2006 IEEE International Conference on Fuzzy Systems
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

A method for the fuzzification of categorical variables

Etienne Jodoin, Carlos Andrés Peña Reyes, *Member, IEEE*, Eduardo Sanchez, *Member*, *IEEE*

Abstract— **Besides the numeric variables which are common in fuzzy modeling, some variables involved in the description of specific behaviors are categorical. Such variables are discrete, have no order a-priori, and most of the time handle a large amount of values (e.g., genes, proteins, countries, religions, etc.). This paper proposes a methodology for the fuzzification of categorical variables which could make part of a larger fuzzy modeling approach. The proposed solution allows to automatically create fuzzy membership functions for nominal categorical variables. We study some parameters so as to better assess their possible effect on the final outcome of the whole fuzzy modeling process.**

Index Terms – **categorical data, fuzzification, fuzzy systems.**

I. Introduction

Fuzzy logic has the ability to deal with uncertainty, it presents a human-friendly linguistic notation and it has proven approximation capabilities. Usually, fuzzy systems process c*ontinuous* variables. However, many real-world problems involve processing *categorical* variables. There are two primary types of scales. Variable having categories without a natural ordering are called *nominal*. Many categorical variables *do* have ordered categories. Such variables are called *ordinal* [5].

Fuzzification of continuous variables is often straightforward. The order relation within continuous variables puts an implicit assumption that helps associating linguistic concepts. With an order relation, we specify a certain context that circumscribes the range of values a variable can take and thus, the fuzzification is possible. But what about categorical variables? For ordinal one, it is as easier as for *continuous* one since they own an order relation. The problem arises when one wants to fuzzify nominal categorical variables. How to fuzzify variables that don't have order a-priori?

Before answering this question, one would probably like to know the advantages of fuzzifying categorical variables. It actually lies in the number of categories. Of course, fuzzify a binary variable is not relevant. On the other hand, the fuzzification of a categorical variable containing hundred of categories could be very beneficial. For example, let's take the variable "country". There are currently 192 states (countries) recognized by the United Nations [31]. If we maintain a large database containing this variable, it could be interesting to extract knowledge in the form of rules. The order of magnitude could be considerably reduced if we were able to create four of five fuzzy groups instead of using all the countries to predict the system. It could be more interpretable since these groups would represent linguistic concepts such as continents, geographic areas or levels of development. We could also try to predict if a mushroom is edible or poisonous by its odor. If we have nine possible odors, we could find three or four groups of odor instead. The complexity would be reduced by half.

This paper presents a novel method to build membership functions that could handle both continuous and categorical variables. In particular, for nominal categorical variables, the method first proposes a context by applying multidimensional fuzzy clustering to a dataset. Then it projects the clusters obtained on the axis of the variable to be fuzzified. These projections form the new membership functions of the variable. Finally, after applying several modifications intended to increase interpretability, it reorders the categories and membership functions according to their similarities.

A fuzzy inference system is composed of 4 main components as described in [1]: (1) Knowledge base, (2) Fuzzifier, (3) Rule inference engine, (4) Defuzzifier. Our work will focus on the algorithm that builds the membership functions used in the fuzzifier. Several algorithms such as Mendel-Wang method, Population based algorithms, Support Vector Machine, Fuzzy Decision Trees, Neuro-Fuzzy system, etc., might be used to create a system of rules.

II. Fuzzification of Categorical Variables

As it was said earlier, the fuzzification of continuous variables might be straightforward since we already have an order relation that directly regroups values into coarser grains. Besides, we also should have a specific context that helps to define linguistically coherent groups of values (i.e., membership functions). However, for nominal categorical variables, such order relation is not present. We need more information to regroup categories and reduce the granularity of the variable. Generally speaking, to execute the fuzzification of categorical variables we need a context that permits the emergence of one or several relations. Provided that this context exists, we can imagine the following forward process for building the fuzzy variables:

Context → Relation(s) → Groups → Fuzzification.

For many problems, this forward process can be performed by hand, based on (human) specific knowledge about the variables and the problem. Nevertheless, in our approach, we assume that the human-provided context is hard to obtain or it doesn't exist at all. We propose, thus, a reverse process to create the fuzzy membership functions,

This work was supported by the Swiss Federal Institute of Technology at Lausanne (EPFL) and Novartis Institute of Biomedical Research (NIBR).

E. Jodoin is with the Swiss Federal Institute of Technology at Lausanne. (email: etienne.jodoin@gmail.com)

C.A. Peña Reyes is with the Novartis Institute for Biomedical Research, (email: carlos_andres.pena@novartis.com, c.pena@ieee.org)

E. Sanchez is with the Swiss Federal Institute of Technology at Lausanne, (email: eduardo.sanchez@epfl.ch)

0-7803-9489-5/06/$20.00/©2006 IEEE

which is based in the assumption that available data is rich enough to provide an adequate context.

The solution consists of 4 main steps: (1) multidimensional fuzzy clustering of the database, (2) cluster projection on the variables, (3) Membership function consolidation, (4) reordering of the categories and membership functions.

A. *Multidimensional fuzzy clustering of the database*

This step finds fuzzy groups of similar objects or data points. Each receives a membership value according to the clusters generated. We assume that the set of objects to be clustered is stored in a table **T** defined by a set of variables A_1, …, A_m with domains D_1, …, D_m, respectively. Each object in **T** is represented by a tuple $t \in D_1 \times ... \times D_m$. A domain D_j is defined as categorical if it is finite and unordered, e.g., for any a, $b \in A_j$, either $a=b$ or $a \neq b$. Therefore, we don't consider ordinal categorical variables or we simply remove their order relation.

An object t in **T** can be logically represented as a conjunction of variables-category pairs $[A_1=x_1] \wedge [A_2=x_2] \wedge ... \wedge [A_m=x_m]$, where $x_j \in D_j$ for $1 \leq j \leq m$. We represent t as a vector $[x_1, x_2, …, x_m]$. We consider every object has exactly m variables.

Our multidimensional fuzzy clustering for categorical variables finds a partition **C** of **T** into k non-empty fuzzy clusters C_1, C_2, …, C_k with $\bigcup_{i=1}^{k} C_i = \mathbf{T}$ but $C_i \cap C_j$ not necessarily equal to $\varnothing$. An object t_i receives a value $w_{i,l}$ included between 0 and 1 that tells its membership value to the cluster C_l. 0 means no membership to the cluster and 1 a total membership. A two-dimensional matrix **W** with elements w_{li} representing a membership value of an object t_i for a cluster C_l, gives the results of a multidimensional fuzzy clustering applied on **T**.

B. *Cluster projection on the variables*

The next step is to project **W** on the axes of the categorical variables to represent the clusters on the variables. But **W** alone doesn't give enough information about the categories. We need **T** to know which object t has which category. The projection of cluster C_j on a category x_i of a variable A_i ($x_i \in D_i$) is the maximum membership value w_{hj} in **W** of an object t_h in **T** that have a category x_i for its variable A_i and a cluster C_j:

$$w'_{A_i=x_i,C=C_j} = \max(w_{hj} \mid t_h.A_i = x_i) \quad (1)$$

Therefore, all categorical variables A_i have a matrix $\mathbf{W'_i}$ representing the projections of all the clusters on all the categories contained in domain D_i. $\mathbf{W'_i}$ is a two-dimensional matrix composed of elements w'_{ji} representing value of category x_i for the variable A_i, to the cluster C_j. Once the projection is made on A_i we call the lines that represent the projections of the clusters, membership functions.

C. *Membership function consolidation*

Several problems could arise after the cluster projection that disturbs linguistic concepts beneath the membership functions. As we can see in figure 1.a, the membership functions obtained could be too numerous, lead to redundancy or present too weak membership values. Moreover, the cluster projection does not guarantee that cumulated membership values for a given category equal one. This third step tries to avoid these problems by applying stepwise the following functions: (1) Global thresholding, (2) Inclusion subject to a minimal margin, (3) Local thresholding and (4) Complementary function.

1) *Global thresholding*: To eliminate too weak clusters after the projection, if a cluster C_l never has a truth value higher than a certain level gth, it is automatically eliminated. In other words if $\max(w'_{li} \mid \forall i) < gth$, C_l is automatically eliminated. $0 \leq gth \leq 1$
2) *Inclusion subject to a minimal margin*: This function will merge membership functions that are similar so as to reduce redundancy and assure that no more than a certain number of membership functions are present. Two parameters are used in this function: margin range (mr) and maximum membership functions (mmf). It first merges all pairs of membership functions in which one of them includes the other. This inclusion is subject to a minimal margin mr. Mathematically, it can be expressed as: **If** $(\mu_k - \mu_j) \leq mr \mid \forall D_i$ **then** $Merge(\mu_k,\mu_j)$.
$Merge(\mu_k, \mu_j) = \mu_n = (\mu_k + \mu_j)/2$
$0 \leq mr \leq 1$. Where μ_k and μ_j are two different membership functions of a variable A_i that has the domain D_i. μ_n is the membership function created after the "*merge*" operation. The next step continues to merge membership functions until it reaches a given number ,mmf, of membership functions by gradually increasing the margin range. $1 \leq mmf \leq \infty, mmf \in integer$. For a given variable A_i with domain D_i, if this criterion is higher than $|D_i|$, it merges until it reaches a number of membership functions smaller or equal to $|D_i|$. Figure 1.b. shows the results of such a procedure on the membership functions obtained in figure 1.a. For this example we used fd=0.1 and mmf=3.
3) *Local thresholding*: After the fusion, this operation affects the membership values of each category. First, it sets a membership function value to 0 on a given category if it is smaller than a certain value lth. $0 \leq lth \leq 1$. Then, it assures that only a given number, mmc, of membership values can be assigned for a given category (i.e. the mmc membership functions that

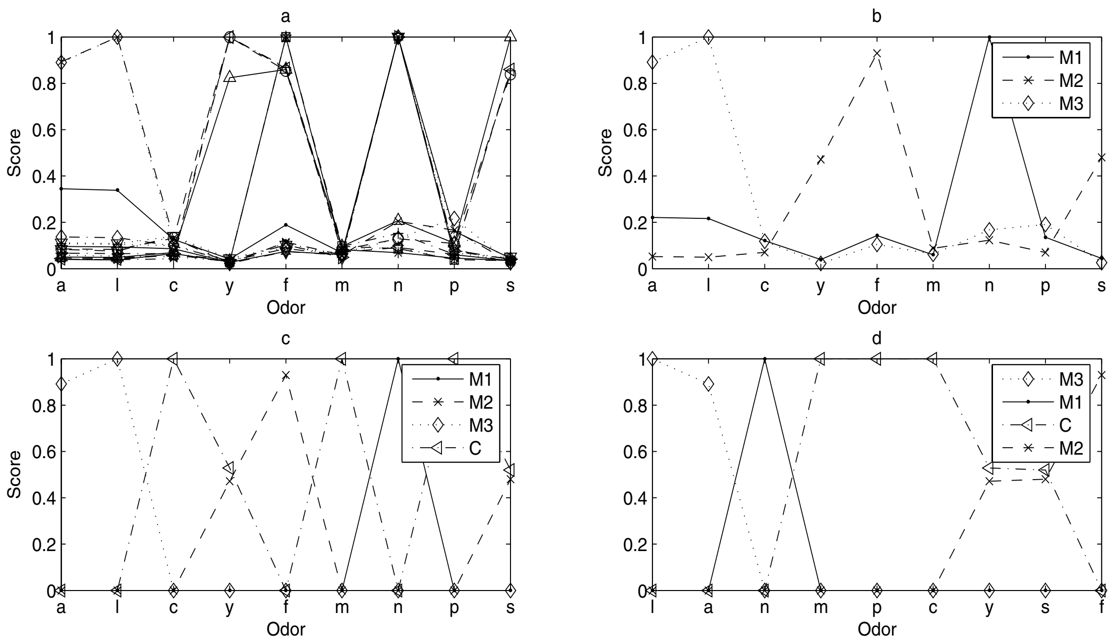


Fig. 1. Different steps of the fuzzifier construction algorithm for the variable "odor" in the mushrooms problem, A) After projection of the clusters. B) After applying, *mr* and *mmf* parameters. C) After applying *lth* and the complementary function. D) Final membership functions after reordering. The different odors are : (a) almond, (l) anise, (c) creosote, (y) fishy, (f) foul, (m) musty, (n) none, (p) pungent, (s) spicy.

4) achieve the highest membership values for that category). $0 \leq mmc \leq \infty, mmc \in integer$.
5) *Complementary function*: If cumulated membership values for a given category are smaller than 1-*lth*, an additional membership function μ_C assures its completeness. Figure 1.c. shows the results after applying the complementary function and the local thresholding. The complementary function has the label "C". *lth* was set to 0.3 and *mmc* was not applied. For a category x_i, the membership value of $\mu_C(x_i)$ is :

$$\mu_C(x_i)=1-\sum_{j=1}^{|C|} w'_{ji} \ . \ 0 \leq \mu_C(x_i) \leq 1 \qquad (2)$$

D. Reordering categories and membership functions

The previous steps modify the membership functions contained in the matrices **W'ᵢ**, but they do not regroup similar categories and membership functions to assure linguistic coherences. This step reorders the categories and membership functions by similarity. The final output of this operation are reordered matrices **O_W'ᵢ** that creates a pseudo-order between the different categories and between the membership functions. To do so, we use the bond energy algorithm [5]:

1) *Initialization*: Given a N x M matrix A (where N are the lines and M the columns), select one column and put it into the first column of output matrix oA, i=1
2) *Iteration Step i* : Place one of the remaining n-i columns in one of the i+1 possible positions in the output matrix, that makes the largest **contribution** to the global neighbor affinity measure. To measure the contribution, it is important to normalize the matrix. Contribution of column w'_k when placing between w'_i and w'_j :

$$Cont(w'_i, w'_k, w'_j) = Bond(w'_i, w'_k) + Bond(w'_k, w'_j) - Bond(w'_i, w'_j) \qquad (3)$$

$$Bond(w'_x, w'_y) = \sum_{z=1}^{N} Aff(w'_{xz}, w'_{yz}) \qquad (4)$$

$$Aff(w'_{an}, w'_{bn}) = w'_{an} * w'_{bn} \qquad (5)$$

Remark : Bond(Null, w'ₖ) = 0, Where Null is the edge of the matrix

We first apply the bond energy algorithm to reorder the columns (categories) and then for the rows (membership functions). The resulting ordered matrices **O_W'ᵢ** fully describe a mapping for the fuzzifier. The final membership functions obtained for the variable "odor" of the mushroom problem (see section III), are shown in figure 1.d. It is important to note that this method should let the user modify the membership functions generated. In fact, given that it is based on a multidimensional clustering, it should find relations not necessarily contained in the data. The algorithm could also give results that aren't coherent with the user's knowledge.

E. Rule inference engine

Now that we can perform the fuzzification of all our variables, it is possible to generate a system of fuzzy rules to be used in the rule inference engine. The rules of such system are expressed as logical implications (i.e., in the form of **if** … **then** statements). Several approaches are possible. We will use a relatively simple one inspired by Mendel-Wang [23], described as follows:

1. *Initialization*: Given an input table **T,** we first perform the fuzzification of **T** using the matrices **O_W'**$_i$. We then obtain a fuzzified database table **fT**. The list that contains all the rules of the system, Rule_sys, is empty.
2. *For each instance ft of **fT**:*
   2.1. Find the rule **r** that produces the maximum implication degree d with $d \geq mwth$.
   2.2. If Rule_sys doesn't have a rule **r1** that own the same antecedent group of **r**, add **r** to Rule_sys with its consequent.
   2.3. Else, take the rule **r1** that have the same antecedents group of **r** and include the consequent of **r** in **r1**.
3. Finally for each rule **r** of Rule_sys, determine the consequent by choosing the majority candidate.

mwth is a threshold value for generating a rule. If the rule that produce the maximum implication degree for instance *ft* in **fT** gives an implication degree smaller than *mwth*, then we don't generate a rule and we directly step to the next instance. This solution provides very accurate results. However it most of the time, handle a large number of rules that are dependant on the database of the problem. It, thus, has the problem of generalizing the system of rules generated.

III. Experimental results

In this section, we apply our algorithm to solve two problems. We study the effects of some of the different parameters explain in the third step of the membership function building algorithm (section II.C). The multidimensional categorical fuzzy clustering algorithm is ***fuzzy K-modes*** [16]. The number of clusters varies depending on the database. However, it must be high enough as it must be representative for the projection on many variables. Since *fuzzy K-modes* can present slight differences between two runs because of its random initialization phase, it is executed only once per problem in order to assure a complete independence between the clustering algorithm and the other steps of section II.

For each problem, we study the effects of the following parameters:

- Margin range (*mr*)
- Maximum membership functions (*mmf*)
- Local threshold (*lth*)

The remaining parameters are set according to the following consideratios: the "global thresholding" parameter, *gth*, stays at 0.5. The "local thresholding" parameter, *mmc*, is always ignored since we vary the "maximum membership functions" (*mmf)* parameter. The Mendel-Wang threshold (*mwth*) of section II.E stays at 0.6 which is a reasonable value. The variation of each parameter is illustrated in two figures: the first one shows the variation of the score (classification rate) and the second, its number of rules. For each problem, three pairs of figures are presented.

A. Mushrooms

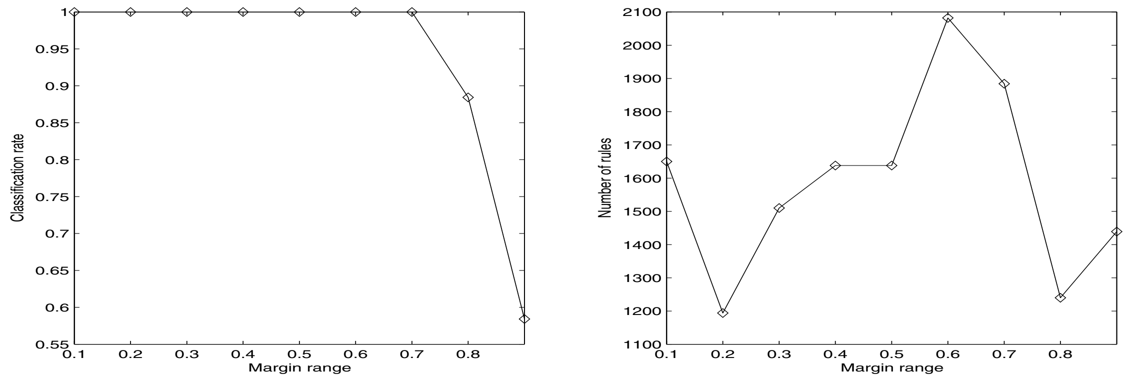
This database [30] consists of 22 input attributes of different gilled mushrooms. All of them are either edible or poisonous. The 22 inputs are all categorical attributes. 8125 instances have been classified in this database. The task of our fuzzy inference system is to identify mushrooms that are edible or poisonous. We use 16 clusters for the *fuzzy K-modes* algorithm.

We observe in figures 2, 3 and 4 that the classification rate doesn't seem to be disturbed by the parameters under study. It is possible to reduce the number of rules without affecting accuracy. The number of rules seems better for a lower the margin range parameter in figure 2.b. In figure 3.b. we note a growth of the rules if we increase the maximum amount of membership functions. Having more membership functions force the functions to cover a smaller space represented for the variable. What we can observe in figure 3 is systems with more than 2 membership functions per variable will try to cover the space the same way as with 2 membership functions. They need more rules to describe the same behavior. The parameter *lth* (section II.C.3) is stable until it reaches 0.6. After that, its amount of rules gets smaller. A high threshold value will let pass only membership functions that are strongly represented for a given category. The resulting functions will become closer to a crisp function than a fuzzy one. More crisp sets reduce the number of membership values a category can take. It is also more probable that a given category will not be represented with a membership function. The complementary function (section II.C.4) will therefore represent more categories and will reduce the possibilities an antecedent can have.

As figures 2, 3 and 4 show that we can easily obtain 100% classification rate under the described set up, this problem doesn't allow us to assess adequately the effects of the parameters studied. In consequence, we present below a more difficult classification problem involving categorical variables.

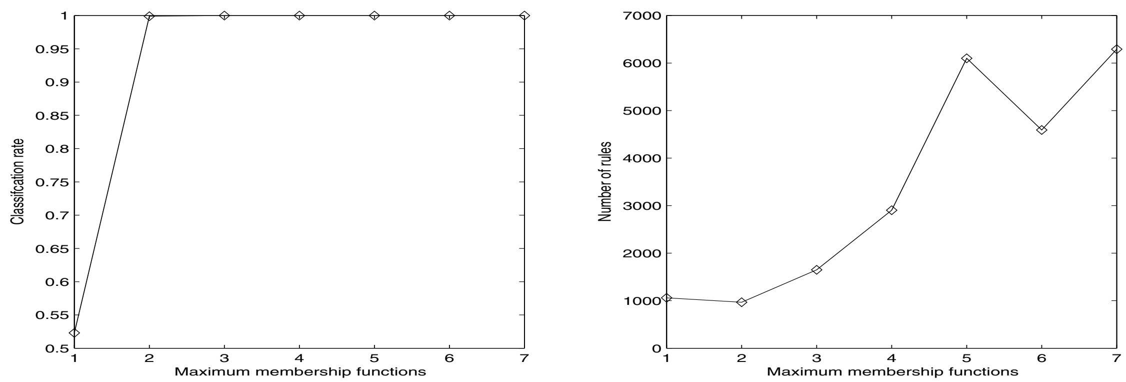
B. Catalonia mammography database

This database was collected at the Duran y Reynals hospital in Barcelona. It consists of 15 input attributes and a diagnosis result indicating whether or not a carcinoma was detected after a biopsy. The 15 input attributes include three clinical characteristics and two groups of six radiological features, according to the type of lesion found in the mammography: mass or microcalcifications [1]. Consequently, we separated the database in two cases: instances with mass lesion and microcalcifications lesion. 375 instances have been classified as microcalcifications and 106 as mass. All the input attributes are proceed as categorical. The task of our fuzzy inference system is to



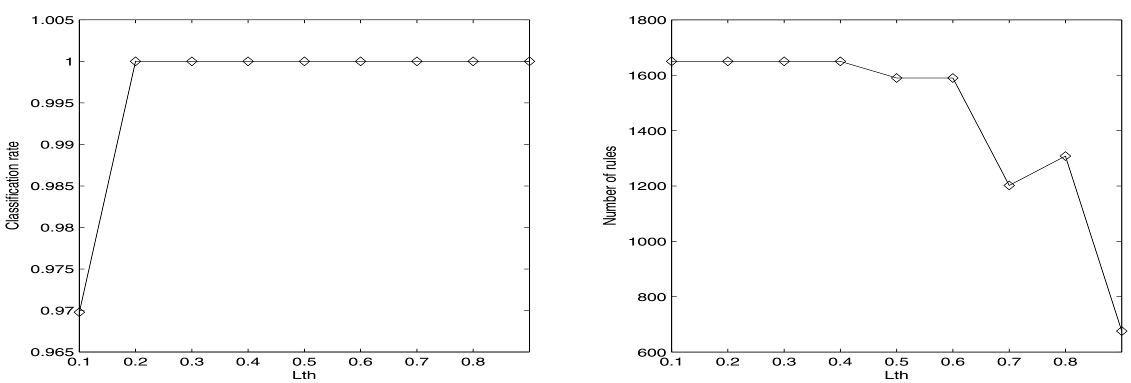
A B

Fig. 2. Variation of the classification rate (A) and number of rules (B) with respect to the margin range (*mr*) for mushrooms problem. *mmf*=3, *lth*=0.3



A B

Fig. 3. Variation of the classification rate (A) and number of rules (B) with respect to the maximum membership function (*mmf*) for mushrooms problem. *mr*=0.1, *lth*=0.3



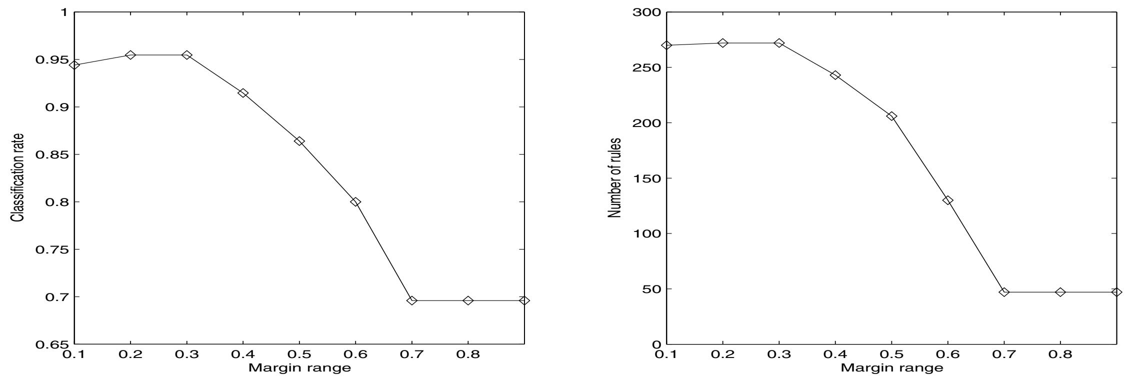
A B

Fig. 4. Variation of the classification rate (A) and number of rules (B) with respect to the local thresholding parameter, *lth*, for mushrooms problem. *mr*=0.1, *mmf*=3

detect instances that have a carcinoma or not. Figures 5, 6 and 7 shows the results obtained for microcalcifications. Note that results for mass lesions are very similar and thus not presented here.

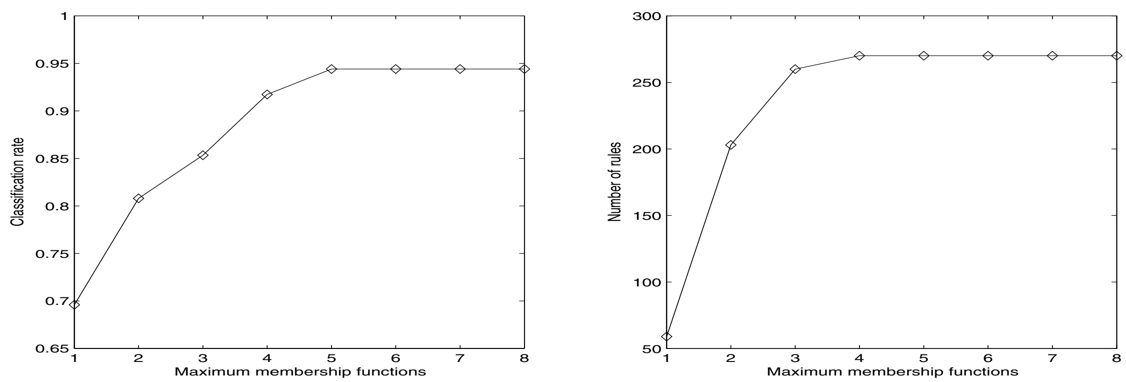
Microcalcifications : We have 9 input values, three clinical and the six radiological features. We use 20 clusters for the *fuzzy K-modes* algorithm.

The most accurate result we obtained was 97.07% but the system had at least 269 rules. As we can see in figure 5.a., a small margin range gives better results since it will merge functions that are more overlapped. This could mean functions that are linguistically similar. A high margin range could merge too much different functions. It could also lead to membership functions that cover too much categories and will not bring new knowledge. The number of rules decrease when the margin range increase in figure 5.b., simply because there will be more fusion and therefore, less membership functions. As with mushrooms, more membership functions gives more rules in figure 6.b. Accuracy is better with more membership functions as it provides a finer granularity in the rule universe that will gives more flexibility. Yet, *mmf* higher than 5 will probably



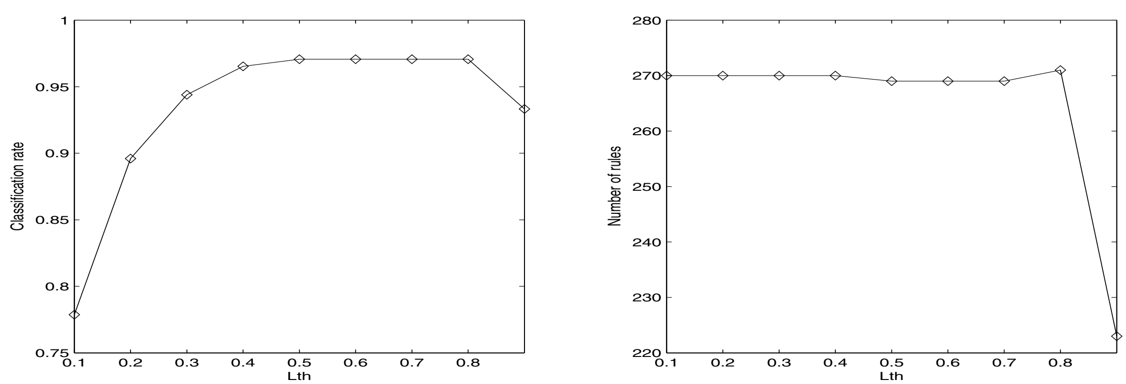
A B

Fig. 5. Variation of the classification rate (A) and number of rules (B) with respect to the margin range (*mr*) for microcalcifications problem. *mmf*=5, *lth*=0.3.



A B

Fig. 6. Variation of the classification rate (A) and number of rules (B) with respect to the maximum membership function (*mmf*) for microcalcifications problem. *mr*=0.1, *lth*=0.3.



A B

Fig. 7. Variation of the classification rate (A) and number of rules (B) with respect to the local thresholding parameter, *lth*, for microcalcifications problem. *mr*=0.1, *mmf*=5.

create redundant membership functions as it doesn't show better results. *A* high level of *lth* gives better outcomes (figure 7.a). However, between 0.5 and 0.8, the performances are identical. The membership functions values were probably situated below 0.5 and above 0.8. At 0.9, accuracy is reduced since it probably eliminates membership functions that were strongly represented. The categories represented by these functions are included in the complementary function and provide a smaller range of possibilities for the rules.

C. Discussion

The fuzzy inference system proposed shows very accurate results but are largely due to the Mendel-Wang algorithm. Nonetheless, the other parameters refine in accuracy and interpretability.

A small margin range (*mr*), lower than 0.4, generally gives better results. Intuitively, it is normal to merge two functions with a small distance because they are probably linguistically similar. Merging functions that are too much different will harm in *distinguishability*, a semantic criterion described in [1]. Higher values of "maximum membership functions" (*mmf*) provides more precision. However, it could lead to redundant functions that will describe the same behavior as a smaller system. It also increases the number of rules. Again in [1], another criterion, the *number of linguistic labels,* should be compatible with the number of conceptual entities a human being can handle. It should not exceed 7 ± 2 distinct terms. In figures 4, and 7, the local thresholding parameter, *lth*, has to be higher than the global thresholding parameter, *gth*, to obtain better results. With *gth*=0.5, a value of *lth* between 0.6 and 0.8 seems to provide the best performances. But we also slightly observe that a too high *lth* will affect accuracy. A strong local thresholding will create almost crisp membership functions and thus crisp rules which is the opposite of fuzzy logic. It could also deform too much the original shapes created by the cluster projections. We still believe that an *lth* smaller than *gth* should be appropriate because a local thresholding should be a refinement of the global thresholding. We recommend values of *lth* around 0.3 for *gth*=0.5.

IV. CONCLUSION

The major contribution of this work was to furnish a method to builds membership functions for categorical variables. The membership functions created seem to satisfy well the semantic criteria described in [1]. A good example is exposed in figure 1. Only *complementarity* is not satisfied in a lot of cases. Another advantage is that it is a constructive algorithm with different independent parts. It is possible to change one of them without affecting the others. For example, we could change the categorical fuzzy clustering algorithm without having repercussions on the other steps of the method. Finally, it can be viewed as a universal fuzzifier that handles all kind of variables. Since the ones studied are discrete, a small range of values could generate crisp membership functions. A higher amount of categories would probably exhibit smoother shapes. However, more studies are needed to reveal the power of our method.

The Mendel-Wang algorithm is considered as very precise. Nevertheless, its major flaw is its dependency on the dataset. It creates a huge system of rules that represents the training data instead of smaller and more general rules. Thus, it has been proven that Mendel-Wang approach is not good for generalization. Other methods such as Genetics algorithms, Swarm intelligence [9][27], Support Vector Machines [26], Fuzzy Decision Trees [21] and even Neuro-Fuzzy networks should be tried as rule system construction algorithms. Another interesting strategy would be to improve the interpretability of the systems generated by the Mendel-Wang approach.

Several researches should be made to improve our fuzzifier construction algorithm. First, it would be interesting to try more actual categorical clustering algorithms such as [10][11][13] or [15] and transform them into fuzzy clustering algorithms. Clustering incomplete data would help to increase the reliability of the system. Interesting solutions are proposed in [20] for fuzzy *c*-Means and could probably be used for fuzzy *k*-Modes. More statistics should be revealed for the different parameters of section II.C. with more data. Other operations such as the level of significance exposed in [7] or using the center of area (COA) of a membership function to give an importance degree according to a certain category could be integrated to the method to help solving the problems explained in section II.C.

Problems could arise if the database could not be clustered. The major flaw of this algorithm is its complete dependency of the membership functions on the clusters generated. Moreover, even if a database can be clustered and gives accurate results, the linguistic concepts beneath the membership functions could be totally foolish.

REFERENCES

[1] C. A. Peña Reyes, *Coevolutionary Fuzzy Modeling*, Lecture Notes in Computer Science, Vol. 3204, Springer, Berlin, 2004.

[2] J. S. Roger Jang, N. Gully, *MATLAB Fuzzy Logic Toolbox : User's Guide*, Book, Version 2, Natick, Mass : MathWorks, 2000.

[3] D. Hand, H. Mannila, P. Smyth, *Principles of Data Mining,* Book, MIT Press, Cambridge, Mass., 2001.

[4] I. H. Witten, E. Frank, *Data Mining: Practical machine learning tools and techniques with java implementations*, Book, Morgan Kaufmann Publisher, San Francisco, California, 1999.

[5] A. Agresti, *Categorical Data Analysis*, Book, Second Edition, Wiley-Interscience, New-York, 2002.

[6] W. McCormick, P. Schweitzer, T. White, *Problem decomposition and data reorganisation by a clustering technique*, Operations Research, 20(4): 741-751, July, 1972.

[7] M. R. Emami, I.B. Türksen, A. A. Goldenberg, *Development of a systematic methodology of fuzzy logic modeling*, IEEE Transactions on Fuzzy Systems, 6(3): 346-361, August, 1998.

[8] S. Guillaume, *Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review*, IEEE Transactions on Fuzzy Systems, 9(3): 426-443, June, 2001.

[9] M. Dorigo, G. D. Caro, L. M. Gambardella, *Ant Algorithms for Discrete Optimization*, Artificial Life, 5(3): 137-172, 1999.

[10] H. Zengyou, X. Xiaofei, D. Shengchun, *Squeezer: An Efficient Algorithm for Clustering Categorical Data*, Journal of Computer Science and Technology, 17(5): 611-624, Sept., 2002.

[11] P. Andritsos, P. Tsaparas, R. J. Miller, K. C. Sevcik, *LIMBO: Scalable Clustering of Categorical Data*, Hellenic Database Symposium, 2003.

[12] V. Ganti, J. Gehrke, R. Ramakrishnan, *CACTUS: Clustering Categorical Data Using Summaries*, Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, 73-83, 1999.

[13] M. J. Zaki, M. Peters, I. Assent, T. Seidl, *CLICKS: an effective algorithm for mining subspace clusters in categorical datasets***,** Conference on Knowledge Discovery in Data, 736-742, 2005.

[14] E. H. Han, G. Karypis, V. Kumar, B. Mobasher, *Clustering based on association rule hypergraphs*, Departement of computer science, University of Minnesota, 1997.

[15] D. Barbarà, Y. Li, J. Couto, *COOLCAT : an entropy-based algorithm for categorical clustering*, Proceedings of the eleventh international conference on Information and knowledge management, McLean, Virginia, USA, 582-589, 2002.

[16] Z. Huang, M. K. Ng, *A fuzzy k-Modes Algorithm for Clustering Categorical Data*, IEEE Transactions on Fuzzy Systems, 7(4): 446-452, Aug., 1999.

[17] O. M. San, V. N. Huynh, Y. Nakamori, *An Alternative Extentsion of the k-Means Algorithm for Clustering Categorical Data*, Int. J. Appl. Math. Comput. Sci., 14(2): 241-247, 2004.

[18] S. Guha, R. Rastogi, K. Shim, *ROCK : A Robust Clustering Algorithm for Categorical Attributes*, Information Systems, 25(5): 345-366, 2000.

[19] M. E. S. Mendes Rodrigues, L. Sacks, *A Scalable Hierarchical Fuzzy Clustering Algorithm for Text Mining*, Proc. of the 4th International Conference on Recent Advances in Soft Computing, RASC'2004: 269-274, Nottingham, UK, Dec. 2004.
[20] R. J. Hathaway, J. C. Bezdek, *Fuzzy c-Means Clustering of Incomplete Data*, IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, 31(5): 735-744, Oct., 2001.
[21] C. Z. Janikow, *Fuzzy Decision Trees: Issues and Methods*, IEEE Transactions on Systems, Man and Cybernectics-Part B: Cybernetics, 28(1): 1-14, Feb., 1998.
[22] L. X. Wang, J. M. Mendel, *Generating Fuzzy Rules by Learning from Examples*, IEEE Transactions on Systems, Man and Cybernectics, 22(6): 1414-1427, Nov./Dec., 1992.
[23] S. M. Chen, S. H. Lee, C. H. Lee, *A new method for generating fuzzy rules from numerical data for handling classification problems*, Applied Artificial Intelligence, 15: 645-664, 2001.
[24] J. Casillas, O. Cordon, F. Herrera, *Improving the Wang and Mendel's Fuzzy Rule Learning Method by Inducing Cooperation Among Rules*, Proceedings of the 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference, Madrid, Spain, 1682-1688, 2000.
[25] *STA 6938 Lecture 5 Prepare Categorical Variables*, (http://dms.stat.ucf.edu/sta6714/lec5/STA6714Lecture5.pdf)
[26] Y. Chen, J. Z. Wang, *Support Vector Learning for Fuzzy Rule-Based Classification Systems*, IEEE Transactions on Fuzzy Systems, 11(6): 716-728, Dec., 2003.
[27] C. R. Mouser, S. A. Dunn, *Comparing genetic algorithms and particle swarm optimization for an inverse problem exercise*, Australian and New-Zealand Industrial and Applied Mathematics Journal, 46(E): 89-101, 2005.
[28] E. H. Mamdani, *Application of fuzzy algorithms for control of a simple dynamic plan*, Proceedings of the IEEE, 121(12): 1585-1588, 1974.
[29] E.H. Mamdani, S. Assilian. *An experiment in linguistic synthesis with a fuzzy logic controller*. International Journal of Man-Machine Studies, 7(1):1-13, 1975.
[30] C. J. Merz, P. Merphy, *UCI Repository of Machine Learning Databases,* 1996. (http://www.ics.uci.edu/~mlearn/MLRRepository.html).
[31] Wales J., *Wikipedia, the free encyclopedia*,2001 (http://en.wikipedia.org/wiki/Country#Characteristics_of_a_country)