Automated Fuzzy Model Generation Through Weight and Fuzzification Parameters' Optimization

Markos G. Tsipouras, *Student Member, IEEE*, Themis P. Exarchos, *Student Member, IEEE*, and Dimitrios I. Fotiadis, *Senior Member, IEEE*

***Abstract*—In this paper we explore the use of weights in the generation of fuzzy models. 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 a set of 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 both sigmoid function parameters and 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 classification problems, obtained from the UCI machine learning repository, and the obtained classification accuracy is high.**

I. Introduction

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 [1,2]. A fuzzy model can be created by defining an initial crisp model (set of rules) and then fuzzifing 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.

Manuscript received December 15, 2007. This work was supported in part by the European Commission within the NOESIS project: Platform for wide scale integration and visual representation of medical intelligence (IST-2002-507960).

M. G. Tsipouras is with the Unit of Medical Technology and Intelligent Information Systems, Dept. of Computer Science, University of Ioannina, Ioannina, Greece, GR 45110 (e-mail: markos@cs.uoi.gr).

T. P. Exarchos is with the Unit of Medical Technology and Intelligent Information Systems, Dept. of Computer Science, University of Ioannina, Ioannina, Greece, GR 45110. (e-mail: exarchos@cc.uoi.gr).

D. I. Fotiadis is the director of the Unit of Medical Technology and Intelligent Information Systems, Dept. of Computer Science, University of Ioannina, Ioannina, Greece, GR 45110 (+30-26510-98803; fax: +30-26510-97092; e-mail: fotiadis@cs.uoi.gr).

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 genetic algorithms [3], fuzzy rules with simulated annealing [4], multicriteria decision analysis with genetic algorithms [5], fuzzy rules with modified controlled random search [6], etc. Also, several research attempts exist in the literature, which integrate data mining techniques with fuzzy modeling. 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 [7-11], (ii) induction of a fuzzy decision tree, integrating fuzzy techniques during the tree construction [12-17], (iii) induction of a crisp decision tree, extraction of a set of rules from it and fuzzyfication of these rules [18].

In the first category (fuzzyfication of a crisp decision tree), Jeng *et al.* [7] proposed the integration of fuzzy theory into the regular inductive learning method for single dimension decision problems. Suarez and Lutsko [8] proposed a fuzzyfication of a CART decision tree. Olaru *et al.* [9] proposed the fuzzyfication of a crisp regression tree. Chen and Jeng [10] extended the method proposed in [7], integrating fuzzy theory into the regular inductive learning method for multi dimensional decision problems. Crockett *et al.* [11] constructed fuzzy decision trees, fuzzyfying crisp decision trees induced from the data using the C4.5 algorithm.

In the second category (integration of fuzzy techniques during the tree construction), Yuan and Shaw [12] introduced a heuristic algorithm for generating fuzzy decision trees, similar to the ID3 method, based on the measurement of the classification ability. Ichihashi *et al.* [13] proposed a method for inducing fuzzy decision trees, based on the fuzzy ID3 algorithm, inducting expert's partial knowledge. Apolloni *et al.* [14] presented a method for learning fuzzy decision trees, using recurrent neural networks to suggest the next move during the descent along the branches of the tree. Janikow [15] provided a detailed investigation for fuzzy decision trees, combining fuzzy representation with symbolic decision trees. Wang *et al.* [16]

investigated the optimization of fuzzy decision trees and proposed a branch-merging algorithm for FDT generation. Tsang *et al*. [17] proposed a methodology to improve the learning accuracy of fuzzy decision trees, using hybrid neural networks.

Concerning the third category, Abonyi *et al*. [18] proposed a method for fuzzyfication of rules extracted from decision trees, induced by the C4.5 algorithm, and optimized and simplified the fuzzy set of rules using genetic algorithms.

In most of these approaches, the ID3 tree induction algorithm is employed, while in some works the CART or the C4.5 tree induction algorithms are used. Fuzzy modeling has been treated with several different approaches concerning the fuzzyfication of the input variables, the construction of the inference engine and the defuzzyfication procedure. Different approaches for the T_{norm} and S_{norm} functions have been employed. All approaches include an optimization stage in order to tune the parameters entering the fuzzy models to fit a specific problem, described by a dataset. The majority of the published works employ genetic algorithms in this stage.

Concerning the use of weights in fuzzy modeling, 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) [19-21]; (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 [22].

In this work, we use a three stage methodology for automated fuzzy model generation. 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.

II. METHODOLOGY

The three stage methodology, used to automatically generate the fuzzy model, is presented in Fig. 1. In the current work, 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.

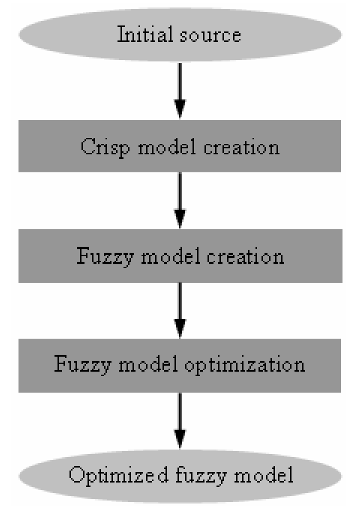

Fig. 1. The three-stage methodology used for the automated fuzzy model generation.

A. 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 [23]. 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:

$$Cond_{i,j}^{c}\left(x,\theta_{i,j}^{c}\right) = \bigwedge_{k=1}^{K_{i,j}} g_c\left(a_{i,j,k},\theta_{i,j,k}^{c}\right), \quad (1)$$

where $a_{i,j,1},a_{i,j,2},...,a_{i,j,K_{i,j}}$, are the features that are encountered in the path, $\theta_{i,j,1}^{c},\theta_{i,j,2}^{c},...,\theta_{i,j,K_{i,j}}^{c}$ 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:

$$r_{i,j}^{c}\left(x,\theta_{i,j}^{c}\right): Cond_{i,j}^{c}\left(x,\theta_{i,j}^{c}\right) \rightarrow y_i, \quad (2)$$

with $i=1,...,I$ (I is the number of classes), $j=1,...,J_i$ (J_i is the number of conditions that predict the i^{th} class) and $k=1,...,K_{i,j}$ ($K_{i,j}$ 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:

$$R_i^c\left(x,\theta_i^c\right): \overset{J_i}{\underset{j=1}{\vee}} r_{i,j}^c \to y_i . \quad (3)$$

Based on the above, the crisp model M^c is defined as follows:

$$M^c\left(x,\Theta^c\right)=F^c\left(R_1^c,R_2^c,...,R_I^c\right), \quad (4)$$

where Θ^c is a vector including all parameters entering the crisp model ($\Theta^c=\left\{\theta_1^c,\theta_2^c,...,\theta_I^c\right\}$), 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).

There are two important aspects to consider when constructing the set of rules of a rule based classifier. First, the set of rules should be composed by mutually exclusive rules. The rules in a set are mutually exclusive if a record triggers only one rule. This property ensures that every record is covered by at most one rule in the set. Second, the set of rules should be exhaustive. A set of rules has exhaustive coverage if there is a rule for each combination of feature values. This property ensures that every record is covered by at least one rule in the set. Together, these properties ensure that every record is covered by exactly one rule. Both these properties characterize the decision trees induced using the C4.5 algorithm, and thus the set of rules that is generated from the decision tree. Based on the above, the decision function F^c is defined as $F^c\left(R_1^c,R_2^c,...,R_I^c\right)=y_i$, if $R_i^c\left(x,\theta_i^c\right)$ is true.

B. *Fuzzy model generation*

The crisp model is based on axis-parallel decision boundaries. This is a limitation that can be treated with the fuzzyfication of the crisp rules, which introduces flexibility in the decision boundaries [18]. In addition, each feature test is considered of the same "power", i.e. each one of them is considered equally important with all others. The same applies for crisp rules and class rules. These assumptions do not accurately reflect the properties of real-world datasets where, for example each class, and thus its respective rule, has its own importance, depending on the domain of application and the specific dataset concerning this domain. To overcome this limitation, appropriate vectors of weights are introduced into the fuzzy model. Based on the above, the crisp model (M^c) is transformed into a fuzzy model (M^f) as follows:

1. The sigmoid function is used as fuzzy membership function instead of the crisp membership function. The sigmoid function is defined as:

$$g^f\left(a,\theta^f\right)=\left(1+e^{\theta^{1,f}\left(a-\theta^{2,f}\right)}\right)^{-1}, \quad (5)$$

where θ^f is a vector containing all parameters used in the sigmoid function, $\theta^f=\left\{\theta^{1,f},\theta^{2,f}\right\}$, (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 $r_{i,j}^f\left(x,\theta_{i,j}^f,w_{i,j}^l\right)$ is defined as:

$$r_{i,j}^f\left(x,\theta_{i,j}^f,w_{i,j}^l\right)=Cond_{i,j}^f\left(x,\theta_{i,j}^f,w_{i,j}^l\right), \quad (6)$$

where $Cond_{i,j}^f$ is a fuzzy condition, defined as:

$$Cond_{i,j}^f=\overset{K_{i,j}}{\underset{k=1}{min}}\left(w_{i,j,k}^l\cdot g^f\left(a_{i,j,k},\theta_{i,j,k}^f\right)\right), \quad (7)$$

where $\theta_{i,j}^f=\left\{\theta_{i,j,k}^f\right\}$ and $w_{i,j}^l=\left\{w_{i,j,k}^l\right\}$, $k=1,...,K_{i,j}$. $\theta_{i,j}^f$ is the set of parameters entering the $Cond_{i,j}^f$, $w_{i,j,k}^l$ is a local weight (used for a single conjunct) and $w_{i,j}^l$ is a vector containing all weights used in the $Cond_{i,j}^f$.

2. Each fuzzy class rule $R_i^f\left(x,\theta_i^f,w_i^g,w_i^l\right)\to y_i$ is defined as:

$$R_i^f\left(x,\theta_i^f,w_i^g,w_i^l\right)=\overset{J_i}{\underset{j=1}{max}}\left(w_{i,j}^g\cdot r_{i,j}^f\left(x,\theta_{i,j}^f,w_{i,j}^l\right)\right), \quad (8)$$

where $\theta_i^f=\left\{\theta_{i,j}^f\right\}$, $w_i^g=\left\{w_{i,j}^g\right\}$ and $w_i^l=\left\{w_{i,j}^l\right\}$, $j=1,...,J_i$. $w_{i,j}^g$ is a global weight (used for each rule), θ_i^f is a set of all parameters entering R_i^f, w_i^g is the set of all global weights of the R_i^f and w_i^l is the set of all local weights of R_i^f.

3. Finally, the fuzzy model, M^f, is defined as:

$$M^f\left(x,\Theta^f,W\right)=\overset{I}{\underset{i=1}{max}}\left(w_i^c\cdot R_i^f\left(x,\theta_i^f,w_i^g,w_i^l\right)\right), \quad (9)$$

where w^c is a class weight, Θ^f is defined as: $\Theta^f=\left\{\theta_i^f\right\}$, $i=1,...,I$ and W is a set containing all weights introduced in the fuzzy model: $W=\{w_i^c, w_i^g, w_i^l\}$.

Thus, based on Eq. (6)-(9), the fuzzy model is defined as:

$$\begin{aligned} M^f\left(x,\Theta^f,W\right) &= \max_{i=1}^{I}\left(w_i^c \cdot R_i^f\left(x,\theta_i^f,w_i^g,w_i^l\right)\right) \\ &= \max_{i=1}^{I}\left(w_i^c \cdot \max_{j=1}^{J_i}\left(w_{i,j}^g \cdot r_{i,j}^f\left(x,\theta_{i,j}^f,w_{i,j}^l\right)\right)\right) \\ &= \max_{i=1}^{I}\left(w_i^c \cdot \max_{j=1}^{J_i}\left(w_{i,j}^g \cdot \min_{k=1}^{K_{i,j}}\left(w_{i,j,k}^l \cdot g^f\left(a_{i,j,k},\theta_{i,j,k}^f\right)\right)\right)\right) \\ &= \max_{i=1,j=1}^{I,J_i}\left(\min_{k=1}^{K_{i,j}}\left(w_i^c \cdot w_{i,j}^g \cdot w_{i,j,k}^l \cdot g^f\left(a_{i,j,k},\theta_{i,j,k}^f\right)\right)\right) \end{aligned} \quad (10)$$

This equation denotes the implicit input-output formula of the fuzzy model.

The transformation of the crisp model to the respective fuzzy model greatly depends on the selection of the fuzzifier (fuzzy membership function), the T_{norm} and S_{norm} and the defuzzifier; if specific combinations among these are chosen then known solutions from the literature can be used to express the explicit mathematical input-output of the fuzzy model [1,24]. Also, the transformation is straightforward and thus, it can be performed easily in a fully automated manner. Furthermore, the parameters of a monotonic fuzzy membership function can be set to resemble the crisp membership function, while the class weight can remain the same as in the crisp class rules.

C. 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:

$$F\left(\Theta^f,W,D_{train}\right)=trace\left(X\right)/\left|D_{train}\right|, \quad (11)$$

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 [25] have 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 $\theta_{i,j,k}^{1,f}\sim N\left(5,1\right)$ or $\theta_{i,j,k}^{1,f}\sim -N\left(5,1\right)$, if the crisp membership function decreases or increases, respectively. The definition of the $\theta_{i,j,k}^{1,f}$ and $\theta_{i,j,k}^{2,f}$ is made so as the fuzzy membership function initially resembles the crisp one. All the fuzzy model's weights are initialized as: $w_i^c\sim U\left(0.95,1.05\right)$, $w_{i,j}^g\sim U\left(0.95,1.05\right)$, $w_{i,j,k}^l\sim U\left(0.95,1.05\right)$. Again, the values of the weights are considered as 1 in the crisp model and the above initialization is made so as the initial fuzzy model resembles the crisp one.

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 $W=\{w_i^{c*}, w_{i,j}^{g*}, w_{i,j,k}^{l*}\}$. The employment of local optimization technique along with the described initialization of the optimization parameters ensures that the final solution will be relatively close to the one obtained from the initial set of rules, maintaining in this way its transparency and interpretability.

III. Results

Table I presents the datasets which were employed from the UCI machine learning repository [26], 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 of the small number of missing values; none or very few values are missing in each dataset. Missing values are replaced by the average of the column (attribute).

TABLE I
DESCRIPTION OF DATASETS USED FOR THE EVALUATION OF THE PROPOSED METHODOLOGY

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

The ten fold stratified cross validation method was used for the evaluation [27]. The procedure was applied to each fold, generating ten different decision trees and, subsequently, ten different crisp and fuzzy models. Decision trees were implemented using the C4.5 algorithm. Post pruning was employed, using the pessimistic error rate based method (sub-tree replacement). The confidence factor for pruning was set to 0.25 and the minimum number of instances in a leaf was 2.

Table II presents the total accuracy (average accuracy of the ten folds) for all five datasets, for the decision tree and all stages of the proposed methodology. In addition, overall results are presented, for each stage, in terms of average accuracy. The average accuracy for all employed datasets is 78.42%, for the decision tree, while the results obtained in each optimization stage of the proposed methodology are 78.36%, 79.7%, 80.39% and 80.31%, for each of the four stages, respectively. As it is shown in Table II, the introduction of weights in the fuzzy models and their optimization improves the classification accuracy of the decision tree.

IV. DISCUSSION

In this paper we perform an analysis concerning weights in fuzzy modeling. Fuzzy models are automatically generated, 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, 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, and (iii) optimization of the fuzzy model's parameters. Additionally, three sets of weights are introduced and optimized using a four-stage optimization strategy. Results are presented for five datasets obtained from the UCI machine learning repository.

The proposed realization, which falls in the third of the categories which are mentioned in the introduction (i.e. induction of a crisp decision tree, extraction of a set of rules from it and fuzzyfication of these rules), presents several similarities with the approaches from the other two categories (i.e. induction of a crisp decision tree from the data and then its fuzzyfication, resulting into a fuzzy decision tree or induction of fuzzy decision tree, integrating fuzzy techniques during the tree construction). However, an important difference is that in a fuzzy decision tree, each node receives a single value for each of its parameters, thus having the same decision making functionality for all child-leaves. In the case of rule extraction from the tree, each node is part of more than one rule (i.e. it is included as a conjunct in all the rules corresponding to its child-leaves) and receives different parameter values for each one of them. This increases the optimization time required compared to the fuzzy decision trees due to the larger number of parameters which are introduced. However, it also bestows a major advantage on the decision making procedure of the fuzzy rules compared to the fuzzy decision trees since the larger number of parameters allows the fuzzy rules to be more flexible and thus more adaptable to a specific dataset. In addition, the complexity of the decision making process remains the same (the number of nodes that have to be parsed to reach a decision is the same with the number of conjuncts of the respective rule).

In the proposed methodology, the fuzzy model is created by fuzzyfing an initial crisp model (set of rules). The initial crisp set of rules can be either provided by experts, thus leading to a knowledge based fuzzy model, or extracted directly from the data, using data mining methods, and thus leading to a data driven fuzzy model. In the latter case, several rule mining techniques can be employed. The proposed methodology can employ in its first stage any rule mining technique. In the current approach we selected decision trees with the C4.5 algorithm, for extraction of the initial set of rules, since C4.5 is considered as a state of the art tree induction algorithm, presenting several advantages, such as the extraction of a mutually exclusive and exhaustive

TABLE II
CLASSIFICATION ACCURACY (%) FOR THE DECISION TREE AND THE FOUR OPTIMIZATION STAGES OF THE PROPOSED METHODOLOGY

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±10.28	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
average	78.42	78.36	79.7	80.39	80.31

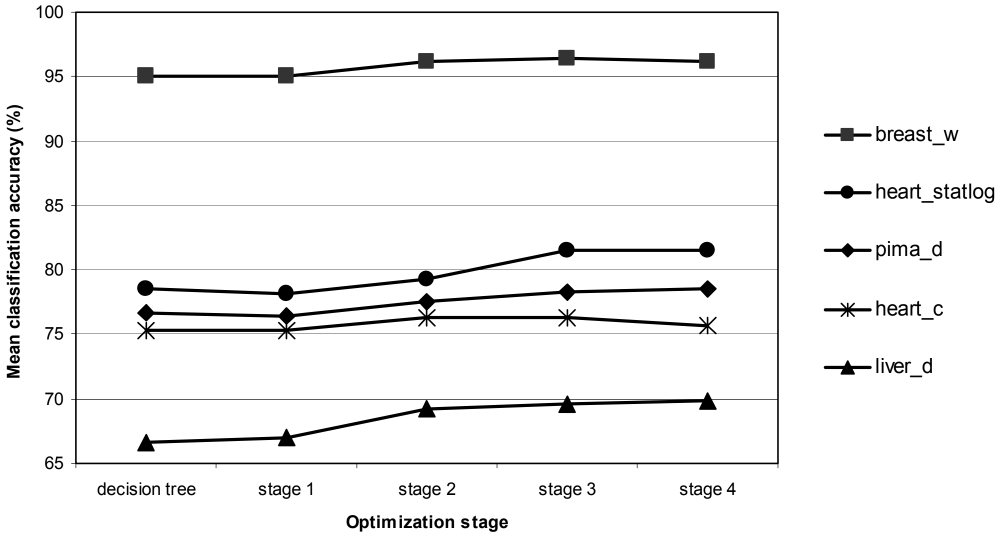

Fig. 2. Graphical representation of the obtained classification accuracy results (%).

set of rules. Also, its low computational complexity is another advantage of C4.5 [28]. However, any other rule mining technique can be integrated in the first stage of the methodology as well. It should be mentioned that the quality of the generated fuzzy model, greatly depends on the quality of the initial crisp set of rules.

The employed optimization procedure is sequential; each set of parameters included in the fuzzy model is optimized independently. This four-stage sequential optimization procedure usually results to a suboptimal solution compared to an optimization procedure that optimizes the fuzzy model with respect to all sets of parameters to a single-stage. However, the number of parameters included in each set varies greatly. Additionally, the importance of each set is not the same; class weighs are more important than the other sets of weighs since they greatly affect the behavior of the fuzzy model. A single-stage optimization procedure tends to optimize only the parameters of high importance, leaving the rest parameters unaffected.

Concerning the evaluation procedure, five benchmark datasets has been employed, with different number of instances and attributes, having different classification complexity. Also, all evaluation experiments were conducted using ten-fold stratified cross validation, which is considered as the most reliable approach for evaluation [27], which is an advantage compared to other approaches presented in the literature, which have been evaluated using different evaluation strategies [8,18,29,30]. These features ensure that the evaluation is adequate to demonstrate the advantages/disadvantages of our methodology and fully exploit its potential.

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 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. In Fig. 2, a graphical representation of the classification accuracy obtained in the decision trees and the four stages of the proposed methodology, for all five datasets, is presented. 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 III 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. [8], Abonyi et al. [18], Crockett *et al.* [11] and Olaru *et al.* [9] 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 III: 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 Olaru *et al.* In general, the results obtained in this work are comparable or better than those reported in the literature.

Suarez *et al.* [8] used three common datasets for the evaluation of two different architectures, reporting an average 82.73% accuracy for the first and 81.4% accuracy for the second, respectively. Both architectures present lower average results compared with this work. It should be mentioned that the evaluation performed in [8] is based on 10 different randomly selected training-test sets while in the proposed work, 10 fold stratified cross validation is used,

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY (%) RESULTS PRESENTED IN THE LITERATURE

datasets	Suarez et al. [8]	Suarez et al. [8]	Crockett et al. [11]	Abonyi et al. [18]	Olaru et al. [9]	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

which is considered more reliable. The approach proposed by Crockett *et al*. [11] was also evaluated using the same three datasets, presenting slightly better overall results compared to the proposed work (0.51%). The work of Abonyi *et al*. [18] uses two common datasets, reporting average accuracy 84.94%, which is lower than the accuracy reported from proposed method (using the same two datasets). The evaluation performed in [18] is based on 5 fold cross validation. In [18] a model simplification stage is included, thus the generated fuzzy models are simpler (i.e. have less fuzzy rules) than the ones generated in this study. Finally, the method presented by Olaru *et al*. [9] reports lower average results compared to the proposed methodology.

V. CONCLUSION

In this work we focused on weight analysis of fuzzy models. The fuzzy models structure is based on fundamental definitions of its aspects, i.e. fuzzy membership function, T_{norm} and S_{norm} definitions and defuzzyfication technique. Future work will focus on employing different definitions of each of these elements and thus defining more complex fuzzy models. Also, since the current approach of the methodology was evaluated using only biomedical datasets, further evaluation of the methodology is needed with additional datasets in order to fully exploit its potential. Additionally, the employment of different local or global optimization techniques, as long as 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.

REFERENCES

[1] L.X. Wang, *A course in fuzzy systems and control*, Upper Saddle River NJ, Prentice Hall PTR, 1997.

[2] L.H. Tsoukalas, R.E. Uhrig, *Fuzzy and neural approaches in engineering*, John Wiley & Sons, 1997.

[3] W. Pedrycz, J.V. de Oliveira, "An algorithmic framework for development and optimization of fuzzy models", *Fuzzy sets and systems,* vol. 80, pp. 37-55, 1996.

[4] J.M. Garibalti, E.I. Ifeachor, "Application of simulated annealing fuzzy model tuning to umbilical cord acid-base interpretation", *IEEE Trans. Fuzzy Systems*, vol. 7, pp. 72-84, 1999.

[5] Y. Goletsis, C. Papaloucas, D.I. Fotiadis, A. Likas, L.K. Michalis, Automated ischemic beat classification using genetic algorithms and multicriteria decision analysis, *IEEE Transactions on Biomedical Engineering*, vol. 51(10), pp. 1717–1725, 2004.

[6] M.G. Tsipouras, C. Voglis, D.I. Fotiadis, "A Framework for Fuzzy Expert System Creation – Application to Cardiovascular Diseases", *IEEE Transactions on Biomedical Engineering*, vol. 54(11), pp. 2089–2105, 2007.

[7] B. Jeng, Y.-M. Jeng, T.-P. Liang, "FILM: a fuzzy inductive learning method for automated knowledge acquisition", *Decision Support Systems*, vol. 21, pp 61–73, 1997.

[8] A. Suarez, F. Lutsko, "Globally optimal fuzzy decision trees for classification and regression", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21(12), pp. 1297-1311, 1999.

[9] C. Olaru, L. Wehenkel, "A complete fuzzy decision tree technique", *Fuzzy sets and systems*, vol. 138, pp. 221–254, 2003.

[10] Y.-T. Chen, B. Jeng, "MFILM: a multi-dimensional fuzzy inductive learning method", *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 17(3), pp. 267–281, 2005.

[11] K. Crockett, Z. Bandar, J. O'Shea, D. Mclean, "On Constructing a Fuzzy Inference Framework using Crisp Decision Trees", *Fuzzy Sets and Systems*, vol. 157, pp. 2809–2832, 2006.

[12] Y. Yuan, M.J. Shaw, "Induction of fuzzy decision trees", *Fuzzy Sets and Systems*, vol. 69, pp. 125–139, 1995.

[13] H. Ichihashi, T. Shirai, K. Nagasaka, T. Miyoshi, "Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning", *Fuzzy Sets and Systems*, vol. 81, pp. 157–167, 1996.

[14] B. Apolloni, G. Zamponi, A.M. Zanaboni, "Learning fuzzy decision trees", *Neural Networks*, vol. 11, pp. 885–895, 1998.

[15] C.Z. Janikow, "Fuzzy decision trees: issues and methods", *IEEE Trans. Systems Man and Cybernet.*, vol. 28(1), pp. 1–14, 1998.

[16] X. Wang, B. Chen, G. Qian, F. Ye, "On the optimization of fuzzy decision trees", *Fuzzy Sets and Systems*, vol. 112, pp. 117-125, 2000.

[17] E.C.C. Tsang, X.Z. Wang, Y.S. Yeung, "Improving learning accuracy of fuzzy decision trees by hybrid neural networks", *IEEE Trans. Fuzzy Systems*, vol. 8(5), pp. 601–614, 2000.

[18] 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*, pp. 1-21, 2003.

[19] S.M. Chen, "A new approach to handling fuzzy decision making problems", *IEEE Trans. Systems Man Cybernet.*, vol. 18, pp. 1012-1016, 1988.

[20] D.L. Hudson, M.E. Cohen, "The role of approximate reasoning in medical expert system", in A. Kandel (ed.), *Fuzzy Expert Systems*, CRC Press, pp. 165-179, 1991.

[21] H.C. Tseng, D.W. Teo, "Medical expert system with elastic fuzzy logic", *3rd IEEE Internat. Conf. on Fuzzy Systems*, vol. 3, 1994, pp. 2067-2071.

[22] D.S. Yeung, E.C.C. Tsang. "Weighted fuzzy production rules", *Fuzzy sets and systems*, vol. 88, pp. 299-313, 1997.

[23] J.R. Quinlan. C4.5: *Programs for Machine Learning*, California, Morgan Kauffman, 1993.

[24] X.-J. Zeng, M.G. Singh, "Approximation accuracy analysis of fuzzy systems as function approximators", *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 44-63, 1996.

[25] 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*, vol. 9(1), pp. 112-147, 1998.

[26] 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.

[27] R. Kohavi. "A study of cross-validation and bootstrap for accuracy estimation and model selection", *14th International Joint Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann, 1995, pp. 1137 – 1143.

[28] J.R. Quinlan, "Improved use of continuous attributes in C4.5", *Journal Artificial Intelligence Res,* vol. 4, pp. 77–90, 1999.

[29] T.P. Exarchos, M.G. Tsipouras, C.P. Exarchos, C. Papaloukas, D.I. Fotiadis, L.K. Michalis, "A Methodology for the Automated Creation of Fuzzy Expert Systems for Ischaemic and Arrhythmic Beat Classification Based on a Set of Rules Obtained by a Decision Tree", *Artificial Intelligence in Medicine*, vol. 40(3), pp. 187-200, 2007.

[30] M.G. Tsipouras, T.P. Exarchos, D.I. Fotiadis, "Integration of global and local knowledge for fuzzy expert system creation – Application to Arrhythmic Beat Classification", *29th IEEE EMBS Annual International Conference*, Lyon, France, pp. 3840-3843, 2007.