, we compare these results obtained with the results obtained by other classic algorithms for machine learning (Decision Trees, Neural Network, Logic Regression) and the method of machine learning developed at the department of *Ciencias de la Computacion e Inteligencia Artificial in Granada*. We apply them all over the same division of the examples base. The KBS developed obtains the best result, (99.4% as compared to 98.7%), moreover the rules that it obtains are the simplest rules.

6 Conclusion

In this paper, we present a methodology for developing fuzzy KBS with the following features:

- The design phase is partially automated.
- It may be applied to machine learning.
- It makes use of non-monotonic reasoning.
- The design is modular.

Moreover, it has been applied to a very simple example of machine learning where it has obtained highly significant results.

References

[1.] Campbell, J.G. Fuzzy Rule-Based Software Modules - Programmers' Guide. Interactive Systems Centre, *Report isc/94021 /R*, University of Ulster, 1995.

[2.] Castro, J.L., Trillas, E., Zurita, J.M. Expert Validation. *Proceedings of the sixth IFSA Congress*, Sao Paulo, 1995, 41-44.

[3.] Castro, J.L., Delgado, M. Fuzzy Systems with defuzzification are Universal Approximators. *IEEE Trans. On Syst, Man. and Cybern.* 26-01, 1996, 149-152.

[4.] Castro, J.L., Trillas, E., Zurita, J.M. Non Monotonic Fuzzy Reasoning. To appear in *Fuzzy Sets and Systems.*

[5.] Castro, J.L., Castro-Schez, J.J., Zurita, J.M. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. To appear in *Fuzzy Sets and Systems.*

[6.] Gonzalez, A. A learning methodology in uncertainty and imprecise environments. *International Journal of Intelligence Systems*, 10, 1995, 357-371.

[7.] Hall, L.O., Kandel, A. *Designing Fuzzy Expert Systems.* Verlag TÜV Rheinland, Germany, 1986.

[8.] Hirota, K. *Industrial Applications of Fuzzy Technologies.* Springer Verlag, 1993.

[9.] Hwang, G.J. Knowledge Acquisition for fuzzy Expert Systems. *International Journal of Intelligent Systems,* 10, 1995, 541-560.

[10.] IPMU96. *Proceedings of the sixth IPMU Congress*, Granada, España, 1996.

[11.] Kandel, A. *Fuzzy Expert Systems.* Kandel Eds, CRC Press, 1992.

[12.] Sestito, S., Dillon, T. Knowledge Acquisition of Conjunctive Rules Using Multilayered Neural Networks. *International Journal of Intelligent Systems*, 8, 1993, 779-805.

[13.] Sugeno, M., Tanaka, K. Successive identification of a fuzzy model and its applications to prediction of a complex systems. *Fuzzy Sets and Systems*, 42, 1991, 315-334.

[14.] Wang, L., Mendel, J.M. Generating fuzzy rules by learning from examples. *IEEE Trans. On Syst, Man. and Cybern.* 22, 1992, 1414-1427.

[15.] Zadeh, L.A. The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems*, 11, 1983, 197-227.

[16.] Zadeh, L.A. Fuzzy Logic. *Neural Network and Soft Computing*, CACM, 1994, 37-43.