WEIGHT ANALYSIS AND OPTIMIZATION IN FUZZY MODELING

M.G. TSIPOURAS, T.P. EXARCHOS, D.I. FOTIADIS
Unit of Medical Technology and Intelligent Information Systems,
Department of Computer Science,
University of Ioannina, GR 45110 Ioannina, GREECE

In this paper we propose the use of a set of weights in fuzzy modelling, the class weights, which are assigned to each class of a classification problem. We automatically generate a fuzzy model, using a three-stage methodology: (i) generation of a crisp model from a decision tree, induced from the data, (ii) transformation of the crisp model into a fuzzy one, and (iii) optimization of the fuzzy model's parameters. Based on this methodology, the generated fuzzy model includes the Θ^f parameters, which are all the parameters included in the sigmoid functions. In addition, local, global and class weights are included, thus the fuzzy model is optimized with respect to all these parameters (Θ^f, local, global and class weights). The class weight introduction, which is a novel approach, grants to the fuzzy model the ability to identify the individual importance of each class and thus more accurately reflect the underlying properties of the classes under examination, in the domain of application. The above described methodology is applied to five known medical classification problems, obtained from the UCI machine learning repository, and the obtained classification accuracy is high.

1. Introduction

Fuzzy logic is the extension of the classical crisp (binary) logic into a multivariate form. Fuzzy logic is closer to the human logic, thus being able to deal with real world noisy and imprecise data [1]. Fuzzy models experience several advantages, compared to crisp ones, mainly being more flexible on the decision boundaries, and thus characterized by their higher ability to adjust to a specific domain of application and more accurately reflect its particularities. A fuzzy model can be created by defining an initial crisp model (set of rules) and then fuzzyfing it. This approach is a complex task since several issues must be defined for the fuzzy model to be generated. First, the origin of the rules must be addressed, which determines the philosophy of the method; if expert's knowledge is used then the generated fuzzy model will be knowledge-based while, if data mining techniques are employed then a data-driven fuzzy model will be generated. In the fuzzyfication step, there are several fundamental features related to the definition of the fuzzy model, such as the fuzzy membership function, the fuzzy operators, the defuzzyfication approach and the use of weights. Following this approach, the generated fuzzy model resembles the decision making processes of the initial crisp model and thus its parameters must be tuned before being able to identify the particularities of a specific problem. This “tuning” can be performed using parameter optimization.

Several approaches have been proposed in the literature for the development of knowledge –based fuzzy models. In most of them the model is trained using a known optimization technique, i.e. fuzzy rules with simulated annealing [2], fuzzy rules with modified controlled random search [3]. Also, several research attempts exist in the literature, which integrate data mining techniques with fuzzy modeling towards the generation of a fuzzy model. More specifically, the presented approaches can be classified into three main categories: (i) induction of a crisp decision tree from the data and then its fuzzyfication, resulting into a fuzzy decision tree [4-7], (ii) induction of a fuzzy decision tree, integrating fuzzy techniques during the tree construction [8,9], (iii) induction of a crisp decision tree, extraction of a set of rules from it and fuzzyfication of these rules [10].

Concerning use of weights in fuzzy modelling, two approaches have been presented: (i) local weights, which are used to indicate the relative degree of importance of a proposition contributing to its consequent, thus one local weight is assigned to each fuzzy conjunct. Local weights play an important role in many real world problems. For example, in medical diagnostic systems it is common to observe that a particular symptom combined with other symptoms may lead to a possible disease and thus it is important to assign a local weight to each symptom in order to show the relative degree (weight) of each symptom leading to the consequent (a disease) [11,12]; (ii) global weights, which are used to represent the relative degree of importance of each rule's contribution, thus one global weight is assigned to each fuzzy rule [13].

In this work, we use a specific realization of a previously reported methodology for automated fuzzy model generation [14] which includes three stages. Initially, a crisp model is created, then it is transformed into the respective fuzzy model, and finally, all parameters of this fuzzy model are optimized. During the transformation of the crisp model into the respective fuzzy, several new parameters are introduced, corresponding to the fuzzyfication of the decision boundaries. In addition, three sets of weights are employed in the fuzzy model, local and global weights and class weights, which are introduced for the first time, and indicate the relative importance of each class. All three sets of weights are also optimized. The integration of all three types of weights (local, global and class) in a single fuzzy model grants additional flexibility and thus the model is more adaptable to fuzzy decision boundaries and can more accurately identify the underling properties of a specific application domain.

2. Materials and Methods

In our realization, the crisp model is generated from a decision tree, induced from the data. The transformation of the crisp model into a fuzzy one is made using the sigmoid function, as fuzzy membership function, the *min* and *max* operators for T_{norm} and S_{norm} functions, respectively, and the maximum defuzzifier. The optimization of the fuzzy model's parameters is conducted using a simplex-based local optimization technique.

2.1. *Crisp model creation*

In order to construct the crisp model, an initial set of rules must be extracted from an annotated dataset, thus a rule-mining technique is employed based on decision trees with the C4.5 inductive algorithm [15]. The produced tree can be easily transformed into a set of rules, as follows:

1. One crisp rule $r_{i,j}^{c}\left(x,\theta_{i,j}^{c}\right)$, having a crisp condition $Cond_{i,j}^{c}$, is created for every leaf of the tree, by parsing the tree from the root node to that leaf. The feature tests encountered along the path form the conjuncts of the condition: $Cond_{i,j}^{c}\left(x,\theta_{i,j}^{c}\right) = g_c\left(a_{i,j,1},\theta_{i,j,1}^{c}\right)\wedge g_c\left(a_{i,j,2},\theta_{i,j,2}^{c}\right)\wedge\ldots\wedge g_c\left(a_{i,j,K_{i,j}},\theta_{i,j,K_{i,j}}^{c}\right)$, where $a_{i,j,1},a_{i,j,2},\ldots,a_{i,j,K_{i,j}}$, are the features that are encountered in the path, $\theta_{i,j,1}^{c},\theta_{i,j,2}^{c},\ldots,\theta_{i,j,K_{i,j}}^{c}$ are the respective parameters and $g_c(\cdot)$ is the crisp membership function. The class label at the leaf node is assigned to the rule consequent: $r_{i,j}^{c}\left(x,\theta_{i,j}^{c}\right): Cond_{i,j}^{c}\left(x,\theta_{i,j}^{c}\right)\rightarrow y_i$, with $i=1,\ldots,I$ (I is the number of classes), $j=1,\ldots,J_i$ (J_i is the number of conditions that predict the i^{th} class) and $k=1,\ldots,K_{i,j}$ ($K_{i,j}$ is the number of conjuncts of the j^{th} condition that predicts the i^{th} class).

2. A crisp class rule R_i^c is created for each class y_i, using all crisp rules that have as consequent this class: $R_i^c\left(x,\theta_i^c\right):\left(r_{i,1}^{c}\vee r_{i,2}^{c}\vee\ldots r_{i,J_i}^{c}\right)\rightarrow y_i$.

Based on the above, the crisp model M^c is defined as follows: $M^c\left(x,\Theta^c\right)=F^c\left(R_1^c,R_2^c,\ldots,R_I^c\right)$, $\Theta^c=\left\{\theta_1^c,\theta_2^c,\ldots,\theta_I^c\right\}$, where F^c is a function that combines the outcomes of all R_i^c crisp class rules and results to one of the classes (decision function).

2.2. *Fuzzyfication of the crisp model*

The crisp model (M^c) is transformed into a fuzzy model (M^f) as follows: (i) the sigmoid function is used as fuzzy membership function instead of the crisp membership function. The sigmoid function is defined as: $g^f\left(a,\theta^f\right)=\left(1+e^{\theta^{1,f}\left(a-\theta^{2,f}\right)}\right)^{-1}$, where θ^f is a vector containing all parameters used in the sigmoid function, $\theta^f = \{\theta^{1,f}, \theta^{2,f}\}$, (ii) the binary AND and OR operators are replaced with T_{norm} and S_{norm} functions, defined as min and max operators [1], respectively, and (iii) the F^c function is replaced with a defuzzification function F^f; the defuzzifier was selected as the maximum operator [1].

According to the above, each fuzzy rule $r_{i,j}^f\left(x, \theta_{i,j}^f\right)$ is defined as: $r_{i,j}^f\left(x, \theta_{i,j}^f\right) = Cond_{i,j}^f\left(x, \theta_{i,j}^f\right)$, where $Cond_{i,j}^f$ is a fuzzy condition, defined as: $Cond_{i,j}^f = min\left(w_{i,j,1}^l \cdot g^f\left(a_{i,j,1}, \theta_{i,j,1}^f\right), ..., w_{i,j,K_{i,j}}^l \cdot g^f\left(a_{i,j,K_{i,j}}, \theta_{i,j,K_{i,j}}^f\right)\right)$, and w^l is a local weight. Each fuzzy class rule $R_i^f\left(x, \theta_i^f\right) \to y_i$ is defined as: $R_i^f\left(x, \theta_i^f\right) = max\left(w_{i,1}^g \cdot r_{i,1}^f, w_{i,2}^g \cdot r_{i,2}^f, ... w_{i,J_i}^g \cdot r_{i,J_i}^f\right)$, where w^g is a global weight. Finally, M^f is defined as: $M^f\left(x, \Theta^f, W\right) = max\left(w_1^c \cdot R_1^f, w_2^c \cdot R_2^f, ..., w_I^c \cdot R_I^f\right)$, where w^c is a class weight, Θ^f is defined as: $\Theta^f = \{\theta_1^f, \theta_2^f, ..., \theta_I^f\}$ and W is a set containing all weights introduced in the fuzzy model: $W = \{w_i^c, w_{i,j}^g, w_{i,j,k}^l\}$. Thus: $M^f\left(x, \Theta^f, W\right) = \max_{i=1,j=1}^{I,J_i}\left(\min_{k=1}^{K_{i,j}}\left(w_i^c \cdot w_{i,j}^g \cdot w_{i,j,k}^l \cdot \left(1 + e^{\theta_{i,j,k}^{1,f}\left(a_{i,j,k} - \theta_{i,j,k}^{2,f}\right)}\right)^{-1}\right)\right)$. This equation denotes the implicit input-output formula of the fuzzy model.

2.3. *Parameter optimization*

The fuzzy model $M^f\left(x, \Theta^f, W\right)$ is optimized with respect to its parameters Θ^f and W, using a training dataset (D_{train}). For this purpose, a cost function is used, defined as: $F\left(\Theta^f, W, D_{train}\right) = trace(X) / \left|D_{train}\right|$, where X is the confusion matrix, and $\left|D_{train}\right|$ is the size (number of patterns) included in the D_{train}. A local optimization technique and the Nelder-Mead simplex search method [16], has been employed. Nelder-Mead simplex search method is an unconstrained nonlinear local optimization technique, which attempts to find a minimum of a scalar function of several variables, starting from an initial estimate (initial point). The method does not use numerical or analytical computation of the gradient. The initial point was defined setting $\theta_{i,j,k}^{2,f} = \theta_{i,j,k}^c$ ($\theta_{i,j,k}^c$ are defined from the decision tree) and $\theta_{i,j,k}^{1,f} : N(5,1)$ or $\theta_{i,j,k}^{1,f} : -N(5,1)$, if the crisp membership function decreases or increases, respectively. All the fuzzy model's weights are initialized as: $w_i^c \sim U(0.95,1.05)$, $w_{i,j}^g \sim U(0.95,1.05)$, $w_{i,j,k}^l \sim U(0.95,1.05)$.

Optimization was performed using a hybrid four-stage optimization strategy:

Stage 1. Set $w_{i,j,k}^l = \mathbf{1}$, $w_{i,j}^g = \mathbf{1}$, $w_i^c = \mathbf{1}$ and initialize Θ^f.
Optimize M^f with respect to Θ^f (resulting to Θ^{f*}).

Stage 2. Set $\Theta^f = \Theta^{f*}$, $w_{i,j}^g = \mathbf{1}$, $w_i^c = \mathbf{1}$ and initialize $w_{i,j,k}^l$.
Optimize M^f with respect to $w_{i,j,k}^l$ (resulting to $w_{i,j,k}^{l*}$).

Stage 3. Set $\Theta^f = \Theta^{f*}$, $w_{i,j,k}^l = w_{i,j,k}^{l*}$, $w_i^c = \mathbf{1}$ and initialize $w_{i,j}^g$.
Optimize M^f with respect to $w_{i,j}^g$ (resulting to $w_{i,j}^{g*}$).

Stage 4. Set $\Theta^f = \Theta^{f*}$, $w_{i,j,k}^l = w_{i,j,k}^{l*}$, $w_{i,j}^g = w_{i,j}^{g*}$ and initialize w_i^c.
Optimize M^f with respect to w_i^c (resulting to w_i^{c*}).

The result of the optimization procedure is the optimized fuzzy model, containing optimal values for its parameters $M^f\left(x,\Theta^{f*},W^*\right)$, with $W = \left\{w_i^{c*}, w_{i,j}^{g*}, w_{i,j,k}^{l*}\right\}$.

3. Results

In order to evaluate the proposed methodology several well known datasets obtained from the UCI machine learning repository [17] were employed. Table 1 presents all datasets that were employed, along with the number of samples included, the number of attributes used in each dataset and the number of classes. These datasets were selected because they belong to the biomedical domain and in addition none or very few values are missing in each dataset.

Table 1: Datasets used for evaluation of the proposed methodology [17].

Dataset	Samples	Attributes	Classes
Wisconsin breast cancer (breast_c)	699	9	2
Cleveland Heart Disease (heart_c)	303	13	2
Heart disease (heart_statlog)	270	13	2
BUPA liver disorders (liver_d)	345	6	2
Pima Indian diabetes (pima_d)	768	8	2

Based on these datasets, evaluation was performed in terms of classification accuracy. The ten fold stratified cross validation method, was used for the evaluation. The procedure was applied to each fold, generating ten different decision trees and, subsequently, ten different fuzzy models. Table 2 presents the obtained results for all datasets, of the initial decision trees along with the results obtained in each stage of the classification procedure, in terms of average classification accuracy of the ten folds and standard deviation. The classification accuracy results are also presented graphically in Fig. 1.

Table 2: Average classification accuracy and standard deviation results obtained from the initial decision trees and the four optimization stages, for the five datasets.

Dataset	**Decision tree**	**Stage 1**	**Stage 2**	**Stage 3**	**Stage 4**
breast_c	94.99±2.05	94.99±2.05	96.14±1.91	96.42±1.81	96.14±2.24
heart_c	75.28±10.2	75.28±10.28	76.27±10.46	76.27±10.46	75.61±10.73
heart_statlog	78.52±6.94	78.15±6.86	79.26±7.45	81.48±6.3	81.48±6.3
liver_d	66.65±5.08	66.94±5.04	69.26±5.53	69.55±5.88	69.83±5.74
pima_d	76.68±5.17	76.42±5.09	77.59±5.28	78.25±5.45	78.51±5.11

4. Discussion and conclusions

In this paper we perform an analysis concerning weights in fuzzy modeling. Fuzzy models are automatically generated and three sets of weights are introduced. A four-stage optimization strategy is employed to define optimal weight values. Results are presented for five datasets related to biomedical problems, obtained from the UCI machine learning repository.

The introduction of the class weights as long as the incorporation of all sets of weights in a single fuzzy model and their optimization, are novel features of

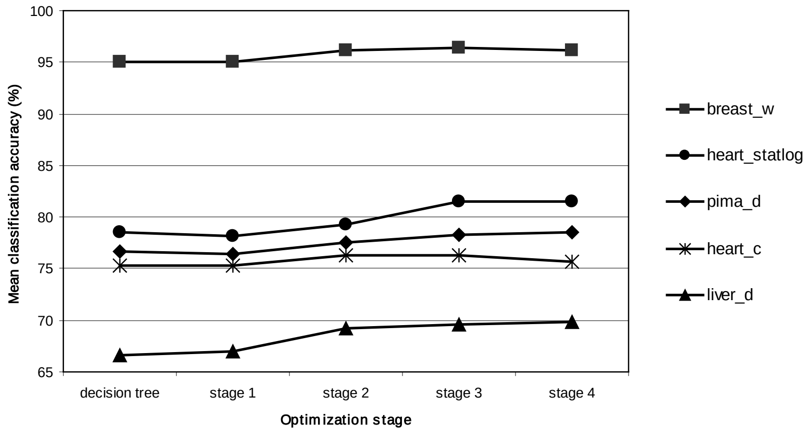

Figure 1. Graphical representation of the obtained classification accuracy results (%).

this work. The obtained results indicate that the fuzzyfication of the crisp model generated from the initial decision tree, and the optimization of the parameters introduced in the fuzzy model (Θ^f, w^c, w^g and w^l) increase the classification accuracy of the initial decision trees. More specifically, for the liver_d, pima_d and heart_statlog datasets, the average classification accuracy gradually increases with the optimization of each weight set, while for the breast_c and heart_c the average classification accuracy increases until stage 3, while the optimization stage 4 causes a decrease. However, in all cases the average classification accuracy of the initial decision tree is improved: an increase of 1.15%, 2.96%, 1.83%, 0.33% and 3.18% is reported for breast_w, heart_statlog, pima_d, heart_c and liver_d datasets, respectively, while the average increase for all employed datasets is 1.89%.

Table 3 presents a comparison of the results obtained by similar approaches presented in the literature. The three datasets included (breast_c, heart_c and pima_d) are those that are reported in at least two of these research attempts while the works of Suarez et al. [4], Abonyi et al. [10], Crockett et al. [7] and Olaru et al. [5] have presented results for at least two datasets, which are also used in the evaluation of the proposed methodology. Overall accuracy results (mean values) are also presented in Table 3: in the first line of the overall section, the mean accuracy corresponds to all three datasets, in the second to the datasets employed by Abonyi et al. and the third line corresponds to the datasets employed by Crockett et al. In general, the results obtained in this work are comparable or better than those reported in the literature.

Table 3: Comparison of classification accuracy results presented in the literature.

datasets	Suarez et al. [4]	Suarez et al. [4]	Crockett et al. [7]	Abonyi et al. [10]	Olaru et al. [5]	This work
heart_c	77.6	74	**77.7**		74.41	75.61
pima_d	74.8	74.3	78.5	73.05	74.43	**78.51**
breast_c	95.8	95.9	95.58	**96.82**		96.14
Overall (mean)	82,73	81,40	83,93			83,42
				84,94		87,33
					74,42	77,06

Future work will focus on the employment of different local or global optimization techniques. The importance of the order of the optimization stages has not been evaluated; optimizing the weight set in a different order may result to different classification results. Also, the reduction of the fuzzy model's complexity can be handled during the decision tree pruning or by pruning the fuzzy rules.

Acknowledgement

This research is part funded by the program "Heraklitos" of the Operational Program for Education and Initial Vocational Training of the Hellenic Ministry of Education under the 3rd Community Support Framework and the European Social Fund.

References

1. L.X. Wang, A course in fuzzy systems and control, NJ, Prentice Hall PTR, 1997.
2. W. Pedrycz, J.V. de Oliveira, An algorithmic framework for development and optimization of fuzzy models, Fuzzy sets and systems 80 (1996) 37-55.
3. M.G. Tsipouras, C. Voglis, D.I. Fotiadis, A Framework for Fuzzy Expert System Creation – Application to Cardiovascular Diseases, IEEE Trans. Biomed. Eng. (accepted for publication).
4. A. Suarez, F. Lutsko, Globally optimal fuzzy decision trees for classification and regression, IEEE Trans. Pattern Anal. Machine Intel. 21 (12) (1999) 1297–1311.
5. C. Olaru, L. Wehenkel, A complete fuzzy decision tree technique, Fuzzy Sets Syst. 138 (2003) 221-254.
6. W. Pedrycz, Z.A. Sosnowski, Genetically Optimized Fuzzy Decision Trees, IEEE Trans. Systems, Man Cybernetics – part B, 35 (2005) 633-641.
7. K. Crockett, Z. Bandar, J. O'Shea, D. Mclean, On Constructing a Fuzzy Inference Framework using Crisp Decision Trees, Fuzzy Sets Syst. 157 (2006) 2809– 2832.
8. C.Z. Janikow, Fuzzy decision trees: issues and methods, IEEE Trans. Systems Man Cybernet. 28(1) (1998) 1–14.
9. X. Wang, B. Chen, G. Qian, F. Ye, On the optimization of fuzzy decision trees, Fuzzy Sets Syst. 112 (2000) 117-125.
10. J. Abonyi, H. Roubos, F. Szeifert, Data-Driven Generation of Compact, Accurate, and Linguistically Sound Fuzzy Classifiers Based on a Decision Tree Initialization, International Journal of Approximate Reasoning (2003) 1-21.
11. D.L. Hudson, M.E. Cohen, The role of approximate reasoning in medical expert system, in A. Kandel (ed.), Fuzzy Expert Systems, CRC Press (1991) 165-179.
12. H.C. Tseng, D.W. Teo, Medical expert system with elastic fuzzy logic, 3rd IEEE Internat. Conf. on Fuzzy Systems, 3 (1994) 2067-2071.
13. D.S. Yeung, E.C.C. Tsang, Weighted fuzzy production rules. Fuzzy Sets Syst. 88 (1997) 299-313.
14. M.G. Tsipouras, T.P. Exarchos, D.I. Fotiadis, A. Kotsia, K.V. Vakalis, K.K. Naka, L.K. Michalis, Automated diagnosis of coronary artery disease based on data mining and fuzzy modelling, IEEE Trans. Inf. Tech. Biomed. (accepted for publication).
15. J.R. Quinlan. C4.5: Programs for Machine Learning. CA, Morgan Kauffman, 1993.
16. J.C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright, Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions, SIAM Journal of Optimization, 9(1) (1998) 112-147.
17. A. Asuncion, D.J. Newman. UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 2007.