Logic-based granular prototyping

[1]Andrzej Bargiela, [2]Witold Pedrycz, [3]Kaoru Hirota

[1] Department of Computing and Mathematics
The Nottingham Trent University, Nottingham NG1 4BU, UK
(andre@doc.ntu.ac.uk)
[2] Department of Electrical & Computer Engineering
University of Alberta, Edmonton, Canada
(pedrycz@ee.ualberta.ca)
and
Systems Research Institute, Polish Academy of Sciences
01-447 Warsaw, Poland
[3] Dept. of Computational Intelligence and Systems Science
Interdisciplinary Graduate School of Science and Engineering
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama-city 226-8502, Japan

Abstract

A fuzzy logic based similarity measure is introduced as a criterion for the identification of structure in data. An important characteristic of the proposed approach is that cluster prototypes are formed and evaluated in the course of the optimization without any a-priori assumptions about the number of clusters. The intuitively straightforward compound optimization criterion of maximizing the overall similarity between data and the prototypes while minimizing the similarity between the prototypes has been adopted. It is shown that the partitioning of the pattern space obtained in the course of the optimization is more intuitive than the one obtained for the standard FCM. The local properties of clusters (in terms of the ranking order of features in the multi-dimensional pattern space) are captured by the weight vector associated with each cluster prototype. The weight vector is then used for the construction of interpretable information granules.

Keywords: Granular prototyping, clustering, logic-based optimization, data mining.

1. Introduction

Investigation of the various clustering techniques has received much attention from researchers in data mining [1, 2, 12, 13], pattern recognition [4, 8] and fuzzy modeling [3, 7, 14, 17, 18, 19]. The reason for that is that clusters represent information granules that lift the analyst onto a higher level of abstraction compared to the level represented by the individual data items. When dealing with complex problems, granulation of the problem information domain is of crucial importance for the development of approximations to system behavior. Granular computing is a methodology that explicitly recognizes the fundamental importance of information processing at the various levels of abstraction. As such, it relies on efficient and effective techniques for identification of granular prototypes in data and granular constructs, in general. There have been several pursuits along this lines [9], [14], [16] yet the area is still in its early development stage. This study contributes a novel approach to granular clustering, one that is based on optimization of a fuzzy logic -based similarity measure.

One of the objectives that we have set in this study was to achieve a sequential procedure in which prototypes emerge from data in the order of their significance. This will avoid the need for any pre-judging of what structure is present in the data. The second objective was to ensure the explorative capabilities of the logic-based prototyping in the whole of the pattern space. This is achieved by defining a suitable performance index (objective function). And the final objective was to achieve a characterization of the prototypes that guides formation of interpretable granules.

In the paper, we proceed with a top-down presentation by first discussing the essence of the method and then elaborating on all pertinent details. The experimental part of the study consists of a 2-dimensional pattern intended to illustrate the proposed clustering and granulation mechanisms. We compare the results to those obtained with the Fuzzy C-Means (FCM), which is considered as a de-facto standard in fuzzy clustering.

2. Problem formulation

The problem formulation comprises several main components such as a format of data, a form of the performance index and a general organization of the search for data.

In this study, we are concerned with data (patterns) distributed in an n-dimensional [0,1] hypercube. In what follows, we will be treating the data as points in $[0,1]^n$, say $\mathbf{x} \in [0,1]^n$. In general we are concerned with N patterns (data points) $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_N$. The “standard” objective of the clustering method (no matter what is its realization) is to reveal a structure in the data set and to present it in a readable and easily comprehensible format. In general, we consider a collection of prototypes to be a tangible and compact reflection of the overall structure. In the approach undertaken here we adhere to the same principle. The prototypes representing each cluster are selected as some elements of the data set. Their selection is realized in such a way that they (a) match (represent) the data to the highest extent while (b) being evidently distinct from each other. These two requirements are represented in the objective function guiding the clustering process. In the sequel we define the detailed components of the optimization. Since the elements in the unit hypercube can be viewed as fuzzy sets, we can take advantage of well-known logic operations developed in this domain. The notion of similarity (equality) between membership grades plays a pivotal role and this concept is crucial to the development of the clustering mechanisms.

2.1. Expressing similarity between two fuzzy sets

The measure of similarity between two fuzzy sets (in this case a datum and a prototype) $\mathbf{x}=[x_1\ x_2 \ldots x_n]^T$ and $\mathbf{v}$ $(=[v_1\ v_2 \ldots v_n]^T)$ is defined by incorporating the operation of matching ($\equiv$) encountered in fuzzy sets. The following definition will be used

$$\mathrm{sim}(\mathbf{x},\mathbf{v}) = \mathop{\mathrm{T}}_{i=1}^{n}(w_i^2 s(x_i \equiv v_i)) \tag{1}$$

In the above, T( ) and s( ) denote a t-norm and s-norm, respectively. The weights (w_i) quantify an impact of each coordinate of the feature space $[0,1]^n$ on the final value of the similarity index sim( ). When convenient, we will be using a notation sim($\mathbf{x}$, $\mathbf{v}$; $\mathbf{w}$) to emphasize the role played by the weight vector. The similarity between two membership grades is rooted in the fundamental concept of similarity (or equivalence) of two fuzzy sets (or sets). Given two membership grades a and b, (the values of a and b are confined to the unit interval), a similarity level a $\equiv$ b is computed in the form

$$a \equiv b = (a \rightarrow b)t(b \rightarrow a) \tag{2}$$

where the implication operation ($\rightarrow$) is defined as a residuation (ϕ-operator) [5], [6] that is

$$a \rightarrow b = \sup\{c \in [0,1] \mid atc \le b\} \tag{3}$$

The above expression of the residuation is induced by a certain t-norm. The implication models a property of inclusion; referring to (3) we note that it just quantifies a degree to which a is *included* in b. The *and* connective used in (2) translates it into a verbal expression

(a is included in b) *and* (b is included in a) (4)

which in essence quantifies an extent to which two membership grades are equal. As a matter of fact, the origin of this definition traces back to what we know well in set theory: we say two sets A and B are equal if A is included in B and B is included in A. Moving on with the definition, the visualization of the similarity treated as a function of “a” with “b” regarded as a parameter of this index is included in Figure 1. As expected, it attains 1 if and only if “a” is equal to “b”. The function decreases when moving away from “b”. It is however quite asymmetric where this asymmetry arises as a consequence of the implication operations being used in the definition. Note also that the change in the t-norm in the basic definition (2) does not affect the form of the similarity index.

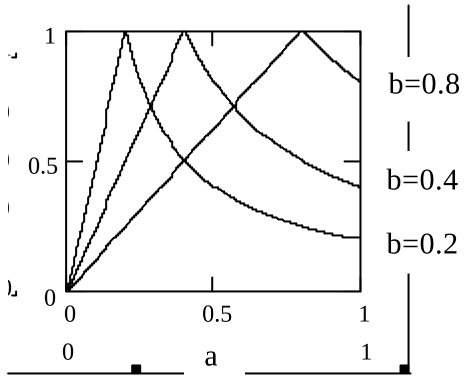


Figure 1. The similarity index a $\equiv$b regarded as a function of a for selected values of b; the residuation is induced by the product operation, a→b = min(1,b/a)

2.2. Performance index (objective function)

Performance index reflects the character of the underlying clustering philosophy. In this work we have adopted performance index that can be concisely described in the following manner. A prototype of the first cluster $\mathbf{v}_1$ is selected as one of the elements of the data set ($\mathbf{v}_1 = \mathbf{x}_j$ for some j=1,2,…,N) so that it maximizes the sum of the similarity measures of the form

$$\sum_{k=1}^{N} \mathrm{sim}(\mathbf{x}_k, \mathbf{v}_1; \mathbf{w}_1) \Rightarrow \mathrm{Max}_{\mathrm{v}_1, \mathrm{w}_1} \tag{5}$$

with sim($\mathbf{x}_k$, $\mathbf{v}_{1,}\mathbf{w}_1$) defined by (1). Once the first cluster (prototype) has been determined (through a direct search across the data space with a fixed weight vector and subsequent optimization of the weights treated as another part of the optimization process), we move on to the next cluster (prototype) $\mathbf{v}_2$ and repeat the cycle. The form of the objective function remains the same throughout the iterative process but we combine now the maximization of the sum of similarity measures (5) with a constraint on the relatrive positioning of the new prototype. The point is that we want this new prototype, say $\mathbf{v}_2$, not to “duplicate” the first prototype by being too close to it and thus not representing any new part of the data. To avoid this effect, we now consider the expression of the form

$$(1 - \mathrm{sim}(\mathbf{v}_2, \mathbf{v}_1; \mathbf{0}))\sum_{k=1}^{N} \mathrm{sim}(\mathbf{x}_k, \mathbf{v}_2; \mathbf{w}_2) \tag{6}$$

where the first factor 1- sim($\mathbf{v}_1$, $\mathbf{v}_2$; 0) expresses the requirement of $\mathbf{v}_2$ to be as far apart from $\mathbf{v}_1$ as possible. The above expression has to be maximized with respect to $\mathbf{v}_2$ and this optimization has to be carried out with the weight vector ($\mathbf{w}_2$) involved. In the sequel, we proceed with the determination of the third prototypes $\mathbf{v}_3$, etc. In general, the optimization of the L-th prototype follows the expression

$$\begin{aligned} \mathrm{Q(L)} &= (1 - \mathrm{sim}(\mathbf{v}_L, \mathbf{v}_{L-1}; \mathbf{0})(1 - \mathrm{sim}(\mathbf{v}_L, \mathbf{v}_{L-2}; \mathbf{0})\ldots \\ &\ldots (1 - \mathrm{sim}(\mathbf{v}_L, \mathbf{v}_1; \mathbf{0}))\sum_{k=1}^{N} \mathrm{sim}(\mathbf{x}_k, \mathbf{v}_L; \mathbf{w}_L) \end{aligned} \tag{7}$$

As noted, this expression takes into account all previous prototypes when looking for the current prototype. Interestingly, the performance index to be maximized is a decreasing function of the prototype index, that is $L_1 < L_2$ implies that $Q(L_1) \le Q(L_2)$.

Another observation of interest is that the first prototype constitutes the best representative of the overall data set. Subsequent prototypes are, in effect, the best representatives for the more detailed partitions of data.

So far, we have not touched the issue of the optimization of the weight vector associated with the prototype that is an integral part of the overall clustering. The next section provides a solution to this problem.

3. Prototype optimization

Let us concentrate on the optimization of the performance index in its general form given by (7). Apparently the optimization consists of two phases, that is (a) the determination of the prototype ($\mathbf{v}_L$) and the optimization of the weight vector ($\mathbf{w}_L$). These two phases are intertwined yet they exhibit a different character. The prototype is about enumeration out of a finite number of options (patterns in the data set). The weight optimization has not been formulated in detail and now requires a prudent formulation as a constraint type of optimization (without any constraint the task may return a trivial solution).

Referring to (7) we observe that it can be written down in the form

$$\mathrm{Q(L)} = \mathrm{G}\sum_{k=1}^{N} \mathrm{sim}(\mathbf{x}_k, \mathbf{v}_L; \mathbf{w}_L) \tag{8}$$

Note that the first part of the original expression does not depend on $\mathbf{w}_L$ and can be treated as constant with this regard,

$$\mathrm{G} = (1 - \mathrm{sim}(\mathbf{v}_L, \mathbf{v}_{L-1}; \mathbf{0})(1 - \mathrm{sim}(\mathbf{v}_L, \mathbf{v}_{L-1}; \mathbf{0})\ldots(1 - \mathrm{sim}(\mathbf{v}_L, \mathbf{v}_1; \mathbf{0})) \tag{9}$$

We impose the following constraint on $\mathbf{w}_L$ requesting that its components are located in the unit interval and sum up to 1,

$$\sum_{j=1}^{n} w_{Lj} = 1 \tag{10}$$

The optimization of (8) with respect to $\mathbf{w}_L$ for a fixed prototype $\mathbf{v}_L$ is expressed as

$$\max \mathrm{Q(L)} = \mathrm{G}\sum_{k=1}^{N} \mathrm{sim}(\mathbf{x}_k, \mathbf{v}_L; \mathbf{w}_L) \tag{11}$$

subject to $\sum_{j=1}^{n} w_{Lj} = 1$

The detailed derivations of the weight vector is done through the technique of Lagrange multipliers. First, we form an augmented form of the performance index

$$V = G\sum_{k=1}^{N}\left\{\mathop{T}_{j=1}^{n}(w_j^2 s(x_{kj} \equiv v_{Lj}))\right\} - \boldsymbol{l}(\sum_{j=1}^{n} w_{Lj} - 1) \tag{12}$$

To shorten the expression, we introduce the notation $u_{ks} = x_{ks} \equiv v_{Ls}$. The derivative of V taken with respect to w_{Ls} (the s-th coordinate of the weight vector) is set to zero and the solution of the resulting equations gives rise to the optimal weight vector

$$\frac{dV}{dw_{Ls}} = 0 \qquad \frac{dV}{d\boldsymbol{l}} = 0 \tag{13}$$

The derivatives can be computed once we specify t- and s-norms. For the sake of further derivations (and ensuing experiments), we consider a product and probabilistic sum as the corresponding models of these operations. Furthermore we introduce the abbreviated notation $u_{ks} = x_{ks} \equiv v_s$ for k=1, 2, …, N and s=1, 2, …,n. Taking all of these into account, we have

$$\frac{dV}{dw_s} = G\sum_{k=1}^{N}\frac{d}{dw_s}\{A_{ks} w_s^2 s u_{ks}\} - \boldsymbol{l} = 0 \tag{14}$$

where

$$A_{ks} = \mathop{T}_{\substack{j=1\\ j\neq s}}^{n}(w_j^2 s u_{ks})$$

The use of the probabilistic sum (s-norm) in (14) leads to the expression

$$\begin{aligned}&\frac{d}{dw_s}\{A_{ks} w_s^2 s u_{ks}\} = A_{ks}\frac{d}{dw_s}(w_s^2 + u_{sk} - w_s^2 u_{sk}) = \\ &= 2A_{ks} w_s (1 - u_{ks})\end{aligned} \tag{15}$$

and, in the sequel

$$\frac{dV}{dw_s} = 2Gw_s \sum_{k=1}^{N} A_{ks}(1 - u_{ks}) - \boldsymbol{l} = 0 \tag{16}$$

From (16) we have

$$w_s = \frac{\boldsymbol{l}}{2G\sum_{k=1}^{N} A_{ks}(1 - u_{ks})} \tag{17}$$

The form of the constraint, $\sum_{j=1}^{c} w_j = 1$, produces the following expression

$$\frac{\boldsymbol{l}}{2}\sum_{j=1}^{c}\frac{1}{G\sum_{k=1}^{N} A_{kj}(1 - u_{kj})} = 1 \tag{18}$$

or

$$\frac{\boldsymbol{l}}{2} = \frac{1}{\sum_{j=1}^{c}\frac{1}{G\sum_{k=1}^{N} A_{kj}(1 - u_{kj})}} \tag{19}$$

Finally inserting (19) into (17) the s-th coordinate of the weight vector reads as

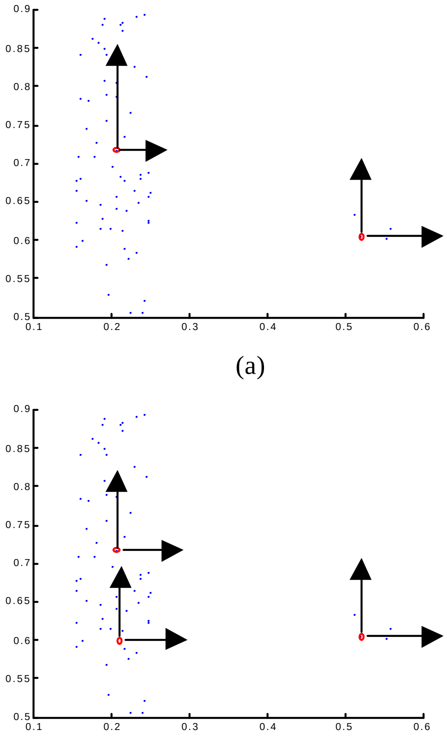
$$w_s = \frac{1}{\sum_{j=1}^{c}\frac{\sum_{k=1}^{N} A_{ks}(1 - u_{ks})}{\sum_{k=1}^{N} A_{kj}(1 - u_{kj})}} \tag{20}$$

Summarizing the algorithm, it essentially consists of two steps. We try all patterns as a potential prototype, for each choice optimize the weights and find a maximal value of Q(L) out of N options available. The one that maximizes this performance index is treated as a prototype. It comes with an optimal weight vector $\mathbf{w}_L$. Each prototype comes with its own weight vector that may vary from prototype to prototype. Bearing in mind the interpretation of these vectors we can say that they articulate the "local" characteristics of the feature space of the patterns. The lower the value of the weight for a certain feature (variable), the more essential the corresponding feature is.

4. Illustrative example

The logic-based granular prototyping is illustrated here on a synthetic 2-dimensional data. We restrict ourselves to showing just 2 and 3 prototypes but, of course, the prototyping could be carried out to resolve finer detail in data. In this particular example however, where we have only two distinct data groupings and the performance index stabilizes for the number of prototypes greater than 2. This means that the structure of data has already been captured with two prototypes. The data, with corresponding 2 and 3 prototypes identified by the algorithm is depicted in Figure 2.

It is worth noting that the weights vector associated with the prototypes gives an idea about the importance of individual features for each prototype. We indicate these vectors in Figure 2, overlaying them onto the data plot. It is worth pointing out that moving from 2- to 3 prototypes changes the geometry of the clusters in the larger data grouping and this is duly reflected in the relative values of the weights for the two features.



(a)

(b)

Figure 2. Synthetic data set with two unbalanced data groupings. (a) two prototypes, (b) three prototypes; with corresponding weights.

In order to assess the relative merits of the logic -based granular prototyping we compare the partitioning of the data achieved with the similarity measure to the one obtained with standard FCM. The results for 2 and 3 prototypes are illustrated in Figures 3-6.

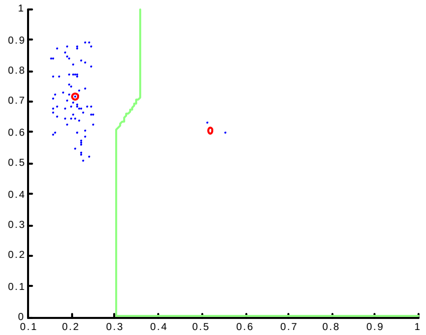


Figure 3. Pattern space partitioning with logic-based prototyping for 2 prototypes.

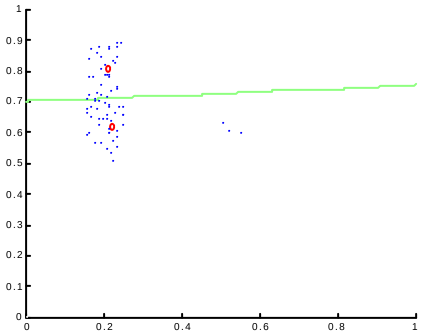


Figure 4. Pattern space partitioning with FCM for 2 prototypes.

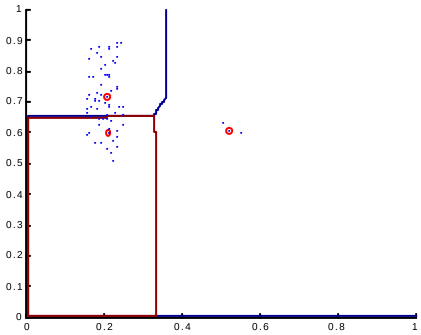


Figure 5. Pattern space partitioning with logic-based prototyping for 3 prototypes.

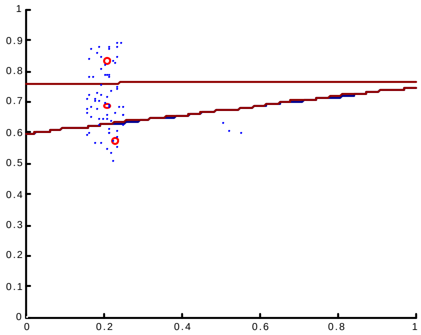


Figure 6. Pattern space partitioning with FCM for 3 prototypes.

It is clear that the inherent property of FCM, that of allowing larger data structures to dominate the smaller ones in the pattern space is amply demonstrated on through the partitioning of the space. Of course it is possible to restrict this happening by including further components in the FCM objective function dealing with penalizing the proximity of prototypes, but this only goes to emphasize the advantage of taking an alternative approach as proposed here.

5. Conclusions

We have proposed a new logic-based granular prototyping that displays a number of unique features:

- The algorithm is sequential in that there is no need to pre-specify the number of prototypes (thus guess the structure of data) and the algorithm continues to search for prototypes that highlight a progressively finer detail in data until the performance index 'flattens out'. The most 'essential' prototypes are identified first and further prototypes are identified in the course of the optimization.
- The identification of prototypes is based on well-defined fuzzy sets operations and the logic-based character of processing gives easily interpretable results.
- The weight vector associated with the prototypes gives a good indication of the relative importance of individual features for each prototype thus allowing easy granular interpretation of prototypes.

We should stress, however although the properties of the logic-based prototyping are attractive for data analysis, the organization of the search for the structure as arranged here could be computationally intensive. This is particularly so for large data sets. Consequently this method should be considered as a complement to other clustering techniques.

Acknowledgments

The support from Engineering and Physical Sciences Research Council of UK (EPSRC), the Natural Sciences and Engineering Research Council of Canada (NSERC) and ASERC (Alberta Software Engineering Research Consortium) is gratefully acknowledged.

6. References

[1] M.R. Anderberg, *Cluster Analysis for Applications*, Academic Press, N. York, 1973.

[2] A. Bargiela, Interval and ellipsoidal uncertainty models, In: W. Pedrycz (ed.) *Granular Computing*, Physica Verlag, Heidelberg, 2001,pp.23-57.

[3] A. Bargiela, W. Pedrycz, Recursive information granulation: Aggregation and interpretation issues, *IEEE Trans. SMC-B*, to appear.

[4] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York, 1981.

[5] B. Bouchon-Meunier, M. Rifqi, S. Bothorel, Towards general measures of comparison of objects, *Fuzzy Sets and Systems*, 84, 2, 1996, 143-153.

[6] A. Di Nola, S. Sessa, W. Pedrycz, E. Sanchez, *Fuzzy Relational Equations and Their Applications in Knowledge Engineering*, Kluwer Academic Press, Dordrecht, 1989.

[7] M.Delgado, A.F. Gomez-Skarmets, F. Martin, A methodology to model fuzzy systems using fuzzy clustering in a rapid-prototyping approach, *Fuzzy Sets and Systems*, vol. 97, no.3, 1998, 287-302.

[8] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, 2nd edition, J. Wiley, N. York, NY, 2001.

[9] B. Gabrys, A. Bargiela, General fuzzy Min-Max neural network for clustering and classification, *IEEE Trans. on Neural Networks*, Vol. 11, No. 3, pp. 769-783, 2000.

[10] K. Hirota, W. Pedrycz, Directional fuzzy clustering and its application to fuzzy modeling, *Fuzzy sets and systems*, 80, 3, 1996, 315-326.

[11] F. Hoppner et al., *Fuzzy Cluster Analysis*, J. Wiley, Chichester, 1999.

[12] A. Kandel, *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley, Reading, MA, 1986.

[13] W. Pedrycz, *Computational Intelligence: An Introduction*, CRC Press, Boca Raton, Fl, 1997.

[14] W. Pedrycz, A. Bargiela, Granular clustering: a granular signature of data, *IEEE Trans. on Systems, Man, and Cybernetics –B*, 2002, to appear

[15] T. A. Sudkamp, R. J. Hammell II, Granularity and specificity in fuzzy function approximation, in Proc. *NAFIPS-98*, 105-109, 1998.

[16] P. K. Simpson, Fuzzy Min-Max neural networks – Part1: Classification, *IEEE Trans. on Neural Networks*, Vol 3, No. 5, pp. 776-86, September 1992.

[17] L. A Zadeh, Fuzzy sets and information granularity, In: M.M. Gupta, R.K. Ragade, R.R. Yager, eds., *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 1979, 3-18.

[18] L. A. Zadeh, Fuzzy logic = Computing with words, *IEEE Trans. on Fuzzy Systems*, vol. 4, 2, 1996, 103-111.

[19] L. A. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 1997, pp. 111-117.