Knowledge-Based Systems 124 (2017) 105–132

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Designing rule-based fuzzy systems for classification in medicine

Marco Pota*, Massimo Esposito, Giuseppe De Pietro

National Research Council of Italy – Institute for High Performance Computing and Networking (ICAR), Via P. Castellino 111, 80131 Naples, Italy

ARTICLE INFO

Article history:
Received 13 July 2016
Revised 3 February 2017
Accepted 4 March 2017
Available online 6 March 2017

Keywords:
Classification
Learning
Fuzzy models
Medicine
Rule-based classifiers

ABSTRACT

Decision Support Systems (DSSs) based on Fuzzy Logic are gaining increasing research interest in order to solve classification problems in a wide range of application fields, especially in medicine, where the chance of presenting classification results together with a clear explanation and with a measure of the associated uncertainty is highly appealing. However, designing a fuzzy system is a thorny process, requiring many steps to be accomplished, from the knowledge extraction and representation, to the inference process, until the presentation of results. Therefore, this paper proposes a general procedure for constructing rule-based fuzzy classifiers, according to the system characteristics of performance and interpretability required by the specific application, which can be used with any type of data, and is particularly useful for the medical field requirements. The proposed procedure is based on the naïve Bayes approximation, therefore, the optimization of necessary parameters is performed only once and separately for each variable, thus resulting computationally fast, while later steps of the procedure enable to calculate more complicate models and choose the best one, without any further optimization. Moreover, the choices of all degrees of freedom of the design, associated with the variables constituting the model, their fuzzy partitions, the rule base construction, and the inference process, are suggested in this paper. Some of them are motivated by general considerations regarding systems applied in the medical ambit. Some other design choices depend on the dataset and on the application. In order to provide an objective way for choosing these degrees of freedom, some parameters for defining the required trade-off between performance and interpretability are proposed here. The application of the proposed procedure is guided by showing a running example, using data of the Wisconsin Breast Cancer Dataset. For different values of the trade-off parameters, optimal interpretability, or first-rate performance, or acceptable interpretability and performance are obtained, with respect to the best fuzzy systems applied on the same dataset. Finally, the procedure is applied on a number of benchmark datasets, and outstanding results are achieved in terms of performance, with respect to the best classification methods of the state-of-the-art.

1. Introduction

Classification problems in many fields of application can be tackled by artificial intelligence, for the construction of automatic classification systems. Recent research has focused on new learning approaches, e.g., deep neural architectures [1], showing improvements of results in terms of performance, also for medical field applications. On the other hand, in a field like medicine, or other fields with the same characteristics, the system results are often required together with a clear explanation and a measure of uncertainty associated with them. With this respect, the role of rule-based systems employing fuzzy logic [2] is of particular importance, because of its ability of presenting the model in a transparent and even interpretable manner. Moreover, the use of fuzzy logic naturally offers the possibility of modelling uncertainty associated both with input measurement and with the relationship between input features and output class. In this paper, data-driven construction of fuzzy models for classification problems in the medical ambit is regarded, aiming at extracting knowledge from supervised data pertaining known samples, for classifying further unknown samples.

In order to design a fuzzy classifier, all the different aspects of the system should be adjusted by the developer, which are the set of influencing *variables*, the *fuzzy partitions* of each variable, the *rule base* connecting fuzzy sets to different classes, and the *inference* process, by means of which each data sample is associated to one or more classes. Therefore, the system design requires a number of settings, which are called here *degrees of freedom*.

Depending on the application, the system settings should follow specific requirements for a correct system design, e.g., if *interpretability* is desired, then settings implying system simplification should be preferred, while if the best possible *performance* is re-

* Corresponding author.
E-mail address: marco.pota@na.icar.cnr.it (M. Pota).

http://dx.doi.org/10.1016/j.knosys.2017.03.006
0950-7051/

quired, then many complications can be added to the system. Performance and interpretability mainly follow opposite trends; therefore, the proper trade-off should be decided to guide the design process. Moreover, on the one hand, different measures could be considered to evaluate the performance, since while some applications need a *simple classifier* (SC), which presents a single class as a result, others need a *confidence-weighted classifier* (CWC), which presents, as a result, a set of classes with respective confidences. On the other hand, interpretability can be regarded by different viewpoints, which are *complexity* and *semantic interpretability* [3,4], and each of these characteristics can be more or less important for the final application. Particular requirements are usually desired in the medical field; some of them can be considered valid for any application in this field, while others are expected to be defined by the medical staff that represents the final users of the system.

The aim of this work is to individuate a procedure for designing fuzzy classification systems for medical applications. This procedure should take into account all the main aspects of a fuzzy classifier, and should be general enough to contemplate different possible requirements desired by the users, within the ambit of medical applications.

Many works exist on fuzzy classifiers, and, in particular, different types of systems were proposed for medical applications. However, many authors developed systems by imposing some settings without motivating their design choices, while only some attempts were done on evaluating the specificities of each choice. For example, at the best of our knowledge, the possibility of using different shapes of fuzzy sets, often triangular or Gaussian, and different types of T-norm and S-norm used for inference, has been studied only few times with rational motivations [5–8]. Therefore, some structural decisions to be made during the design of fuzzy classifiers have not been extensively considered yet, nor addressed in particular to medical applications.

In some previous works, fuzzy systems were developed mainly for specific applications. Many authors developed fuzzy systems by proposing some improvements regarding particular steps of the knowledge extraction process, particularly devoted to performance maximization, through the optimization of fuzzy partitions parameters and rule base, and fixing the other settings. In some works, settings regarding different aspects of the system were taken into account; among them, some [3,8,9] are worth to be mentioned. In [8], authors attempted to optimize various degrees of freedom; however, they only considered the SC performance as final goal. Other authors [3,9] focused on the definition of interpretability of fuzzy systems. However, either performance of the system or its interpretability have been considered in previous works, while very few attempts have been done to take into account both perspectives at the same time.

Starting from the last decade, few works were based on both performance and interpretability. In [10], a genetic multi-objective approach was proposed, but only the number of rules was accounted for interpretability. In [11], a similar approach was used to generate fuzzy rule-based systems with different trade-offs between accuracy and interpretability in regression problems; however, interpretability was measured by only using the number and length of rules. Multi-objective learning was also used in [12], where adaptive defuzzification induce weighted rules, but interpretability is still intended only from the complexity point of view, and by considering only the number of rules, the number of rules with weights associated and the average number of rules triggered by each example. The FisPro implementation [13] was proposed, which allows designing fuzzy systems, aiming at both performance and interpretability. However, it only maximizes performance of SCs, and it only includes some constraints for interpretability. Moreover, it considers different degrees of freedom, like methods for optimizing the fuzzy partitions (even if they are chosen only among unsupervised methods, therefore they do not provide the best partitions for classification or regression purposes), different rules learning and simplification schemes (even if they do not consider some characteristics of rules interpretability), different norms for input combination and for aggregation (even if they are limited to min and product T-norms and max and sum S-norms), and different defuzzification operators (even if only the winner-takes-all strategy is proposed for classification problems). However, many other degrees of freedom were not considered, like different shapes of fuzzy sets (which are possible to draw by hand in the proposed system, but optimization is contemplated only for trapezoidal fuzzy sets), or the possibility of using weights. In [14], partition integrity is considered as an objective together with complexity (and accuracy); however, the integrity maximization only forces partitions towards uniform granulation, while complexity is measured only by the rules number. [15] focused on fuzzy classification rules, to obtain systems with a predefined level of compromise between its performance and interpretability; however, while in this case a squared error was considered for performance (which is more suitable for CWCs), again interpretability is measured only by considering rules number and length. The same reductive set of characteristics is taken into account also in [16]. Finally, the most recent work [17], improving [15], presents a multi-objective genetic approach to design interpretability-oriented fuzzy rule-based classifiers; however, it uses an interpretability measure based on the average length of rules, the number of fuzzy sets and the number of inputs used in the rule base, which are still not satisfying, as better explained later.

Different degrees of freedom were already analysed by authors in preliminary works [18,19], by pursuing mainly performance [18], and by refining the definition of some properties ensuring interpretability [19].

This paper proposes a novel procedure, aimed at designing fuzzy systems for classifying data, by taking into account all the most important degrees of freedom at the same time. In particular, medical data are taken into account, which in principle are not different from those regarding other domains. However, the models constructed for medical applications should have particular characteristics, taken into account in the proposed procedure for fuzzy system design, as well as good classification performance: (i) interpretability, in order to explicitly encode knowledge about relations between features and classes, which can thus be validated by domain experts and can aid in associating a clear explanation with the system results; (ii) produce classification results together with an accurate measure of their confidence, in order to treat each patient by also considering the uncertainty regarding his/her real class, which is particularly important in medical domain.

Therefore, in this paper it is shown how some settings can be directly identified for general purposes of the medical domain, while other settings can be chosen based on the desired requirements of specific applications. All the design choices are discussed by taking into account different possible outcomes, among performance measures and interpretability properties, at the same time. Moreover, some parameters are proposed, which can be used to individuate the trade-off between performance and interpretability.

In more detail, all the aspects of design are considered here, which are the choice of variables of interest, the fuzzy partitioning of continuous ones, the rule base construction, and the choice of inference settings. The corresponding choices of the developer for the system design are summarized in Table 1. The viewpoints used to evaluate different degrees of freedom are also reported in Table 1. The following outcomes are mainly evaluated: classification performance, system complexity, and semantic interpretability.

The additions of complications are discussed in order to decide whether performance improvements are great enough to justify

Table 1
Degrees of freedom.

Main aspect	Degree of freedom	System evaluation
Variables	Variables number	Performance, complexity
	Variables type	Performance, semantic interpretability
	Variables selection	Performance
Fuzzy sets	Fuzzy sets number	Performance, complexity
	Fuzzy sets shape	Performance, complexity, semantic interpretability, differentiability
	Fuzzy sets position	Performance
Rules	Antecedents weights	Performance, semantic interpretability
	Antecedents number	Performance, complexity
	Consequents type	Performance, semantic interpretability
	Rule weights	Performance, semantic interpretability
	Rules number	Performance, complexity
	Rule base completeness	Performance, complexity, semantic interpretability
	Rules reduction/selection	Performance, complexity, semantic interpretability
Inference	Norms type	Performance, semantic interpretability, differentiability
	Defuzzification type	Performance, differentiability

the associated increase of complexity and/or loss of semantic interpretability, depending on the priority of the application requirements. This priority is encoded by the proposed trade-off parameters, combining performance and interpretability requirements. As a consequence, it is shown how all the degrees of freedom can be found automatically, following the reported considerations and the fixed trade-off parameters.

The proposed procedure is based on the naïve Bayes approximation, where interactions are assumed to be negligible, therefore, the optimization of necessary parameters is performed separately for each variable, and the procedure results scalable, requiring a number of parameters linear in the number of variables, thus simple and performing.

A well-known dataset, i.e., the Wisconsin Breast Cancer Dataset (WBCD) [20], is used as a proof of concept, for showing the application of the proposed procedure, and the usefulness of obtained models for classification in medicine. Results obtained by fixing different values of trade-off parameters are compared among them and with different existing fuzzy approaches, previously applied on the same dataset, in terms of performance and interpretability. Moreover, a number of benchmark datasets regarding classification problems in medicine are used, in order to compare the performance that can be obtained by applying the proposed procedure with the performance of some of the best existing classification methods.

This paper is organized as follows. In Section 2, a kind of tutorial regarding fuzzy systems for classification and relative degrees of freedom is presented, while the criteria for evaluating a classification system in medicine are briefly reviewed in Section 3. In Section 4, the proposed approach for designing a fuzzy system for classification in medicine is explained, and in Section 5 its application is shown, and results on different datasets are compared with the best of the state-of-the-art. Finally, Section 6 concludes the work.

2. Fuzzy systems for classification

A classifier allows extracting knowledge from a training dataset made of N data samples described by vectors $\mathbf{z}_i = \{\mathbf{x}_i, y_i\}$, with $i = 1, \ldots, N$, where $\mathbf{x}_i = \{x_i^{(1)}, \ldots, x_i^{(n)}\}$ are the values of the input variables $X^{(j)}$, $j = 1, \ldots, n$, defined in respective Universes Of Discourse, $x_i^{(j)} \in UOD^{(j)}$, n is the number of input variables used by the model, and y_i are the values of the output variable Y, defined on a set of classes, $y_i \in \{c_1, \ldots, c_K\}$. Once constructed, it should allow classifying an incoming data sample $\mathbf{x} = \{x^{(1)}, \ldots, x^{(n)}\}$. A further set of data can be available for testing, or training and test sets can be obtained from the available dataset. The classification can be done differently, depending on the required type of output, by a simple classifier (SC), which assigns to the data sample one of the classes, or by a confidence-weighted classifier (CWC), which gives as a result a set of classes with respective confidences or probabilities. Without losing generality, the output can be represented as a fuzzy set depending on $\mathbf{x}$:

$$\hat{y}(\mathbf{x}) = \frac{y_1(\mathbf{x})}{c_1} + \cdots + \frac{y_K(\mathbf{x})}{c_K}, \tag{1}$$

where $\frac{y_k(\mathbf{x})}{c_k}$ (with $k = 1, \ldots, K$) defines an element c_k of the fuzzy set $\hat{y}(\mathbf{x})$ having membership grade $y_k(\mathbf{x})$, and $+$ stands for union of elements. In case of SC, one y_k is 1 and all the others are 0, while in case of CWC, different non-zero y_k can be presented as a result.

Many degrees of freedom of a fuzzy system have to be settled by the developer. Some of them regard the knowledge representation: if it is configured, as data-driven, knowledge has to be extracted starting from the training dataset and this consists in different steps, namely variables choice, fuzzy partitioning, rules extraction, and, eventually, optimization of some pieces of knowledge. Other aspects regard the inference process. In the following, all the main degrees of freedom are pointed out.

2.1. Variables

In order to build a fuzzy system, the most influencing variables have to be chosen. This implies the need to set the number of variables, to select some of them, and to choose using original or derived variables. In particular, the choice of appropriate variables enables: (i) to find the simplest possible model, (ii) to avoid noise or collinearity coming from unnecessary variables, (iii) to avoid measuring unnecessary features.

One of the most important degrees of freedom of a fuzzy classifier is the number n of variables to use for building the model. The developer is expected to directly fix n, or indirectly some other number related to it. Alternatively, a maximum number of variables $n_{\max}$ could be settled. This choice influences performance and complexity, as better explained later.

Each Universe Of Discourse $UOD^{(j)}$ can be of different nature, and here two types are distinguished: continuous (i.e., samples are represented by pseudo-continuous numbers, or by meaningfully ordered discrete numbers with sufficiently numerous different elements), or categorical (i.e., samples are represented by symbols or by meaningless numbers, or by only 2–3 different numbers). Since this paper is dedicated to fuzzy systems, continuous input variables are mainly considered in the following. However, let us assume that n_{cont} variables $X^{(j)}$, $j = 1, \ldots, n_{cont}$, included in the model are continuous, while $n_{cat} = n - n_{cont}$ variables $X^{(j)}$, $j = n_{cont} + 1, \ldots, n$, are categorical. The latter can be included in a fuzzy model in a straightforward way, since they play the same

role of continuous variables, even if they cannot be fuzzified, as better explained later.

Original variables can be used to build a model. Alternatively, variables derived from them (like linear combinations or non-linear terms) can be preferred. Therefore, the developer is expected to make a qualitative choice, which influences performance and semantic interpretability, as better explained later.

Once the number and type of variables have been chosen, the selection of the best n variables should be done. The developer is expected to choose a method for variables selection, in order to obtain the best performance.

Some methods select the influent variables based on statistical tests, which consider the individual discrimination ability of a variable, e.g., in case of classification, the values of a variable can be divided into those associated to the samples of different classes, and a Wilcoxon–Mann–Whitney [21] test can be performed to check whether these subsets of values are distributed according to the same distribution or not. If not, then the variable can be considered influent. However, these methods are based on single variables and mainly take into account only linear relations. Some multivariate methods were proposed to analyse discriminative power of more variables at the same time, as shown in [22]. However, all these approaches only discriminate variables based on some fixed threshold, generally representing the probability that each variable is influent or not, and do not take into account $n_{\max}$. For example, if a variable $x^{(2)}$ is a linear function of $x^{(1)}$, and each of them can be used to build a perfect classifier, a statistical method would find that both are useful, but would not choose one of them, even if $n_{\max} = 1$. An analysis of collinearity or non-linear correlations could be performed to eliminate some variables, based for example on the calculation of Pearson coefficients [23], but, also in this case, a threshold should be fixed to eliminate variables with a high Pearson coefficient, moreover, since the coefficient is defined for couples of variables, it is not simple to choose which variables have to be eliminated.

Depending on the application, the objective could be different from performance, or even multi-objective, often a combination of performance and some other measure. The most useful way to select variables is surely to choose them by evaluating the objective outcome associated with different sets of variables. In this perspective, stepwise procedures consider different models varying for one variable at a time, and choose the best one based on the calculated objective. Among them, backward elimination starts from the complete set of variables and deletes the one associated with the smallest worsening of some type of performance, while forward selection starts with the best variable and adds the best one among the others (also decision trees perform this kind of selection). Both procedures stop when performance variation is under or over a fixed threshold. However, these approaches often do not agree on the same set of the best variables. This is due to the fact that the influence of a variable also depends on the other variables constituting the model. Therefore, for example, if a variable $x^{(3)}$ is comprised in the dataset and it is linearly dependent from $x^{(1)}$ and $x^{(2)}$, and, by itself, it can be used to build a perfect classifier, then, a forward selection method would choose it (and only it) among all the variables, thus finding the simplest perfect classifier, while a backward elimination method could eliminate it and keep the other two, since any combination of two among the three variables can be used to build a perfect classifier, thus finding a perfect classifier using a higher number of variables. On the other hand, if $x^{(3)}$ is not comprised in the dataset, then a forward selection method could find many other variables that singularly classify better than $x^{(1)}$ and $x^{(2)}$, thus presenting difficulties to obtain a perfect classifier, while a backward elimination method would never eliminate one of them, thus finding the simplest perfect model. As a consequence, none of these approaches ensures to find the best set of n variables.

An alternative method was used in [24], which overcomes the drawbacks of selecting variables by using statistical tests and stepwise procedures, based on a brute force research of the best one of all possible sets of n variables. However, it is computationally feasible only with a small number of variables.

2.2. Fuzzy sets

Each fuzzy partition is made of a collection of fuzzy sets defined on a continuous variable, representing the terms of the associated linguistic variable. Therefore, the range $UOD^{(j)}$ of each selected continuous variable $X^{(j)}$, with $j = 1, \ldots, n_{cont}$, is partitioned into a certain number M_j of fuzzy sets $\hat{F}^{(j)}_{m_j}$, described by membership functions $\mu^{(j)}_{m_j}$ with $m_j \in \{1, \ldots, M_j\}$ and modelling terms of the associated linguistic variable. Each data sample $\mathbf{x} = \{x^{(1)}, \ldots, x^{(n)}\}$ belongs to the fuzzy set $\hat{F}^{(j)}_{m_j}$ with membership grade $\mu^{(j)}_{m_j}(x^{(j)})$.

The cardinality of each continuous variable corresponds to the number of fuzzy sets constituting the fuzzy partition, M_j. This or some other related number should be chosen by the developer, which surely influences performance and complexity.

On the other hand, each categorical variable $X^{(j)}$, $j = n_{cont} + 1, \ldots, n$, can assume M_j different values. In this case, M_j cannot be manipulated. Even if each value corresponding to $m_j \in \{1, \ldots, M_j\}$ does not represent a proper fuzzy set but a singleton, these values play in the fuzzy model the same role of linguistic terms of continuous variables. Indeed, for $j = n_{cont} + 1, \ldots, n$, each data sample $\mathbf{x} = \{x^{(1)}, \ldots, x^{(n)}\}$ belongs to the singleton $\hat{F}^{(j)}_{m_j}$ with membership grade 1 if $x^{(j)}$ corresponds to m_j, 0 otherwise. Therefore, categorical variables can be simply included in the fuzzy model, just as particular cases, playing the same role of continuous variables.

The shape used for describing fuzzy sets represents a qualitative choice to be made, which influences performance, complexity, and semantic interpretability. Triangular, trapezoidal and Gaussian MFs are the most widely used in literature. However, only a few works have discussed about the opportunity of using a certain shape for MFs [5–7,18,25,26], and results seem to be application-dependent. In order to ensure the minimum properties for semantic interpretability of each fuzzy set, here one-dimensional, normal, continuous, convex, and unimodal MFs are considered. Moreover, the existence of the leftmost and rightmost fuzzy sets is assumed, and they are called here “shoulders”, while the others are “internal”.

In first instance, not necessarily symmetrical shapes are considered, while orthogonality of different fuzzy sets of each partition is assumed. Therefore, binary, triangular, trapezoidal and sigmoidal shapes are considered. The advantages of fuzzy sets with respect to binary MFs representing crisp intervals (which are discontinuous) are widely recognized [27,28]. Fuzzy numbers, having triangular MFs, can be viewed as particular cases of fuzzy intervals, having trapezoidal MFs. Trapezoidal MFs are very similar to smoother sigmoidal MFs. Moreover, Gaussian and quadratic bell-shaped MFs could also be considered. Using these functions, each (internal) fuzzy set is symmetric, but they are not orthogonal.

In Fig. 1, different types of partition are shown, all with a number of terms $M = 3$ and approximately the same positions (see Appendix B).

The position of fuzzy sets is determined by a set of parameters. In particular, a whole partition with M fuzzy sets can be described by $M - 1$ parameters for binary orthogonal MFs, M parameters for triangular orthogonal MFs, $2(M - 1)$ parameters in case of trapezoidal or sigmoidal orthogonal MFs, and $2M$ parameters in case

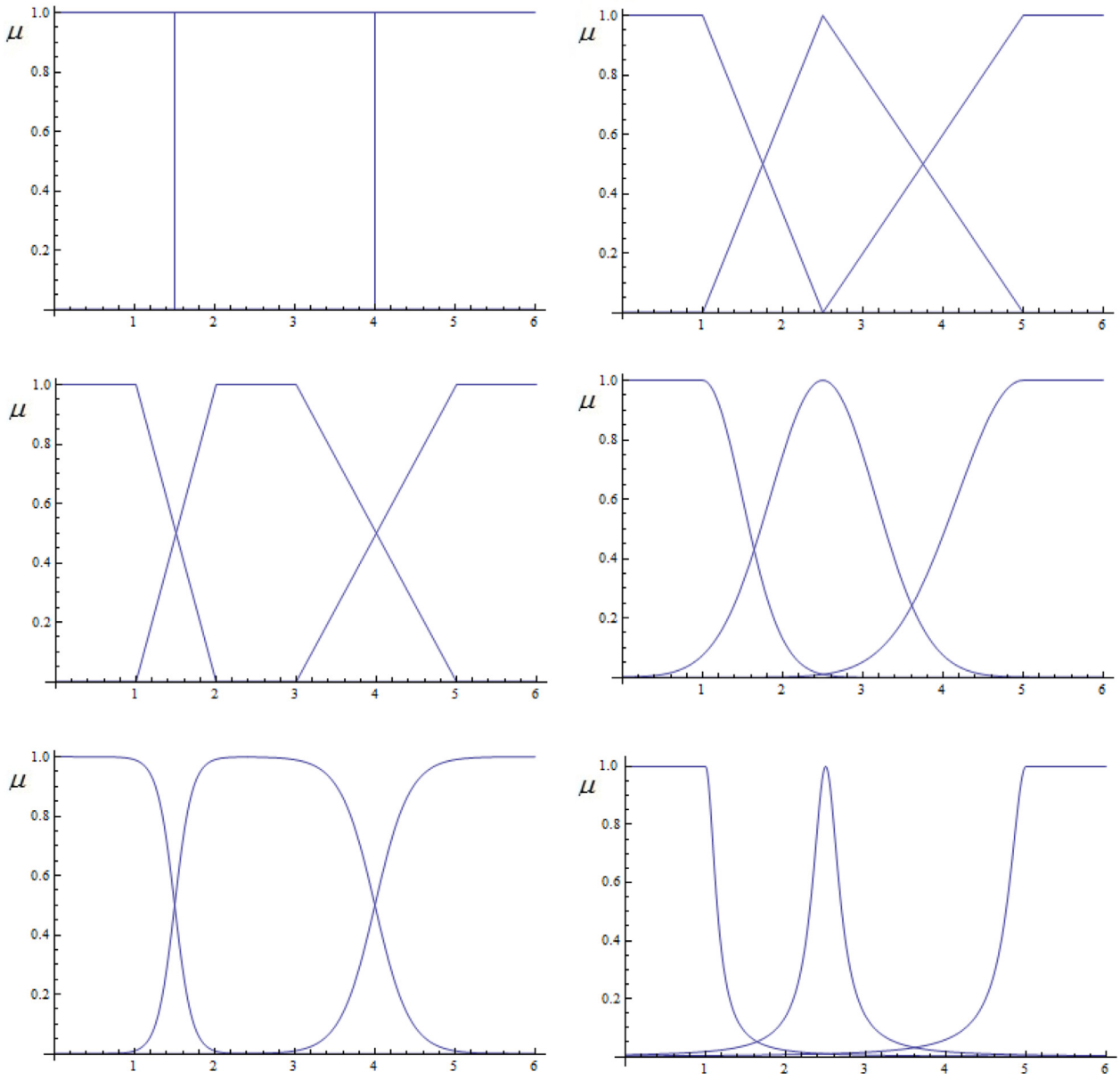

Fig. 1. Fuzzy partitions with (a) binary, (b) triangular, (c) trapezoidal, (d) Gaussian, (e) sigmoidal, and (f) quadratic membership functions.

of Gaussian or quadratic bell-shaped MFs. The method for finding the convenient values of these parameters should be chosen by the developer. The position of fuzzy sets is particularly influent on the system performance. Different methods were used, particularly dedicated to locate the position of fuzzy sets. First, a rough partition of an input variable domain into sharp intervals can be defined: (i) by using intervals obtained from expert's knowledge; (ii) by dividing the domain into a number of equally spaced intervals [29]; (iii) by decision tree algorithms [30]; (iv) by unsupervised clustering methods [31]; (v) by supervised clustering methods [32]. In these cases, specified intervals can be fuzzified by assigning a fixed shape to the corresponding membership functions. However, the prior definition of this shape starting from crisp intervals is a suboptimal solution. Alternatively, an input variable domain can be roughly partitioned by directly estimating fuzzy sets by (i) a priori definition; (ii) equal distribution. However, roughly defined fuzzy sets can be improved in order to maximize their matching with data. For this scope, there are unsupervised methods: (i) unsupervised fuzzy clustering methods [33], partition coefficient maximization [33], partition entropy minimization [33], data equalization [34], and so on. However, since these are mainly unsupervised methods then resulting fuzzy sets are not the best for classification. On the contrary, supervised methods exist, to optimize fuzzy sets positions, in order to maximize the "goodness" of the DSS. Through these methods, either fuzzy sets positions once a rule base has been extracted, or fuzzy partitions and a rule base at the same time, can be obtained. The most noticeable are: (i) fuzzy decision trees [35]; (ii) supervised fuzzy clustering [36,37]); (iii) genetic algorithms [38,39]; (iv) neural networks [8,40]; (v) statistical approaches [41–43], like Likelihood Fuzzy Analysis (LFA) that is described in detail in Section 5.1.4.

2.3. Rules

A rule base can be written as a set of R rules r_ρ with $\rho = 1, \ldots, R$:

$$\text{if } x^{(1)} \text{ is } \hat{F}^{(1)}_{(\rho)} \text{ and/or} \ldots \text{and/or } x^{(n)} \text{ is } \hat{F}^{(n)}_{(\rho)} \text{ then } y \text{ is } \hat{C}_{(\rho)}. \quad (2)$$

In particular, let us name *combinatorial* the rule base made of a set of rules $r_{\{m_1,\dots,m_n\}}$, with $\{m_1,\dots,m_n\} \in \{1,\dots,M_1\} \times \cdots \times \{1,\dots,M_n\}$:

$$\begin{cases} \text{if} x^{(1)} \text{ is } \hat{F}_1^{(1)} \text{ and} \dots \text{and } x^{(n)} \text{ is } \hat{F}_1^{(n)} \text{ then } y \text{ is } \hat{C}_{\{1,\dots,1\}} \\ \dots \\ \text{if } x^{(1)} \text{ is } \hat{F}_{M_1}^{(1)} \text{ and} \dots \text{and } x^{(n)} \text{ is } \hat{F}_{M_n}^{(n)} \text{ then } y \text{ is } \hat{C}_{\{M_1,\dots,M_n\}} \end{cases}. \tag{3}$$

The rule base extraction is a fundamental step of system construction. Some methods exist which are dedicated to this step, the most important being the following: (i) fast prototyping algorithm [44] implements all possible rules, whose antecedent parts are made of each possible combination of fuzzy set of different variables, while the consequent parts are calculated on the top of the training set; (ii) the Wang & Mendel method [45] implements a rule for each data item of the training set. The former method has the drawback of being computationally expensive, while in the latter, results are influenced by data errors. Both the approaches need a successive step for reducing the rule base dimension. As stated before, other methods are also used to obtain fuzzy partitions and a rule base at the same time [32,35–37,41–43].

Each antecedent of a rule base is made of a series of premises "$x^{(j)}$ is $\hat{F}_{(\rho)}^{(j)}$", where $\hat{F}_{(\rho)}^{(j)}$ corresponds to a fuzzy set of the jth partitions, $\hat{F}_{(\rho)}^{(j)} \in \{\hat{F}_1^{(j)},\dots,\hat{F}_{M_j}^{(j)}\}$, and the premises are combined by logic connectives. The number of antecedents in each rule (2), ANT_ρ, or the mean number of rule antecedents in the rule base, ANT, cannot be decided directly, since they depend on the variables number, the rules number, and the type of rule base. In case of a combinatorial rule base, the rule antecedents are as follows:

$$ANT = ANT_\rho = n. \tag{4}$$

Different importance can be assigned to antecedents of each rule by assigning rule weights. This has consequences on performance and on semantic interpretability.

In the consequent part, $\hat{C}_{(\rho)}$ is a fuzzy set [46,47]. While in case of a regression [47], consequents are fuzzy sets defined in the continuous domain of the output variable, in case of a classification, fuzzy sets are defined on the same set of nominal values which the output variable can assume. Then, each consequent can be written as:

$$\hat{C}_{(\rho)} = \frac{C_{\rho-1}}{c_1} + \cdots + \frac{C_{\rho-K}}{c_K}. \tag{5}$$

Here, the consequents are distinguished into *simple consequents*, if among $C_{\rho-1},\dots,C_{\rho-K}$ there is only one value equal to 1 and the others are null, and *fuzzy consequents*, if more values are non-null. Therefore, a simple consequent is made of a singleton whose support is a class:

$$\hat{C}_{(\rho)} = \frac{1}{c_{(\rho)}}, \tag{6}$$

where $c_{(\rho)} \in \{c_1,\dots,c_K\}$, while a fuzzy consequent is made of a proper fuzzy set (5). While models employing simple consequents are properly called "linguistic", those employing fuzzy consequents are also called "relational" fuzzy models [48]. Among the methods already cited for extracting the rule base, the fast prototyping algorithm [44] and the Wang & Mendel method [45] only search for simple rule consequents, while the LFA method [42] optimizes fuzzy consequents. Alternatively, neural networks and genetic algorithms could be effective.

A rule base can be complete or not complete. This has consequences on performance, complexity, and semantic interpretability. A fuzzy rule base is complete [44] if all the input universes are covered by rules. In case it is incomplete, some samples may exist that fire no rule, therefore they cannot be processed. In a combinatorial rule base, all antecedent combinations $\{m_1,\dots,m_n\}$ are used, corresponding to regions of the variables domain, therefore it allows modelling the system behaviour in the whole domain covered by the dataset, i.e., it is a complete rule base. While the method of fast prototyping algorithm [44] and the LFA method proposed by the authors [42] use combinatorial rule bases, therefore they ensure to extract a complete rule base, the Wang & Mendel method [45] and many others do not ensure it.

The number of rules R constituting the rule base, or its maximum $R_{\max}$, or some other number related to it should be fixed. This influences performance and complexity. In case of a combinatorial rule base,

$$R = \prod_{j=1}^{n} M_j. \tag{7}$$

Some methods exist to reduce the rules number, by joining different rules (reduction) or eliminating some of them (selection). If rules reduction is used, different rules with the same consequent and common antecedents are joined, e.g., if the only linguistic terms for $x^{(2)}$ are *low* and *high*, then the rules

$$\begin{cases} \text{if } x^{(1)} \text{ is } low \text{ and } x^{(2)} \text{ is } low \text{ then } y \text{ is } c_1 \\ \text{if } x^{(1)} \text{ is } low \text{ and } x^{(2)} \text{ is } high \text{ then } y \text{ is } c_1 \\ \dots \end{cases} \tag{8}$$

can be transformed into

$$\begin{cases} \text{if } x^{(1)} \text{ is } low \text{ then } y \text{ is } c_1 \\ \dots \end{cases}. \tag{9}$$

Any preferred procedure for realizing as much reduction as possible is surely convenient, since the rule base keeps being complete, therefore complexity decreases without worsening neither performance nor semantic interpretability. On the other hand, if rules selection is used, some rules are completely eliminated. Some procedures for rules selection were implemented in FisPro [49]. However, in this case, the rule base becomes incomplete, therefore while complexity decreases, both performance and semantic interpretability become worse.

A different relative importance can be assigned to each rule of a rule base by using rule weights. The different impact of rules on the result can be modelled [8] by substituting the aggregation S-norm with a weighted S-norm, as specified in the next section. The choice of using weighted or non-weighted rule base influences performance and semantic interpretability. In case rule weights are used, they should also be optimized by the preferred method. To the best of our knowledge, the only methods previously employed for this aim are neural networks [8] and an ad-hoc method developed by the authors [42], however genetic algorithms could be effective as well.

2.4. Inference

The inference process is performed as follows. Each data sample $\mathbf{x} = \{x^{(1)},\dots,x^{(n)}\}$ fires the rule r_ρ with a strength (in the hypothesis of "and" connectives)

$$FS_\rho(\mathbf{x}) = \underset{j=1,\dots,n}{T}\left[\mu_{(\rho)}^{(j)}\left(x^{(j)}\right)\right], \tag{10}$$

while the implication of the consequence is usually modelled as:

$$\hat{IMP}_\rho(\mathbf{x}) = T\left[FS_\rho(\mathbf{x}), \hat{C}_{(\rho)}\right], \tag{11}$$

and different implications are aggregated as:

$$\hat{AGG}(\mathbf{x}) = \underset{\rho=1,\dots,R}{S}\left[\hat{IMP}_\rho(\mathbf{x})\right], \tag{12}$$

where T is a T-norm and S is an S-norm. Note that FS_ρ in (10) is a number, while $\hat{IMP}_\rho$ in (11) and $\hat{AGG}$ in (12) are fuzzy sets:

$$\hat{IMP}_\rho(\mathbf{x}) = \frac{IMP_{\rho-1}(\mathbf{x})}{c_1} + \cdots + \frac{IMP_{\rho-K}(\mathbf{x})}{c_K} \tag{13}$$

$$\hat{AGG}(\mathbf{x}) = \frac{AGG_1(\mathbf{x})}{c_1} + \cdots + \frac{AGG_K(\mathbf{x})}{c_K}, \tag{14}$$

having membership grades

$$IMP_{\rho-k}(\mathbf{x}) = T\left[FS_\rho(\mathbf{x}), C_{\rho-k}\right] \tag{15}$$

$$AGG_k(\mathbf{x}) = \underset{\rho=1,\ldots,R}{S}\left[IMP_{\rho-k}(\mathbf{x})\right]. \tag{16}$$

In case of weighted rules, the S-norm in (16) is substituted by a weighted S-norm [8]:

$$WS[a_1,\ldots,a_R; W_1,\ldots,W_R] = S[W_1a_1,\ldots,W_Ra_R], \tag{17}$$

where all $W_1,\ldots,W_R$ must be in the interval [0, 1] [8]. Finally, a defuzzification step is required to get the inference result (1), where the resulting activation of each class is given by a generic function Φ of the aggregations:

$$y_k(\mathbf{x}) = \Phi\left(k, \hat{AGG}(\mathbf{x})\right). \tag{18}$$

Different types of T norms, S norms and defuzzification operators can be used. To the best of our knowledge, only few attempts were made before to evaluate some of them by considering the final performance of SCs [8], while no attempt by considering performance of CWCs.

Many types of T-norms and S-norms were developed. The choice of a type for each norm used in the inference process can influence the performance. In particular, minimum, product, Łukasiewicz and drastic T-norms, and the respective S-norms, are the most used. Some parameterized families of norms were developed, which smoothly change one norm into the others by varying just one parameter. Soft norms were also proposed [50], which are a middle way between each norm and the arithmetic mean. To the best of our knowledge, very few attempts [8] were done on deciding which type is the most suitable for different applications. However, [8] chose the norms by considering the final performance of SCs, while no attempt was made by considering the performance of CWCs.

A defuzzification operator Φ should be chosen in Eq. (18). This can influence performance of different types of system (SCs or CWC), and the system differentiability. A review on defuzzification procedures was provided by [51]. In case of regression, the defuzzification can be accomplished in different ways [51]. In case of classification, usually the class k which takes the greatest AGG_k is chosen (winner-takes-all strategy), and if there is a dead heat, one class is randomly chosen (random choice of maxima) [51]. In this case:

$$y_k(\mathbf{x}) = \begin{cases} 1, k = \arg\max\limits_{k=1,\ldots,K}\left[AGG_k(\mathbf{x})\right] \\ 0, otherwise \end{cases}. \tag{19}$$

However, in case of CWCs, the result should be constituted by different classes with respective confidence grades, then the defuzzification can be implemented as a normalization [8], alias a fuzzy mean [51], therefore:

$$y_k(\mathbf{x}) = \frac{AGG_k(\mathbf{x})}{\sum_{\kappa=1}^{K} AGG_\kappa(\mathbf{x})}. \tag{20}$$

3. Evaluation of a fuzzy classifier

In this section, different criteria used to evaluate the "goodness" of a fuzzy system are presented, dividing them into different evaluation perspectives: performance, interpretability, and differentiability.

3.1. Performance

In order to evaluate systems performance, here both SCs and CWCs are considered. Therefore, different performance indexes are reported which are widely used for these systems. Moreover, a qualitative evaluation is presented, following [18], regarding the characteristics of the *class activation functions* (CAFs), describing the activation of each class $y_k(\mathbf{x})$ as a function of the input $\mathbf{x}$.

Different indices exist that are usually employed in medical field. For example, *sensitivity* and *specificity* are often used, or equivalent couples of measures. However, these measures suffer of different problems: (i) they can be used only for dichotomous classification; (ii) they are related to the peculiar sense of negative and positive classes; (iii) one cannot fix in advance that he/she wants a high sensitivity rather than a high specificity. Therefore, other measures like *accuracy* summarizing them are more suitable to measure the goodness of any classifier. The Area Under Curve (AUC) of the Receiving Operator Curve (ROC) is also used in some works. However, this measure is not properly a measure of the goodness of a classifier, since it is based on the performance of a series of classifiers, varying for the threshold separating classes. Even if a series of classifiers reaches a high AUC, the choice of the right classifier still depends on both specificity and sensitivity, and thus can be chosen only by considering a combination of them, like accuracy. A scalar function of classification results should be defined by the user, depending on the application.

The systems performances are proposed here to be evaluated in terms of two well-known indexes, i.e., *classification error* (*CE*) [52] and *squared classification error* (*SCE*) [13]:

$$CE = 1 - \frac{1}{N}\cdot\sum_{i=1}^{N}\gamma_i \tag{21}$$

$$SCE = \frac{1}{N}\cdot\frac{1}{K}\cdot\sum_{i=1}^{N}\sum_{k=1}^{K}\left(y_k(\mathbf{x}_i) - \delta_i^k\right)^2, \tag{22}$$

where N is the number of the test set samples, γ_i is 1 if the ith sample is correctly classified and 0 otherwise, $y_k(\mathbf{x}_i)$ is the activation of the kth class for the ith sample, comprised in the interval [0, 1], and δ_i^k is 1 if the correct class for the ith data sample is the kth one and 0 otherwise. Both *CE* and *SCE* should be minimized.

It can be proved that *CE* is minimized when

$$\arg\max_k y_k(\mathbf{x}_i) = \arg\max_k P(c_k|\mathbf{x}_i), \tag{23}$$

for any i, and *SCE* in minimized when

$$y_k(\mathbf{x}_i) = P(c_k|\mathbf{x}_i), \tag{24}$$

for any i and k. Clearly, if (24) stands, also (23) is verified.

While *CE* is simply the fraction of wrongly classified data items, *SCE* takes into account that high (low) confidence should be assigned to right (wrong) solutions. Therefore, while *CE* can be used to measure the classification accuracy, which is $1 - CE$, *SCE* can be used to evaluate the goodness of the classification uncertainty calculated by the system. Therefore, a SC should minimize *CE*, while a CWC aims at minimizing *SCE*.

Moreover, in case when a model is studied with maximum two input variables, the behaviour of 2D lines or 3D surfaces representing the CAFs can be graphically evaluated. They plot the values of each class activation y_k obtained by the model, as a function of the values of the input variables $\mathbf{x}$. These graphics can give qualitative information about the influence of some system settings, by a simple visual description. In the following application (e.g., Fig. 3), where classes "benign" (c_B) and "malignant" (c_M) are distinguished, the surface representing the resulting activation of c_B given different values of the input variables, i.e., $y_B(\mathbf{x})$, is plotted in green, and

that representing $y_M(\mathbf{x})$ in red, and they are shown together. The lateral view of the images is given when the height of the surfaces is to be shown, in order to evaluate the goodness of a CWC. The top view is preferred to emphasize the separation of the variables space into regions associated to different classes, in order to evaluate a SC; e.g., if $y_B(\mathbf{x}) > y_M(\mathbf{x})$, then the point $\mathbf{x}$ of the space is assigned to the class c_B and in the top view it is green. The set of points $\mathbf{x}$ where $y_B(\mathbf{x}) = y_M(\mathbf{x})$ can also be individuated, and it is called *interclass separation*.

3.2. Interpretability

In order to describe and briefly review the criteria to evaluate the interpretability of fuzzy inference systems, a taxonomy proposed in [3] and refined in [4] is used here. In particular, two types of difficulties associated to the process of understanding the system can be individuated, which are *semantic interpretability* [3,4] and *complexity* [3] (named "readability" in [4]). On the other hand, four levels of the system whose interpretation is needed were individuated [4], which are the *single fuzzy sets*, the *fuzzy partitions*, the *single rules*, and the whole *rule base*. Two further levels are proposed here to integrate this taxonomy: the *variables* used to describe sample features, and the *inference* process.

The complexity of the system should be minimized, and it increases with the *number of variables* used by the model, the *number of parameters* used to represent each fuzzy set, the *number of fuzzy sets* (or categories) for each variable (mean cardinality, *M*), the *number of antecedents* in each rule (*ANT*), the total *number of rules R* [4], and the *average number of firing rules* (*ANFR*), having firing strength higher than ε_I during the inference process [3,4,53].

Semantic interpretability can be taken into account by imposing constraints or optimizing some measures, at all the abstraction levels. A comprehensive list of criteria for evaluating semantic interpretability is reported and commented here.

Firstly, let us distinguish between *magnitude-related* linguistic terms like "high temperature", and *value-related* linguistic terms like "around 37 °C" [43].

Regarding the variables, those employed by the system can be of different *type*. It is straightforward that a system comprising only original variables, as available in the dataset, is more interpretable than another system using their linear combinations and/or nonlinear terms [53].

Regarding each fuzzy set, *one-dimensionality, normality, continuity, convexity*, and *unimodality* are required constraints for semantic interpretability as terms of a linguistic variable [3,4,53]. All these characteristics are considered in this work. On the contrary, the *symmetry* condition for each fuzzy set [54] should be required only for value-related linguistic terms [19].

Different fuzzy sets of a fuzzy partition should satisfy *distinguishability, coverage* and *proper ordering* [3,4,53]. In particular, for distinguishability and coverage some measures were proposed which should be minimized [55]:

$$J_D = \frac{1}{N}\sum_{i=1}^{N}\left((SUM_p(x_i)-1)^2 \cdot H(SUM_p(x_i)-1)\right) \tag{25}$$

$$J_C = \frac{1}{N}\sum_{i=1}^{N}\left((SUM_p(x_i)-\varepsilon_C)^2 \cdot H(\varepsilon_C - SUM_p(x_i))\right), \tag{26}$$

where

$$SUM_p(x) = \left(\sum_{m=1}^{M}\mu_m^p(x)\right)^{1/p} \tag{27}$$

$$H(\xi) = \begin{cases} 0, \xi < 0 \\ 1, \xi \geq 0 \end{cases}, \tag{28}$$

ε_C is a threshold fixed for coverage, M is the cardinality, and p is a positive integer (in [55], $p = 1$). Proper ordering should be imposed by considering the strong condition of relation preservation defined in [53], since the corresponding weak condition [53] seems not enough. Moreover, the presence of *leftmost and rightmost* fuzzy sets [53] should be ensured. The usefulness of complementarity among fuzzy sets:

$$\forall x, S - norm[\mu_1(x), ..., \mu_M(x)] = 1 \tag{29}$$

or of their *orthogonality* (when the S-norm is the bounded sum) is controversial [43,53]. However, [19] concluded that orthogonality is a very useful property since it ensures perfect distinguishability, coverage and proper ordering at the same time. Some works also considered the presence of reference fuzzy sets in correspondence of *special elements* (e.g., 37 °C, or the zero element if any) [53], and these should be the only value-related linguistic terms [19], but should be required only if specifically stated by the problem. Regarding the condition of *uniform granulation* [53], it is not considered here since it does not improve semantic interpretability of the partition [19].

The semantic interpretability of each rule depends on the *type of antecedents*, the presence of *antecedent weights*, and on the *type of consequent*. Some works underlined differences of consequent types, e.g., stated that a fuzzy consequent is more understandable than a 0-order Takagi–Sugeno one (a number) [53], which is valid for regression systems. In [19], the authors proposed an order of information levels, noticing their relation with interpretability, thus ordered both rule antecedents and consequents interpretability, comprising the case of antecedent weights. Therefore, while the use of fuzzy antecedents for continuous variables is the basis of fuzzy systems, the use of antecedent weights heavily complicates interpretability. On the other hand, the case of consequents (5) made of a fuzzy set defined on different classes is slightly less interpretable than the case of a singleton.

If the whole rule base is considered, syntactic interpretability is evaluated by checking *Modus Ponens, consistency*, and *locality* of the rule base [3,4,53]. Moreover, other qualitative settings can be accounted, as different rule interpretations (conjunctive or disjunctive, associated with different models for implication and aggregation operators) [53], in case of multiple output the difference between multi-input-single-output (MISO) and multi-input-multi-output (MIMO) systems [53], and so on. Here, all these characteristics are considered as established. In particular, MISO systems with conjunctive interpretation are considered. Instead, the *completeness* of the rule base should be ensured for semantic interpretability. This ensures that all data points are covered by some rule. However, in case of not orthogonal partitions, rule base completeness is not enough to ensure data coverage; therefore, a coverage index *CI* was defined [13], corresponding to the fraction of samples firing at least one rule with strength over ε_R. Moreover, the opportunity of using *rule weights* [8,18,56,57] can be evaluated, since they slightly reduce the system interpretability [56,57] but can improve system performances [8,18].

The interpretability of the inference process depends on the *operators* chosen for the T-norms in (10) and (15), the S-norm in (16), and the defuzzification operator in (18), which should be as simple as possible, e.g., minimum, product, and Łukasiewicz T-norms, maximum, sum, and Łukasiewicz S-norms, and both winner-takes-all strategy and fuzzy mean defuzzification operators can be considered simple enough.

3.3. Differentiability

Even if it is not a usual requirement for a good system, the property of *differentiability* is considered here at different levels.

Firstly, the eventual use of gradient descent methods for optimization requires that each MF should be differentiable [8,58], and all the inference process should be made of differentiable operators.

Moreover, here, the differentiability of CAFs describing $y_k(\mathbf{x})$, which derives from that of MFs and inference operators, is also considered as a requisite able to ensure system generality, which means that the system should be applicable to any type of data. Indeed, a differentiable CAF can approximate any real class posterior probability function (CPPF) $P(c_k|\mathbf{x})$, while a non-differentiable CAF cannot approximate a differentiable CPPF. The same consideration can be made about the interclass separation, which should be differentiable to ensure system generality.

Therefore, all the operators used in (10), (15), (16) and (18), comprising two T-norms, an S-norm and the operator Φ, should be differentiable. Differentiability of CAFs and of the interclass separation can be easily checked by visual evaluation of CAFs plots.

4. The proposed design procedure

In this section, an approach to design rule-based fuzzy systems for classification in medical ambit is proposed.

As explained in Section 4.1, some characteristics of the classifier are directly individuated by considering general motivations based on usual requirements of medical domain experts. Other characteristics depend on the dataset and on the specific requirements of the application. Therefore, some parameters are proposed in Section 4.2 to formulate particular application requirements. In the proposed general algorithm, described in Section 4.3, the way for saturating the application-dependent degrees of freedom and the remaining choices is suggested, while the optimization method is left to the designer. The efficiency of the algorithm is evaluated in Section 4.4.

4.1. General settings in medicine

In order to be used in the medical ambit, classification systems are required: (i) to describe the model by a transparent knowledge base, giving to the physician a clearly interpretable and logic justification of the classification process [28]; (ii) to enrich the assignment of each patient to a class with the associated uncertainty, giving to the physician a measure to evaluate the confidence with which each classified case should be handled. Fuzzy Logic [2] has widely demonstrated its capability in supporting decisions for different medical classification problems, such as [43,54,55]; in particular, the use of fuzzy logic enables to obtain a transparent knowledge base. However, the interpretability of the system and the measure of confidence of results are not foregone. Therefore, the design of a fuzzy classifier is based here on the following hypotheses:

- the semantic interpretability is often a crucial requirement of a system for classification in medicine;
- a trade-off between good performance and low complexity is usually desired;
- a measure of the confidence of the classification results is usually needed, i.e., a CWC should be built.

As a consequence, a fuzzy system in medicine should be constructed by selecting the degrees of freedom in the following manner:

- the proper number of variables depends on the desired trade-off between performance and complexity;
- original variables should be used for semantic interpretability;
- the variables selection method depends on the total number of variables: for few variables, the best couples or triplets of variables can be found by a brute force approach, while for numerous variables the forward selection method results faster than backward elimination, therefore, for general applicability, the forward selection should be chosen;
- the proper number of fuzzy sets for each variable depends on the desired trade-off between performance and complexity;
- the proper shape of fuzzy sets depends on data, however, based on considerations made in Section 5.1.2, only sigmoidal MFs result appropriate in general;
- optimization of fuzzy sets positions can be made by the preferred method, among those reported in Section 2.2;
- antecedent weights should be avoided for semantic interpretability;
- antecedents number depends on other choices;
- fuzzy consequents should be used instead of simple ones, in order to build a CWC, and a method among those suggested in Section 2.3 should be used to optimize them;
- rule weights can be used to improve performance, even if they slightly reduce semantic interpretability, and in this case the method for their optimization can be chosen among those reported in Section 2.3; alternatively, rule weights can be avoided if performance is not improved enough; the choice of using them depends on the desired trade-off between performance and semantic interpretability;
- rules number correspond to that of a combinatorial rule base, since neither rules reduction nor selection can be used (see following points);
- rule base completeness should be ensured for semantic interpretability;
- rules reduction cannot be used for rules with fuzzy consequents, while rules selection should be avoided, in order to keep rule base completeness;
- any types of norms can be chosen, given that they are simple and differentiable, however, product T-norms and Łukasiewicz S-norm allow to use the procedure presented below;
- defuzzification should be implemented as a normalization, in order to build a CWC.

Optimization of fuzzy sets positions, rule consequents and rule weights should be made at the same time, therefore, the same method should be employed. The aim is to minimize *SCE*, in order to optimize performance of a CWC.

4.2. Trade-off parameters

The choice of some of the degrees of freedom depends on a trade-off between opposite trends of system performance and interpretability. The developer should decide about these trade-off settings in advance, and here some parameters are proposed to help him/her.

If the number of variables n increases, the system performances can surely improve. On the other hand, the system is undoubtedly complicated by the presence of an increasing number of variables, which increases complexity directly and by increasing the mean number of antecedents. In order to consider that the system complexity is directly related to n, then the corrected rate Qn between the performance, measured as $1 - SCE$, and the number of variables n should be maximized:

$$Qn = \frac{1 - SCE_n}{n + 1 + 1/q_n}, \tag{30}$$

where SCE_n is the squared classification error obtained with n variables, and q_n is a parameter to fix, with $0 < q_n \leq 1$, which controls the desired trade-off: if it is near 0, the performance is considered

mostly, while if it increases to 1, greatest consideration is given to the complexity.

If M_j is increased, the highest granularity of the input allows better approximating the output, thus improving the system performances. On the other hand, the system is undoubtedly complicated by the presence of an increasing number of linguistic terms for each variable, which increases the mean cardinality and the total number of rules, therefore increases the complexity. In order to consider that the system complexity is directly related to the number of fuzzy sets, then the corrected rate *QM* between the performance and the number of fuzzy sets should be maximized:

$$QM = \frac{1 - SCE_M}{M - 1 + 1/q_M}, \tag{31}$$

where SCE_M is the squared classification error obtained with the single variable partitioned into M fuzzy sets, and q_M is a parameter to fix, with $0 < q_M \leq 1$, which controls the desired trade-off: if it is near 0, the performance is considered mostly, while if it increases to 1, greatest consideration is given to the complexity.

The choice of using rule weights can improve performance. On the other hand, they reduce semantic interpretability. In order to consider that the loss of system interpretability should be compensated by a certain performance increase, then the corrected performance *QW* should be maximized:

$$QW = \begin{cases} (1 - q_W) \cdot (1 - SCE_W) & \text{if rule weights are used} \\ 1 - SCE_{NW} & \text{otherwise} \end{cases}, \tag{32}$$

where SCE_W is the squared classification error obtained with rule weights, SCE_{NW} is obtained without rule weights, and q_W is a parameter to fix, with $0 \leq q_W \leq 1$, which controls the desired trade-off: if it is 0, the performance is considered mostly (and weights are probably used), while if it increases to 1, greatest consideration is given to the semantic interpretability (and weights are surely avoided).

4.3. Algorithm

The complete procedure proposed here for designing rule-based fuzzy classifiers for medical applications is outlined by Algorithm 1.

The inputs of the algorithm are the following. Some are manual settings, depending on the application, i.e., the dataset, comprising N samples, each described by *nvar* features and associated to a class $c_1, \ldots, c_K$; the maximum number of fuzzy sets for each partition $M_{\max}$ (a maximum of 3 fuzzy sets is suggested, however, it can be increases up to 5 or more if desired, but optimization become more difficult as this number increases, and experience can drive to the observation that a too much fine granulation does not imply performance improvement); the maximum number of variables comprised in the model $n_{\max}$ (the total number of variables *nvar* is suggested, given that it is not too high); the three parameters q_n, q_M, and q_W, proposed in Section 4.2 (proper ranges are reported). Other settings are suggested with general purposes, i.e., the shape of fuzzy sets (*MFshape*) (the sigmoidal shape is suggested, for reasons explained in Section 5.1.2); the T-norms for calculating firing strength (10) (*FS-T-norm*) and implication (15) (*IMP-T-norm*) (product T-norms should be used, for reasons explained in the following); the S-norm for aggregation (16) (*AGG-S-norm*) (Łukasiewicz S-norm should be used, for the same reasons); the operator Φ for defuzzification (18) (*DEF*) (normalization is suggested, in order to obtain a CWC); and, finally, the preferred *Method* for optimizing fuzzy sets positions, rule consequents and rule weights (Likelihood Fuzzy Analysis (LFA), Neural Networks (NN), and Genetic Algorithms (GA) are suggested). The output of the algorithm is the fuzzy classification system (*FuzzyClassifier*).

FS-T-norm, IMP-T-norm, AGG-S-norm and *DEF* are summarized in Algorithm 1 as *inference*.

Firstly, the prior probabilities of classes $P(c_k)$, $k = 1, \ldots, K$, are calculated, which are used in successive steps.

Then, the worst possible classifier with fuzzy consequent is considered, which comprises no variable ($vars_0 = \{\}$) and the only rule

$$y \text{is} \frac{P(c_1)}{c_1} + \cdots + \frac{P(c_K)}{c_K}, \tag{33}$$

with fuzzy consequent as defined in (5). This model gives, for any $\mathbf{x}$,

$$y_k(\mathbf{x}) = P(c_k). \tag{34}$$

The associated *SCE* and *Qn* are calculated by (22) and (30), respectively.

Afterwards, different models are found comprising each single variable j. For each continuous variable, different models are found with a number of fuzzy sets M_j going from 2 to $M_{\max}$. According to (2) and (5), the generic rule of the jth single variable model is made as follows, with $m_j = 1, \ldots, M_j$:

$$\text{if } x^{(j)} \text{ is } \hat{F}^{(j)}_{m_j} \text{ then } y \text{ is } \frac{C^{(j)}_{m_j-1}}{c_1} + \cdots + \frac{C^{(j)}_{m_j-K}}{c_K}, \tag{35}$$

where $\sum_{k=1}^{K} C^{(j)}_{m_j-k} = 1$. Each model is found, for continuous variables, by optimizing the parameters of fuzzy sets positions and the rule consequents $C^{(j)}_{m_j-k}$ (rule weights are redundant for single variable models), through the chosen *Method*. For each categorical variable assuming M_j values, $C^{(j)}_{m_j-k}$ is fixed equal to $P(c_k|m_j)$. In both cases, *SCE* is maximized. Then, *SCE* is calculated for evaluating each model by (22) and *QM* by (31). Hence, for each variable, the optimal number of fuzzy sets is found by maximizing *QM*, and fuzzy sets positions and rule consequents are chosen accordingly.

At this point, the best single variable $j(1)$ is chosen, based on *SCE*, and the associated model ($vars_1 = \{j(1)\}$) is evaluated by *Qn* (30).

A forward selection method for variables choice is proposed here for general purposes, which is more suitable for a high *nvar*. Accordingly, as the number of variables increases to ν, the best variable is added to the set of variables constituting the model ($vars_\nu = vars_{\nu-1} \cup \{j(\nu)\}$), and the same thing happens adding one variable at a time, but only while the new *Qn* value is higher with respect to the previous one, $Qn_\nu > Qn_{\nu-1}$, and the maximum number of variables has not been reached.

For models comprising more than one variable, the same positions of fuzzy sets optimized for single variables are retained in this procedure, since interactions are assumed to be negligible. Moreover, it is proposed here to calculate rule consequents and rule weights of models comprising more than one variable by simply applying some operations on rule consequents optimized for single variables. Therefore, in order to obtain models with more than one variable, no optimization step is needed, thus avoiding very long computation time. Suppose that the generic rule of a n-variables model is made as follows:

$$\left(W_{\{m_1,\ldots,m_n\}}\right) \text{ if } x^{(1)} \text{ is } \hat{F}^{(1)}_{m_1} \text{ and } \ldots \text{ and } x^{(n)} \text{ is } \hat{F}^{(n)}_{m_n} \text{ then } y \text{ is } \frac{C_{\{m_1,\ldots,m_n\}-1}}{c_1} + \cdots + \frac{C_{\{m_1,\ldots,m_n\}-K}}{c_K}, \tag{36}$$

with weight $W_{\{m_1,\ldots,m_n\}}$. Then, the following operations can be made to obtain rule consequents and weights of multiple variables models from the consequents of the single variable models (35):

$$W_{\{m_1,\ldots,m_n\}} = \frac{\sum_{k=1}^{K} \varpi_{\{m_1,\ldots,m_n\}-k}}{\sum_{\eta_1=1}^{M_1} \cdots \sum_{\eta_n=1}^{M_n} \sum_{k=1}^{K} \varpi_{\{\eta_1,\ldots,\eta_n\}-k}}, \tag{37}$$

Algorithm 1.

// Manual and general settings
Input: {dataset, M_{max} (3), n_{max} (nvar), q_n ($0<q_n\leq 1$), q_M ($0<q_M\leq 1$), q_W ($0\leq q_W\leq 1$), Method (LFA or NN or GA)}
U
{MFshape (sigmoid), FS-T-norm (product), IMP-T-norm (product), AGG-S-norm (Łukasiewicz), DEF (normalization)}

Output: {FuzzyClassifier}

Start

inference = {FS-T-norm, IMP-T-norm, AGG-S-norm, DEF}

// Calculate prior probabilities
For k = 1, ..., K
$P(c_k) = N_k / N$
EndFor

// Write and evaluate model with 0 variables
$\nu = 0$
$vars_0 = \{\}$
$rulebase_0$ = "y is $P(c_1)/c_1+...+P(c_K)/c_K$"
$FuzzyClassifier_0 = \{vars_0, rulebase_0, inference\}$
SCE_0 = SCE calculated on $\{dataset, FuzzyClassifier_0\}$
$Qn_0 = (1 - SCE_0) / (0 + 1 + 1 / q_n)$

// Search models comprising each single variable
$\nu = 1$
For j = 1,...,nvar,
// Optimization of positions of fuzzy sets and fuzzy consequents of models with 1 variable and different cardinalities
If variable j is continuous Then
For M_j = 2, ..., M_{max},
$\{positions_{j,Mj}, consequents_{1,j,Mj}\}$ = optimized by Method based on {dataset, j, M_j, MFshape)
EndFor

// Write and evaluate models with 1 variable and different cardinalities
For M_j = 2, ..., M_{max},
$vars_{1,j} = \{j\}$
$fuzzysets_{j,Mj}$ = M_j fuzzy sets with MFshape and $positions_{j,Mj}$
$rulebase_{1,j,Mj}$ = single variable rules with optimized $consequents_{1,j,Mj}$
$FuzzyClassifier_{1,j,Mj} = \{j, fuzzysets_{j,Mj}, rulebase_{1,j,Mj}, inference\}$
$SCE_{1,j,Mj}$ = SCE calculated on $\{dataset, FuzzyClassifier_{1,j,Mj}\}$
$QM_{j,Mj} = (1 - SCE_{1,j,Mj}) / (M_j - 1 + 1 / q_M)$
EndFor

// Choose best cardinality
$M_j^* = argmax(QM_{j,Mj})$
$fuzzysets_j^* = fuzzysets_{j,Mj^*}$
$consequents_{1,j}^* = consequents_{1,j,Mj^*}$
$SCE_{1,j} = SCE_{1,j,Mj^*}$
Else
$\{consequents_{1,j,Mj}\} = P(c_k|m_j)$
EndIf
EndFor

// Choose, write and evaluate best model with 1 variable
$j(1) = argmin(SCE_{1,j})$
$vars_1=\{j(1)\}$
$rulebase_1 = rulebase_{1,j(1),Mj(1)^*}$
$FuzzyClassifier_1 = \{vars_1, fuzzysets_{j(1)}^*, rulebase_1, inference\}$
$SCE_1 = SCE_{1,j(1)}$
$Qn_1 = (1 - SCE_1) / (1 + 1 + 1 / q_n)$

Algorithm 1. Continued

```
// Forward selection
While (Qnν > Qnν-1 And ν ≤ nmax)
        ν = ν + 1
        If ν ≤ nmax Then

                // Write and evaluate models comprising each further variable, with and without rule weights
                For j = 1,...,nvar, j ∉ varsν-1
                        varsν = Union(varsν-1, {j})
                        consequentsν,j = consequents calculated on {varsν} from consequents1,j* and P(ck)
                        ruleweightsν,j = weights calculated on {varsν} from consequents1,j* and P(ck)
                        weightedrulesν,j = ν-variables rule base with ruleweightsν,j and consequentsν,j
                        weightedFuzzyClassifierν,j = {varsν, fuzzysetsvarsν*, weightedrulesν,j, inference}
                        SCEWν,j = SCE calculated on {dataset, weightedFuzzyClassifierν,j}
                EndFor

                // Choose, write and evaluate best model with ν variables
                j(ν) = argmin(SCEν,j)
                varsν = Union(varsν-1, {j(ν)})
                rulebaseν = weightedrulesj(ν)
                weightedFuzzyClassifierν = {varsν, fuzzysetsvarsν*, rulebaseν, inference}
                SCEWν = SCEWν,j(ν)
                Qnν = (1 - SCEWν) / (ν + 1 + 1 / qn)
        EndIf
EndWhile

// Write and evaluate weighted and non-weighted model
n = ν - 1
vars = varsn
fuzzysets = fuzzysetsvarsn*
consequents = consequentsn,j(n)
weightedrules  = weightedrulesn,j
nonweightedrules = n-variables rule base with no rule weights and consequentsn,j(n)
weightedFuzzyClassifier = {vars, fuzzysets, weightedrules, inference}
nonweightedFuzzyClassifier = {vars, fuzzysets, nonweightedrules, inference}
SCEW = SCEWn,j(n)
SCENW = SCE calculated on {dataset, nonweightedFuzzyClassifierj}
QWW = (1 – qW) · (1 – SCEW)
QWNW = (1 – SCENW)

// Choose weighted or non-weighted model
If QWW > QWNW Then
        rulebase = weightedrules
        SCE = SCEW
Else
        rulebase = nonweightedrules
        SCE = SCENW
EndIf

// Write final model
FuzzyClassifier = {vars, fuzzysets, rulebase, inference}

End
```

$$C_{\{m_1,\ldots,m_n\}-k} = \frac{\varpi_{\{m_1,\ldots,m_n\}-k}}{\sum_{k=1}^{K} \varpi_{\{m_1,\ldots,m_n\}-k}}, \tag{38}$$

where

$$\varpi_{\{m_1,\ldots,m_n\}-k} = \frac{\prod_{j=1}^{n} C_{m_j-k}^{(j)}}{P(c_k)^{n-1}}. \tag{39}$$

It can be proved that, in the naïve Bayes hypothesis, if the consequents of single variable models are optimized (with respect to *SCE*), fuzzy sets are normal and orthogonal, and product T-norms, Łukasiewicz S-norm and fuzzy mean defuzzification are used, then the consequents and weights of a multiple variables model obtained through these operations result optimized as well. The proof is given in Appendix A. Product T-norms and Łukasiewicz S-norm are suggested since they are also simple and differentiable. In Algorithm 1, the operations (37) and (38) allow to calculate $consequents_{\nu,j}$ and $ruleweights_{\nu,j}$, from $consequents_j{}^*$ and $P(c_k)$. However, it is actually not necessary to calculate parameters of all the multiple variables models that should be evaluated, since, through (22) and (A.10), it is possible to evaluate weighted

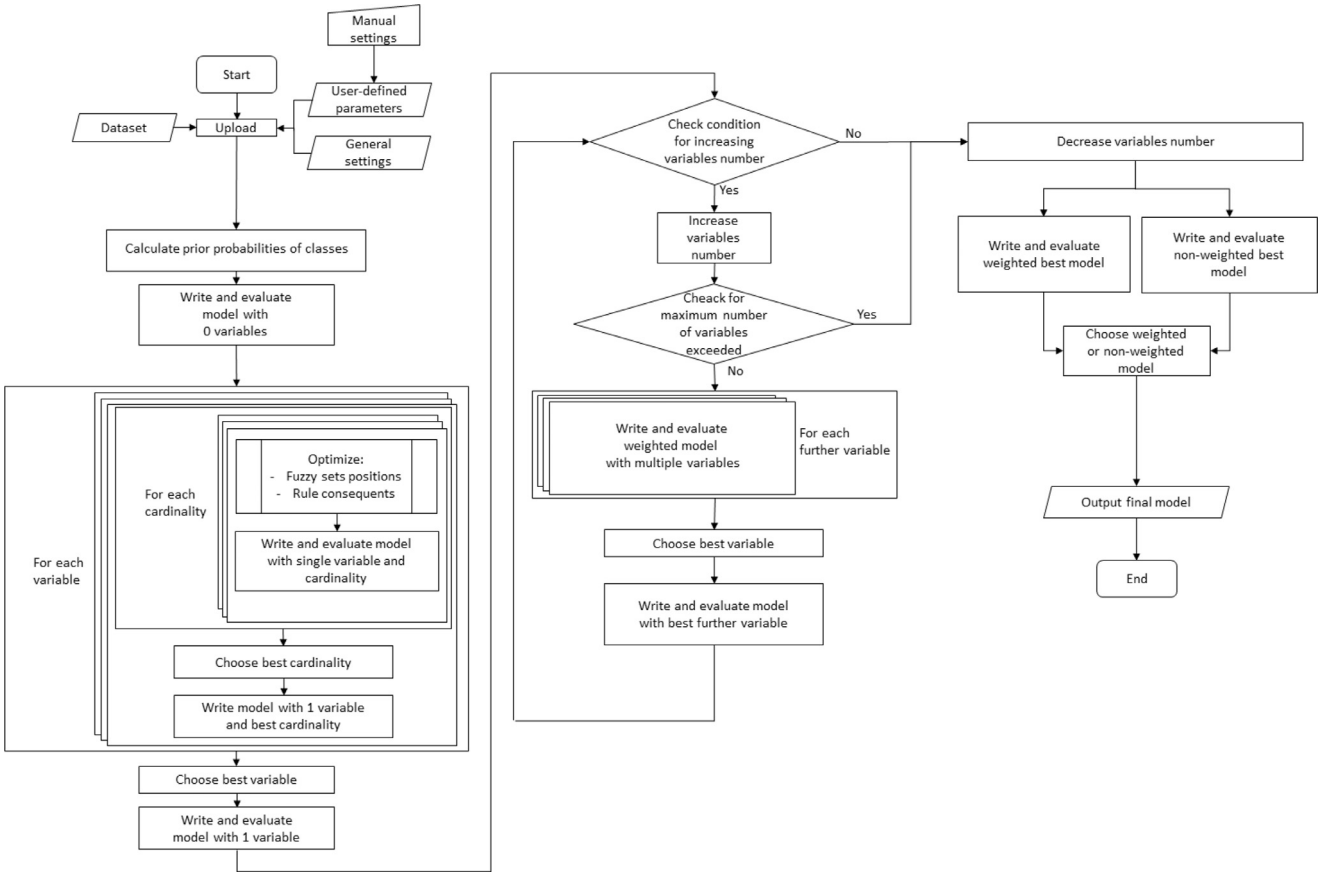

Fig. 2. Block diagram showing the proposed procedure.

models without calculating them, thus speeding up the process. Therefore, the calculation of all models is reported for clarity in Algorithm 1 and Fig. 2, but its computation time is not considered in the following asymptotical analysis (Section 4.4).

The best variable in terms of *SCE* is added to the model if the new associated *Qn* value is higher with respect to the previous one, and so on, until the maximum number of variables is reached, or the performance is not improved enough.

Once the n variables constituting the model are individuated, the parameters of the final model have to be calculated by (37) and (38), *SCE* of weighted model can be calculated by (22) and (A.10), while that of non-weighted model by (22) and (A.7). *QW* of both weighted and non-weighted rule bases are evaluated through (32), and one of them is chosen based on the values of *QW*.

Finally, the selected variables, their fuzzy partitions and the rule base are individuated, which constitute the fuzzy classifier.

The fuzzy classifier can be used to make inference on incoming samples, by computing resulting class activation values. On the other hand, for simple cases (e.g., if the model is made of rules like (35)), the inference can be also deduced by graphical representations of fuzzy partitions and rules directly; e.g., in case in correspondence of a data sample it results $\mu_{m_j}^{(j)}(x_i^{(j)}) = 1$, this can be detected directly from visualization of fuzzy partition of variable $x^{(j)}$, and since antecedent of (35) stands, then the result is directly recognized as the consequent of (35).

4.4. Asymptotical analysis

In this section, in order to evaluate the efficiency of the proposed procedure, the computation time T required for extracting a model from data is evaluated, as a function of the input size S. The input is made of N samples, each described by *nvar* features, represented by $nvar_{cont}$ continuous and $nvar_{cat}$ categorical variables. Therefore, the input size corresponds to $S = N \cdot (nvar_{cont} + nvar_{cat})$, and T is a function of N, $nvar_{cont}$ and $nvar_{cat}$: $T(N, nvar_{cont}, nvar_{cat})$.

In particular, a worst-case analysis is performed, by calculating the upper limiting computation time $Tmax(N, nvar_{cont}, nvar_{cat})$.

Assume that the following times are needed to perform simple operations:

- t_s for a sum;
- t_p for a product or division;
- t_μ to compute a membership function;
- t_c to make a comparison;
- a negligible time for an assignment;
- a negligible time to do a union of vectors;
- $t_{a,m} = t_{a,m} = t_{m1} \cdot m + t_{m2}$ to find *argmax* or *argmin* of a vector with m elements.

Then, the following times are needed to perform calculations required by Algorithm 1:

- $t_{priors} = t_s \cdot N + K \cdot t_p$ to calculate prior probabilities;
- $t_{y,0} = 0$ to calculate $y_k(\mathbf{x})$ of a model without variables by (34), $t_{y,1} = M_j \cdot (t_\mu + t_p + t_s)$ to calculate $y_k(\mathbf{x})$ of a model with one variable by (A.3), $t_{y,n,A.10} = K \cdot 2t_p \cdot n + K \cdot \sum_{j=1}^{n} M_j \cdot (t_\mu + t_p + t_s) + K \cdot t_s + t_p$ to calculate $y_k(\mathbf{x})$ of a model with n variables by using (A.10), $t_{y,n,A.7} = K \cdot \prod_{j=1}^{n} M_j \cdot (t_\mu + t_p) \cdot n + K \cdot \prod_{j=1}^{n} M_j \cdot (2t_p + t_s) + t_p$ by using (A.7);
- $t_{SCE,n} = (t_{y,n} + 2t_s + t_p) \cdot K \cdot N + 2t_p$ to calculate *SCE* of a model;
- $t_{Qn} = t_{QM} = 3t_s + 2t_p$ to calculate *Qn* or *QM*;
- $t_{opt} = t_{m0} + t_{m1} \cdot N$ to optimize positions of fuzzy sets and fuzzy consequents of models with one continuous variable and different cardinalities, where t_{m0} and t_{m1} are specific of the method,

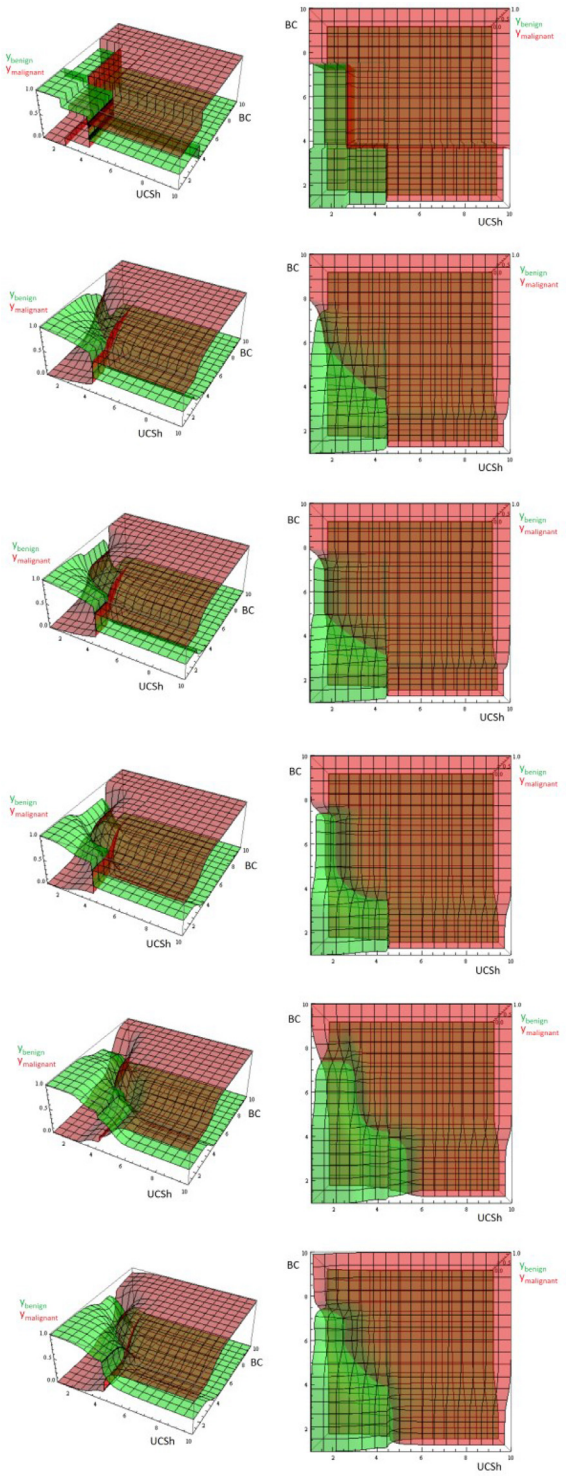

Fig. 3. Class activation functions obtained by using (a) binary, (b) triangular, (c) trapezoidal, (d) sigmoidal, (e) Gaussian, and (f) quadratic bell-shaped MFs.

and since $(M_j + K \cdot M_j - 2)$ is the total number of parameters if sigmoid MFs are used, in the worst case when the maximum number of iterations $Iter_{\max}$ is required for optimization, t_{m0} increases with $Iter_{\max} \cdot K \cdot {M_{\max}}^2$, and t_{m1} increases with $Iter_{\max} \cdot M_{\max}$ for a typical optimization method iteratively updating parameters based on calculated *SCE* (like NN and GA), while t_{m1} is constant for LFA;

- $t_{posts} = 2t_s \cdot N + M_{\max} \cdot K \cdot t_p$ to calculate posterior probabilities of one categorical variable, supposed for simplicity to assume $M_{\max}$ different values;
- $t_{model,n} = 2t_p \cdot K \cdot \prod_{j=1}^{n} M_j \cdot n + (t_s + t_p) \cdot (K+1) \cdot \prod_{j=1}^{n} M_j$ to calculate rule weights and consequents of a model with n variables by (37)–(39); note that, if (A.10) is used to evaluate models, rule weights and consequents have to be computed only one time to compare weighted and non-weighted models and output one of them as the final model;
- $t_{QW} = 3t_s + t_p$ to calculate a couple of *QW*.

Therefore, the total time required to compute the final model in the worst case is:

$$\begin{aligned} &Tmax(N, nvar_{cont}, nvar_{cat}) = t_{priors} + t_{SCE,0} + t_{Qn} \\ &+nvar_{cont} \cdot \left(t_{opt} + \sum_{M_j=2}^{M_{\max}} (t_{SCE,1} + t_{QM}) + t_{\mathrm{a},(M_{\max}-1)} \right) \\ &+nvar_{cat} \cdot t_{posts} + t_{\mathrm{a},nvar} + t_{Qn} + \sum_{\nu=2}^{n_{\max}} \\ &\left(2t_c + t_s + t_c + (nvar - \nu + 1) \cdot t_{SCE,\nu,A.10} + t_{\mathrm{a},(nvar-\nu+1)} + t_{Qn}\right) \\ &+2t_c + t_s + t_c + t_s + t_{model,n_{\max}} \\ &+t_{SCE,n_{\max},A.10} + t_{SCE,n_{\max},A.7} + t_{QW} + t_c. \end{aligned} \tag{40}$$

With some algebra, it can be written as:

$$\begin{aligned} &Tmax(N, nvar_{cont}, nvar_{cat}) = a \cdot nvar_{cont} \cdot N + b \\ &\cdot nvar_{cat} \cdot N + c \cdot N + d \cdot nvar_{cont} + e \cdot nvar_{cat} + f, \end{aligned} \tag{41}$$

where all coefficients increase with K, $M_{\max}$ and $n_{\max}$, and a and d increase also with $Iter_{\max}$. In particular, since c and f increase with ${M_{\max}}^{n_{\max}}$, the procedure suggests to choose $M_{\max} = 3$, and $n_{\max} = nvar$ only if $nvar$ is small, otherwise a lower $n_{\max}$. However, note that the calculated time corresponds to that of the worst case, while in practice usually a little number of variables are worth to be inserted in the model. Anyhow, a, b, c, d, e and f can be considered constant values, depending on the implementation (t_s,t_p, t_a,t_μ, t_c), on the design choices ($M_{\max}$, $n_{\max}$ and t_{method}), and on the problem (K).

The input size grows up linearly as N, $nvar_{cont}$, or $nvar_{cat}$ increase. On the other hand, (41) shows that $Tmax(N, nvar_{cont}, nvar_{cat})$ grows up linearly as N, $nvar_{cont}$ or $nvar_{cat}$ increase. Therefore, the computation time increases linearly with the input size, i.e.,

$$T(S) = \mathrm{O}(S). \tag{42}$$

The demonstrated linear-time solvability proves the computational efficiency of the proposed procedure.

Note that the proposed procedure, using the naïve Bayes hypothesis, involves to separately optimize parameters relative to different variables, and their number is linear with $nvar_{cont}$, therefore it is scalable. In the opposite case of removing the hypothesis and performing optimization of all multiple variables models, the parameters to optimize for each model, relative to different variables, would be coupled, and their number would increase exponentially with $nvar$, thus the optimization would be much more difficult and the algorithm would not be efficient.

5. Application to real data

In this section, the proposed approach is experienced on a number of benchmark datasets. Firstly, the application of the design procedure is shown in Section 5.1, on a dataset taken as example. Then, in Section 5.2, the results of this procedure are compared with the best ones found in literature.

All the applications of the proposed procedure were implemented by using the Mathematica 8.0 software [59].

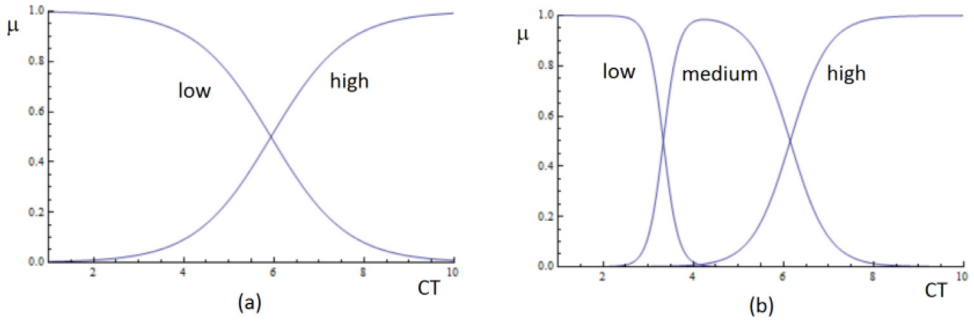

Fig. 4. Fuzzy partitions of the Clump Thickness, with (a) 2 and (b) 3 fuzzy sets.

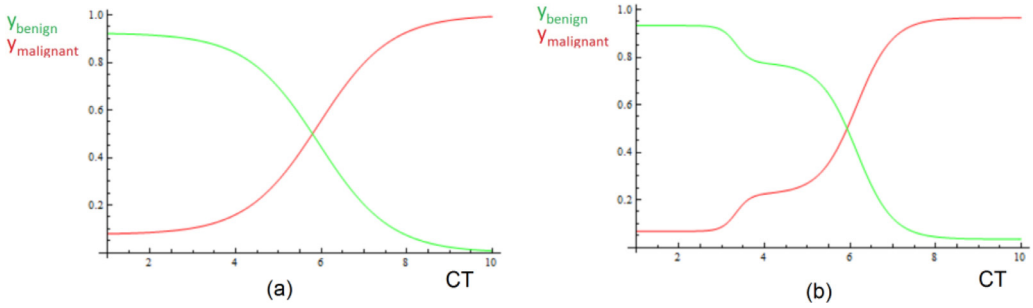

Fig. 5. Class activation functions of models obtained with Clump Thickness partitioned into (a) 2 and (b) 3 fuzzy sets.

5.1. Example of procedure application

Here, the proposed procedure for designing a fuzzy classifier is exemplified on a well-known dataset, described in Section 5.1.1. The choice of some settings of the fuzzy system is driven by the procedure directly, as explained before, and such settings are summarized in Section 5.1.2. For some particular degrees of freedom, the choice is based also on the desired trade-off between performance and interpretability opposite trends, through the proposed ad-hoc parameters. These parameters are fixed in Section 5.1.3, in order to show how they can be used to saturate respective degrees of freedom through the procedure application. In Section 5.1.4, it is also shown an example of a comparison between the results of different optimization methods.

5.1.1. Dataset

The Wisconsin Breast Cancer Dataset, available in UCI Machine Learning Repository [20], is used here as a proof of concept. This dataset was chosen because many works tested knowledge extraction methods on it, and in particular, different fuzzy systems [8,38,39] were used to obtain respective results, which can be used to compare results of the present work.

Each one of the 699 samples is described by 9 variables (with some missing values), corresponding to the following sample features: Clump Thickness (*CT*), Uniformity of Cell Size (*UCSi*), Uniformity of Cell Shape (*UCSh*), Marginal Adhesion (*MA*), Single Epithelial Cell Size (*SECS*), Bare Nuclei (*BN*), Bland Chromatin (*BC*), Normal Nucleoli (*NN*), and Mitoses (*Mi*). Moreover, 458 samples are labelled as *benign* (c_B) and 241 as *malignant* (c_M). The output of the classification system consists in a couple of values, y_B and y_M, indicating the respective activation of classes.

5.1.2. General settings

The general settings of the system are summarized in Table 2.

Most of the general settings motivations are explained in the previous section: original variables, no antecedent weights, and rule base completeness (implying no rules selection) are used for semantic interpretability; the forward selection method is suggested in order to face cases with many variables; fuzzy consequents (implying no rules reduction) and fuzzy mean defuzzification are used to obtain results with confidences; Product T-norm and Łukasiewicz S-norm are simple, differentiable, and functional to the proposed procedure.

Table 2
General settings.

Main aspect	Degree of freedom	Settings
Variables	Number	–
	Type	Original
	Selection method	Forward selection
Fuzzy sets	Average number	–
	Shape	Sigmoidal
	Method to optimize positions	–
Rules	Antecedents weights	No
	Antecedents average number	–
	Consequents type	Fuzzy
	Rule weights	–
	Rules number	–
	Rule base completeness	Yes
	Rules reduction/selection	No
Inference	Norms type	Product / Łukasiewicz
	Defuzzification type	Normalization

Here, let us motivate the choice of sigmoidal fuzzy sets shape. Different shapes of fuzzy sets are compared, which are represented by binary, triangular, trapezoidal, sigmoidal, Gaussian and quadratic bell-shaped MFs. Keeping in mind the difference in the number of parameters depending on the shape, the equations of MFs used here, reported in Appendix B, are functions of the same set of parameters, in order to compare partitions with same positions but different shapes, and to check differences in the properties depending only on the shape. In order to make this comparison, a 2-variables model is considered, comprising 2 variables, vars = {*UCSh*, *BC*} for example, with 3 fuzzy sets for each variable (therefore, 2 antecedents for each of the 9 rules), with positions, rule consequents and rule weights optimized by the LFA method. Performance, complexity, semantic interpretability and differentia-

Table 3
Results for different shapes of fuzzy sets.

	Binary	Triangular	Trapezoidal	Sigmoidal	Gaussian	Bell-shaped
CE	0.059	0.059	0.056	0.056	0.070	0.060
SCE	0.049	0.042	0.039	0.041	0.054	0.048
Parameters	4	6	8	8	12	12
ANFR	1.00	1.80	1.59	1.57	1.50	1.23
J_D	0	0	0	0	0.01	≈0
J_C	0	0	0	0	0.04	0.24
CI	1.00	1.00	1.00	1.00	0.92	0.46
Proper ordering	Strong	Strong	Strong	Strong	Weak	Weak

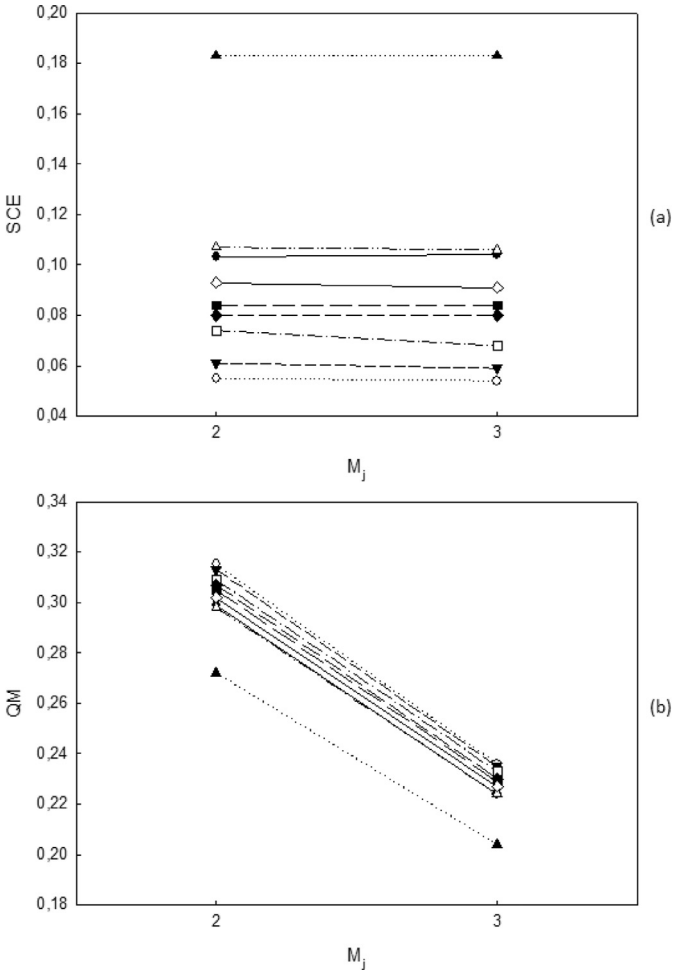

Fig. 6. (a) *SCE* of models comprising each single variable with different cardinalities. (b) Choice of fuzzy sets number for each variable based on *QM*.

bility depend on the chosen shape. *CE*, *SCE*, number of parameters, average number of firing rules *ANFR* (calculated with $\varepsilon_I = 0.01$), distinguishability index J_D, partitions coverage index J_C (with $\varepsilon_C = 0.5$), rules coverage index *CI* ($\varepsilon_R = 0.25$), and the satisfied proper ordering condition are reported for different systems in Table 3, in order to individuate the best shape. Moreover, in Fig. 3, CAFs obtained by using MFs of different shapes are shown.

The performances of different systems depend on how well data trends can be approximated by different shapes. This surely depends on data; however, some general observations can be drawn from results of this example. Firstly, Gaussian and quadratic bell-shaped MFs, with respect to the others, present in this case worse performances; indeed, in general, for these MFs, *SCE* is not ensured to be optimized for multiple variables models by the proposed procedure, since fuzzy sets are not orthogonal, and as a consequence, also *CE* is not optimal. Moreover, binary MFs present in this example *CE* similar to others; however, in general, they are able to individuate interclass separation not differentiable and (due to original variables used) parallel to axes; therefore, they have not general applicability. In addition, binary MFs present in this case high *SCE*, since, in general, CAFs cannot be smooth. Finally, in this case performances of the triangular MFs are worse than those of trapezoids, since, in general, the interclass separation and the CAFs of triangular MFs cannot be as complicate as the trapezoidal case with same cardinality, while requiring higher cardinality and more parameters to reach the same results [19]. Regarding interpretability, Gaussian and quadratic bell-shaped MFs present in this case, with respect to the others, similar complexity, due to a generally higher number of parameters, and lower *ANFR*. However, in general, they present a worse semantic interpretability, due to not optimal J_D, J_C, *CI* and proper ordering. Thus, the best results in terms of performance and interpretability are obtained by trapezoidal and sigmoidal shapes. Between trapezoidal and sigmoidal MFs, numerical results obtained in this case are very similar. However, in general, CAFs obtained by sigmoidal shape are also differentiable, thus presenting general applicability. Therefore, sigmoidal MFs should be chosen for design, to obtain best performance and interpretability, and general applicability.

5.1.3. Procedure application

In the previous section, some inputs were fixed for general purposes. The others, which regard the specific medical application, are supposed here to be fixed as follows. Suppose that, as suggested in case of not too many variables (9 in this case), $M_{\max} = 3$, $n_{\max} = \text{nvar}$ are fixed. Moreover, suppose that the following parameters are chosen for the specific application: regarding the trade-off between performance and number of variables, $q_n = 0.01$, regarding the trade-off between the performance and the number of fuzzy sets for each partition, $q_M = 0.5$, and regarding the trade-off between performance and the loss of interpretability due to the use of rule weights, $q_W = 0.25$. Finally, suppose that the method chosen for optimization is the LFA.

Firstly, the prior probability of each class is calculated, resulting: $P(c_B) = 0.655$ and $P(c_M) = 0.345$.

Starting from a null number of variables $v = 0$, therefore an empty *vars* set, the rule base results:

$$y \text{ is} \begin{cases} 0.34 \ \textit{malignant} \\ 0.66 \ \textit{benign} \end{cases}, \tag{43}$$

the error $SCE_0 = 0.226$ and the index $Qn_0 = \frac{1-SCE_0}{0+1+1/q_n} = 0.00766$.

Now, one variable at a time is considered.

For example, for the first variable (Clump Thickness), the partition in 2 fuzzy sets is performed. Positions and rule consequents are optimized here by the chosen LFA method; however, any other method can be applied, aiming at maximizing the performance of a single-variable classifier; even results of different optimization methods can be compared if desired, as detailed in Section 5.1.4. The sigmoidal fuzzy sets are thus defined, as shown in Fig. 4(a),

Table 4
Performance of single variable models with different numbers of fuzzy sets.

Cardinality	Performance	*CT*	*UCSi*	*UCSh*	*MA*	*SECS*	*BN*	*BC*	*NN*	*Mi*
2	*SCE*	0.102	0.052	0.058	0.105	0.084	0.072	0.080	0.093	0.188
	QM	**0.299**	**0.316**	**0.314**	**0.298**	**0.305**	**0.309**	**0.307**	**0.302**	**0.271**
3	*SCE*	0.103	0.051	0.058	0.104	0.084	0.068	0.080	0.092	0.187
	QM	0.224	0.237	0.236	0.224	0.229	0.233	0.230	0.227	0.203

Table 5
Performance of models with different numbers of variables.

Variables number	Performance	Variables set	*CT*	*UCSi*	*UCSh*	*MA*	*SECS*	*BN*	*BC*	*NN*	*Mi*
1	*SCE*	{} +	0.102	0.052	0.058	0.105	0.084	0.072	0.080	0.093	0.188
	Q_n		0.00880	**0.00929**	0.00924	0.00877	0.00898	0.00909	0.00902	0.00890	0.00796
2	*SCE*	{*UCSi*} +	0.039	–	0.044	0.050	0.050	0.034	0.043	0.050	0.068
	Q_n		0.00932	–	0.00928	0.00923	0.00922	**0.00938**	0.00929	0.00923	0.00905
3	*SCE*	{*UCSi, BN*} +	0.029	–	0.034	0.038	0.033	–	0.035	0.035	0.045
	Q_n		0.00934	–	0.00929	0.00925	0.00930	–	0.00928	0.00928	0.00918

and the rule base is obtained:

$$\begin{cases} \text{if } CT \text{ is } low \text{ then } y \text{ is } \begin{cases} 0.92 \ \ benign \\ 0.08 \ malignant \end{cases} . \\ \text{if } CT \text{ is } high \text{ then } y \text{ is } malignant \end{cases} \quad (44)$$

Then, the partition in 3 fuzzy sets is performed by optimizing positions and rule consequents by the chosen LFA method. The sigmoidal fuzzy sets are thus defined, as shown in Fig. 4(b), and the rule base is obtained:

$$\begin{cases} \text{if } CT \text{ is } low \text{ then } y \text{ is } \begin{cases} 0.93 \ benign \\ 0.07 \ malignant \end{cases} \\ \text{if } CT \text{ is } medium \text{ then } y \text{ is } \begin{cases} 0.78 \ benign \\ 0.22 \ malignant \end{cases} . \\ \text{if } CT \text{ is } high \text{ then } y \text{ is } \begin{cases} 0.03 \ benign \\ 0.97 \ malignant \end{cases} \end{cases} \quad (45)$$

CAFs of the models built with 2 and 3 fuzzy sets are compared in Fig. 5. *SCE* and *QM* can be calculated for both situations: for 2 fuzzy sets, $SCE_{1,1,2} = 0.102$ and $QM_{1,2} = \frac{1-SCE_{1,1,2}}{2-1+1/q_M} = 0.299$, while for 3 fuzzy sets, $SCE_{1,1,3} = 0.103$ and $QM_{1,3} = \frac{1-SCE_{1,1,3}}{3-1+1/q_M} = 0.224$. Since $QM_{1,2} > QM_{1,3}$, then $M_1{}^* = 2$, i.e., the partition made of 2 fuzzy sets is chosen for the first variable.

In Table 4, *SCE* and *QM* values are reported for each single variable, and for 2 or 3 fuzzy sets. All the variables are partitioned in this case in only 2 fuzzy sets, since, for each variable j, the best value (reported in bold) of *QM* is $QM_{j,2}$.

Fig. 6(a) confirms that, for most of the variables, as M_j increases, *SCE* is constant or decreases, i.e., performance is equal or better, as expected. However, due to the chosen value of q_M, *QM* decreases, as clear in Fig. 6(b); as a consequence, $M_j{}^* = 2$ is chosen for all variables.

Once the proper number of fuzzy sets is found for each variable, the fuzzy partitions represented in Fig. 7 are obtained, which are retained for the following steps. The corresponding *SCE* values are reported in the first row of Table 5. The best *SCE* is found for the variable *UCSi*. The corresponding rule base is:

$$\begin{cases} \text{if } UCSi \text{ is } low \text{ then } y \text{ is } \begin{cases} 0.97 \ \ benign \\ 0.03 \ \ malignant \end{cases} \\ \text{if } UCSi \text{ is } high \text{ then } y \text{ is } \begin{cases} 0.04 \ \ benign \\ 0.96 \ \ malignant \end{cases} \end{cases} . \quad (46)$$

The corresponding *SCE* is $SCE_1 = 0.052$, while *Qn* results $Qn_1 = \frac{1-SCE_1}{1+1+1/q_n} = 0.00929$. Since it is greater than Qn_0, the model with 1 variable vars = {*UCSi*} is preferable with respect to that without variables.

Since the maximum number of variables has not been reached, the forward selection can go ahead. Therefore, all the models comprising *UCSi* and any other variable are built: rule consequents and rule weights are found by (37) and (38).

For each combination of 2 variables, comprising *UCSi*, the corresponding values of *SCE* and *Qn* are reported in Table 5. The model presenting the best *SCE* value is chosen, which comprises the variables vars = {*UCSi*, *BN*}. The corresponding *Qn* value is compared with the best one with 1 variable: since *Qn* increases, then the model with 2 variables is preferred.

The same procedure is done for more than 2 variables, and the results are reported in Table 5. The *Qn* value does not increase by using 3 variables, with respect to 2 variables, even if *SCE* decreases in some cases.

In Fig. 8(a), it is shown that for each n the choice of the best model is based on the lower *SCE*. Moreover, it results clear that, as the number of variables comprised in the model increases, *SCE* value decreases, i.e., a better performance is obtained. However, due to the chosen q_n value, a maximum of *Qn* is obtained in correspondence of $n = 2$, as clear in Fig. 8(b); as a consequence, a model comprising 2 variables is chosen.

For the best set of variables, *SCE*of both weighted and non-weighted rule bases are compared. The weighted or the non-weighted model is chosen, based on *QW*.

In this case, for the couple of variables vars = {*UCSi*, *BN*}, the following rule bases are compared:

$$\begin{cases} (0.31\ ) \text{if } UCSi \text{ is } low \text{and } CT \text{ is } low \text{ then } y \text{ is } \begin{cases} 0.99 \ \ benign \\ 0.01 \ \ malignant \end{cases} \\ (0.04) \text{ if } UCSi \text{ is } low \text{ and } CT \text{ is } high \text{ then } y \text{ is } \begin{cases} 0.49 \ \ benign \\ 0.51 \ \ malignant \end{cases} \\ (0.06) \text{ if } UCSi \text{ is } high \text{ and } CT \text{ is } low \text{ then } y \text{ is } \begin{cases} 0.28 \ \ benign \\ 0.72 \ \ malignant \end{cases} \\ (0.59) \text{ if } UCSi \text{ is } high \text{ and } CT \text{ is } high \text{ then } y \text{ is } malignant \end{cases} \quad (47)$$

and

$$\begin{cases} \text{if } UCSi \text{ is } low \text{ and } CT \text{ is } low \text{ then } y \text{ is } \begin{cases} 0.99 \ benign \\ 0.01 \ malignant \end{cases} \\ \text{if } UCSi \text{ is } low \text{ and } CT \text{ is } high \text{ then } y \text{ is } \begin{cases} 0.49 \ \ benign \\ 0.51 \ \ malignant \end{cases} . \\ \text{if } UCSi \text{ is } high \text{ and } CT \text{ is } low \text{ then } y \text{ is } \begin{cases} 0.28 \ \ benign \\ 0.72 \ \ malignant \end{cases} \\ \text{if } UCSi \text{ is } high \text{ and } CT \text{ is } high \text{ then } y \text{ is } malignant \end{cases} \quad (48)$$

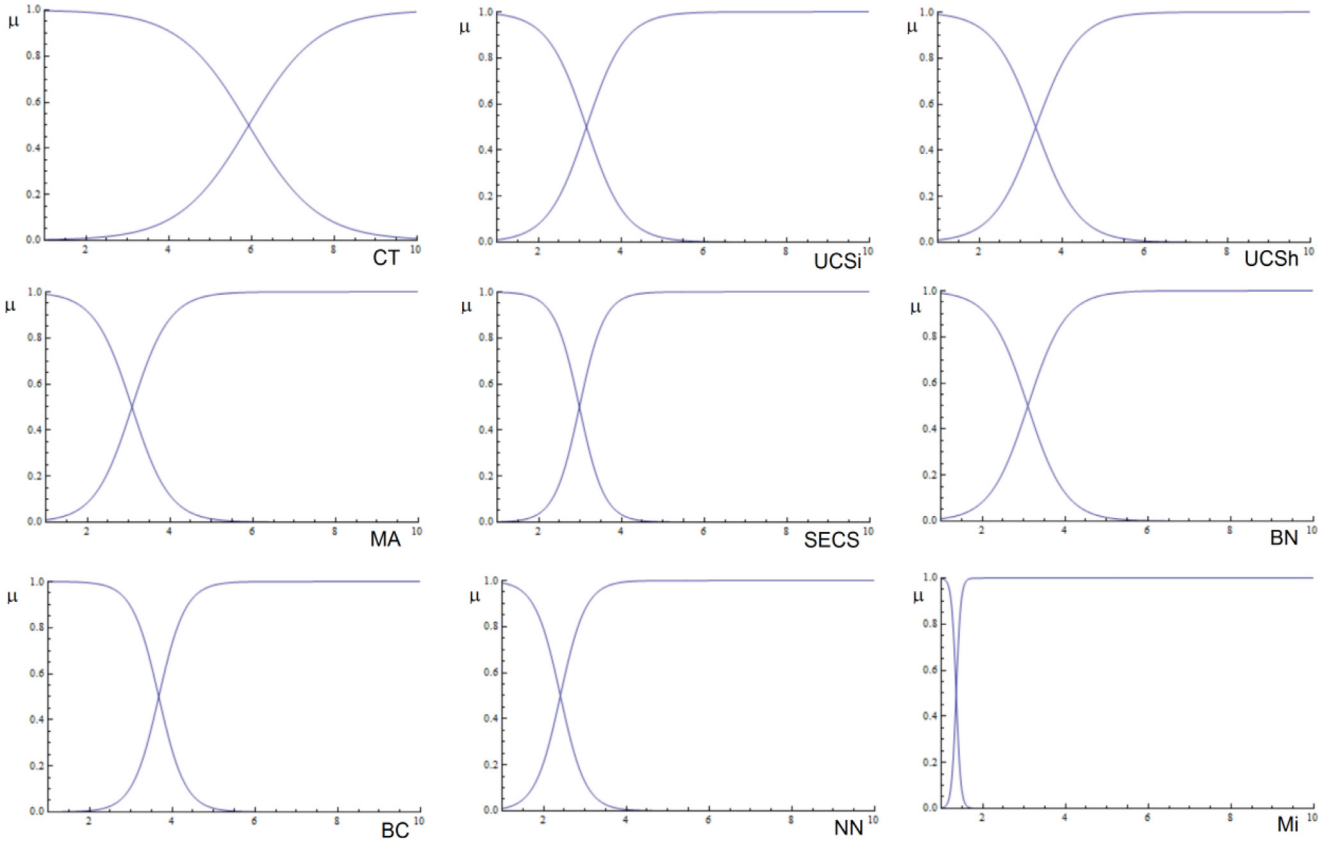

Fig. 7. Fuzzy partitions of all variables.

The *SCE* values in different cases result respectively: $SCE_W = 0.034$ and $SCE_{NW} = 0.041$: as expected, weighted model has higher performance. The corresponding *QW* values are $QW_W = (1 - q_W)(1 - SCE_W) = 0.724$, and $QW_{NW} = 1 - SCE_{NW} = 0.959$. As a consequence, the non-weighted model is chosen.

Finally, the best model, according to the chosen values of trade-off parameters, is found, made of 2 variables, vars $= \{UCSi, BN\}$, each partitioned into 2 fuzzy sets, without rule weights. The corresponding fuzzy partitions are represented in Fig. 7, and the rule base corresponds to (48).

The resulting model has characteristics which are particularly appealing for medical field applications. Firstly, it should be underlined that this type of model allows to obtain confidence-based results. Moreover, it can be noticed that the model (48) is more than transparent, it is also simple, and highly interpretable, since it formalizes concepts very similar to the human reasoning, making use of well-defined linguistic terms (reported in Fig. 7). A model like this represents high-level knowledge, therefore, it allows to easily understand relations between features and classes, enhancing the domain knowledge if properly endorsed by experts, and to give explanations to each classified case. For example, in order to show how clear feature-classes relations are, it is clear from the model (48) that as *UCSi* increases, the probability of malignant class increases, and the same for *BN*. Moreover, in order to show the interpretability of the inference process, suppose that the following values are measured for a sample to be classified: $UCSi = 6$ and $BN = 8$; from Fig. 7, it can be recognized that in this case "*UCSi* is *high*" and "*BN* is *high*"; this conditions are those of the forth rule, whose consequent is "*y* is *malignant*"; therefore, the results of the system can be in this case directly inferred from the model without calculation. However, different models can be found, still transparent, but with different complexity, associated with the number of variables and number of linguistic terms, and/or a little more difficult semantics, associated with the presence of rule weights, depending on the desired trade-off between performance and interpretability.

5.1.4. Comparison of optimization methods

In order to show an example of a comparison among different optimization methods, which can be performed by the developer for choosing the best one, for this application, LFA and NN optimization methods are compared, thus the fuzzy sets positions, rule consequents and rule weights of a model are found by using both of them.

Both methods were applied on the WBCD dataset. In particular, 2 variables were used, i.e., *UCSh* and *BC*, since they do not present missing values, which would be a problem for NN optimization, and each variable was partitioned into 2 or 3 fuzzy sets. Here, it is explained how the fuzzy sets positions and rules consequents of single variables models can be optimized using both methods. Then, these parameters can be used to build any model by (37) and (38).

The LFA method for model construction from data was recently developed by authors [42], and named "Likelihood-Fuzzy Analysis". In a first step, the likelihood functions are calculated from the dataset in a pointwise manner, by using the kernel functions method to calculate the univariate probability distributions $P(x^{(j)}|c_k)$ and then the Bayes theorem to get the class posterior probabilities $P(c_k|x^{(j)})$, in correspondence of N^* points $x^{(j)}$ ($N^* = 100$ in the following). Then, each range of the original variables is partitioned into a collection of M_j fuzzy sets described by MFs $\mu_{m_j}^{(j)}(x^{(j)})$, $m_j = 1, \ldots, M_j$, such that the kth class posterior probabilities are approximated by a linear combination of these MFs by

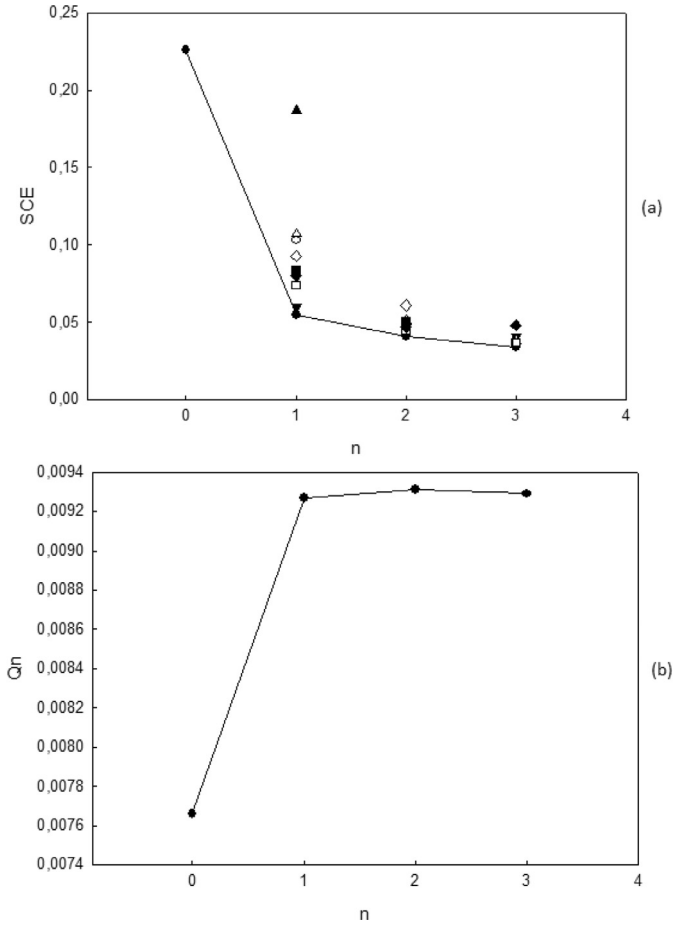

Fig. 8. (a) Forward selection of variables based on *SCE* of models. (b) Choice of the number of variables based on *Qn*.

means of the coefficients $\lambda_{m_j-k}^{(j)}$:

$$P\left(c_k|x^{(j)}\right) \cong \sum_{m_j=1}^{M_j} \left(\lambda_{m_j-k}^{(j)} \cdot \mu_{m_j}^{(j)}\left(x^{(j)}\right)\right), \tag{49}$$

with $k = 1, \ldots, K$ and $j = 1, \ldots, n$. Given the number of fuzzy sets, the coefficients $\lambda_{m_j-k}^{(j)}$ and the MFs parameters are optimized to satisfy (49). The consequents of the single variable rule base are fixed equal to the coefficients:

$$C_{m_j-k}^{(j)} = \lambda_{m_j-k}^{(j)}. \tag{50}$$

In order to obtain a NN for optimization, the fuzzy classifier can be represented as shown in Fig. 9. In this neural network representation, the inputs are the values of the input variables. In the first layer, the input values are transformed into membership grades of the fuzzy sets; in particular, each input $x^{(j)}$ is processed by a number of nodes equal to the cardinality of the associated variable, and in each node it is transformed into the grade of membership $\mu_{m_j}^{(j)}(x^{(j)})$ to the relative fuzzy sets (for continuous variables) or singleton (for categorical variables). Membership grades can be weighted by antecedent weights, and then feed the layer of firing strength of rules; in this layer, each of the R nodes corresponds to a rule, and the outputs of the previous layer, eventually weighted, corresponding to the membership grades to the fuzzy sets constituting the antecedents of the ρth rule, are multiplied, to calculate rule firing strength values $FS_\rho(\mathbf{x})$ (Eq. (10) with product T-norm). The firing strengths feed the implication layer, where each node corresponds again to a rule, and is fed by the output of the corresponding node; here, the corresponding firing strengths and fuzzy consequents are multiplied to obtain rule implication values $I\hat{M}P_\rho(\mathbf{x})$ (Eq. (11) with product T-norm). Fuzzy implications of rules can be multiplied by rule weights, and then all of them feed the only node constituting the aggregation layer; here, rules implications, eventually weighted, are summed up to calculate the aggregation result $A\hat{G}G(\mathbf{x})$ (Eq. (12) with Łukasiewicz S-norm). Finally, the resulting $AGG_k(\mathbf{x})$ values feed the last node constituting the defuzzification layer, where they are defuzzified in one node where a normalization is performed (Eq. (20)). The outputs of the last node are the resulting activation grades of classes.

Fuzzy sets positions, antecedent weights, fuzzy consequents and rule weights of a combinatorial model are the parameters that should be optimized at the same time.

Since the proposed procedure implies to optimize parameters relative to different variables separately, and to not use antecedent weights, and since for single variable models the rule weights are not needed, the NN structure can be simplified, as shown in Fig. 10.

In the simplified case, the parameters representing the fuzzy partitions and the consequents of single variable models are optimized, separately for each variable.

The error of the output nodes is computed as the *SCE* of the classifier. Each parameter π is updated in each epoch τ by applying:

$$\pi(\tau+1) = \pi(\tau) - \eta\frac{\partial SCE}{\partial \pi}, \tag{51}$$

where the learning rate is set to $\eta = 1$ during the first 10 epochs, then $\eta = 0.1$, since this results effective, and the partial derivatives are computed numerically by the previous epochs. Starting values are fixed in correspondence of the equally distributed positions and their small modifications. Convergence is assumed when *SCE* variation in successive epochs is less than 0.0001. The algorithm needed to converge respectively, for *UCSh* and *BC* variables, 167 and 187 epochs for $M_j = 2$, and 94 and 80 epochs for $M_j = 3$.

In the following, the LFA and NN optimization methods are compared in terms of results and of computation time.

Optimal settings tuned by LFA and NN are reported in Table 6. The corresponding performance values are calculated for both single variable models and for the model comprising both variables, obtained by applying (37) and (38), and for cardinalities equal to 2 and 3. CAFs are shown in Fig. 11.

From Table 6, it can be evinced that for this application the LFA method results better than NN, since the best performances are gained by using the LFA method for all the models. Therefore, in this case, the positions of fuzzy sets and the fuzzy rule consequents found by the LFA method should be preferred for both the single variable models, with $M_j = 2$ and $M_j = 3$.

However, in general, as different optimization methods are compared on single variable models, for each variable and each cardinality, the optimized parameters corresponding to the best performance value could be chosen.

LFA method required less computation time with respect to NN. In fact, with 4 GHz processor and 16 Gb RAM, it needed about 10 s in total, for calculating optimized parameters from data, while NN required in total about 107 s. Therefore, LFA optimization resulted in this case much faster than NN optimization method. The rest of the procedure required less than 1 s to output the final model.

If a designer chooses to use different methods to optimize single variable models and compare them, the computational time for optimization would be simply the sum of times required for each method, plus a little time for errors comparison. On the other hand, the computation time of the rest of the procedure does not depend on the type or number of optimization methods, as can be evinced by (40). In fact, as described before, after optimization

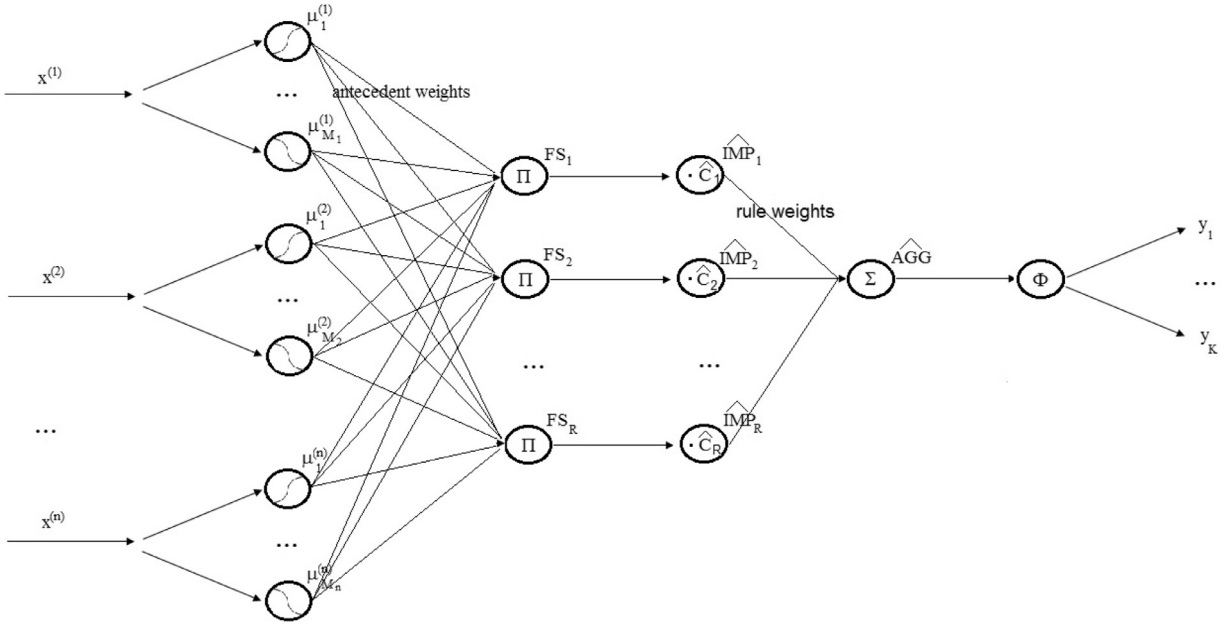

Fig. 9. Neural network representation of a fuzzy classifier.

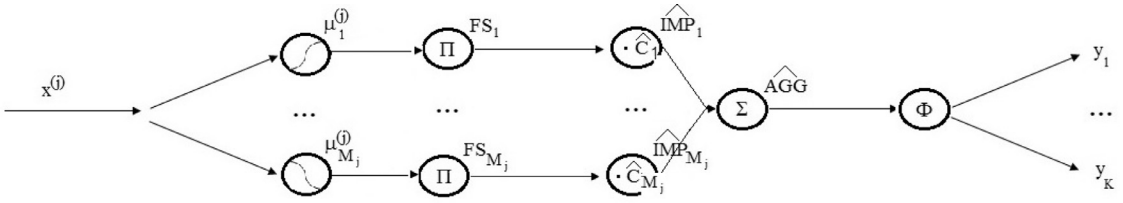

Fig. 10. Neural network representation of the proposed single variable model.

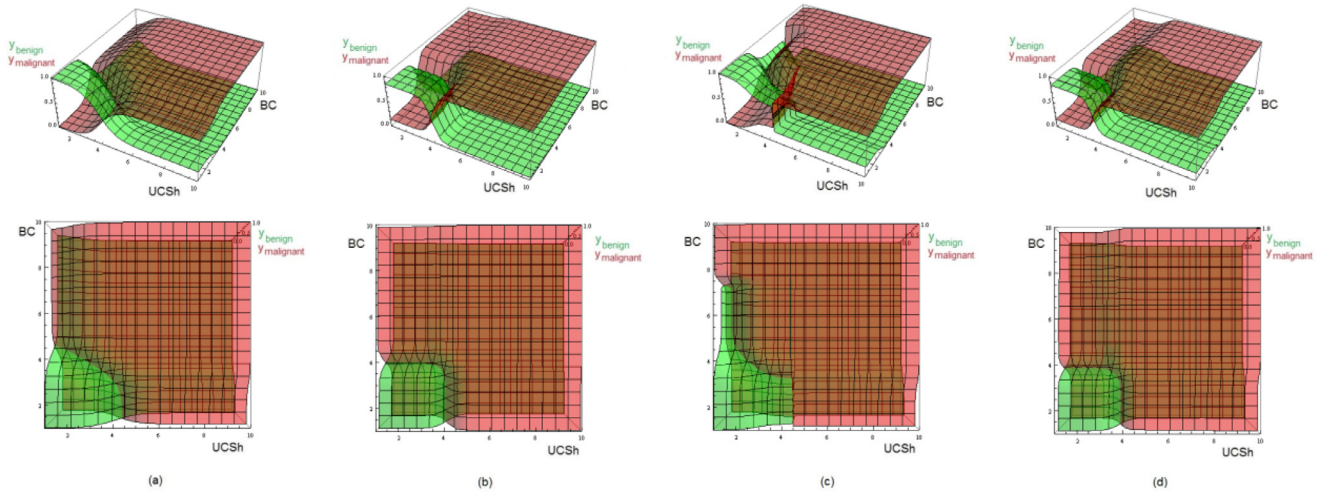

Fig. 11. Class activation functions of models with (a) $M_j = 2$, positions, fuzzy consequents and rule weights optimized by LFA method, (b) $M_j = 2$, fuzzy consequents and rule weights optimized by neural networks, (c) $M_j = 3$, positions, fuzzy consequents and rule weights optimized by LFA method, and (d) $M_j = 3$, fuzzy consequents and rule weights optimized by neural networks.

made for each variable and each cardinality, the best cardinality for each variable is chosen based on *QM*, the parameters of multiple variables models are calculated by (37) and (38), and the best model is chosen based on *Qn* and *QW*; therefore, no further optimization step is needed.

5.2. Comparison of the proposed approach results with state-of-the-art

In this section, the results obtained by means of the proposed approach are compared with other well-established approaches. Firstly, in Section 5.2.1, different runs of the proposed procedure are performed on WBCD data, and compared with the best fuzzy systems found in literature, applied on the same dataset, in terms

Table 6
Results of different optimization methods.

			$M_j = 2$		$M_j = 3$	
			LFA	NN	LFA	NN
UCSh	Position parameters		1.00, 5.73	2.91, 4.92	1.00, 4.24, 4.33, 4.42	2.73, 4.55, 6.29, 8.07
	Fuzzy consequents	$C^{(UCSh)}_{low-benign}$	0.97	0.83	0.99	0.82
		$C^{(UCSh)}_{low-malignant}$	0.03	0.17	0.01	0.18
		$C^{(UCSh)}_{medium-benign}$	–	–	0.59	0.16
		$C^{(UCSh)}_{medium-malignant}$	–	–	0.61	0.84
		$C^{(UCSh)}_{high-benign}$	0.06	0.04	0.06	0.13
		$C^{(UCSh)}_{high-malignant}$	0.94	0.96	0.94	0.87
	SCE		0.058	0.075	0.058	0.077
BC	Position parameters		2.20, 5.15	2.87, 4.82	2.30, 4.97, 7.22, 7.90	2.79, 4.48, 6.22, 8.05
	Fuzzy consequents	$C^{(BC)}_{low-benign}$	0.94	0.82	0.93	0.81
		$C^{(BC)}_{low-malignant}$	0.06	0.18	0.07	0.19
		$C^{(BC)}_{medium-benign}$	–	–	0.10	0.14
		$C^{(BC)}_{medium-malignant}$	–	–	0.90	0.86
		$C^{(BC)}_{high-benign}$	0.07	0.05	0.00	0.17
		$C^{(BC)}_{high-malignant}$	0.93	0.95	1.00	0.83
	SCE		0.080	0.089	0.080	0.092
{*UCSh, BC*}	*SCE*		0.041	0.058	0.041	0.059

of performance and interpretability. In Section 5.2.2, instead, the procedure with fixed values of the trade-off parameters is applied on a number of benchmark datasets, and the resulting performances are compared with a number of selected classification methods.

5.2.1. Comparison of performance and interpretability with the best fuzzy systems applied on WBCD

The fuzzy systems obtained by applying the proposed design procedure on WBCD data, fixing different values of the trade-off parameters, are compared here with existing fuzzy systems, built on the same data.

The systems chosen as terms of comparison are not very recent, but to the best of our knowledge, the best fuzzy classifiers in literature previously applied on WBCD in terms of performance and/or interpretability. Each of the systems is designed by the respective procedure to fix all the fuzzy system aspects and degrees of freedom, tailored for the particular application. On the other hand, different runs of the proposed procedure are aimed at maximizing performance, or interpretability, or achieving a compromise between them, without particular assumptions about data. The final results of other procedures and of the proposed one with different settings are regarded, in order to compare the whole design procedures.

The first results used for comparison were obtained in [39]. That work combined fuzzy systems and genetic algorithms, in order to obtain high performance and interpretability at the same time, by maximizing the following fitness (F) function:

$$F = (1 - CE) - \alpha \cdot ANT - \beta \cdot SCE, \tag{52}$$

where α and β were fixed empirically. The authors used min/max norms, and simple rule consequents, nevertheless confidence measures were obtained by using a weighted rule base (with fixed rule weights, without explaining how these were obtained) and fuzzy mean defuzzification. Authors applied that procedure to WBCD dataset, and obtained a system with the best performance among rule-based approaches, the possibility of attributing a confidence measure to the output, and involving a few simple rules. This method allows choosing the maximum number of rules, and among different results, the one with the highest accuracy is reported here. However, the authors do not take the number of variables into account. Moreover, the rule base completeness, and thus the system interpretability, is debatable, since a rule is added, apparently introduced by the "else" connective, but whose firing strength does not depend on the data values, while the other rules are fired only in the *benign* case. Furthermore, the use of empirical parameters means that the approach is not universally applicable. Finally, even if the authors claim that they obtain a confidence measure of the output, it is not evaluated by measuring *SCE*.

The second work used here for comparing results is [8], where authors, by means of a neural network approach, maximized performance of a fuzzy SC. The greatest part of the degrees of freedom of the fuzzy system were optimized. In particular, antecedent weights, simple consequents, rule weights, and parametric norms were used. When applied to WBCD dataset, the results of their approach were better than those of many methods used by other authors and reported for comparison, including Fisher Linear Discriminant Analysis and naïve Bayes approaches. However, authors did not take interpretability nor confidence measure into account.

Another work [38] obtained good performance by applying a fuzzy-genetic approach. The fitness function coincided with the accuracy. Attention was also paid to interpretability, by fixing some constraints. No rule weights and simple rule consequents were used. However, this method only allowed tuning previously provided crisp rules. Moreover, confidence measure was not taken into account.

Regarding the approach proposed here, five different versions of the algorithm are run for this comparison: the first (Setting 1), giving only emphasis to interpretability, is obtained by fixing $q_n = 1$, $q_M = 1$, and $q_W = 1$; the second (Setting 2), which admits a higher number of variables, corresponds to parameters $q_n = 0.01$, $q_M = 1$, and $q_W = 1$; the third (Setting 3) goes towards performance by relaxing the trade-off parameter regarding the number of fuzzy sets, and is obtained by fixing $q_n = 0.01$, $q_M = 0.001$, and $q_W = 1$; the fourth (Setting 4), admitting the use of rule weights, is obtained by fixing $q_n = 0.01$, $q_M = 0.001$, and $q_W = 0.001$; the fifth (Setting 5), completely devoted to performance, corresponds to parameters $q_n = 10^{-6}$, $q_M = 10^{-6}$, and $q_W = 0$. Based on our experience on this dataset, these settings are reported as representative, however, for each application, the user should try different runs of each trade-

off parameter, to find the desired results. Developers are suggested to tune parameters by starting from Setting 1 and then diminishing firstly q_n, then q_M, and finally q_W.

None of the above-mentioned approaches (nor the greatest part of the others published on the same dataset) reports the *SCE* of classification, therefore, the classification confidence cannot be compared. However, in our experiments, low *SCE* values were obtained (e.g., $SCE = 0.021$ in the last run), revealing a good measure of classification confidence of the systems obtained here.

The comparison of the approach presented here with the mentioned ones is made by considering accuracy $acc = (1 - CE)$, sensitivity and specificity. In Table 7, the results are reported, together with a comparison of different settings of all the degrees of freedom (empty fields are not reported in the respective papers). All performance values are obtained on the dataset without samples comprising missing values, and without a proper 10-fold cross-validation, in order to make a comparison with the other results, obtained in the same manner.

The comparison reported in Table 7 shows that, by means of the proposed procedure, different systems can be extracted from the same data. If all the trade-off parameters are set equal to 1 (Setting 1), then a very simple system is obtained: in this case, it is constituted by only 1 rule without variables, assigning prior probabilities to classes (33). If parameter q_n is set to a low value (Setting 2), then the obtained system is made of 2 variables, each partitioned into only 2 fuzzy sets, and 4 non-weighted rules with 2 antecedents; the complexity is still low, but performance is lower than other systems found in previous works. If both parameters q_n and q_M are set to low values (Setting 3), then the variables are partitioned into 2 or 3 fuzzy sets, thus there are 6 non-weighted rules with 2 antecedents; the complexity increases, and performance increases as well. In case also the parameter q_W is set to a low value (Setting 4), then the same previous system is obtained, but with weighted rules; semantic interpretability is not maximized, but performance presents a good improvement. In the last run, values of parameters near to 0 (Setting 5) imply to build a system with very high complexity, but allow obtaining a very high performance, at the highest level of fuzzy systems of the state-of-the-art.

From the results obtained here, it can be seen that, for a range of values of the trade-off parameters, the proposed procedure allows constructing a fuzzy system more interpretable than all the previous ones, given the low complexity (low number of variables, fuzzy sets, antecedents and rules) and the good semantic interpretability (no antecedent weights, fuzzy consequents for confidence-weighted results, optional rule weights, complete rule base). On the other hand, for very low trade-off values, keeping transparency and good semantic interpretability, complexity increases, but a performance can be reached which is better than the best existing fuzzy systems. Different runs of the procedure can let the developer individuate the desired system with respect to interpretability and performance.

Universality (of possible applications, even if tailored for medical field) and adjustability (of performance vs. complexity), coupled with high semantic interpretability, and a good measure of classification uncertainty, have been shown as the best qualities of the proposed approach.

5.2.2. Comparison of performance with the best classification methods applied on different datasets

Even if the performance is not the only goal of a classifier for medical applications, and even if it was shown that our systems are among the best in terms of interpretability and measure of classification uncertainty, in this section, the effectiveness of the proposed approach is evaluated, in designing, if desired, very high performance classifiers. Therefore, benchmark datasets are used to compare performance-oriented results of the approach with the results of representative known methods to build high performance classifiers.

Six different datasets are considered, which are benchmark cases, widely used for comparison, all available in UCI Machine Learning Repository [20]. A brief description of each dataset is given in Table 8.

The wide use of these examples for machine learning methods testing is due to the relatively large *N* and modest *nvar*. However, the sixth (and most recent) dataset regarding Kidney disease presents a larger *nvar*, without compromising the application of the proposed procedure. Some works regarding the application to new medical data requiring knowledge extraction, with particularly small *N* and even larger *nvar*, are currently under study and are reaching promising results.

For each dataset, the proposed procedure was applied, by fixing trade-off parameters to low values ($q_n = 10^{-6}$, $q_M = 10^{-6}$, and $q_W = 0$), in order to design a classifier giving priority to performance.

On the same datasets, a number of well-established methods were applied, in order to compare results of respective classifiers with those obtained by the proposed approach. In particular, the following methods were chosen, since they are available in the Waikato Environment for Knowledge Analysis (WEKA 3.7.5) [60], and they are the most representative of different categories: among logical/symbolic techniques, One-R rule-based classifier [61], and C4.5 decision tree [62]; among statistical learning algorithms, Naïve Bayes (NB) [63]; among instance-based learning, K-nearest neighbour (K-nn) [64], and Support Vector Machine (SVM) [65]; finally, among perceptron-based techniques, multilayer artificial Neural Network (NN) [66]. Each method was run with the default setting of WEKA.

It should be noticed that all the methods used for comparison produce classifiers that are not interpretable, except One-R, C4.5, and the proposed method, which produce rule-based classifiers. Among them, the One-R model has surely the lowest complexity, but is expected to reach the lowest accuracy.

Moreover, while the other classifiers aim at maximizing accuracy, the proposed method is tailored for optimizing the measure of classification uncertainty, and, as a consequence, the accuracy.

All the results presented here are obtained by applying a 10-fold stratified cross-validation, on the whole datasets comprising missing values.

In Table 9, results are shown in terms of the achieved accuracy, averaged over the 10 folds. For each dataset, the best accuracy is shown in bold. In parentheses, the accuracy standard deviation over 10 folds is given for the proposed procedure. Moreover, average accuracy, number of variables and ranking over all the datasets is calculated. Based on the average accuracy, the classifiers are given a ranking order regarding performance.

From results shown in Table 9, it can be evidenced that, for all the considered datasets, the proposed approach allows designing a classifier that presents among the best performance values in terms of classification accuracy. For three of the six datasets, the accuracy is the best among all the classifiers. The Friedman test [67,68], reveals that there are significant differences among all classifiers, with $p < 0.01$. In particular, a post-hoc test (Nemenyi test [69]), to find, among all couples, which couples of classifiers actually present statistically different results, detects significant difference (with $p < 0.05$) when the proposed approach (the best one) is compared with 1-nn and One-R methods (the worst two), and when SVM is compared with 1-nn, while no significant information is detected about the difference between the classifiers of all the other couples. The same conclusions can be drawn if the proposed approach is compared to the other methods, by Bonferroni–Dunn test [70] or Holm's step-down procedure [71]. Moreover, the resulting best average ranking and first position in

Table 7
Comparison of settings and performance of the best fuzzy systems of the state-of-the-art and this paper on WBCD.

Main aspect	Degree of freedom	[39]	[8]	[38]	This work – setting 1 ($q_n = 1$ $q_M = 1$ $q_W = 1$)	This work – setting 2 ($q_n = 0.01$ $q_M = 1$ $q_W = 1$)	This work – setting 3 ($q_n = 0.01$ $q_M = 0.001$ $q_W = 1$)	This work – setting 4 ($q_n = 0.01$ $q_M = 0.001$ $q_W = 0.001$)	This work – setting 5 ($q_n = 10^{-6}$ $q_M = 10^{-6}$ $q_W = 0$)
Variables	Number	9	9	–	0	2	2	2	6
	Type	Original	Original	Original	Original	Original	Original	Original	Original
	Selection method	GA	NN	GA	Forward selection	Forward selection	Forward selection	Forward selection	Forward selection
Fuzzy sets	Average number	2	2	–	0	2	2.5	2.5	2.7
	Shape	Trapezoid	Gaussian	Trapezoid	–	Sigmoid	Sigmoid	Sigmoid	Sigmoid
	Method to optimize positions	GA	NN	GA	–	LFA	LFA	LFA	LFA
Rules	Antecedents weights	No	Yes	No	No	No	No	No	No
	Antecedents average number	3	9	–	0	2	2	2	6
	Consequents type (optimization method)	Simple (GA)	Simple (NN)	Simple (GA)	Fuzzy (LFA)	Fuzzy (LFA)	Fuzzy (LFA)	Fuzzy (LFA)	Fuzzy (LFA)
	Rule weights (optimization method)	Yes (fixed)	Yes (NN)	No	No	No	No	Yes (LFA)	Yes (LFA)
	Rules number	4	512	9	1	4	6	6	324
	Rule base completeness	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Rules reduction / selection	Yes	No	Yes	No	No	No	No	No
Inference	(T-) / (S-) Norms type	Min/max	Parametric	Łukasiewicz/ Łukasiewicz	Product/ Łukasiewicz	Product/ Łukasiewicz	Product/ Łukasiewicz	Product/ Łukasiewicz	Product/ Łukasiewicz
	Defuzzification type	Fuzzy mean	Fuzzy mean	Fuzzy mean	Fuzzy mean	Fuzzy mean	Fuzzy mean	Fuzzy mean	Fuzzy Mean
Performance	Accuracy (%)	97.8	96.6	97.6	65.0	95.6	95.8	96.6	98.0
	Sensitivity	0.99	–	–	0.00	0.93	0.92	0.96	0.99
	Specificity	0.97	–	–	1.00	0.97	0.98	0.97	0.97

Table 8
Benchmark datasets for system performance evaluation.

Dataset	Description	Samples	Variables	Classes
Cancer	Breast cancer cases from University of Wisconsin at Madison Hospital.	699	9	2
Haberman's survival	Cases on the survival of patients who had undergone surgery for breast cancer from University of Chicago's Billings Hospital.	306	3	2
Heart	Patients with heart disease originally obtained from Cleveland Clinic Foundation.	303	13	2
Indian liver	Liver patient records, collected from north east of Andhra Pradesh, India.	583	10	2
Pima diabetes	Patients with signs of diabetes according to the World Health Organization criteria obtained from National Institute of Diabetics and Digestive and Kidney Disease.	768	8	2
Kidney disease	Early stage of Indians Chronic Kidney Disease	400	24	2

Table 9
Accuracy (%) (and its standard deviation over folds) obtained on benchmark datasets, average accuracy, number of variables and ranking over all the datasets, and ranking based on the average accuracy, obtained by different methods.

	Cancer	Haberman's survival	Heart	Indian liver	Pima diabetes	Kidney disease	Average accuracy	Average n	Average ranking	Ranking
One-R	92.70	74.84	72.61	65.87	72.79	92.50	78.55	1.0	5.75	6
C4.5	94.56	71.90	80.20	68.78	73.83	99.00	81.38	6.0	4.67	4
NB	95.99	74.84	84.49	55.75	76.30	95.00	80.39	11.2	3.92	5
1-nn	95.14	67.65	77.89	64.15	70.18	95.75	78.46	11.2	6.00	7
SVM	97.00	73.53	**84.82**	**71.35**	**77.34**	97.75	83.63	11.2	2.25	2
NN	95.28	72.88	80.53	68.27	75.39	**99.75**	82.02	11.2	3.67	3
This work	**97.57**	**76.14**	82.84	71.01	**77.34**	99.50	84.07	4.7	1.75	1
	(1.65)	(3.55)	(7.98)	(1.74)	(4.40)	(1.06)				

the ranking of average performance clarify that the proposed classification method is among the best ones so far established in literature. In addition, even if the trade-off parameters are set with the aim of obtaining a good performance, in many cases very simple models are obtained, involving few variables. At average, the proposed classifiers involve the lowest number of variables, among the other methods, apart from One-R classifier.

6. Conclusion

A novel procedure for designing rule-based fuzzy systems was proposed in this paper, particularly devoted to the data-driven knowledge extraction for classifying medical data. All degrees of freedom which characterize the modelling process were analysed, with the aim of individuating the proper choices of the developer, to obtain a confidence-weighted classifier, and improving the classification performance and the interpretability of the system.

Some of the degrees of freedom are suggested to be fixed, based on general considerations regarding the use of such systems in the medical ambit, such as the need of interpretability and of the measure of output confidence. Other choices depend on particular requirements of the application, in terms of performance and interpretability. Therefore, some parameters were introduced to define the required trade-off between performance and interpretability. These parameters can be used, as suggested by the procedure, for individuating the associated design choices. Finally, the remaining degrees of freedom can be found by using the preferred optimization method; however, the Likelihood-Fuzzy Analysis method results efficient, with respect to genetic algorithms and neural networks, for optimizing fuzzy classifiers.

The proposed procedure is based on the application of naïve Bayes hypothesis to fuzzy systems, and as a consequence, it results much simpler, compared with a full optimization of a rule-based fuzzy system, since it involves a number of parameters linear in the number of variables. Moreover, an asymptotical analysis reveals that the procedure presents in the worst case linear time solvability, therefore it is computationally efficient.

Different applications of the procedure, aiming more at interpretability or performance, are exemplified by using a benchmark dataset regarding the classification of breast cancer patients. The results show that, depending on the choice of the trade-off parameters, a system can be obtained, with optimal interpretability and performance slightly lower than other systems, or with suboptimal but still good interpretability and high performance. Moreover, the design procedure aiming at performance, applied on different datasets, produce respective classifiers with accuracy at the highest level of the state-of-the-art.

Summarizing, the best qualities of the proposed design procedure are:

- Universality of possible applications, even if the approach is tailored for medical field.
- Good measure of classification uncertainty.
- Scalability and computational efficiency.
- Adjustable high to optimal semantic interpretability.
- Adjustability of performance vs. complexity, which allows to build classifiers with very high performance, or the lowest possible complexity, or the desired middle way solution.

Appendix A

This section reports the proof that, given the following hypotheses:

- positions of fuzzy sets $\hat{F}_{m_j}^{(j)}$ and consequents $C_{m_j-k}^{(j)}$, $\forall m_j \in \{1,\dots,M_j\}$ and $\forall k \in \{1,\dots,K\}$, of rule bases (35), with $j = 1,\dots,n$, are optimized with respect to *SCE*,
- product T-norms, Łukasiewicz S-norm and fuzzy mean defuzzification are used for inference,
- normal fuzzy sets are used, i.e., $0 \le \mu_{m_j}^{(j)}(x^{(j)}) \le 1$, $\forall m_j \in \{1,\dots,M_j\}$ and $\forall j \in \{1,\dots,n\}$,

- orthogonal fuzzy sets are used, i.e., $\sum_{m_j=1}^{M_j} \mu_{m_j}^{(j)}(x^{(j)}) = 1,\ \forall j \in \{1,\dots,n\}$,
- the naïve Bayes hypothesis approximately holds, i.e., $p(\mathbf{x}|c_k) \cong \prod_{j=1}^{n} p(x^{(j)}|c_k),\ \forall k \in \{1,\dots,K\}$,
- rule weights and consequents of (36) are calculated by (37)–(39),

then the output of the rule base (36) results optimized with respect to *SCE*.

Firstly, calculate the output of a single variable model in the form (35) by substituting (10), (15) and (16) in (20).

$$\begin{aligned} y_k\left(x^{(j)}\right) &= \frac{AGG_k\left(x^{(j)}\right)}{\sum_{\kappa=1}^{K} AGG_\kappa\left(x^{(j)}\right)} \\ &= \frac{\min\left\{1, \sum_{m_j=1}^{M_j}\left(IMP_{m_j-k}\left(x^{(j)}\right)\right)\right\}}{\sum_{\kappa=1}^{K}\min\left\{1, \sum_{m_j=1}^{M_j}\left(IMP_{m_j-\kappa}\left(x^{(j)}\right)\right)\right\}} \\ &= \frac{\min\left\{1, \sum_{m_j=1}^{M_j}\left(FS_{m_j}\left(x^{(j)}\right)\cdot C_{m_j-k}^{(j)}\right)\right\}}{\sum_{\kappa=1}^{K}\min\left\{1, \sum_{m_j=1}^{M_j}\left(FS_{m_j}\left(x^{(j)}\right)\cdot C_{m_j-\kappa}^{(j)}\right)\right\}} \\ &= \frac{\min\left\{1, \sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-k}^{(j)}\right)\right\}}{\sum_{\kappa=1}^{K}\min\left\{1, \sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-\kappa}^{(j)}\right)\right\}}. \end{aligned} \tag{A.1}$$

Since normal and orthogonal fuzzy sets are used, and consequents are $0 \le C_{m_j-k}^{(j)} \le 1$, then

$$y_k\left(x^{(j)}\right) = \frac{\sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-k}^{(j)}\right)}{\sum_{\kappa=1}^{K}\sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-\kappa}^{(j)}\right)}. \tag{A.2}$$

Since orthogonal fuzzy sets are used, and $\sum_{k=1}^{K} C_{m_j-k}^{(j)} = 1$, then, with some algebra:

$$\begin{aligned} y_k\left(x^{(j)}\right) &= \frac{\sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-k}^{(j)}\right)}{\sum_{m_j=1}^{M_j}\sum_{\kappa=1}^{K}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-\kappa}^{(j)}\right)} \\ &= \frac{\sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-k}^{(j)}\right)}{\sum_{m_j=1}^{M_j}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\sum_{\kappa=1}^{K} C_{m_j-\kappa}^{(j)}} \\ &= \sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-k}^{(j)}\right). \end{aligned} \tag{A.3}$$

If, for continuous variables, positions of MFs and consequents of single variable models are optimized in order to minimize *SCE*, i.e., (24) approximately holds, then

$$\sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-k}^{(j)}\right) \cong P\left(c_k|x^{(j)}\right). \tag{A.4}$$

For categorical variables, (A.4) stands by definition.

Now, calculate the output of a weighted multiple variables model, by substituting (10), (15) and (16) in (20).

$$\begin{aligned} y_k(\mathbf{x}) &= \frac{AGG_k(\mathbf{x})}{\sum_{\kappa=1}^{K} AGG_\kappa(\mathbf{x})} = \frac{\min\left\{1, \sum_{\rho=1}^{R}\left(W_\rho\cdot IMP_{\rho-k}(\mathbf{x})\right)\right\}}{\sum_{\kappa=1}^{K}\min\left\{1, \sum_{\rho=1}^{R}\left(W_\rho\cdot IMP_{\rho-\kappa}(\mathbf{x})\right)\right\}} \\ &= \frac{\min\left\{1, \sum_{\rho=1}^{R}\left(W_\rho\cdot FS_\rho(\mathbf{x})\cdot C_{\rho-k}\right)\right\}}{\sum_{\kappa=1}^{K}\min\left\{1, \sum_{\rho=1}^{R}\left(W_\rho\cdot FS_\rho(\mathbf{x})\cdot C_{\rho-\kappa}\right)\right\}} \\ &= \frac{\min\left\{1, \sum_{\rho=1}^{R}\left(W_\rho\cdot\prod_{j=1}^{n}\mu_{(\rho)}^{(j)}\left(x^{(j)}\right)\cdot C_{\rho-k}\right)\right\}}{\sum_{\kappa=1}^{K}\min\left\{1, \sum_{\rho=1}^{R}\left(W_\rho\cdot\prod_{j=1}^{n}\mu_{(\rho)}^{(j)}\left(x^{(j)}\right)\cdot C_{\rho-\kappa}\right)\right\}}. \end{aligned} \tag{A.5}$$

Similarly to the previous case, since normal and orthogonal fuzzy sets are used, $0 \le W_\rho \le 1$ from (37) and $0 \le C_{\rho-k} \le 1$ from (38), then

$$y_k(\mathbf{x}) = \frac{\sum_{\rho=1}^{R}\left(W_\rho\cdot\prod_{j=1}^{n}\mu_{(\rho)}^{(j)}\left(x^{(j)}\right)\cdot C_{\rho-k}\right)}{\sum_{\kappa=1}^{K}\sum_{\rho=1}^{R}\left(W_\rho\cdot\prod_{j=1}^{n}\mu_{(\rho)}^{(j)}\left(x^{(j)}\right)\cdot C_{\rho-\kappa}\right)}. \tag{A.6}$$

Adopting a combinatorial rule base in the form (36), it becomes

$$y_k(\mathbf{x}) = \frac{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(W_{\{m_1,\dots,m_n\}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{\{m_1,\dots,m_n\}-k}\right)}{\sum_{\kappa=1}^{K}\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(W_{\{m_1,\dots,m_n\}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{\{m_1,\dots,m_n\}-\kappa}\right)}. \tag{A.7}$$

Since $\sum_{k=1}^{K} C_{\{m_1,\dots,m_n\}-k} = 1$ from (38), then, with some algebra,

$$\begin{aligned} y_k(\mathbf{x}) &= \frac{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(W_{\{m_1,\dots,m_n\}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{\{m_1,\dots,m_n\}-k}\right)}{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(W_{\{m_1,\dots,m_n\}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot\sum_{\kappa=1}^{K} C_{\{m_1,\dots,m_n\}-\kappa}\right)} \\ &= \frac{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(W_{\{m_1,\dots,m_n\}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{\{m_1,\dots,m_n\}-k}\right)}{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(W_{\{m_1,\dots,m_n\}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\right)}. \end{aligned} \tag{A.8}$$

Now, substituting (37)–(39), and with some algebra,

$$\begin{aligned} y_k(\mathbf{x}) &= \frac{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(\frac{\sum_{\kappa=1}^{K}\varpi_{\{m_1,\dots,m_n\}-\kappa}}{\sum_{\eta_1=1}^{M_1}\cdots\sum_{\eta_n=1}^{M_n}\sum_{\kappa=1}^{K}\varpi_{\{\eta_1,\dots,\eta_n\}-\kappa}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot\frac{\varpi_{\{m_1,\dots,m_n\}-k}}{\sum_{\kappa=1}^{K}\varpi_{\{m_1,\dots,m_n\}-\kappa}}\right)}{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(\frac{\sum_{\kappa=1}^{K}\varpi_{\{m_1,\dots,m_n\}-\kappa}}{\sum_{\eta_1=1}^{M_1}\cdots\sum_{\eta_n=1}^{M_n}\sum_{\kappa=1}^{K}\varpi_{\{\eta_1,\dots,\eta_n\}-\kappa}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\right)} \\ &= \frac{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(\varpi_{\{m_1,\dots,m_n\}-k}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\right)}{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(\sum_{\kappa=1}^{K}\varpi_{\{m_1,\dots,m_n\}-\kappa}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\right)} \\ &= \frac{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(\frac{\prod_{j=1}^{n} C_{m_j-k}^{(j)}}{P(c_k)^{n-1}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\right)}{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(\sum_{\kappa=1}^{K}\frac{\prod_{j=1}^{n} C_{m_j-\kappa}^{(j)}}{P(c_\kappa)^{n-1}}\cdot\prod_{j=1}^{n}\mu_{m_j}^{(j)}\left(x^{(j)}\right)\right)} \\ &= \frac{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(\frac{\prod_{j=1}^{n}\left(C_{m_j-k}^{(j)}\cdot\mu_{m_j}^{(j)}\left(x^{(j)}\right)\right)}{P(c_k)^{n-1}}\right)}{\sum_{m_1=1}^{M_1}\cdots\sum_{m_n=1}^{M_n}\left(\sum_{\kappa=1}^{K}\frac{\prod_{j=1}^{n}\left(C_{m_j-\kappa}^{(j)}\cdot\mu_{m_j}^{(j)}\left(x^{(j)}\right)\right)}{P(c_\kappa)^{n-1}}\right)}. \end{aligned} \tag{A.9}$$

This can also be written as:

$$y_k(\mathbf{x}) = \frac{\frac{\prod_{j=1}^{n}\sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-k}^{(j)}\right)}{P(c_k)^{n-1}}}{\sum_{\kappa=1}^{K}\frac{\prod_{j=1}^{n}\sum_{m_j=1}^{M_j}\left(\mu_{m_j}^{(j)}\left(x^{(j)}\right)\cdot C_{m_j-\kappa}^{(j)}\right)}{P(c_\kappa)^{n-1}}}, \tag{A.10}$$

which is also a computationally fast way to calculate $y_k(\mathbf{x})$, with respect to (A.7), since its complexity is lower, and it does not require to previously compute the parameters of the associated multiple variables model.

Due to (A.4), it results from (A.10):

$$y_k(\mathbf{x}) \cong \frac{\frac{\prod_{j=1}^{n} P\left(c_k|x^{(j)}\right)}{P(c_k)^{n-1}}}{\sum_{\kappa=1}^{K}\frac{\prod_{j=1}^{n} P\left(c_\kappa|x^{(j)}\right)}{P(c_\kappa)^{n-1}}}. \tag{A.11}$$

Applying some algebra and the Bayes' theorem, it results:

$$\begin{aligned} y_k(\mathbf{x}) &\cong \frac{P(c_k)\prod_{j=1}^{n}\frac{P(c_k|x^{(j)})}{P(c_k)}}{\sum_{\kappa=1}^{K}\left(P(c_\kappa)\prod_{j=1}^{n}\frac{P(c_\kappa|x^{(j)})}{P(c_\kappa)}\right)} \\ &= \frac{P(c_k)\prod_{j=1}^{n}\frac{P(x^{(j)}|c_k)}{P(x^{(j)})}}{\sum_{\kappa=1}^{K}\left(P(c_\kappa)\prod_{j=1}^{n}\frac{P(x^{(j)}|c_\kappa)}{P(x^{(j)})}\right)} \\ &= \frac{\frac{P(c_k)\cdot\prod_{j=1}^{n}P(x^{(j)}|c_k)}{\prod_{j=1}^{n}P(x^{(j)})}}{\sum_{\kappa=1}^{K}\frac{P(c_\kappa)\cdot\prod_{j=1}^{n}P(x^{(j)}|c_\kappa)}{\prod_{j=1}^{n}P(x^{(j)})}} \\ &= \frac{P(c_k)\cdot\prod_{j=1}^{n}P(x^{(j)}|c_k)}{\sum_{\kappa=1}^{K}\left(P(c_\kappa)\cdot\prod_{j=1}^{n}P(x^{(j)}|c_\kappa)\right)}. \end{aligned} \tag{A.12}$$

At this point, the naïve Bayes hypothesis of conditional independence plays its role, allowing to write:

$$P(x^{(j)}|c_k) \cong P(x^{(j)}|x^{(j+1)},\ldots,x^{(n)},c_k). \tag{A.13}$$

Therefore,

$$\begin{aligned} y_k(\mathbf{x}) &\cong \frac{P(c_k)\cdot P(x^{(1)}|c_k)\cdots P(x^{(n)}|c_k)}{\sum_{\kappa=1}^{K}\left(P(c_\kappa)\cdot P(x^{(1)}|c_\kappa)\cdots P(x^{(n)}|c_\kappa)\right)} \cong \\ &\cong \frac{P(x^{(1)}|x^{(2)},\ldots,x^{(n)},c_k)\cdots P(x^{(n-1)}|x^{(n)},c_k)\cdot P(x^{(n)}|c_k)\cdot P(c_k)}{\sum_{\kappa=1}^{K}\left(P(x^{(1)}|x^{(2)},\ldots,x^{(n)}c_\kappa)\cdots P(x^{(n-1)}|x^{(n)},c_\kappa)\cdot P(x^{(n)}|c_\kappa)\cdot P(c_\kappa)\right)}, \end{aligned} \tag{A.14}$$

and due to the chain rule of conditional probabilities,

$$y_k(\mathbf{x}) \cong \frac{P(c_k|\mathbf{x})\cdot P(\mathbf{x})}{\sum_{\kappa=1}^{K}P(c_\kappa|\mathbf{x})\cdot P(\mathbf{x})} = \frac{P(c_k,\mathbf{x})}{\sum_{\kappa=1}^{K}P(c_\kappa,\mathbf{x})}. \tag{A.15}$$

Therefore,

$$y_k(\mathbf{x}) \cong P(c_k|\mathbf{x}). \tag{A.16}$$

This final formula (A.16), stating that the output of the multiple variables classifier approximates the class posterior probability, approximately satisfies (24); therefore, it proves that the multiple variables model is optimized with respect to *SCE*.

Appendix B

This section reports the equations, by means of which membership functions $\mu_m(x)$ are modelled here, depending on the fuzzy sets shape, with $m = 1,\ldots,M$, where M is the cardinality of the partition. In all these equations, the membership grade is a function of the same set of parameters $\mathbf{p} = \{p_1,\ldots,p_{2M-2}\}$ with $p_1 < \cdots < p_{2M-2}$. These equations were chosen in order to obtain, in correspondence of a given set of parameters, partitions with different shapes but approximately the same positions of the fuzzy sets (as shown in Fig. 1), in order to properly compare properties depending only on the shape (as reported in Section 5.1.2).

Firstly, the set of parameters $\mathbf{p} = \{p_1,\ldots,p_{2M-2}\}$ is transformed. In case of binary shape, it is transformed into $M-1$ parameters as follows:

$$\mathbf{p}_{binary} = \left\{p_{1,binary},\ldots,p_{m,binary},\ldots,p_{M-1,binary}\right\}, \tag{B.1}$$

where

$$p_{m,binary} = \frac{p_{2m-1}+p_{2m}}{2}, \tag{B.2}$$

and MFs are modelled as:

$$\mu_{1,binary}(x;\mathbf{p}_{binary}) = \begin{cases} 1, & x \le p_{1,binary} \\ 0, & p_{1,binary} < x \end{cases} \tag{B.3}$$

$$\mu_{m,binary}(x;\mathbf{p}_{binary}) = \begin{cases} 0, & x \le p_{m-1,binary} \\ 1, & p_{m-1,binary} < x \le p_{m,binary} \\ 0, & p_{m,binary} < x \end{cases} \tag{B.4}$$

$$\mu_{M,binary}(x;\mathbf{p}_{binary}) = \begin{cases} 0, & x \le p_{M-1,binary} \\ 1, & p_{M-1,binary} < x \end{cases}. \tag{B.5}$$

In case of triangular shape, M parameters are obtained as follows:

$$\mathbf{p}_{triangular} = \left\{p_{1,triangular},\ldots,p_{m,triangular},\ldots,p_{M,triangular}\right\}, \tag{B.6}$$

where

$$p_{m,triangular} = \begin{cases} p_1, & m = 1 \\ \frac{p_{2m-2}+p_{2m-1}}{2}, & 1 < m < M \\ p_{2M-2}, & m = M \end{cases}, \tag{B.7}$$

and MFs are modelled as:

$$\mu_{1,triangular}(x;\mathbf{p}_{triangular}) = \begin{cases} 1, & x \le p_{1,triangular} \\ \frac{p_{2,triangular}-x}{p_{2,triangular}-p_{1,triangular}}, & p_{1,triangular} < x \le p_{2,triangular} \\ 0, & p_{2,triangular} < x \end{cases} \tag{B.8}$$

$$\mu_{m,triangular}(x;\mathbf{p}_{triangular}) = \begin{cases} 0, & x \le p_{m-1,triangular} \\ \frac{x-p_{m-1,triangular}}{p_{m,triangular}-p_{m-1,triangular}}, & p_{m-1,triangular} < x \le p_{m,triangular} \\ \frac{p_{m+1,triangular}-x}{p_{m+1,triangular}-p_{m,triangular}}, & p_{m,triangular} < x \le p_{m+1,triangular} \\ 0, & p_{m+1,triangular} < x \end{cases} \tag{B.9}$$

$$\mu_{M,triangular}(x;\mathbf{p}_{triangular}) = \begin{cases} 0, & x \le p_{M-1,triangular} \\ \frac{x-p_{M-1,triangular}}{p_{M,triangular}-p_{M-1,triangular}}, & p_{M-1,triangular} < x \le p_{M,triangular} \\ 1, & p_{M,triangular} < x \end{cases}. \tag{B.10}$$

In case of trapezoidal shape, $2M-2$ parameters are equal to the original ones:

$$\mathbf{p}_{trapezoidal} = \mathbf{p}, \tag{B.11}$$

and MFs are modelled as:

$$\mu_{1,trapezoidal}(x;\mathbf{p}) = \begin{cases} 1, & x \le p_1 \\ \frac{p_2 - x}{p_2 - p_1}, & p_1 < x \le p_2 \\ 0, & p_2 < x \end{cases} \tag{B.12}$$

$$\mu_{m,trapezoidal}(x;\mathbf{p}) = \begin{cases} 0, & x \le p_{2m-3} \\ \frac{x - p_{2m-3}}{p_{2m-2} - p_{2m-3}}, & p_{2m-3} < x \le p_{2m-2} \\ 1, & p_{2m-2} < x \le p_{2m-1} \\ \frac{p_{2m} - x}{p_{2m} - p_{2m-1}}, & p_{2m-1} < x \le p_{2m} \\ 0, & p_{2m} < x \end{cases} \tag{B.13}$$

$$\mu_{M,trapezoidal}(x;\mathbf{p}) = \begin{cases} 0, & x \le p_{2M-3} \\ \frac{x - p_{2M-3}}{p_{2M-2} - p_{2M-3}}, & p_{2M-3} < x \le p_{2M-2} \\ 1, & p_{2M-2} < x \end{cases}. \tag{B.14}$$

For sigmoidal, Gaussian and quadratic shapes, some constants are previously found:

$$t_s = \ln(1/\varepsilon - 1) \tag{B.15}$$

$$t_G = \sqrt{-2\ln(\varepsilon)} \tag{B.16}$$

$$t_q = \sqrt{1/\varepsilon - 1}, \tag{B.17}$$

where $\varepsilon \ll 1$ (in this paper, $\varepsilon = 0.01$) is a positive constant fixed to approximate those MFs to normal, i.e., $0 < \mu(x) < 1, \exists x | \mu(x) < \varepsilon, \exists x | \mu(x) > 1 - \varepsilon$.

In case of sigmoidal shape, $2M - 2$ parameters are equal to the original ones:

$$\mathbf{p}_{sigmoidal} = \mathbf{p}, \tag{B.18}$$

and MFs are modelled as:

$$\mu_{1,sigmoidal}(x;\mathbf{p}) = 1 - \frac{1}{1 + e^{\left(t_s \cdot \frac{p_2 + p_1}{p_2 - p_1} - 2 \cdot t_s \cdot \frac{x}{p_2 - p_1}\right)}} \tag{B.19}$$

$$\mu_{m,sigmoidal}(x;\mathbf{p}) = \frac{1}{1 + e^{\left(t_s \cdot \frac{p_{2m-2} + p_{2m-3}}{p_{2m-2} - p_{2m-3}} - 2 \cdot t_s \cdot \frac{x}{p_{2m-2} - p_{2m-3}}\right)}} - \frac{1}{1 + e^{\left(t_s \cdot \frac{p_{2m} + p_{2m-1}}{p_{2m} - p_{2m-1}} - 2 \cdot t_s \cdot \frac{x}{p_{2m} - p_{2m-1}}\right)}} \tag{B.20}$$

$$\mu_{M,sigmoidal}(x;\mathbf{p}) = \frac{1}{1 + e^{\left(t_s \cdot \frac{p_{2M-2} + p_{2M-3}}{p_{2M-2} - p_{2M-3}} - 2 \cdot t_s \cdot \frac{x}{p_{2M-2} - p_{2M-3}}\right)}}. \tag{B.21}$$

In case of Gaussian shape, $2M$ parameters are obtained as follows:

$$\mathbf{p}_{Gauss} = \{\eta_{1,Gauss}, \sigma_{1,Gauss}, \ldots, \eta_{m,Gauss}, \sigma_{m,Gauss}, \ldots, \eta_{M,Gauss}, \sigma_{M,Gauss}\}, \tag{B.22}$$

where

$$\eta_{m,Gauss} = p_{m,triangular} \tag{B.23}$$

and

$$\sigma_{m,Gauss} = \begin{cases} \frac{\eta_{2,Gauss} - \eta_{1,Gauss}}{t_G}, & m = 1 \\ \frac{\eta_{m+1,Gauss} - \eta_{m-1,Gauss}}{2t_G}, & 1 < m < M, \\ \frac{\eta_{M,Gauss} - \eta_{M-1,Gauss}}{t_G}, & m = M \end{cases} \tag{B.24}$$

and MFs are modelled as:

$$\mu_{1,Gauss}(x;\mathbf{p}_{Gauss}) = \begin{cases} 1, & x \le \eta_{1,Gauss} \\ \exp\left(-\frac{(x - \eta_{1,Gauss})^2}{2(\sigma_{1,Gauss})^2}\right), & \eta_{1,Gauss} < x \end{cases} \tag{B.25}$$

$$\mu_{m,Gauss}(x;\mathbf{p}_{Gauss}) = \exp\left(-\frac{(x - \eta_{m,Gauss})^2}{2(\sigma_{m,Gauss})^2}\right) \tag{B.26}$$

$$\mu_{M,Gauss}(x;\mathbf{p}_{Gauss}) = \begin{cases} \exp\left(-\frac{(x - \eta_{M,Gauss})^2}{2(\sigma_{M,Gauss})^2}\right), & x \le \eta_{M,Gauss} \\ 1, & \eta_{M,Gauss} < x \end{cases}. \tag{B.27}$$

In case of quadratic bell shape, $2M$ parameters are obtained:

$$\mathbf{p}_{quadratic} = \{\eta_{1,quadratic}, \sigma_{1,quadratic}, \ldots, \eta_{m,quadratic}, \sigma_{m,quadratic}, \ldots, \eta_{M,quadratic}, \sigma_{M,quadratic}\}, \tag{B.28}$$

similarly to the previous case:

$$\eta_{m,quadratic} = p_{m,triangular} \tag{B.29}$$

$$\sigma_{m,quadratic} = \begin{cases} \frac{\eta_{2,quadratic} - \eta_{1,quadratic}}{t_q}, & m = 1 \\ \frac{\eta_{m+1,quadratic} - \eta_{m-1,quadratic}}{2t_q}, & 1 < m < M, \\ \frac{\eta_{M,quadratic} - \eta_{M-1,quadratic}}{t_q}, & m = M \end{cases} \tag{B.30}$$

and MFs are modelled as:

$$\mu_{1,quadratic}(x;\mathbf{p}_{quadratic}) = \begin{cases} 1, & x \le \eta_{1,quadratic} \\ \frac{1}{1 + \left|\frac{x - \eta_{1,quadratic}}{\sigma_{1,quadratic}}\right|^2}, & \eta_{1,quadratic} < x \end{cases} \tag{B.31}$$

$$\mu_{m,quadratic}(x;\mathbf{p}_{quadratic}) = \frac{1}{1 + \left|\frac{x - \eta_{m,quadratic}}{\sigma_{m,quadratic}}\right|^2} \tag{B.32}$$

$$\mu_{M,quadratic}(x;\mathbf{p}_{quadratic}) = \begin{cases} \frac{1}{1 + \left|\frac{x - \eta_{M,quadratic}}{\sigma_{M,quadratic}}\right|^2}, & x \le \eta_{M,Gauss} \\ 1, & \eta_{M,Gauss} < x \end{cases}. \tag{B.33}$$

References

[1] A. Ortiz, J. Munilla, J.M. Górriz, J. Ramírez, Ensembles of deep learning architectures for the early diagnosis of the Alzheimer's disease, Int. J. Neural Syst. 26 (7) (2016) 1650025.
[2] L. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[3] M. J.Gacto, R. Alcalá, F. Herrera, Interpretability of linguistic fuzzy rule-based systems: an overview of interpretability measures, Inf. Sci. 181 (2011) 4340–4360.
[4] C. Mencar, Interpretability of fuzzy systems, in: Fuzzy Logic and Applications, 8256, Springer, Cham, 2013, pp. 22–35.
[5] N. Gupta, S.K. Jain, Comparative analysis of fuzzy power system stabilizer using different membership functions, Int. J. Comput. Electr. Eng. 2 (2) (2010) 262–267.
[6] J.G. Monicka, N. Sekhar, K.R. Kumar, Performance evaluation of membership functions on fuzzy logic controlled AC voltage controller for speed control of induction motor drive, Int. J. Comput. Appl. 13 (5) (2011) 8–12.
[7] J. Zhao, B. Bose, Evaluation of membership functions for fuzzy logic controlled induction motor drive, in: Proceedings of the Twenty-Eighth IEEE Annual Conference of the Industrial Electronics Society, Sevilla, Spain, 2002, pp. 229–234.
[8] L. Rutkowski, K. Cpalka, Flexible neuro-fuzzy systems, IEEE Trans. Neural Netw. 14 (3) (2003) 554–574.
[9] C. Mencar, A.M. Fanelli, Interpretability constraints for fuzzy information granulation, Inf. Sci. 178 (24) (2008) 4585–4618.
[10] R. Alcalà, M.J. Gacto, F. Herrera, J. Alcalà-Fdez, A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems, Int. J. Uncertain. Fuzziness Knowl. Based Syst. 15 (05) (2007) 539.
[11] R. Alcalà, P. Ducange, F. Herrera, B. Lazzerini, F. Marcelloni, A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems, IEEE Trans. Fuzzy Syst. 17 (5) (2009) 1106–1122.
[12] A.A. Márquez, F.A. Márquez, A. Peregrín, A multi-objective evolutionary algorithm with an interpretability improvement mechanism for linguistic fuzzy systems with adaptive defuzzification, in: Proceedings of the IEEE International Conference on Fuzzy Systems, 2010, pp. 18–23.

[13] S. Guillaume, B. Charnomordic, Learning interpretable fuzzy inference systems with FisPro, Inf. Sci. 181 (2011) 4409–4427.
[14] M. Antonelli, P. Ducange, B. Lazzerini, F. Marcelloni, Learning concurrently data and rule bases of Mamdani fuzzy rule-based systems by exploiting a novel interpretability index, Soft Comput. 15 (10) (2011) 1981–1998.
[15] M.B. Gorzałczany, F. Rudziński, Accuracy vs. interpretability of fuzzy rule-based classifiers: an evolutionary approach, Lect. Notes Comput. Sci. 7269 (2012) 222–230.
[16] A. Cano, A. Zafra, S. Ventura, An interpretable classification rule mining algorithm, Inf. Sci. 240 (2013) 1–20.
[17] F. Rudziński, A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers, Appl. Soft Comput. 38 (2016) 118–133.
[18] M. Pota, M. Esposito, Degrees of Freedom and Advantages of Different Rule-Based Fuzzy Systems, in: N.E. Mastorakis, P.M. Pardalos, R.P. Agarwal, L. Kočinac (Eds.), Advances in Applied and Pure Mathematics—Proceedings of the 2014 International Conference on Pure Mathematics, Applied Mathematics and Computational Methods (PMAMCM 2014), Santorini Island, Greece, July 17-21, 2014, Mathematics and Computers in Science and Engineering Series, Vol. 29, 2014, pp. 107–114.
[19] M. Pota, M. Esposito, Insights into interpretability of neuro-fuzzy systems, in: Proceedings of the 16th World Congress of the International Fuzzy Systems Association (IFSA) and the 9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), Gijòn,, Atlantis Press, Spain, 2015, pp. 1427–1434, doi:10.2991/ifsa-eusflat-15.2015.202.
[20] K. Bache, M. Lichman, UCI Machine Learning Repository, School of Information and Computer Science, University of California, Irvine, CA, 2013.
[21] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Stat. 18 (1) (1947) 50–60.
[22] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, B. Prieto, PCA filtering and probabilistic SOM for network intrusion detection, Neurocomputing 164 (2015) 71–81.
[23] K. Pearson, Notes on regression and inheritance in the case of two parents, Proc. R. Soc. Lond. 58 (1895) 240–242.
[24] M. Pota, E. Scalco, G. Sanguineti, G.M. Cattaneo, M. Esposito, G. Rizzo, Early classification of parotid glands shrinkage in radiotherapy patients: a comparative study, Biosyst. Eng. 138 (2014) 77–89.
[25] D. Wu, Twelve considerations in choosing between Gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers, in: Proceedings of the IEEE World Congress on Computational Intelligence, Brisbane, Australia, 2012, pp. 1–8.
[26] V. Barrile, G. Bilotta, Object-oriented analysis applied to high resolution satellite data, WSEAS Trans. Signal Process. 4 (3) (2008) 68–75.
[27] T.E. Rothenfluh, K. Bögl, K.-P. Adlassnig, Representation and acquisition of knowledge for a fuzzy medical consultation system, in: P.S. Szczepaniak, P.J.G. Lisboa, J. Kacprzyk (Eds.), Fuzzy systems in Medicine, Studies in Fuzziness and Soft Computing, 41, Physica-Verlag HD, 2000, pp. 636–651.
[28] J.M. Alonso, C. Castiello, M. Lucarelli, C. Mencar, Modelling interpretable fuzzy rule-based classifiers for medical decision support, in: R. Magdalena, E. Soria, J. Guerrero, J. Gómez-Sanchis, A.J. Serrano (Eds.), Medical Applications of Intelligent Data Analysis: Research Advancements, IGI Global, 2012, pp. 255–272.
[29] S. Guillaume, Designing fuzzy inference systems from data: an interpretability-oriented review, IEEE Trans. Fuzzy Syst. 9 (2001) 426–443.
[30] J.R. Quinlan, Induction on decision trees, Mach. Learn. 1 (1986) 81–106.
[31] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Symposium on Mathematical Statistics and Probability, Berkeley, 2001, pp. 281–297.
[32] C.F. Eick, N. Zeidat, Z. Zhao (2005). Supervised clustering – algorithms and benefits, Technical Report Number UH-CS-05–10, IEEE.
[33] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, NorwellMAUSA, 1981.
[34] W. Pedrycz, Fuzzy equalization in the construction of fuzzy sets, Fuzzy Sets Syst. 64 (1994) 21–30.
[35] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets Syst. 65 (1995) 125–139.
[36] J. Abonyi, F. Szeifert, Supervised fuzzy clustering for the identification of fuzzy classifiers, Pattern Recognit. Lett. 24 (2003) 2195–2207.
[37] M. Pota, M. Esposito, G. De Pietro, Hybridization of possibility theory and supervised clustering to build DSSs for classification in medicine, in: Proceedings of the Twelvth International Conference on Hybrid Intelligent Systems (HIS), Pune,, IEEE, India, 2012, pp. 490–495.
[38] M. Esposito, I. De Falco, G. De Pietro, An evolutionary-fuzzy DSS for assessing health status in multiple sclerosis disease, Int. J. Med. Inf. 80 (12) (2011) e245–e254.
[39] C.A. Pena-Rees, M. Sipper, A fuzzy-genetic approach to breast cancer diagnosis, Artif. Intell. Med. 17 (1999) 131–155.
[40] M.-J. Huang, Y.-L. Tsou, S.-C. Lee, Integrating fuzzy data mining and fuzzy artificial neural networks for discovering implicit knowledge, Knowl. Based Syst. 19 (6) (2006) 396–403.
[41] M. Pota, M. Esposito, G. De Pietro, Transforming probability distributions into membership functions of fuzzy classes: a hypothesis test approach, Fuzzy Sets Syst. 233 (2013) 52–73.
[42] M. Pota, M. Esposito, G. De Pietro, Approximation of statistical information with fuzzy models for classification in medicine, in: A.M.J. Skulimowski, J. Kacprzyk (Eds.), Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions, Advances in Intelligent Systems and Computing, Springer International Publishing, 2016, pp. 359–371.
[43] M. Pota, M. Esposito, G. De Pietro, Fuzzy partitioning for clinical DSSs using statistical information transformed into possibility-based knowledge, Knowl. Based Syst. 67 (2014) 1–15.
[44] P.Y. Glorennec, Algorithmes D'apprentissage Pour Systèmes D'inférence Floue, Hermes Science Publications, Paris, 1999.
[45] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE. Trans. Syst. Man. Cybern. 22 (1992) 1414–1427.
[46] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man Mach. Stud. 7 (1) (1975) 1–13.
[47] M. Sugeno, Industrial Applications of Fuzzy Control, Elsevier Science Pub. Co, 1985.
[48] R. Babuska, H.B. Verbruggen, An overview of fuzzy modelling for control, Control Eng. Pract. 4 (11) (1996) 1593–1606.
[49] S. Guillaume, B. Charnomordic, A new method for inducing a set of interpretable fuzzy partitions and fuzzy inference from data, Stud. Fuzziness Soft Comput. 128 (2003) 148–175.
[50] R.R. Yager, D.P. Filev, Essentials of Fuzzy Modeling and Control, Wiley, New York, 1994.
[51] W. Van Leekwijck, E.E. Kerre, Defuzzification: criteria and classification, Fuzzy Sets Syst. 108 (1999) 159–178.
[52] S.N. Ghazavi, T.W. Liao, Medical data mining by fuzzy modeling with selected features, Artif. Intell. Med. 43 (2008) 195–206.
[53] C. Mencar, A. Fanelli, Interpretability constraints for fuzzy information granulation, Inf. Sci. 178 (2008) 4585–4618.
[54] P. Pulkkinen, H. Koivisto, A dynamically constrained multiobjective genetic fuzzy system for regression problems, IEEE Trans. Fuzzy Syst. 18 (2010) 61–177.
[55] P. Fazendeiro, J.V. de Oliveira, W. Pedrycz, A multiobjective design of a patient and anaesthetist-friendly neuromuscular blockade controller, IEEE. Trans. Biomed. Eng. 54 (2007) 1667–1678.
[56] R. Mikut, J. Jakel, L. Grall, Interpretability issues in data-based learning of fuzzy systems, Fuzzy Sets Syst. 150 (2005) 179–197.
[57] A. Marquez, F. Márquez, A. Peregrin, A multi-objective evolutionary algorithm with an interpretability improvement mechanism for linguistic fuzzy systems with adaptive defuzzification, in: Proceedings of the IEEE World Congress on Computational Intelligence, 2010.
[58] G. Castellano, A.M. Fanelli, C. Mencar, A neuro-fuzzy network to generate human-understandable knowledge from data, Cognit. Syst. Res. 3 (2002) 125–144.
[59] Wolfram Research, Inc., Mathematica, Version 8.0, Wolfram Research, Inc., Champaign, IL (2010).
[60] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explor. 11 (1) (2009) 10–18.
[61] D.K. Subramanian, V.S. Ananthanarayana, M. Narasimha Murty, Knowledge-based association rule mining using and-or taxonomies, Knowl. Based Syst. 16 (1) (2003) 37–45.
[62] J.R. Quinlan, Improved use of continuous attributes in C4.5, J. Artif. Intell. Res. 4 (1996) 77–90.
[63] G. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Mach. Learn. 9 (4) (1992) 309–347.
[64] D.M. Dutton, G.V. Conroy, A review of machine learning, Knowl. Eng. Rev. 12 (4) (1996) 341–367.
[65] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schoelkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods – Support Vector Learning, MIT Press Cambridge, MA, USA, 1998.
[66] N. O'Connor, M.G. Madden, A neural network approach to predicting stock exchange movements using external factors, Knowl. Based Syst. 19 (5) (2006) 371–378.
[67] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc. 32 (1937) 675–701.
[68] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.
[69] P.B. Nemenyi, Distribution-free multiple comparisons. (Ph.D. thesis), Princeton University, 1963.
[70] O.J. Dunn, Multiple comparisons among means, J. Amer. Statist. Assoc. 56 (1961) 52–64.
[71] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 65–70.