Artificial Intelligence in Medicine (2009) **47**, 25—41

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

An interpretable fuzzy rule-based classification methodology for medical diagnosis

Ioannis Gadaras[*], **Ludmil Mikhailov**

University of Manchester, School of Computer Science, Manchester, M13 9EP, United Kingdom

Received 14 May 2008; received in revised form 5 May 2009; accepted 10 May 2009

KEYWORDS
Fuzzy rule extraction from data;
Hierarchical and flexible input partitioning;
Interpretable and scalable knowledge;
Medical diagnosis

Summary

Objective: The aim of this paper is to present a novel fuzzy classification framework for the automatic extraction of fuzzy rules from labeled numerical data, for the development of efficient medical diagnosis systems.
Methods and materials: The proposed methodology focuses on the accuracy and interpretability of the generated knowledge that is produced by an iterative, flexible and meaningful input partitioning mechanism. The generated hierarchical fuzzy rule structure is composed by linguistic; multiple consequent fuzzy rules that considerably affect the model comprehensibility.
Results and conclusion: The performance of the proposed method is tested on three medical pattern classification problems and the obtained results are compared against other existing methods. It is shown that the proposed variable input partitioning leads to a flexible decision making framework and fairly accurate results with a small number of rules and a simple, fast and robust training process.

1. Introduction

Over the last decades, there have been numerous implementations of computer systems in medicine [1]. Despite the increasing scientific evolution of both information technology and medicine, the inherent uncertainty of the latter makes the fusion of these technologies a rather difficult task. The main sources of this natural imprecision are due to the insufficient understanding of biological mechanisms and their interactions, and the ambiguity of medical results and measurements. Furthermore, many diseases appear in multiple stages, in combination with other similar disorders and with different symptoms of variable extension and sequence. Therefore, it would be essential and very beneficial to ensure fast, accurate and meaningful diagnosis for a number of widespread and fatal diseases. That would additionally improve the effectiveness of medical treatment as well as the speed and accu-

[*] Corresponding author at: Flat 41, Trinity Court, Higher Cambridge St. 41, Manchester, M1 6AR, United Kingdom.
Tel.: +44 7737 53 0964/161 306 3361; fax: +44 161 306 3518.
E-mail addresses: i.gadaras@student.manchester.ac.uk (I. Gadaras), ludi.mikhailov@manchester.ac.uk (L. Mikhailov).

0933-3657/$ — see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2009.05.003

racy of the remedy reaction, affecting the recovery and life expectancy of the patient and the operational efficiency of the medical units.

If we additionally consider the increasingly growing amount of various collected medical data, we can easily appreciate the necessity of its categorization and the expediency of such a classification framework.

The natural evolution of various diseases, the obscure nature of medical data and the intrinsic ambiguity of medical problems require a consistent framework that can handle uncertainty by allowing variable and multiple class memberships and facilitating approximate reasoning. This inevitably makes the fuzzy logic (FL) a valuable tool for depicting medical concepts by treating them as fuzzy sets [2]. In addition, the FL and the utilization of linguistic/fuzzy variables provide a rigorous framework for verbal representation of numerical concepts that can be then embedded in meaningful fuzzy rules. Such rules can be easily comprehended, verified, tuned and possibly expanded by medical experts, and used for the development of classification systems that can be very valuable in the process of medical diagnosis.

Each fuzzy variable is composed of a set of membership functions that determine the degree of fitness of numerical examples to each particular fuzzy set. This way, the FL can represent variable degrees of an illness and symptoms via multiple class memberships and provide an approximate but verbally consistent and accurate inference process.

Two main factors are proved to be critical for the success of any medical diagnosis process, based on fuzzy reasoning:

1. A fast and accurate input partitioning method that attempts to find the soft class boundaries by automatically processing a series of representative examples.
2. A verbally interpretable knowledge representation framework that allows the verification and integration of the generated rules.

In this paper we treat the medical diagnosis as a pattern classification problem and try to match symptoms against diseases by learning from medical data. Thus we can classify potential patients according to numerical values of their symptoms and their degrees of membership in various classes, which are represented as fuzzy sets.

The proposed classification methodology initially identifies fuzzy boundaries of the classes by processing a set of labeled data. Exploring the characteristics of the identified boundaries, membership functions for each class are automatically produced and corresponding fuzzy rules are obtained. When new patterns need to be classified, their numerical attributes are tested against generated knowledge in order to match the symptoms with a rule's antecedent. When that happens, the appropriate rule is executed to identify the most appropriate class.

The paper is structured as follows. In the next section some related methods and technologies are shortly presented. The proposed fuzzy classification method is described in Section 3. Section 4 presents some numerical results of the proposed method against three different medical datasets, followed by the conclusions in Section 5.

2. Related technologies

One of the earliest attempts to formalize medical diagnosis applications was based on a pattern classification approach to identify the decision boundaries of various diseases from data [3]. Expert systems [4,5] were introduced soon after with MYCIN [6], a medical knowledge based system that was using certainty factors to express uncertainty.

The FL was applied in medical systems [7], almost 20 years after its introduction by Zadeh [8], but has recently given birth to various interesting implementations [9,10]. Most of them employ fuzzy clustering algorithms [11] or connectionist neuro-fuzzy methodologies [12] to separate the input space and automatically extract fuzzy rules directly from data. However, those methods have a number of critical disadvantages. Clustering algorithms do not use class labels but usually a gain/loss objective function, yielding sometimes suboptimum results that are sometimes vulnerable to parameters initialization [13,14]. On the other hand, neural learning is usually slow, order dependent and incomprehensible, since the extracted knowledge is represented in terms of numerical weights.

Genetic algorithms (GA) and genetic programming emerged four decades ago but have been recently proven to be fairly successful and popular [15,16]. The GA mimic the process of natural evolution, using the survival of the fittest and natural selection principles for tackling classification/optimization problems. Most of GA perform exhaustive search iterations on a population of candidate results and select a competitive set each time. They attempt to obtain an optimum result by swapping parts, selectively mutating chromosomes that encode the solution and evaluate candidate combinations against a fitness function. This procedure has been proved to be effective, as it is used in natural evolution and is extensively used in fuzzy-genetic applications, however it is usually slow due to its exhaustive character and is very sensitive to various parameters.

Finally, a number of fusion methodologies have also been recently developed, combining some of the above-mentioned technologies. The most popular attempt is probably NEFCLASS neuro-fuzzy classification model [17], a three layer feed-forward neural architecture similar to the fuzzy perceptron [18]. Several similar combinatorial attempts have been proposed over the last decade including neuro-fuzzy systems [19], evolutionary-fuzzy [20] and lastly, neuro-genetic-fuzzy classification frameworks [21].

All the previous mentioned methodologies have their own practical advantages and weaknesses that depend upon the implementation tools, strategy and operational criteria. In contrast, a fairly contemporary methodology named cooperative rules methodology [22], attempts to discuss and focuses on the interpretability and accuracy of the generated knowledge and provides a valid framework to handle it. By using a series of variable fuzzy grid partitions the model produces several fuzzy rules that describe the same input space in a variable granularity. By using a selected set of rules, this feature provides the flexibility to both effectively represent the generated knowledge and achieve a highly accurate classification performance.

Despite the interpretability and the originality of the above approach, it has a considerable disadvantage. It produces an excessive number of fuzzy rules and requires an additional simulated annealing [23] algorithm that attempts to identify the rule set with the best combined accuracy. These two operational attributes seriously affect its speed and complexity and the size of the rule base when handling multidimensional data.

The methodology proposed in this paper is focusing on an alternative manner of modeling the interpretability-accuracy trade off by a more meaningful and simple input space partitioning that generates additional rules only to accurately describe specific overlapping regions.

3. Fuzzy rule-based classification

According to the proposed method, the pattern classes are represented by a possibly overlapping set of n-dimensional hyperboxes (rectangular representations of n-dimensional hyper-cubes). The values of their boundaries are used for generating membership functions and the corresponding fuzzy rules. In contrast to the above-mentioned technologies, the proposed approach allows for overlapping of partitions corresponding to different classes and does not require any optimization component. The classification conflict that occurs is effectively resolved by regarding the overlapping region as a separate fuzzy hyperbox and possibly re-dividing it, to appropriately assign a class to the corresponding testing data-patterns.

The initial phase of the proposed method is based on a novel approach to the generation of hyperboxes using relational algebra operations and the fuzzy set theory for their mapping to appropriate fuzzy rules. All hyperboxes are completely defined by the minimum and maximum values of the input data of each class and for all the input dimensions. The combinations of these min—max points delineate the membership functions and the equivalent fuzzy sets for each class. The Mamdani type of fuzzy rules produced are both linguistically comprehensible and accurate and can be used to classify similar disease patterns. This is achieved by loading similar medical examples, computing the membership values for each existing fuzzy set and finally assigning the pattern to the class with the largest membership.

3.1. Input partitioning

Let us consider a set of training input—output data pairs $[(x_1^{(1)}, x_2^{(1)}, \ldots, x_m^{(1)};\ y^{(1)}), (x_1^{(2)}, x_2^{(2)}, \ldots, x_m^{(2)};\ y^{(2)}), \ldots, (x_1^{(N)}, x_2^{(N)}, \ldots,\ x_m^{(N)};\ y^{(N)})]$, where $x_k^{(j)}$ is the k-th attribute of the input vector x for the j-th data pair, j = 1, 2, ..., N, and y is a classification label, which can be considered as an output variable. We assume that there is a discrete set of n distinct classes Y_i, i = 1, 2, ..., n, and the classification label of each data pair belongs to one of those classes. Our aim is to produce a set of fuzzy classification rules from the given training input—output data pairs.

Considering the set of all input vectors X_i that produce an output in class Y_i, we can obtain a single hyperbox A_i for the i-th class, which is defined by the minimum $_{ik}$ and maximum V_{ik} values of all $x \in X_i$ for the k-th dimension:

$$A_i = \{x \in X_i|_{ik} \le x_k \le V_{ik}, k = 1, 2, \ldots, m\}. \qquad (1)$$

Generally, each hyperbox A_i may contain data from other classes as well and therefore, it may overlap with hyperboxes, corresponding to different classes. If two hyperboxes A_i and A_j overlap, their intersection $A_{ij} = A_i \cap A_j$ forms a new hyperbox A_{ij}, which may or may not contain data from both classes. It is even possible that the intersection hyperbox contains no data at all.

Similarly, we can form overlapping hyperboxes between more than two classes. For example, an overlapping hyperbox between three classes is

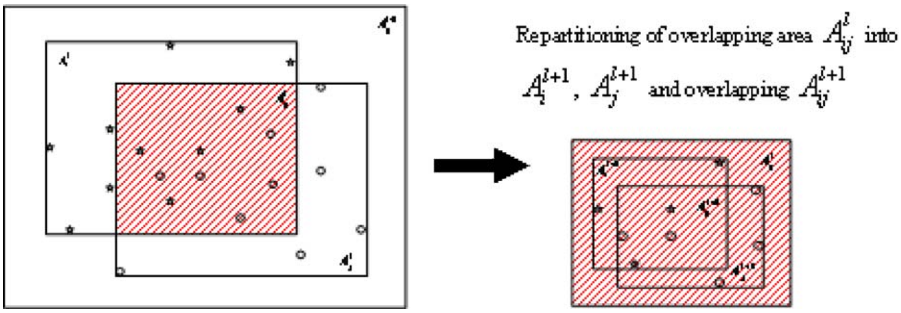
Repartitioning of overlapping area A^l_{ij} into
A^{l+1}_i, A^{l+1}_j and overlapping A^{l+1}_{ij}

Figure 1 The automatic process of hyperbox generation and repartitioning of the overlapping area.

denoted by A_{ijk}, $A_{ijk} = A_i \cap A_j \cap A_k$. For brevity, we are not going to discuss overlapping between more than two class hyperboxes, but the proposed approach could be easily extended to such cases.

Provided that there exists overlapping hyperboxes, a recursive algorithm is applied for further partitioning of the input space. It analyses the overall partitioning, obtained from the previous iteration and repartitions the overlapping sections. Each overlapping hyperbox A^l_{ij}, obtained during the *l-th* iteration, could be further partitioned into two new hyperboxes from both classes A^{l+1}_i and A^{l+1}_j, such that $A^{l+1}_i \subset (A^l_i \cap A^l_j), A^{l+1}_j \subset (A^l_i \cap A^l_j)$, if some conditional criteria are not satisfied.

An example of the iterative partitioning process is shown in Fig. 1.

As a result, for each *i*-th classification class Y_i, $i = 1, 2, \ldots, n$, a hierarchy of nested hyperboxes $A^0_i, A^1_i, A^2_i, \ldots, A^L_i$, is generated iteratively, where *L* denotes the depth of the repartitioning process and number of iterations—see Fig. 2. The hierarchy represents the particular input pattern space in a granular fashion, so that each next level represents more detailed partitioning.

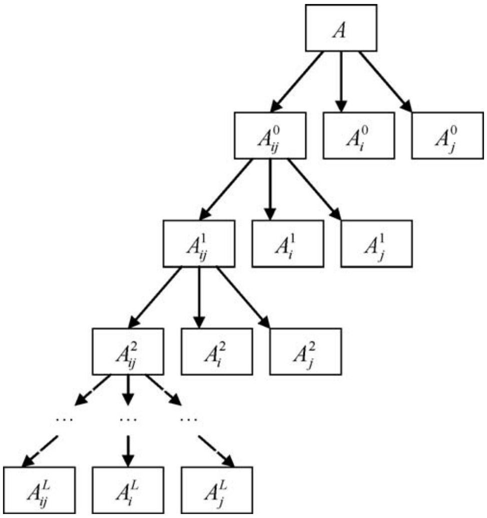

Figure 2 The iterative partitioning fashion of the overlapping region.

The partitioning of any overlapping area is based on some conditional criteria, which determine the partition granularity and the overall number of iterations.

The first criterion examines the density of the input data in the overlapping area A^l_{ij}, related to the overall number of data in A_{ij}. The relative density R^l_{ij} is given by

$$R^l_{ij} = \frac{D(A^l_i \cap A^l_j)}{D(A^1_i \cup A^1_j)},$$

where the operator $D(.)$ denotes the number of input patterns in the specified area.

A value of R^l_{ij} close to zero indicates that the amount of data in A^l_{ij} is very small, related to all data in $A^1_i \cup A^1_j$ and therefore no further partitioning is required. The stopping condition is defined by a threshold value *Th*1, which is set by the decision maker. If $R^l_{ij} > Th1$, an additional grouping of the intersection will be triggered, if $R^l_{ij} < Th1$, the partitioning of A^l_{ij} is terminated.

When the partitioning process finalises, another criterion is applied to assign appropriate weights to the rules that correspond to the intersection. This criterion examines the population of the intersection area in terms of the different classes:

$$S^l_{ij} = \frac{|D_i(A^l_i \cap A^l_j) - D_j(A^l_i \cap A^l_j)|}{D(A^l_i \cap A^l_j)},$$

where $D_i(.)$ and $D_j(.)$ are the numbers of input patterns from class *i* and *j* in the overlapping area.

The ratio S^l_{ij} takes values between zero (when A^l_{ij} is equally populated with data from both classes), and one (when the overlapping area does not contain data from one of the two classes). The second criterion manages the type of the weights of the weighted rules. When $S^l_{ij} < Th2$, which means that the overlapping area is equally populated, distance-based weights are used. Otherwise, when the intersection is dominantly populated by one class, $S^l_{ij} > Th2$, density weights that favour that class are used in the rules consequents. The value of the threshold parameter *Th*2 seriously influences

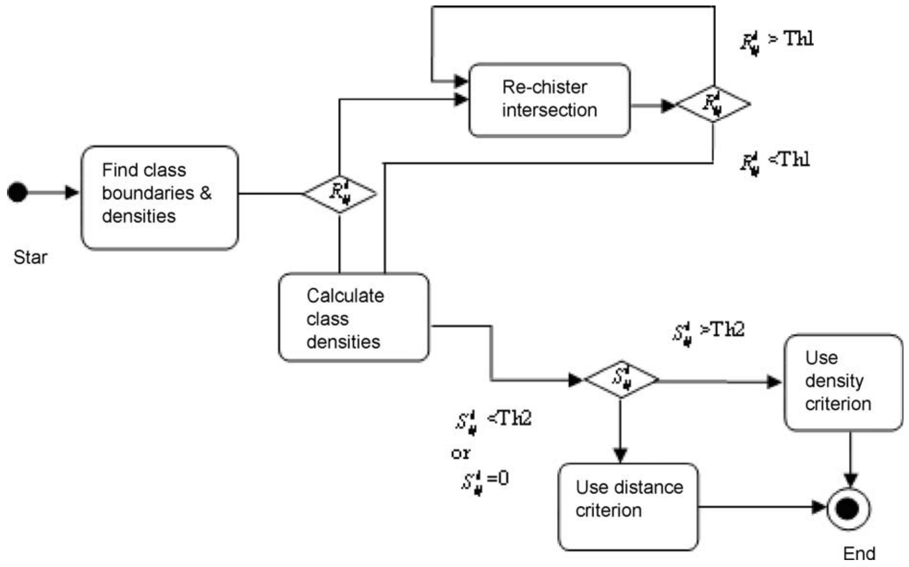

Figure 3 The activity diagram of the feature partitioning process using the two threshold values.

the classification accuracy, and it should be appropriately chosen by the user.

The iterative partitioning process can also be formally depicted in the activity diagram, shown in Fig. 3.

The final result of the recursive partitioning approach is the generation of two different types of regions in the input feature space: regions, containing data from one single class only, and regions with relatively small amount of data of different classes. The regions of mixed data are identified in the list of 'completed' partitions and are generally obtained during the last iterations of the partitioning process.

3.2. Fuzzy rules generation

For each hyperbox A^l_i, containing data from a single class Y_i only, we can introduce a fuzzy rule of the type:

$$\text{IF } x \text{ is in } A^l_i \text{ THEN } y \text{ is in } Y_i \tag{2}$$

If the hyperbox A^l_i overlaps with A^l_j, the fuzzy rule for the i-th class takes the form:

$$\text{IF } x \text{ is in } A^l_i \text{ and } x \text{ is not in } A^l_{ij} \text{ THEN } y \text{ is in } Y_i \tag{3}$$

For each 'completed' overlapping hyperbox A^l_{ij} an additional rule is generated, which has a double-consequent part, as the hyperbox might contain data from two classes. It is possible, however, for the hyperbox to contain data from one class only, then $S^l_{ij} = 1$. Generally, the amount of data in any 'completed' overlapping hyperbox is very small, compared to the single class hyperboxes A^l_i, l = 0, l, ..., L, which is ensured by the relative density criterion and the stopping condition $R^l_{ij} < Th1$. Therefore, without a significant lost of accuracy, we can introduce additional weights, representing the relative densities of the data of each class in the corresponding hyperbox. The linguistic form of that rule is

$$\begin{aligned}&\text{IF } x \text{ is in } A^l_{ij} \text{ THEN } y \text{ is in } Y_i \text{ when } w_i > w_j,\\ &\quad \text{OR } y \text{ is in } Y_j \text{ when } w_i < w_j,\end{aligned} \tag{4}$$

where w_i and w_j are the weights of the consequents, calculated by

$$w_i = \frac{D_i(A^l_{ij})}{D(A^l_{ij})}, \quad w_j = \frac{D_j(A^l_{ij})}{D(A^l_{ij})}.$$

In practice, in the inference process, in the case where $w_i > w_j$, the rule that results to Y_i will be triggered, or else when $w_i < w_j$, the rule that produces Y_j will be fired instead. That effectively means that both rules of type (3) will be generated for both classes but only one of them will be fired depending to the result of $w_i < w_j$, the values of which were obtained from the training process.

When the density indexes of both classes $D_i(A^l_{ij})$ and $D_j(A^l_{ij})$ are equal, $w_i = w_j$, distance-based weights are calculated. According to that, an input vector in the intersection region is classified to the 'nearest' class by simply comparing its Euclidean distances d_i and d_j from the classes-centers (centroids) C_i and C_j. The centroids are calculated as arithmetic means for each k-th dimension, over all the points of the cluster. For example, if the intersection hyperbox contains N points from the class Y_i, $x^{(p)} = (x_1^{(p)}, x_2^{(p)}, \ldots, x_m^{(p)})$, $p = 1, 2, \ldots, N$, then the centroid of the i-th cluster is defined by

$$C_i = (x_1^{c_i}, x_2^{c_i}, \ldots, x_m^{c_i}), \qquad x_k^{c_i} = \frac{x_k^{(1)} + x_k^{(2)} + \cdots + x_k^{(N)}}{N}, \quad k = 1, 2, \ldots, m.$$

Consequently, the rule corresponding to the intersection region takes the form:

$$\text{IF } x \text{ is in } A_{ij}^{l} \text{ THEN } y \text{ is in } Y_i \text{ when } d_j > d_i$$
$$\text{OR } y \text{ is in } Y_j \text{ when } d_i > d_j, \qquad (5)$$

where the Euclidean distances from the vector x to the centroids are calculated as:

$$d_i = \sqrt{(x_1 - x_1^{C_i})^2 + (x_2 - x_2^{C_i})^2 + \cdots + (x_m - x_m^{C_i})^2},$$
$$d_j = \sqrt{(x_1 - x_1^{C_j})^2 + (x_2 - x_2^{C_j})^2 + \cdots + (x_m - x_m^{C_j})^2}.$$

The coexistence of single and double-consequent fuzzy rules was first proposed by Nozaki et al. [24] resulting in two separate knowledge bases, that can be combined during the inference process. Cordon and Herrera [21] have also used an analogous rule structure in order to improve the accuracy and interpretability of the fuzzy model. Here we use the similar type of rules but instead of using evolutionary criteria for selecting the appropriate rule, a fractional variable is introduced to control the generation of additional rules and assign an appropriate consequent. The ability to choose from the two classes provides the necessary flexibility for the classification process to effectively handle the overlapping parts of the hyperbox classes.

As a result, these indexes simply represent the dominance of each class in the common region and determine which of the two consequents will be assigned to the rule. Weighted fuzzy rules have been extensively used in the past [25] to guarantee the execution of the appropriate rule and a similar concept have been applied in the proposed method.

For the reasoning process, a rule with dual consequents is equivalent to 2 additional rules with single consequents. For example, the rule (4) is substituted by the following two rules:

$$\text{IF } x \text{ is in } A_{ij}^{l} \text{ AND } w_i > w_j \text{ THEN } y \text{ is in } y_i,$$
$$\text{IF } x \text{ is in } A_{ij}^{l} \text{ AND } w_i < w_j \text{ THEN } y \text{ is in } Y_j. \qquad (6)$$

3.3. Fuzzy inference process

In pattern classification, the degree of membership of a 'new' input pattern x for rules given by (2) and (3), depends on the relative position of the input's coordinates regarding the generated partitions. Hence, we can naturally assume that the degree of membership of an input to a specific class is equal to one if it lies inside the classes' hyperbox and decreases as it moves away from its boundaries. The fuzzy boundaries of each hyperbox can be represented graphically as a 'generalization area', whose contour surface is parallel to the hyperbox. All inputs within the generalization area partially belong to the corresponding hyperbox and therefore their degree of membership is less than 1 but greater than 0. That kind of fuzzy boundaries can be represented mathematically by trapezoid membership functions, as they can represent a full membership by their upper base and the linearly decreasing degree of membership with their sloped sides.

For a class partition A_i^l, which does not intersect with another class partition, the membership function is described by the following equation:

$$m_i^l(x_k) = \min\{[1 - \max(0, \min(1, \gamma_k({}_{ik}^{l} - x_k)))],$$
$$[1 - \max(0, \min(1, \gamma_k(x_k - V_{ik}^l)))]\} \qquad (7)$$

where γ_k is the sensitivity parameter for the k-th input component x_k.

Fig. 4 shows graphically the generalization area of a class partition A_i^l, which does not intersect with another one, the corresponding trapezoidal membership function and its parameters. For each input vector that lies inside the hyperbox A_i^l, all its components satisfy x_k the inequalities ${}_{ik}^{l} < x_k < V_{ik}^l$, $k = 1, \ldots, m$ and have degrees of membership $m_i^l(x_k)$ equal to one. When the input vector is outside the class hyperbox A_i^l, but near its borders, the degree of membership of some components x_k (or for all of them) is positive, but less than one, $0 < m_i^l(x_k) < 1$. Such input vectors satisfy the double-side inequality ${}_{ik}^{l} - 1/\gamma_k < x_k < V_{ik}^l + 1/\gamma_k$, and therefore belong to the generalization region around the class hyperbox. And finally, if there is at least one component x_k such that ${}_{ik}^{l} - 1/\gamma_k \geq x_k \geq V_{ik}^l + 1/\gamma_k$, the corresponding degree of membership is $m_i^l(x_k) = 0$ and the input vector is outside the generalized region.

By varying the value of γ_k we can achieve reasonable tuning of the membership functions, since we can expand or contract the generalization region around the crisp class hyperbox.

When the hyperbox A_i^l overlaps with A_j^l, the membership functions for each area A_i^l, we introduce a similar but non-symmetrical membership function of the type:

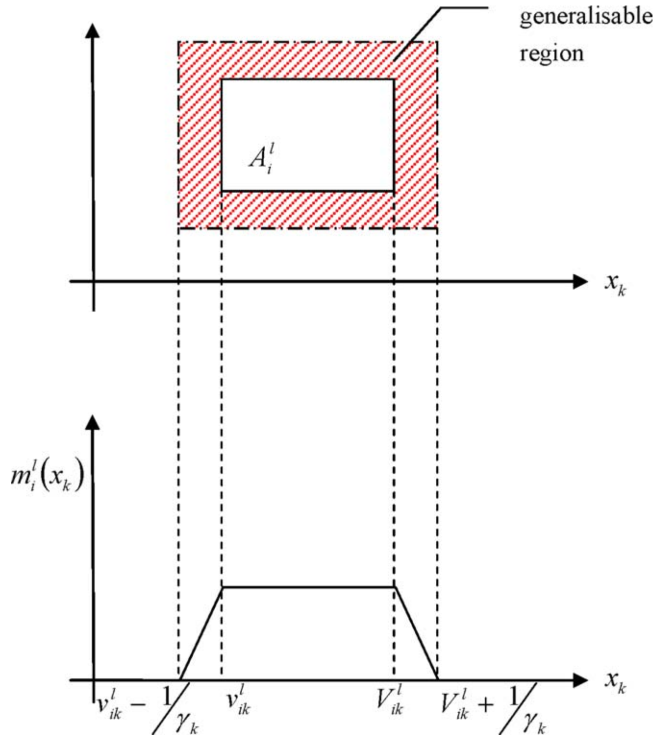

Figure 4 Class boundaries and a membership function of a two-dimensional hyperbox.

$$m_i^l(x_k) = \min\{[1 - \max(0, \min(1, \gamma_k({}^{l}_{ik} - x_k)))],$$
$$[1 - \max(0, \min(1, (1/(V_{ik}^l + 1/\gamma_k - {}^{l}_{jk}))(x_k - V_{ik}^l)))]\} \quad (8)$$

Fig. 5 illustrates a case with two overlapping hyperboxes, and their non-symmetrical membership functions, defined by (8).

For each class hyperbox A_i^l with a membership function given by (7), a rule R_i^l of the type (2) is generated. For the overlapping case, shown in Fig. 5, two new rules of the type (3) R_i^l and R_j^l are generated. In order to define the degree of membership $d_{R_i^l}(x)$ of each rule R_i^l, we use the *min* operator. The rule degree of membership is calculated by taking the minimum of the membership

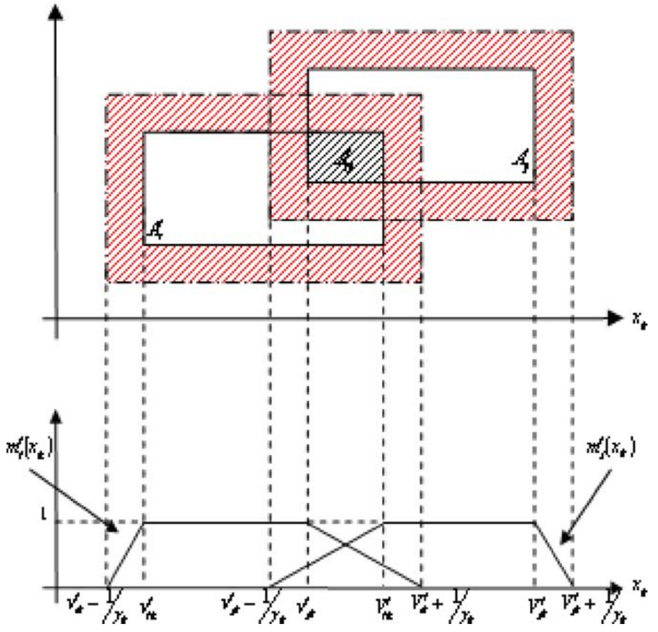

Figure 5 Class boundaries and membership functions of two overlapping hyperboxes.

values of the fuzzy value rule antecedent for a given input:

$$d_{R_i^l}(x) = \min_{k=1,2,\ldots,m}\{m_i^l(x_k)\}\,. \tag{9}$$

The min operator ensures a rule degree of membership equal to one for all input vectors that fully belong to the corresponding class hyperbox. Otherwise, for input vectors which are not completely in the class hyperbox or are in the intersection area $A_{i\,j}^l$ (if it exists), the rule degree of membership x is less than one. When the input vector partially belongs to two different classes, the corresponding rule degrees of membership $d_{R_i^l}(x)$ and $d_{R_j^l}(x)$ for both rules are different. The degrees of membership of the rules in this case depend also on the values of the sensitivity parameters γ_k.

Compared to other classification approaches, there is no need to compute a combined output of the rules. In the proposed classification application the rules do not cooperate since we obtain iteratively one cluster for each class. Hence there is no need for an aggregation function.

Finally, at the conceptual level, the algorithm generates two single consequent rules for $A_i^l \neg (A_i^l \cap A_j^l)$ and $A_j^l \neg (A_i^l \cap A_j^l)$, and a weighted double-consequent of type (4) for the intersection $(A_i^l \cap A_j^l)$. In practice, two rules are generated by each grouping iteration. For a testing example that belongs to the intersection, one of the two single-consequent rules of the type (6) will be executed, depending on the values of w_i and w_j that are calculated during the training process. As a double-consequent rule is equivalent to two single-consequent rules [26], by checking their weights the inference algorithm decides which rule to execute.

4. Application and performance evaluation

The performance of the algorithm is measured against three main criteria: the classification accuracy, the operational simplicity and the expressiveness of the generated model.

By accuracy we mean the classification performance not only to a constrained artificial environment but also to real world applications. That is mainly due to the fact that the alignment of the training patterns may be considerably different from the arrangement of the testing in real world data and hence yielding a considerable error. In addition to that, we should consider the intrinsic obscurity of medical data and the numerous external factors that influence the criticality of the disease, which cannot always be equally depicted in the available data or be consistently modeled. To achieve an effective measure and present the accuracy of our approach, a series of tests with various training/testing configurations have been prepared. That effectively means that the system should be able to successfully infer on real world cases that may deviate from the training patterns, in a consistent and meaningful way. As a result, the generalization ability and the interpretability of the generated model are strongly related to the overall system accuracy and thus should be mutually studied.

Naturally, the model simplicity is usually disproportionate to the accuracy since the simpler the model, the fewer the number of rules; and hence less accurate the description of the feature space which usually result in lower accuracy levels. The model simplicity, represented by the number of generated rules and model parameters, is yet another important characteristic since complicated, black-box models are difficult to handle and tune.

By expressiveness or interpretability we describe two main attributes. Firstly, we want to express the ability of the system to generalize on real world cases, thus being able to classify patterns that possibly lie outside the generated training boundaries. This is guaranteed by the selected membership functions that extend the crisp class boundaries with a fuzzy region of gradually decreasing membership allowing to meaningfully handle remote examples. Secondly, we express the interpretability of the generated knowledge both in terms of rule format, using rules that can be linguistically expressed and comprehended, and in terms of rule base structure, allowing the use of more detailed rules when data allows it. This is achieved by introducing the two repartitioning threshold values that iteratively group any intersection when the relevant data allows, which is controlled by the decision maker.

A vital aspect of the accuracy limitations of classification algorithms regards the efficiency and requirements of the training and testing process and the configuration of the equivalent datasets. In order to objectively and effectively measure the accuracy of the proposed method, two different training/testing configurations composed by a series of individual tests were developed for each dataset and for the two best sensitivity parameter values. These values were chosen after a series of independent experiments and are different for each dataset depending on the size of the hyperbox classes and their demand for expansion. In the first configuration, the training dataset is composed of a minimum set of selected boundary values and the testing by the rest of the data. This way, we can achieve maximum accuracy with minimum training requirements. The second configuration is composed by ten individual tests, five tests for each best value of γ_k, where the training and the testing dataset are randomly selected[1] and approximately equal in size. This configuration was chosen in order to achieve neutral training process and resemble random, real world circumstances where the testing data have no relation to the training ones, also demonstrated in [19,27].

Ideally, in a confined real world scenario, the training dataset should be carefully selected so as to include the data-points with the minimum and maximum values for all the dimensions and hierarchies. This is because these specific values will order the geometrical characteristics of the generated class boundaries that are also used for the rule description. In fact, a dataset containing only these polar values will be optimum for training purposes.

Finally, the training and testing requirements in terms of execution time provide a useful indication of the speed and simplicity of the method that are increasingly vital for processing large and multidimensional datasets and are consequently added in the results.

In all cases, training was performed with a single pass through the dataset, for two different sensitivity parameter values and with appropriate threshold values for R^l_{ij} and S^l_{ij} that vary according to the data under consideration. The sensitivity parameter of a fuzzy membership function controls the slope of the trapezoid sides and hence the expansion of its lower base and the generalization region around the cluster, while the re-grouping threshold parameter $Th1$—the number of grouping iterations and rules. The value of the parameter $Th2$ is fixed to 1/8, a relatively small value, which means that distance-based weights will be calculated only when the intersected class densities are almost equal. This way we might avoid the further computational load occurred by the calculation of multiple and multidimensional Euclidean distances.

4.1. Results presentation

In order to thoroughly test the functional characteristics of the algorithm, three different datasets with increasing dimensionality were considered. These are Wisconsin Breast Cancer, Pima Indian Diabetes and Bupa Liver data composed by nine, eight and six dimensions respectively, obtained from the UCI repository of machine learning databases [28]. The number of dimensions and pattern classes is highly important, since it considerably affects the complexity and computational efficiency of the classification process.

4.1.1. Wisconsin Breast Cancer Dataset

The Wisconsin Breast Cancer Dataset was compiled by Medical College of Wisconsin and has been widely used to test the functionality of many classification and rule extraction methods. It is composed by nine numerical attributes, describing nine different blood ingredients[2] and two different output classes, describing the nature of the cancer, either malign or benign. The original dataset is composed of 699 observations of which 16 were deliberately excluded due to incomplete descriptions of all nine dimensions. From the remaining 683 patterns, 444 belong to the benign class, 239 to the malign class while 252 belong to the intersection of the two classes. It is consequently a fairly dimensional dataset where the two classes coincide considerably.

Three iterations were performed during the training process to adequately describe the feature space, resulting in a rather dense partitioning and a set of six fuzzy rules. The number of iterations and rules can be manually controlled by setting suitable threshold values for R^l_{ij}, $Th1 = 1/10$ for both configurations of Wisconsin data.

Initially, we obtained the best values of the sensitivity parameter γ_k, by varying it in a wide range of values, and selecting the two best values of this parameter. Then, as it was mentioned before, we carried out two tests for configuration 1, one for each best value of γ_k, and 10 tests of configuration 2, five for each best γ_k. In the first arrangement 50 boundary observations were chosen for the training and the rest for testing. The first configuration was implemented to analyze the optimum performance of the algorithm in a confined environment, where the decision maker has full control and awareness of both the training and testing data. In the second experimental arrangement, the arbitrary choice of the training and testing data provides us with a good indication of how the system would operate in a real world situation where the range of the testing data is unexpected. This is another example where the generalization capabilities of the methodology are

[1] The training/testing datasets were compiled by consecutive selection of examples every 10, 20, 30, 40 and 50 observations. E.g. training data for Wisconsin: [1—10], [21—30], ..., [681—683], leaving the rest [11-20], [31—40], ..., [671—680] for testing.

[2] These are: Clump Thickness (UC), Uniformity of Cell Size (UC), Uniformity of Cell Shape (UC), Marginal Adhesion (MA), Single Epithelial Cell Size (SE), Bare Nuclei (BN), Bland Chromatin (BC), Normal Nucleoli (NN), Mitoses (Mit.) ranging from 1 to 10 and Class: (0 for benign, 1 for malignant).

Table 1 Experimental results and parameters of configuration 1 for breast cancer data.

Experiment	Sensitivity parameter	Number of iterations	Tr./test. data	Error/accuracy	Number of rules
1	$\gamma_k = 3$	1	50/633	103/633—83.7%	2
1	$\gamma_k = 3$	2	50/633	32/633—94.93%	4
1	$\gamma_k = 3$	3	50/633	12/633—98.2%	6
2	$\gamma_k = 4$	1	50/633	75/633—88.1%	2
2	$\gamma_k = 4$	2	50/633	39/633—93.8%	4
2	$\gamma_k = 4$	3	50/633	17/633—97.3%	6

Table 2 Experimental results and parameters of configuration 2 for breast cancer data.

Experiment	Sensitivity parameter	Number of iterations	Tr./test. data	Error/accuracy	Max number of rules
1	$\gamma_k = 3$	3	340/343	12/343—96.5%	6
2	$\gamma_k = 3$	3	340/343	13/343—96.2%	6
3	$\gamma_k = 3$	3	330/353	12/353—96.2%	6
4	$\gamma_k = 3$	3	320/363	14/363—96.2%	6
5	$\gamma_k = 3$	3	300/383	15/383—95.9%	6
6	$\gamma_k = 4$	3	340/343	13/343—96.2%	6
7	$\gamma_k = 4$	3	340/343	13/343—96.2%	6
8	$\gamma_k = 4$	3	330/353	14/353—96.1%	6
9	$\gamma_k = 4$	3	320/363	15/363—95.9%	6
10	$\gamma_k = 4$	3	300/383	17/383—95.4%	6

highly important and also considerably affect the classification accuracy.

The operational performance for the two best values of γ_k is shown in Table 1.

As we can clearly observe from Table 1, with broader generalization boundaries, $V^l_{jk} \pm (1/\gamma_k)$, we can get slightly improved accuracy when different testing data is considered. That obviously happens because the expansion of the fuzzy boundaries of the class improves its inclusion capabilities, which is crucial when the crisp margins are not optimal. As it was also expected, the accuracy level increases, following the rising number of groupings and rules that attempt to describe the conflicting intersection in a more descriptive manner. Ten further experiments have been performed for the second configuration, five for each value of γ_k, but with different and randomly selected datasets, as can be seen in Table 2.

It is obvious that the amount of fuzzy rules we finally obtain only depends on the number of classes and generated groupings, and consecutively on the number of the hierarchies produced by the re-clustering mechanism.

We can additionally compare the accuracy differentiation between the two configurations and assess the importance of the training process. The hyperboxes corresponding to the boundary values for each dimension that can be selected during the training process are very inclusive and hence produce optimum classification results. On the contrary, when the hyperboxes are generated randomly, they cannot embrace every possible example of that class and hence rely on the generalization area around the crisp hyperbox.

Four fuzzy rules were produced by the first two iterations, which are shown in Table 3.

In the case where a third iteration is triggered, another two additional and more specific rules are produced, which describe the same pattern space in a more finely granulated manner, see Table 4.

As it can be observed, we have named the membership functions in a consecutive manner from *low* to *high*. The intermediate membership functions, which refer to an intersection region, are identified by the 'med' label, combined with the label of the neighbouring functions, indicating the level of the hierarchy by equivalent linguistic components.

The schematic and numerical representation of the generated membership functions for all nine input features and for the first two hierarchies used in rule R1, R2, R3.1 and R3.2 can be seen in Fig. 6.

The membership functions for the nested hierarchies are generated in the same manner and within the intersection but for simplicity and legibility reasons their numerical and linguistic values have not been added.

Table 3 The four rules generated by the first two iterations for the breast cancer data.

Rule no.	CT	UC	UC	MA	SE	BN	BC	NN	Mit.	Class
R1	Low	Low	Low	Low	Low	Low	Low	Low	Low	Benign
R2	High	High	High	High	High	High	High	High	High	Malign
R3.1	Low—med	Low—med	Low—med	Low—med	Low—med	Low—med	Low—med	Low—med	Low—med	Benign
R3.2	Med—high	Med—high	Med—high	Med—high	Med—high	Med—high	Med—high	Med—high	Med—high	Malign

4.1.2. Pima Indian Diabetes Dataset

The particular data is composed of 768 patterns of nine numerical attributes and is a selected part of a larger data set held by the National Institutes of Diabetes and Digestive and Kidney Diseases, USA. The values of these medical attributes come from Pima-Indian women potential patients, 21 years old or more living near Phoenix, Arizona, USA. The class variable takes the values '0' or '1', indicating a negative and positive test for diabetes, respectively. Their other eight clinical features are:

1. Number of times pregnant
2. Plasma glucose concentration a 2 h in an oral glucose tolerance test
3. Diastolic blood pressure (mmHg)
4. Triceps skin fold thickness (mm)
5. Two hour serum insulin (μU/ml)
6. Body mass index
7. Diabetes pedigree function
8. Age (years)

As done before, two main training/testing configurations were implemented, composed of 2 and 10 tests respectively, to accommodate both values for parameters γ_k, and threshold values. In particular, the value of the sensitivity parameter γ_k had to be decreased to 1/3 and 1/2, to expand the fuzzy boundaries of the classes following equivalently increased crisp values of the boundaries. For $Th1 = 1/10$, three grouping iterations have been triggered, the results of which are shown in Table 5.

As was mentioned regarding the previous dataset, the accuracy inevitably rises when we consider additional groupings of intersections and the equivalent fuzzy rules. In addition, when the value of γ_k declines, more examples are correctly classified due to the expansion of the generalization area.

For the second random configuration, an additional fourth iteration was triggered to obtain sufficient classification accuracy by setting *Th*1 value equal to 1/15. Table 6 illustrates the functional characteristics of the second training—testing arrangement by using four grouping iterations.

Allowing for up to four grouping iterations, we managed to obtain an adequate set of hierarchical rules and obtain highly accurate and comparative results of 92.26% accuracy as an average of the 10 individual runs. On Table 7 we have appended the fuzzy values for each feature that define the membership to the equivalent fuzzy set and compose the classification rules.

The additional linguistic label of rule R3.3.3.1 and R3.3.3.2 corresponds to the additional grouping iteration. The values of the linguistic terms denote the range of numerical values with 'medium' refer-

Table 4 The two additional rules generated by the third iteration for the breast cancer data.

Rule no.	CP	UC	UC	MA	SECS	BN	BC	NN	Mit.	Class
R3.3.1	Low—med—med	Low—med—med	Low—med—med	Low—med—med	Low—med—med	Low—med—med	Low—med—med	Low—med—med	Low—med—med	Benign
R3.3.2	Med—med—high	Med—med—high	Med—med—high	Med—high—high	Med—med—high	Med—med—high	Med—med—high	Med—med—high	Med—med—high	Malign

ring to the intersection, and the number of terms represent the overlapping hierarchy.

4.1.3. Bupa Liver Dataset

Our third example, BUPA Liver Disorders Dataset was created by BUPA Medical Research Ltd. and donated to the Irvine collection by Richard Forsyth. Each record in the dataset is composed by readings supplied by a single male individual.

Composed of 345 instances, each of six numerical, continuous attributes:

1. MCV: mean corpuscular volume
2. AlkPh: alkaline phosphotase
3. AlAm: alamine aminotransferase
4. AsAm: aspartate aminotransferase
5. GamGT: gamma-glutamyl transpeptidase
6. Drno: number of half-pint alcoholic beverages or equivalent units drunk per day.

An additional attribute is normally used to specify the class that equivalently describes the criticality or existence of the liver malfunction. The first five attributes are results from blood tests and are thought to be factors sensitive to liver disorders influenced by excessive alcohol usage, while the sixth is an indication of daily alcohol consumption.

In that particular dataset, the overlapping is rather extended and the setting $Th1 = 1/6$ resulted to four clustering iterations that adequately grouped our training data. That equivalently results to four rule hierarchies and twelve fuzzy classification rules that are configured in a nested structure. As in the previous example, the value of the sensitivity parameter γ_k was set to 1/3 and 1/2 as the numerical values of diabetes and liver data are of the same range.

In the first training and testing configuration, the testing data set was composed of 300 patterns,

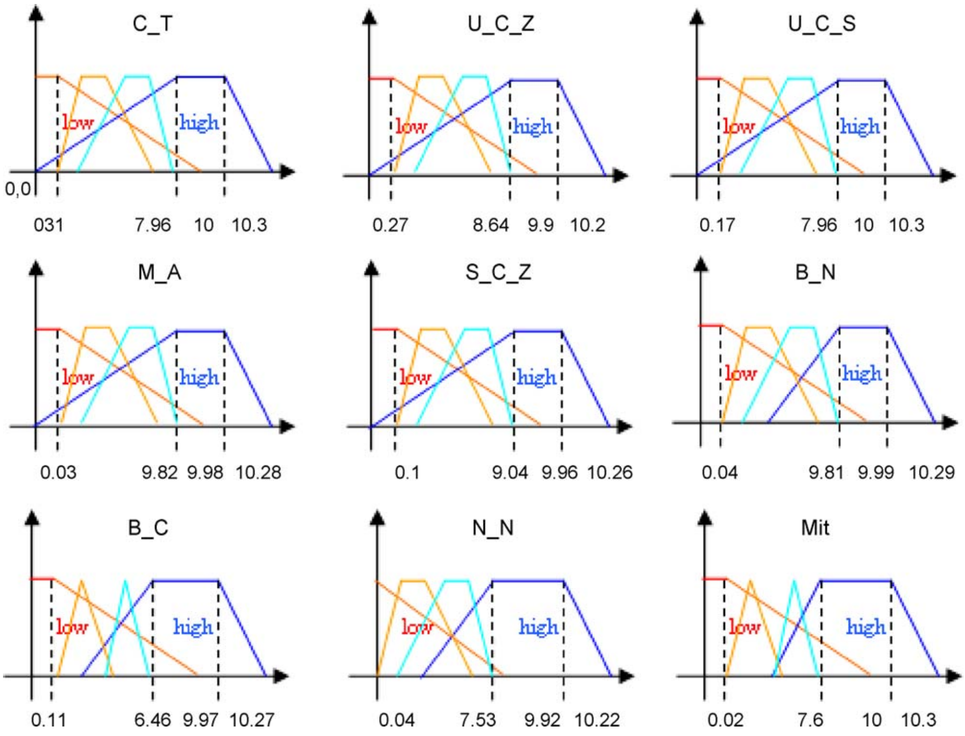

Figure 6 The generated membership functions for nine features and two hierarchies of the first configuration for the Wisconsin data.

Table 5 Experimental results and parameters of configuration 1 for the diabetes data.

Experiment	Sensitivity parameter	Number of iterations	Tr./test. data	Error/accuracy	Number of rules
1	$\gamma_k = 1/3$	1	68/700	139/700—81.2%	2
1	$\gamma_k = 1/3$	2	68/700	79/700—89.7%	4
1	$\gamma_k = 1/3$	3	68/700	45/700—93.6%	6
2	$\gamma_k = 1/2$	1	68/700	143/700—80.5%	2
2	$\gamma_k = 1/2$	2	68/700	82/700—89.3%	4
2	$\gamma_k = 1/2$	3	68/700	47/700—93.2%	6

Table 6 Experimental results and parameters of configuration 2 for the diabetes data.

Experiment	Sensitivity parameter	Number of iterations	Tr./test. data	Error/accuracy	Max number of rules
1	$\gamma_k = 1/3$	4	380/388	31/388—92.1%	8
2	$\gamma_k = 1/3$	4	380/388	28/388—92.8%	8
3	$\gamma_k = 1/3$	4	378/390	30/390—92.1%	8
4	$\gamma_k = 1/3$	4	360/408	29/408—92.9%	8
5	$\gamma_k = 1/3$	4	350/418	31/418—92.6%	8
6	$\gamma_k = 1/2$	4	380/388	35/388—91.08%	8
7	$\gamma_k = 1/2$	4	380/388	29/388—92.5%	8
8	$\gamma_k = 1/2$	4	378/390	31/390—92.1%	8
9	$\gamma_k = 1/2$	4	360/408	31/408—92.5%	8
10	$\gamma_k = 1/2$	4	350/418	34/418—91.9%	8

while the training by 45 selected examples that contained the boundary values for each class and hence providing an optimal separation arrangement. This is depicted in Table 8.

In the second configuration, we carried out 10 separate tests with five different randomly selected training/testing data and for both selected values of parameter γ_k, the results of which can be viewed in Table 9. As expected, the arbitrary configuration achieved lower classification accuracy of 89.9%, as an average of the ten tests, for the same level of grouping due to the randomly selected class boundaries.

Table 10 shows the eight fuzzy rules produced by all four grouping iterations. As expected, the rules have the same, meaningful structure as in the pre-

Table 7 The eight rules generated by all four iterations for the diabetes data.

Rule no./feature	NP	PGC	DBP	TSFT	2HSI	BMI	DPF	Age	Class
R1	High	High	Low	High	High	High	High	Low	1
R2	Low	Low	High	Low	Low	Low	Low	High	0
R3.1	Med—high	Med—high	Med—high	Med—low	Med—high	Med—high	Med—high	Med—low	1
R3.2	Med—low	Med—low	Med—low	Med—high	Med—low	Med—low	Med—low	Med—high	0
R3.3.1	Med—med—high	Med—med—low	Med—med—high	Med—med—low	Med—med—high	Med—med—high	Med—med—high	Med—med—low	1
R3.3.2	Med—med—low	Med—med—high	Med—med—low	Med—med—high	Med—med—low	Med—med—low	Med—med—low	Med—med—high	0
R3.3.3.1	Med—med—med—high	Med—med—med—high	Med—med—med—high	Med—med—med—high	Med—med—med—high	Med—med—med—high	Med—med—med—high	Med—med—med—high	1
R3.3.3.2	Med—med—med—low	Med—med—med—low	Med—med—med—low	Med—med—med—low	Med—med—med—low	Med—med—med—low	Med—med—med—low	Med—med—med—low	0

Table 8 Experimental results and parameters of configuration 1 for the liver data.

Experiment	Sensitivity parameter	Number of iterations	Tr./test. data	Error/accuracy	Number of rules
1	$\gamma_k = 1/3$	1	45/300	71/300—76.4%	3
1	$\gamma_k = 1/3$	2	45/300	59/300—80.4%	6
1	$\gamma_k = 1/3$	3	45/300	51/300—83%	9
1	$\gamma_k = 1/3$	4	45/300	22/300—92.7%	12
2	$\gamma_k = 1/2$	1	45/300	76/300—75.7%	3
2	$\gamma_k = 1/2$	2	45/300	61/300—79.7%	6
2	$\gamma_k = 1/2$	3	45/300	52/300—82.7%	9
2	$\gamma_k = 1/2$	4	45/300	29/300—90.4%	12

Table 9 Experimental results and parameters of configuration 2 for the liver data.

Experiment	Sensitivity parameter	Number of iterations	Tr./test. data	Error/accuracy	Max number of rules
1	$\gamma_k = 1/3$	4	170/175	17/175—90.3%	12
2	$\gamma_k = 1/3$	4	160/185	18/185—90.3%	12
3	$\gamma_k = 1/3$	4	150/195	20/195—89.8%	12
4	$\gamma_k = 1/3$	4	160/185	18/185—90.3%	12
5	$\gamma_k = 1/3$	4	150/195	19/195—90.3%	12
6	$\gamma_k = 1/2$	4	170/175	18/175—89.7%	12
7	$\gamma_k = 1/2$	4	160/185	19/185—89.7%	12
8	$\gamma_k = 1/2$	4	150/195	21/195—89.3%	12
9	$\gamma_k = 1/2$	4	160/185	18/185—90.3%	12
10	$\gamma_k = 1/2$	4	150/195	20/195—89.7%	12

vious case, which allows differentiating the rules from each other according to the hierarchy they belong to.

4.2. Comparative analysis

The proposed algorithm creates membership functions for all fuzzy features, in contrast with some feature elimination approaches [20], and a variable number of rules for every dataset. Its functionality for the Wisconsin data is compared against a number of eminent classification methods, ranging from neuro-fuzzy approaches, Nauck and Kruse [17] and Wang and Lee [27], to evolutionary, Chang and Lilly [20], and alternative methodologies, like that of Abonyi and Szeifert [29], based on decision trees initialization. The comparison results are shown in Table 11.

Regarding the results obtained from the second configuration; the proposed method is outperformed by some other methods with respect to its accuracy. The accuracy, however, can be increased after careful selection of the training data. The main advantage of the proposed methodology is the simplicity of its training process and the comprehensibility of the extracted rules. The average speed of the training, 114 ms, and testing process, 1343 ms, indicate another positive operational characteristic that cannot be used for comparison since similar figures are rare in literature. All the tests were performed on a Pentium 3 machine with 512 MB RAM.

The proposed method is compared against some other popular and diverse methods using the diabetes data—see Table 12.

From this table we can observe that the proposed method has the best accuracy, compared to other methods. The additional iteration that was triggered for the second experimental configuration resulted in an additional computational load for the training process of 146 ms, while not affecting the testing, that lasted approximately 1360 ms.

Finally, the processing of the liver lasted 1269 ms, 97 ms for the training and 1172 ms for the testing process. Some additional comparative results and characteristics can be observed in Table 13.

We can easily observe that the obtained classification results are fairly comparable to the rest of the modern classification methods without any need for parameter tuning or initialization of a membership function, or the use of an additional fuzzy clustering component. Moreover, the proposed classification algorithm attempts to achieve fairly good classification accuracy without any optimization component. This can be explained as a result of the iterative minimization of the overlying area that is normally responsible for the classification conflict. With a very fast, non-order-dependent learning process, it only requires three parameters and can rapidly generate a reasonable set of rules, the

Table 10 The eight rules generated by all four grouping iterations for the liver data.

Rule no./feature	MCV	AlkPh	AlAm	AsAm	Gam GT	DrNo	Class
R1	High	Low	Low	Low	Low	High	Normal
R2	Low	High	High	High	High	Low	Critical
R3.1	Med—high	Med—high	Low—med	Low—med	Low—med	Low—med	Normal
R3.2	Low—med	Low—med	Med—high	Med—high	Med—high	Med—high	Critical
R3.3.1	Med—med—high	Med—med—high	Med—med—high	Med—med—high	Med—med—high	Med—med—high	Normal
R3.3.2	Low—med—med	Low—med—med	Low—med—med—	Low—low—med	Low—low—med	Low—med—med	Critical
R3.3.3.1	Med—med—med—high	Med—med—med—high	Med—med—med—low	Med—med—med—low	Med—med—med—high	Med—med—med—low	Normal
R3.3.3.2	Med—med—med—low	Med—med—med—low	Med—med—med—high	Med—med—med—high	Med—med—med—low	Med—med—med—high	Critical

Table 11 The comparative results for the breast cancer data.

Technique	Accuracy	Tr./test. dataset size	No. of rules/ features	Comments
SANFIS, Wang and Lee [27]	96.3—97.47%	342/341 patterns	2 rules/ 9 features	Improved results after optimization
VISIT, Chang and Lilly [20]	96.5%	400/283 patterns	3 rules/ 2 features	Needs initialization of memb. functions, 100 learning iterations needed
NEFCLASS, Nauck and Kruse [17]	95.06%	All observations/ all observations	4 rules/ 9 features	Long training, more than 10 conditions, and rule pruning
Abonyi and Szeifert [29]	95.57%	342/341 ids	2 rules/ 9 features	Needs parameter initialization and three to four conditions
Proposed method	96.08% (avg. conf. 2)	(300—340)/ (343—383) patterns	6 rules/ 9 features	Fast training, 114 ms, no initialized knowledge, thee parameters

Table 12 Comparative results for the diabetes data.

Technique	Accuracy	Tr./test. dataset size	Number of rules/ features	Comments
ART-MAP [30]	66—81%	576/212 patterns	N.A./8 features	Order-dependent learning, plurality voting and vigilance param.
PROAFTN [31]	71.3—76%	N.A.	N.A./8 features	Params. Initialization, weights and discrimination thresholds
BZ/IZ-Value Meas. [32]	75.03—85.3%	N.A.	N.A./8 features	GA, incremental learning, 17 params. discriminant functions
GF-SVM [33]	70—76%	Five different shuffles, popul. size = 100, 30 generations	N.A./8 features	G.A. feature transformation, kernel function optimization
Proposed method	92.26%	(350—380)/ (388—418)	8 rules/8 features	Fast, no initialized knowledge, three param.

Table 13 Comparative results for the liver data.

Technique	Accuracy	Tr./test. dataset size	Number rules/ features	Comments
FBP-NN [34]	79.5%	N.A.	N.A./8 features	GA net. Param., fuzzy back propagation, 10 params.
BZ/IZ-Value Meas. [32]	69.5—84.06%	N.A.	N.A./8 features	GA, incremental learning, 17 params. discriminant functions
GF-SVM [33]	61—76%	Five different shuffles, popul. size = 100, 30 generations	N.A./8 features	G.A. feature transform., kernel function optimization
NF-BSP [35]	79.7%	122/123 randomly selected	55—98 rules/ 8 features	Gradient descent method, two params.
Proposed method	89.9%	(150—170)/(175—195) randomly selected	8 rules/8 features	Fast, no initialized knowledge, three param.

number and format of which can be implicitly controlled by the user.

The linguistic expressiveness and flexibility of the proposed approach composes its main advantage over other methods that may use a 'black box', neural topology to formulate the rules in terms of incomprehensible weights or require complicated optimization components to refine their knowledge. As a result, it achieves a meaningful, simple, transparent and fast learning process that is necessary for a critical medical decision.

5. Conclusions

The current paper presents a detailed methodology for the generation of fuzzy classifiers that can learn from numerical labeled medical data in a meaningful, iterative and consistent manner. The power of the proposed method lies on its simple partitioning process that does not need any initial knowledge. The method does not use any optimization component or any other time consuming learning strategy, such as back propagation and expansion/contraction process, but produces relatively accurate classification results. In addition, the hierarchical structure of the generated rule base greatly enhances the interpretability and flexibility of the model.

The flexibility of the iterative fuzzy grouping methodology makes it ideal for the categorization of critical medical cases where real world accuracy is elusive and the interpretation and integration of medical knowledge is necessary. Offering a meaningful input partitioning, the proposed method can be also regarded as a consistent, data driven accuracy-interpretability framework that enables the decision maker to choose between a small set of general rules and a larger set of more accurate rules, for the same area. That particular characteristic would be essential for the diagnosis and beneficial an experienced medical decision maker.

In addition, due to the very simple learning process that avoids the lengthy connectionist weight learning and the blind search genetic programming methodology, our approach has a considerably shorter learning phase. That learning period is normally, strongly dependent on the number of additional grouping iterations but rarely lasts more than a second when processing 300—400 training examples. Finally, the learning process always converges even when the overlapping area is equally populated, due to the alternative distance-based criterion, increasing the robustness of the system.

Trying to always resolve the classification conflict by generating numerous partitions or developing complicated cluster boundaries, it could not always guarantee improved classification performance [36]. This is simply because in a real world scenario, the distribution of the testing data is usually slightly different from the testing ones. Additionally, developing complicated boundaries for achieving higher accuracy may have a considerable effect on the computational complexity and generalization power of the approach. In the current document, we focus on adequately handling the intersections and possibly producing denser partitioning when the data and the decision maker indicate so. In the real world, datasets usually overlap to some extent and that is what the proposed method attempts to illustrate and handle in a consistent and meaningful manner.

References

[1] Akay M, Cohen M, Hudson D. Fuzzy sets in life science. Fuzzy Sets Syst 1997;90:219—24.

[2] Steimann F. On the use and usefulness of fuzzy sets in medical AI. Artif Intell Med 2001;21:131—7.

[3] Patrick E, Stemock F, Shen L. Review of pattern recognition in medical diagnosis and consulting relative to a new system model. IEEE Trans Syst Man Cybernet 1974;1:1—16.

[4] Leung KS, Felix Wong WS, Lam W. Applications of a novel fuzzy expert system shell. Expert Syst 1989;6(1):2—10.

[5] Liao SH. Expert systems methodologies and applications—a decade review form 1995 to 2004. Expert Syst Appl 2005;28:93—103.

[6] Shortliffe EH. Computer-based medical consultations, MYCIN. New York: Elsevier/North-Holland; 1976.

[7] Phuong NH, Kreinovich V. Fuzzy logic and its application in medicine. Int J Med Inform 2001;62:165—73.

[8] Zadeh LA. Fuzzy sets. Inform Control 1965;8:338—53.

[9] Belacel N, Boulassel MR. Multicriteria fuzzy assignment method: a useful tool to assist medical diagnosis. Artif Intell Med 2001;21:201—7.

[10] Chen SM. A weighted fuzzy reasoning algorithm for medical diagnosis. Decis Support Syst 1994;11:37—43.

[11] Yager R, Filev D. Approximate clustering via mountain method. IEEE Trans Syst Man Cybernet 1994;24:1279—84.

[12] Wang L-X, Mendel J. Generating fuzzy rules by learning from examples. IEEE Trans Syst Man Cybernet 1992;6:1414—27.

[13] Ruspini E. A new approach to clustering. Inform Control 1996;15(1969):22—3.

[14] Bezdek JC, Erilch R, Full WE. FCM: the fuzzy c-means clustering algorithm. Comput Geosci 1984;10:191—203.

[15] Setnes M, Roubos H. GA-fuzzy modelling and classification: complexity and performance. IEEE Trans Fuzzy Syst 2000;8:509—22.

[16] Pena-Reyes CA, Sipper M. A fuzzy genetic approach to breast cancer diagnosis. Artif Intell Med 1999;17:131—55.

[17] Nauck D, Kruse R. Obtaining interpretable fuzzy classification rules from medical data. Artif Intell Med 2003;16:149—69.

[18] Pal SK, Mitra S. Multilayer perceptron, fuzzy sets, and classification. IEEE Trans Neural Netw 1992;3:683—97.

[19] Simpson PK. Fuzzy min—max neural network. Part 1. Classification. IEEE Trans Neural Netw 1992;3:776—86.

[20] Chang X, Lilly JH. Evolutionary design of fuzzy classifier from data. IEEE Trans Syst Man Cybernet 2004;34:1894—906.

[21] Cordon O, Herrera F. A proposal for improving the accuracy of linguistic modelling. IEEE Trans Fuzzy Syst 2000;3: 335—44.

[22] Casillas J, Cordon O, Herrera F. COR: a methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. IEEE Trans Syst Man Cybernet 2002;4:526—37.

[23] Selim SZ, Alsultan K. A simulated annealing algorithm for the clustering problem. Pattern Recogn Lett 1991;24:1003—8.

[24] Ishibuchi H, Nozaki K, Tanaka H. Distributed representation of fuzzy rules and its application to pattern classification. Fuzzy Sets Syst 1992;52:21—32.

[25] Nozaki K, Ishibuchi H, Tanaka H. A simple but powerful heuristic method for generating fuzzy rules from numerical data. Fuzzy Sets Syst 1997;86:251—70.

[26] Alcalá R, Casillas J, Cordón O, Herrera F. Linguistic modeling with weighted double-consequent fuzzy rules based on cooperative coevolutionary learning. Integr Comput-Aided Eng 2003;4:343—55.

[27] Wang JS, Lee G. Self-adaptive neuro-fuzzy inference system for classification applications. IEEE Trans Fuzzy Syst 2002;6: 790—802.

[28] UCI Machine Learning Repository, http://www.archive.ics.uci.edu/ml/datasets.html (last accessed: 15. 25 March 2009).

[29] Abonyi J, Szeifert F. Supervised fuzzy clustering for the identification of fuzzy classifiers. Pattern Recogn Lett 2003;24:2195—207.

[30] Loo CK, Rao MVC. Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy ARTMAP. IEEE Trans Knowl Data Eng 2005;11:1589—93.

[31] Belacela N, Ravala HB, Punnenc AP. Learning multicriteria fuzzy classification method PROAFTN from data. Comput Oper Res 2007;34:1885—98.

[32] Chiena BC, Linb JY, Yang WP. Learning effective classifiers with Z-value measure based on genetic programming. Pattern Recogn 2004;37:1957—72.

[33] Jin B, Tang YC, Zhang YQ. Support vector machines with genetic fuzzy feature transformation for biomedical data classification. Inform Sci 2007;177:476—89.

[34] Choi B, Bluff K. Genetic optimisation of control parameters of a neural network. In: Proceedings of second international conference on artificial neural networks and expert systems. Dunedin, New Zealand: IEEE Press; 1995. p. 174—7.

[35] Gonçalves LB, Vellasco MMBR, Pacheco MAC, de Souza FJ. Inverted hierarchical neuro-fuzzy BSP system: a novel neuro-fuzzy model for pattern classification and rule extraction in databases. IEEE Trans Syst Man Cybernet 2006;36: 236—48.

[36] Gadaras I, Mikhailov L, Lekkas S.In: Proceedings of FUZZ-IEEE, international conference on fuzzy systems. London, UK: IEEE Press; 2007. p. 1—6.