

# Automated Fuzzy-Clustering for Doctus Expert System

---

Zoltán Baracscai  
Doctus KBS

Hungary  
[zoltan@doctus.info](mailto:zoltan@doctus.info)

Viktor Dörfler

Department of Management Sciences  
University of Strathclyde  
Graham Hills Building,  
40 George Street, G1 1QE Glasgow  
United Kingdom  
[viktor@mansi.strath.ac.uk](mailto:viktor@mansi.strath.ac.uk)

**Abstract – Our Knowledge-Based Expert System Shell 'Doctus'<sup>1</sup> is capable of deduction also called rule-based reasoning and of induction, which is the symbolic version of reasoning by cases<sup>2</sup>. If connected to databases or data warehouses the inductive reasoning of Doctus is also used for data mining. To handle numerical domains Doctus uses statistical clustering algorithm.**

**We define the problem in three steps: how to perform a clustering, which is neither rigid nor sensitive to noise, benefiting from the properties of the application domain, reducing the complexity as much as possible, and supplying the decision maker with useful information enabling the possibility of interaction?**

**In this paper we present the conception of Automated Fuzzy-Clustering using triangular and trapezoidal Fuzzy-sets, which provides overlapping Fuzzy-set covering of the domain.**

## I. FUZZY CLUSTERING FOR SYMBOLIC ES – WHY?

We investigate the expert systems in supporting the business decision making process. Let's first examine the domain of the application, to map characteristics that are important to choose the appropriate tool for support. We are dealing with decision making of a leader and of a manager on the expert level of knowledge and higher, who are to considering much of soft information and hard data, and use heuristic processes to take the decisions.

First there is a need to discover the properties of the heuristic processes in comparison to other processes:

1. At **deterministic processes** there is an expected value only with no dispersion. It is determined what output follows a particular input, it will happen in 100% of repetitions. Small changes on the input will result in small changes on the output, which can be calculated precisely. Deterministic processes can be met e.g. in classical physics (mechanics of not microscopic, but also not astronomy sized bodies).

2. Output of a **stochastic process** can be described with its expected value and its dispersion, which is smaller at least one order of magnitude. Small changes on the input will result in small changes on the output, which can be

estimated, though not calculated. Disciplines of stochastic processes are e.g. quantum physics and chemistry.

3. At **chaotic processes** the dispersion is in similar order of magnitude then the expected value. A small change on the input can result in huge changes on the output. To chaotic processes belong the field of biology and especially the genetics. It is interesting to read Russel [7] writing about instable states of balance (e.g. a ball balanced on the tip of the needle), which obviously belongs to domain of chaos, even he could never heard about chaos theory.

4. Decision maker in **heuristic processes** chooses an alternative not knowing the output. For the previous three processes a more or less formed mathematical toolbar is available (though it is quite underdeveloped for the chaos). In opposite, the heuristic processes cannot be handled exactly. This is the territory e.g. of the psychology and all disciplines (and each problem) concerning human as a person. We should not be misled by the result of a psychoanalysis, which finds the "cause" of a certain spiritual problem. If there was a real causal relation, everybody affected by the "cause" should have been facing the same problem. That is not the truth. In heuristic processes there are a plenty of affects that are not (and cannot be!) taken into account. Such phenomena show a great disorder, which cannot be described as stochastic or even chaotic. It can be stated, that the conditions of the central limit theorem are not satisfied – if they were, then the process would be stochastic. The central limit theorem fails as the causes are not independent and not all of them are known. In this kind of processes the same input may result in very different outputs. If wanted to use the description similar to the previous ones, it should be said that there is only dispersion without an expected value (this is not a correct formulation as the experienced decision maker will guess the output, though he uses unknown heuristics in his tacit knowledge).

It is not goal of this paper to examine if either we do not know enough about the process of business decision making therefore considering it to be heuristic, or it has inherent feature, which makes it heuristic. However, as quantum physics showed that there are processes with inherent uncertainty, which makes them stochastic and we expect a similar situation for the heuristic processes.

Accepting that decision making is a heuristic process, we excluded the possibility of optimization and other OR solutions.

There are two main streams of expert system, concerning the knowledge representation they use, the symbolist and the connectionist. The essence of the **symbolist conception** is to store the knowledge in a

---

<sup>1</sup> [www.doctus.info](http://www.doctus.info)

<sup>2</sup> Originally the term of case-based reasoning was inherited from quantitative decision support. Its essence was – and in much software it still peeps out behind the symbolic solution veil – to define some metrics and distance built on it, which will be the similarity. For a new case the nearest one – that means the most similar – is searched from the case-base. In symbolic logic cases described with the same rules are considered to be similar.

knowledge-base, its elements are symbols which are connected by logical (if... then) rules. The **connectionist conception** emphasizes functioning on sub-symbolic level. There is no need to treat every symbol separately — we may calculate instead. Two important realizations of connectionist conception are neural networks (NN) and Fuzzy-logic (FL). For us the question is not to decide which conception is better or which one gives generally better solutions, but which is appropriate for what, or how they can be usefully combined.

The first advantage of **symbolic representation** is it's humane.<sup>3</sup> In the symbolic knowledge-base of an expert system we can put the knowledge in form as we talk or think of it. Therefore we get to the second advantage, which is the transparency, easy modification and fine-tuning of the knowledge base. A disadvantage is that numerical signs can also be treated only as symbols, so if we want to use numerical data, first we have to transform them into symbols. If there are many symbols; there will be a plenty of rules, which is a real disadvantage. Today, this is not a problem of computing capacity. The expert-level knowledge is few thousand of cognitive patterns that mean few thousand rules, which does not a trouble to modern software. Though it is hard to acquire a lot of rules from the expert; the use of multi-step reasoning helps. Further disadvantage is that the expert has to articulate the rules, thus there is no access to the tacit knowledge. For this reason in symbolic approach it is unrealizable to represent the common sense; we are all masters of it. We cannot articulate much of it; most of our common sense waggles between focal skills and focal intuition (thus tacit). We cannot explain how to ride a bike; how to bypass a puddle or what do we do if a pram is pushed in front of us. Though, we do these without any trouble. The impossibility of extracting the tacit knowledge is valid for the deductive reasoning, i.e. for the rule-based reasoning (RBR). The solution: instead of acquiring rules, acquire the cases of experience, from which the software deduces the rules. This is called induction or reasoning by cases.

The **Fuzzy-logic** representation uses Fuzzy-sets as elements. They differ from conventional sets (crisp sets) in approach. Instead of deciding if something is or is not element of a set, we decide how much it belongs to a set. This is formally similar to probability, though the essence differs from it. This is sub-symbolic representation indeed. The existence of the sub-symbolic level is definitely indispensable. Zadeh [9] has originally developed the Fuzzy-sets for handling concepts with sub-symbolic level (e.g. cold, very warm, short and fast). Great feature of Fuzzy-logic is its capability to make relation between two levels: between quantifiable and un-quantifiable, so between arithmetic and logic. The FL needs fewer rules (then the symbolic representation) due to its sub-symbolic functioning. This is valid for the given number of attributes. Though there is a slip. In symbolic representation we have e.g. the attribute "clothing" with values "slubberer", "casual", "elegant" "extravagant" and "formal". It would mean five attributes in FL, for which

---

<sup>3</sup> The symbolic logic is the only solution that does not quantify the user's preferences. E.g. the person whose knowledge is being modelled thinks that the beautiful is better value than the ugly. Nobody thinks that the beautiful is 3.6 times better than the ugly. Using symbolic logic we do not state something like that.

we have to decide to what extent a case belongs to each of them. Disadvantage is that in the multi-step reasoning the Fuzzy-sets "flatten out". This makes the FL convenient to fast approximate functions and for creating efficient control loops. Since it converts quantifiable into something less concrete, the FL is easy to use in noisy environment as well as when the result of measuring is imprecise.

For expert systems based on symbolic logic, the numeric input is to be transformed to symbols. There are more solutions to do that. The easiest is if instead of saying numbers, the expert tells something like "too much", "not enough", etc. However, this cannot be automated. Numeric data are usually stored in databases or data warehouses, so it would be desirable to have automated transformation with special regard to the huge amounts of data stored.

There are a variety of statistical clustering algorithms available though they are very rigid and sensitive to noise. Rigidity means that the boundaries of the clusters are strict; so elements near the boundaries may belong to different clusters, even if they are much more similar to each other than to other elements in their own clusters. There is no smooth transition between the clusters, which leads to the "paradox of the pile of sand", i.e. when how many grains of sand have to be taken from the pile to stop being a pile. Sensitivity to noise means that if the majority of the elements are near to one of several centres (forming several sets) though there are some elements far from these centres, either the number of clusters increases dangerously or the structure of the clusters degenerates. Therefore the problem statement at this stage is that **it is not known, how to perform clustering, which is neither rigid nor sensitive to noise.**

As fuzzy logic handles the "paradox of the pile of sand" [11] it offers itself to be an adequate solution. The idea of the fuzzy clustering is not new at all. The first solution was introduced by Ruspini in 1969 [8] and since then loads of articles were produced.<sup>4</sup> The fuzzy clustering algorithms matured and now we have really robust solutions e.g. [3]. Why then we not only choose one of the existing algorithms instead of developing a new conception? The robust fuzzy clustering solutions usually have a "parameter", which is to be estimated. We want to presume a decision maker with no understanding of statistics, fuzzy logic or clustering. Therefore we cannot have an appropriately chosen parameter. The other reason is, that according to the application domain there is no need to go with all the features provided by the existing fuzzy clustering solution in return the complexity can be reduced. On the other hand some conditions are to be fulfilled strictly, thus some of the common solutions of reducing computing capacity may not be applied.

There is usually a huge amount of numeric data stored in databases or data warehouses. These are to be retrieved and clustered automatically. As the resulting fuzzy sets are to be handled together with soft information given on ordinal or nominal scales — where no measure or distance is defined — frequent coverage of the numeric input is needed. A question arises: can we expect Ruspini partitions? Probably some domains are appropriate for infrequent coverage — where the advantages of the fuzzy interpolation may be exploited — could be identified,

---

<sup>4</sup> Google finds 9,840 web pages for "fuzzy clustering" 23/06/03.

though it is out of the area of investigation in this paper.

There is no need for multi-step fuzzy reasoning, as since transformed to appropriate scale the fuzzy sets will be processed together with values of symbolic attributes using “if... then” rules. Of course, the symbolic part of reasoning may be multi-stepped.

As once the numeric input is transformed into fuzzy sets they are further handled with symbolic logic a strange situation evolves that there is no need to know the centres of the clusters. This can also reduce the complexity.

The nowadays attempts usually aim to give a general solution to clustering, where the different modes of reduction of complexity and/or computing time are adjusted through different parameters. We suggest a new approach to develop particular conceptions for different application domains – of course not for each application separately – benefiting from the handling complexity at the appropriate level.

The definition of our problem is now modified: **how to perform a clustering, which is neither rigid nor sensitive to noise, benefiting from the properties of the application domain, reducing the complexity as much as possible?**

## II. KBS SHELL DOCTUS

Our Knowledge Expert System Shell Doctus uses symbolic representation to reason about *cases*. Cases can be anything that we can describe from all important aspects. In Doctus it is done with *attributes* and their *values*. One value<sup>5</sup> of every attribute is assigned to each of the cases. The values of the attributes are connected with “if... then” rules.

In deduction (there are about 200 software pieces in the world capable of that) rules are formulated by the *expert* of the domain. The *knowledge engineer* acquires the rules from the expert. The knowledge acquisition in form of rules has two conditions: expert level knowledge is needed, which means few thousand of cognitive patterns [1] and [2]; and explicit knowledge is needed, which means that the expert has to be able to put it into words – to formulate the “if... then” rules [5]. When the deductive knowledge-base is built the execution of rules for a case gives an evaluation – a decision proposal. To avoid redundant storage and to facilitate the articulation of rules the principle of hierarchical reasoning is implemented. The expert and the knowledge engineer build a hierarchy of attributes so the number of rules could be reduced and therefore they are easier to handle. We call this hierarchy a *deductive graph*. While Doctus does the forward chaining in traditional way, the backward chaining is supported with an explanatory option.

Induction is used (about a dozen of the previously mentioned software packages are capable of induction), if the expert cannot articulate the rules but he is experienced enough (a few dozen cases with outcomes); this experience can be used to reason about a new case. The case base is use to extract rules from the experience. We use this approach, because we do not want to quantify concepts.

Induction in Doctus is based on a modified ID3 algorithm, ancestor of which was originally introduced by Quinlan [6]: Let’s presume that all cases form a disordered set, where the order is defined as homogeneity by *benchmark* values (values of outcome attribute), which means that cases in one subset have the same benchmark value. The attribute is searched, which contributes the most to the order. The attributes are taken one-by-one forming subsets according to their values. Their strength in making order is measured by an entropy-gain (informativity) calculating algorithm<sup>6</sup>. The most informative attribute is chosen and the first level subsets are formed according to its values. These subsets are further divided using the same algorithm until all subsets are homogenous by benchmark values. This qualitative classification of the cases by the benchmark values usually appears in the form of decision tree. In Doctus it is called the *modeling graph*, which can be converted into production rules.

Induction is the symbolic machine learning. Doctus uses non-incremental unsupervised learning. It is also supplemented with Knowledge Import feature, which makes possible to retrieve both hard data and soft information from external sources. To make the shell an efficient data mining tool, there is only a need for a robust automated cluster-analysis.

Currently Doctus is equipped with k-means statistical clustering algorithm which uses Euclidean metrics, and is fully automated. This clustering is performed on numeric data during the retrieval, thus for the cases the knowledge base store only to which cluster they do belong.

Clusters are described by two parameters, their centres and the dispersion. (Fig. 1) This information is superfluous to the decision maker, who we presumed to be expert or higher of his own domain, not of statistics. The only data might be useful is how many cases are in particular clusters (see the second column of the dialog box on Fig. 1), which is not part of cluster-analysing. The decision maker might be also interested in place of the borders of the clusters, and he may also want to modify them. This is also to be considered.

Hence our problem definition is refined again: **how to perform a clustering, which is neither rigid nor sensitive to noise, benefiting from the properties of the application domain, reducing the complexity as much as possible, and supplying the decision maker with useful information enabling the possibility of interaction?**

<sup>6</sup> Determining **informativity** ( $I_b$ ) of attribute  $b$  is as follows: Let  $C$  be the set of cases in node,  $a$  the benchmark,  $a_1...a_n$  its values, and  $w_{a1}...w_{an}$  ( $\sum w_{ai} = 1$ ) their rates in set  $C$ . Then entropy of benchmark in set  $C$  can be written:  $E_C = -\sum w_{ai} \log_2 w_{ai}$ . Let  $b_1...b_n$  be the values of attribute  $b$ ,  $\beta$  is a set of them. Disjoint  $\beta$  into not empty subsets  $\beta_1... \beta_m$ . Then  $\cup_i \beta_i = \beta$ . Disjoint  $C$  into subsets  $C_1...C_m$  being attribute  $b$  of all elements of  $C_i$  in  $\beta_i$  for each  $i$ . Let  $w_i$  be the weight of  $C_i$  in  $C$ . ( $\sum w_i = 1$ ). Then  $I_b = E_C - \sum w_i E_{C_i}$ , in words informativity is an increment of entropy resulted from disjoining  $\beta_1... \beta_m$ . Real output of computing is  $I_{bmax}$  of optimal selection. **Density** of it is:  $D_b = w_C I_{bmax} / E_C$ , where  $w_C$  is the number of elements in set  $C$ . Real computing is some more altered by ‘Unknown’, ‘Don’t care’ and distributed values.

<sup>5</sup> One value is default but Doctus can also handle “Unknown”, “Don’t care” and distributed values.

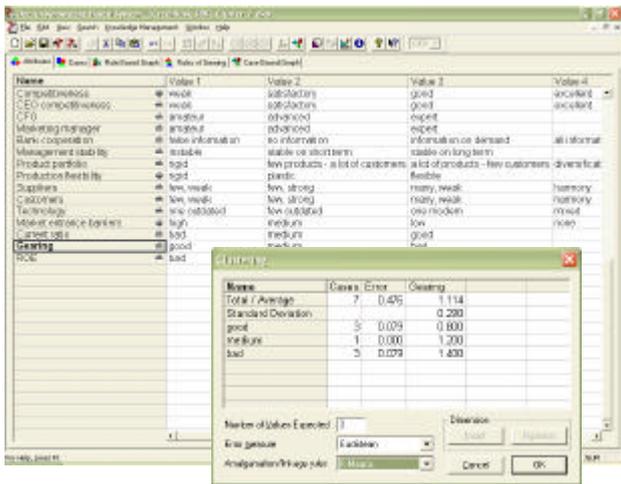


Figure 1: Statistical Clustering in Doctus.

### III. A SYMBOLIC-FUZZY HYBRID – HOW?

The most frequent hybrid representation is when NN's are combined with FL. This occurs when the input of a NN is a Fuzzy-representation. This satisfies the concept of the fittingly chosen function, but it does not cause significant efficiency improvement.

The solutions in which the hybrid's parents are the symbolic representation and the NN's are rare. Namely the two representations are very different, so it is hard to combine them, though there are attempts e.g. [4] and [12].

The combination of the fuzzy logic with symbolic expert systems is almost old as the fuzzy logic itself, as the first this attempt came from Zadeh from 1973 [10]

The fortunate combining of symbolic representation with FL is when we use Fuzzy-logic to convert quantifiable signs into symbols. Alas its reverse use is more common; it is dangerous. This second solution is usually hidden behind a pretty algorithm; it quantifies the concepts against reasonability, that is to say it converts the elements given on ordinary scale onto interval or proportion scale. We quit further examination of the preceding reverse use on the input side.

Better version's biggest trouble begins when a high number numerical signs are to be transferred onto ordinary scale: where the Fuzzy-sets should be placed? Manually it would be easy; anyone could see where the fuzzy sets (i.e. their membership functions) should be (see Fig. 2). However in situations with big amounts, automation is needed. There are FL solutions functioning with infrequently covered fuzzy-sets using rule interpolation.

If we want to handle Fuzzy-sets with symbols jointly, we need frequent coverage. To make the computing easy we use only triangular and trapezoidal fuzzy sets. In expert system applications it would mean four types of fuzzy sets, due to the requirement that ends of the scale should be covered with "open" sets. The used fuzzy set types are called Z (left end of the scale on Fig. 2), S (right end of the Fig 2), ? (triangular) and ? (trapezoidal).

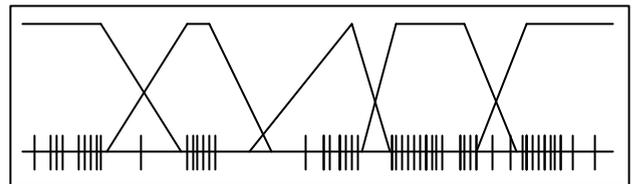


Figure 2: Guesstimated places of the fuzzy sets.

The following method leads to acceptable solution, and it is universal:

If we put the results of measuring on an axis, we can roughly see where the Fuzzy-sets should be located. This cannot be automated. However it is worth to observe how we actually do the guesswork. This observation has led us to the construction of the method. We need to construct a function, for which the inputs are numbers from the previous axis (x). For a particular input, the output of the function will be the reciprocal distance from its neighbour (1), as shown on Fig 3.

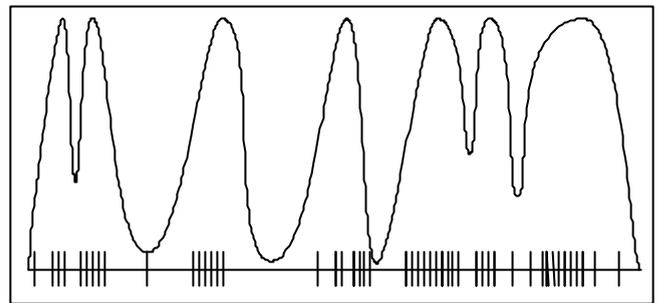


Figure 3: Draft picture of the function.

The method is not sensitive on the chosen neighbour, so for the most correctness, the average of the distances from the left and the right neighbour is chosen. Of course other combination considering e.g. the neighbours' neighbours too, could be also used, though it is not worth doing it.

$$y_i = \frac{1}{\Delta x_i}, \text{ where } \Delta x_i = \frac{(x_{i+1} - x_i) + (x_i - x_{i-1})}{2} \quad (1)$$

From this function triangular fuzzy-sets can be created, so that the top of the triangle is at the local maximum of the function (y' = 0). The bottom point of the triangle is to be at the inflexion point of the neighbour "hill" (y'' = 0). By doing so, we get certainly not bad base. It is important to have overlapping sets; otherwise the result of measuring may be indefinable.

It can occur that the function will have plenty of local maximums. Therefore the function should be smoothed with numerical methods. The reasonable limit is if we maximize the number of "hills" at 5-7. Local maximums may appear so the peak of the top undulates, namely plenty of maximums get very close to each other without going downwards. To "pick the five highest tops" is not the solution.

We should consider the properties of the application domain again. There is always a minimum of the distances from the neighbours. Though this is true for any domain, if measuring distance between cities in miles while the smallest interval is the extent of the smallest elementary particle, it gives no advantage. However, in the domain of

business decisions there are measures as money (L, € \$), number of pieces purchased or number of times of customers' visits. These quantities are not likely to cover many orders of magnitude therefore it is reasonable to find the minimum of  $\Delta x$  and use it to express the distances as integer multiples of it. With this step the function became normalised, with the max value of 1, which makes it more similar to the membership functions of the fuzzy sets.

Integrals of the sections of the function and measuring the distances between the hills can be used to avoid placing several fuzzy sets very near, while leaving the rest of the scale covered with a single one. The other consequence of this step is that the triangular fuzzy sets will not be sufficient to use, also using trapezoidal fuzzy sets better result – nicer picture – is achieved. (Fig. 4) The limit of the number of the fuzzy sets is given by the user. As this process is to be repeated for every each numeric attribute, it has to be computed fast. Therefore the function will not be calculated, only the points will be estimated, that are used for the construction of the fuzzy sets. The precision of the estimation is not larger than the minimum of  $\Delta x$ .

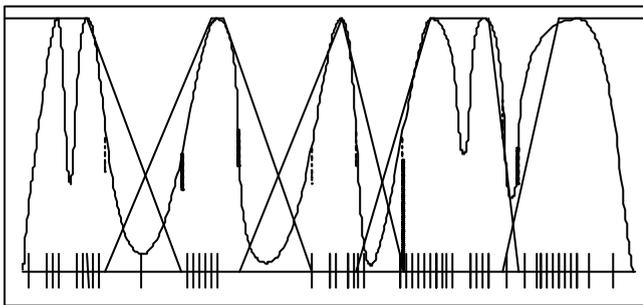


Figure 4: Draft of the fuzzy sets placement by the method

What happens, if the clusters are ready? Triggering off the rules to evaluate a particular case it will frequently happen that the number describing the case falls into more than one fuzzy set. This will make more than one rule to be triggered and the membership function will determine the ratio of the validity of each rule. Therefore it would be easier to use Ruspini partition, as it is likely to have 1 for the sum of membership functions at any point. However it is not known, how to determine if the Ruspini partition is acceptable for a particular decision situation. The other reason that the Ruspini partition is not adopted is that we would like to give the user access to fine-tune the fuzzy sets. Ruspini partition will appear too symmetric to the user if only triangular and trapezoidal fuzzy sets are used, which is essential to save computing capacity.

If there are depending attributes with multiple fuzzy-clustered inputs, operations on fuzzy sets are also to be used to determine the ratio of the validity of the rules. For this stage we adopt traditional maxmin operations defined by Zadeh [9].

Fuzzy reasoning usually follows the steps fuzzyfication > fuzzy operations and/or reasoning > defuzzyfication. The first one is described above; the second one is not needed as the reasoning happens in symbolic logic. Usually the third one is also not needed, though in some situations there can be a need for numerical outputs. For this we use COG defuzzyfication method because of the low computing requirement.

There are other possibilities to successful integration of fuzzy logic into Doctus KBS. In our experience it happened earlier that the knowledge base was a useful thing to discover which are the attributes or rules that different experts or expert groups are having a heated debate. If the experts or expert groups belong have a different area of expertise, they may not understand each other, as different disciplines have different vocabularies. When the SL located the subject of the debate the FL may make it clearer. The approach is the same as if we are building a fuzzy expert system. The experts are to articulate the belongingness of different cases to the range of the attribute. However it is to be done with significantly smaller number of attributes – usually from 60-80 to 2-3 – which is an unarguable advantage. The same approach may be used to discover the overlappings of the terms used in symbolic knowledge bases.

#### IV. CONCLUSION

In this paper a new conception of fuzzy clustering was introduced for symbolic knowledge-based system Doctus. Special properties of the application domain were considered and highly exploited. Other useful elements of an integrated symbolic-fuzzy hybrid system were also highlighted.

The developed conception is under implementation; therefore the conception itself is also likely to change in the near future. New component problems were also identified: it is not known how to find out if the use of Ruspini partitions may be realized and it is not known if the advantages of the infrequent coverage of the domain and the interpolation can be exploited, and which are the conditions for that.

#### V. REFERENCES

- [1] F. Gobet and H. A. Simon, "Recall of random and distorted chess positions: Implications for the theory of expertise", *Memory & Cognition* 24(4), 1996, pp. 493-503.
- [2] F. Gobet and H. A. Simon, "Templates in Chess Memory: Mechanism for Re-calling Several Boards", *Cognitive Psychology* 31, 1996, pp. 1-40.
- [3] J. Leski, "Towards a Robust Fuzzy Clustering", *Fuzzy Sets and Systems*, 137 (2) Elsevier Science, 2003, pp. 215-233.
- [4] K. McGarry, S. Wermter and J. MacIntyre, "Hybrid Neural Systems From Simple Coupling to Fully Integrated Neural Networks", *Neural Computing Surveys*, 2, pp. 62-93, 1999.
- [5] M. Polanyi, *The Tacit Dimension*, Doubleday & Company, Inc, 1966.
- [6] J. R. Quinlan, "The Induction of Decision Trees", *Machine Learning*, 1(1), 1986, pp. 81-106.
- [7] B. A. Russell, *Human Knowledge: Its Scope and Limits*, Routledge, London, 1994.
- [8] E. H. Ruspini, "A New Approach to Clustering", *Information Control*, 15(1), 1969, pp. 22-32.
- [9] L. A. Zadeh, "Fuzzy sets", *Information and Control*, No. 8, 1965, pp. 338-353.

- [10] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Transactions on Systems, Man and Cybernetics* SMC-3, 1973, pp. 28-44.
- [11] L. A. Zadeh, *Fuzzy Logic Applications*, Kluwer Academics, 1992.
- [12] Z.-H. Zhou, Y. Jiang and S.-F. Chen, "A General Neural Framework for Classification Rule Mining", *International Journal of Computers, Systems and Signals*, 1(2), pp. 154-168, 2001.