FuzzyFeatureRank. Bringing order into fuzzy classifiers through fuzzy expressions

Pablo Carmona [a,*], Juan Luis Castro [b]

[a] *Dept. of Computer and Telematics Systems Engineering, University of Extremadura, Spain*
[b] *Dept. of Computer Science and Artificial Intelligence, University of Granada, Spain*

Abstract

This work presents FuzzyFeatureRank, a new feature reduction method inspired on PageRank to reduce the dimensionality of the feature space in supervised classification problems. More precisely, as it relies on a weighted directed graph, it is ultimately inspired on TextRank, a PageRank based method that adds weights to the edges to express the strength of the connections between nodes. The method is based on dividing each original feature used to describe the data into a set of fuzzy predicates and then ranking all of them by their ability to differentiate among classes in the light of the training set. In order to do that, both the information gained by each predicate and their redundancy with other already selected predicates are taken into account. The fuzzy predicates with the best scores can then be used as a reduced input to construct fuzzy classifiers that consider only the preselected predicates to build the antecedents of the fuzzy rules. The novelty of the proposal relies on being an approach halfway between feature selection and feature extraction approaches, being able to improve the discrimination ability of the original features but preserving the interpretability of the new features in the sense that they are fuzzy expressions. The experimental results support the suitability of the proposal.

Keywords: Fuzzy classifiers; Ranking methods; Feature reduction; Feature selection; Feature extraction; TextRank

1. Introduction

In many contexts where a huge amount of information needs to be analyzed, the overwhelming processing requirements can be mitigated by evaluating the usefulness of each piece of information for the goal we are pursuing and selecting the best ones. In supervised classification problems, where the data is expressed through a set of labeled examples (each example represented by a set of feature values and its class label), those techniques fit with the concept of dimensionality or feature reduction.

Two main approaches are used to reach this goal: feature extraction and feature selection [1–3]. While the former is based on projecting the original feature space into a new feature space, the latter directly selects a subset of the best original features using some ranking method to evaluate their relevance.

The feature reduction gained by both approaches not only suppose an improvement of the computational efficiency, but it also often improves the classification performance by increasing its generalization ability. However, each approach has its own drawbacks. On the one hand, the new feature space obtained through feature extraction is usually less interpretable than the original feature space, since the physical meaning of the original features is lost. On the other hand, feature selection is confined to the original features, whose ability to discriminate among classes is expected, in general, to be lower than a set of new features specifically designed to improve this ability.

Fuzzy classifiers are one type of classifiers that use fuzzy rules to describe the relationship between the feature values of an example and its associated class. The main advantages of fuzzy classifiers are their tolerance to imprecise and missing data and their interpretability [4].

In fuzzy classifiers, the domain of each feature is decomposed into a set of fuzzy sets (a fuzzy domain), where each fuzzy set expresses the degrees to which the feature values belong to that set by means of a membership function. This decomposition allows to describe the relationship between features values and classes by means of fuzzy rules where the antecedent part is composed by a set of fuzzy predicates, for example the rule "if *feature*1 is *large* and *feature*2 is *medium* then the *class* is *class*2". There are many ways to decompose the domain of a feature into a fuzzy domain, ranging from handmade solutions where the membership functions are defined by experts to problem independent solutions where the fuzzy sets are equally shaped and distributed along the feature domain. In any case, it is desirable that the method used to design the fuzzy classifier be independent of the technique used to partition each feature into fuzzy predicates, and therefore, can deal with any of them, as it is the case of our proposal.

Many techniques have been proposed in the literature to design fuzzy classifiers, such as heuristic approaches [5,6], neuro-fuzzy techniques [7,8], association rule discovery [4,9], genetic algorithms [10], and based on evolving systems [11] or on support vector machines [12].

However, one of the main problems of fuzzy classifiers relays in the fact that they are affected by the curse of dimensionality, in the sense that both the search space of fuzzy rules to be taken into account during the design of the fuzzy classifier and the number of fuzzy rules that finally describe it, increases quickly with the dimension of the feature space. Concretely, if the number of features is n, the number of fuzzy sets per feature is p and we consider the *don't care* predicate (for more details, see Section 4.2), the number of possible fuzzy rules is $(p+1)^n$. This negatively affects to both the computational cost in the design of the fuzzy classifier and its interpretability.

Several approaches have been proposed to solve this problem, such as the use of neural networks [13], similarity measures [14] or genetic algorithms [15]. Another approach consists in restricting the size of the antecedent on the fuzzy rules being considered [5,6] during the design of the fuzzy classifier, although even with such restricted fuzzy rules, the search space remains intractable for high-dimensional datasets.

In this work, we propose a new the feature reduction method named FuzzyFeatureRank, whose main ideas can be summarize in 1) to translate the problem from the space of feature variables to the space of fuzzy predicates on features, and then apply a feature selection on the new space, and 2) to consider a TextRank-like ranking method for the feature selection on this new space.

Concretely, instead of selecting a set of the original features, the novelty of the proposal relies on considering each possible fuzzy predicate X is A as a new feature that expresses the degrees to which the values of the original feature X belong to the fuzzy set A and then using the ranking method to select a set of promising fuzzy predicates as a preprocessing step before the design of the fuzzy classifier. That way, these preselected fuzzy predicates will be the only ones considered during the design of the fuzzy classifier, thus reducing the number of possible fuzzy rules taken into account. As a result, the search space is reduced and the viability of using fuzzy classifiers when dealing with high-dimensional datasets is increased. Our aim is that the preselection mechanism manages to preserve the best fuzzy predicates, so that the reduction of the search space allows to reduce the computational cost during the design of the fuzzy classifier and obtain classifiers with a better balance of accuracy and interpretability.

As an illustrative example, consider the well known IRIS classification problem, whose objective is to differentiate among three types of iris plants (setosa, versicolor, virginica) based on the features sepal length (SL), sepal width (SW), petal length (PL) and petal width (PW). Assume the same fuzzy domain consisting of 5 fuzzy sets for every feature that represent the fuzzy values {*VS: Very small*, *S: Small*, *M: Medium*, *L: Large*, *VL: Very large*}. The total number of possible fuzzy rules to consider for the design of a fuzzy classifier with 4 fuzzy predicates in the antecedent, including *don't care* predicates, is $6^4 = 1296$ rules.

In our proposal each fuzzy predicate is considered a new feature. For example, the feature SL is decomposed into the new fuzzy features *SL is VS*, *SL is S*, *SL is M*, *SL is L*, and *SL is VL*, where $SLisA$ expresses the degree $A(sl)$ to which the value sl of the original feature SL belongs to the fuzzy set A. For example, if the sepal length of a sample in the dataset is $sl = 0.92$ and this value has a membership degree to the fuzzy set *Large* $L(sl) = 0.7$, the new fuzzy feature *SL is L* will have the value 0.7.

Now, using some feature selection method, a subset of these new features in the form of fuzzy predicates can be preselected. For example, in the IRIS problem, the feature selection could give rise to the preselected fuzzy predicates *pw is VS*, *pw is M*, *pl is L*, *sl is M*, *sl is S*, and *sw is S*. Thus, the number of possible rules considered during design of the fuzzy classifier would be reduced from the 1296 rules to just 36.

FuzzyFeatureRank is inspired by PageRank [16], a well-known ranking algorithm initially used by Google to rank web pages and widely extended to other contexts [17,18]. This algorithm relies on a directed graph with scores associated with its nodes that is iteratively traversed by a random walker making the scores evolve. Once a convergence condition is reached, the final scores are used to rank the web pages. One of the multiple extensions of this algorithm is TextRank [17], a ranking model applied to text processing, whose main difference with respect to PageRank consist in associating weights to the edges of the graph. In the context of TextRank, this weighting allows to express the strength of the connection between text units.

FuzzyFeatureRank also uses a weighted directed graph and, therefore, it is ultimately based on TextRank. The nodes represent fuzzy expressions and each directed edge has a weight that assesses the usefulness of adding the expression in the target node to the expression in the source node. This usefulness is based on the concepts of information gain and mutual information measured through the entropy of fuzzy expressions in the light of the dataset.

The rest of this paper is organized as follows. In Section 2, the PageRank and TextRank models that inspire our ranking methods are described. Section 3 is the core of the paper, where the measures of the entropy, information gain and mutual information between fuzzy predicates are defined, the adaptation of TextRank to our context is explained, and the FuzzyFeatureRank algorithm is described. The experimental results are presented in Section 4, and Section 5 concludes the paper.

2. PageRank and TextRank models

The TextRank model is a graph-based ranking algorithm for text processing inspired by PageRank, a well-known ranking algorithm used by Google to rank web pages.

PageRank [16] shifts the paradigm of web page ranking from analyzing the local information contained in a page to analyzing the global information contained on the link-structure of the World Wide Web. This shift is based on computing the importance of the page by taking into account the links that the page receive (backward links) and the importance of the pages linking to it. To achieve that, a directed graph is built where nodes represent pages and directed edges represent the links from one page to another. Next, an iterative process allows to transfer the importance of the pages connected by links until a convergence condition is reached. Finally, each page will have a score proportional to its importance, this importance being obtained based exclusively on the information embedded in the hyperlink structure, regardless of the content of the page.

A formal definition is the following: Let $G = (\mathcal{N}, \mathcal{A})$ be a directed graph with a set of nodes $\mathcal{N}$ representing web pages and a set of directed edges (arrows) $\mathcal{A}$ representing links between web pages. Let $In(N_i)$ and $Out(N_i)$ the set of nodes that point to and are pointed from node N_i, respectively. Then, the score of a node N_i is defined as:

$$S(N_i) = (1 - d) + d \cdot \sum_{N_j \in In(N_i)} \frac{1}{|Out(N_j)|} \cdot S(N_j), \tag{1}$$

where $d \in [0, 1]$ is a damping factor that models the probability of randomly jumping from one node to another and $| \cdot |$ is the cardinality of a set.

The algorithm starts with arbitrary scores for the nodes and iteratively computes the new scores from (1) until a convergence condition is reached. Usually, this condition is expressed in terms of the difference between scores of two consecutive iterations, being satisfied when this difference is smaller than a threshold τ for every node.

The basic idea behind this ranking method is that of *voting* or *recommendation*: a page that points to another casts a vote or recommendation on that page, so that the importance of a page increases with the number of links that point to it. But, in addition to the number of votes, the importance conferred on a page also depends on the quality of the vote or recommendation issued, defined as the level of importance of the issuing page. Thus, recommendation becomes a mechanism for transmitting the importance of a page to the referenced page, this importance being evenly distributed among all the pages to which it points. Note that in PageRank all links are treated with the same level of importance. It is the importance of the transmitting page, and not of the link, that determines how much of its value is transmitted.

TextRank [17] is an extrapolation of PageRank to the text processing context where the nodes represent text units and the edges represent some relationship between them.

However, the strength of the relationship between the text units may vary depending on the degree of affinity between those text units. That is why it makes sense to extend the PageRank model by adding to the edges weights that represent such a level of affinity. The presence of weights associated with the edges turns out into a non homogeneous transmission of the importance of a node to the nodes to which it points, but proportional to the weight of each edge. The objective of these weights is to reflect the strength of the link between nodes and, ultimately, to define the capacity of the transmission channel. Now, the amount of relevance transmitted depends not only on the level of importance of the emitting node, but also on the capacity (weight) of the edge through which it is transmitted.

Therefore, TextRank considers weighted graphs, where each edge from N_i to N_j is associated with a weight w_{ij} that represents the strength of the connection between text units. In order to integrate these weights into the model, the score function (1) is replaced with the following one:

$$S(N_i) = (1-d) + d \cdot \sum_{N_j \in In(N_i)} \frac{w_{ji}}{\sum_{N_k \in Out(N_j)} w_{jk}} \cdot S(N_j). \tag{2}$$

Since our proposal is also based in a weighted graph, it is ultimately an extension of the TextRank model.

The way to calculate the weights is problem-dependent. For example, in the original TextRank article [17], the method is applied to extract summaries in the form of sentences. In that context, each node represents a sentence and the weights are defined through a measure of similarity between the related sentences based on the number of matching tokens between both. Other similarity measures such as string kernels, cosine similarity or longer common subsequence are also considered.

In any case, the specific definition of the weights used in the context of text processing is not relevant to our proposal, since the context of application of FuzzyFeatureRank will require a new definition that allow to assign a capacity to the transmission channel according to what is intended. This adaptation of weights to the context of FuzzyFeatureRank is described in the next section.

3. FuzzyFeatureRank: a new fuzzy feature reduction method

In supervised classification problems, we consider a set of d classes $\mathcal{C} = \{c_1, \ldots, c_d\}$, n features $\mathcal{F} = \{f_1, \ldots, f_n\}$ and m examples $\mathcal{E} = \{e_1, \ldots, e_m\}$, each example with the form $e_k = ([x_1^k, \ldots, x_n^k], c^k)$, where x_i^k is the value for the feature f_i and c^k is the class label. Each feature f_i is associated with a fuzzy variable X_i whose fuzzy domain is denoted as $\widetilde{\mathcal{X}}_i = \{LX_i^1, \ldots, LX_i^{p_i}\}$, being p_i the number of fuzzy values associated with the variable and being LX_i^j the linguistic label of its jth fuzzy value.[1]

The proposed feature reduction method falls into a category halfway between the feature selection and the feature extraction approaches. This is due to the fact that the underlying idea is not to select a set of the original features but to decompose each of them on a subset of new features in the form of fuzzy expressions and to select a subset of these new features. Concretely, each original feature f_i is decomposed into p_i fuzzy predicates $P_i^j : X_i$ is LX_i^j and, once the method is applied, the best fuzzy predicates are selected as new features.

[1] In case of categorical features, each value of the feature will be represented by a fuzzy singleton.

This allows, on the one hand, to modify the original features in order to improve their ability to distinguish among classes and, on the other hand, to have an interpretable description of these new features, in the sense that they are fuzzy expressions.

3.1. Entropy and mutual information of fuzzy predicates

In order to apply a TextRank-like ranking method we need to establish the weight between every pair of fuzzy predicates. These weights might measure the usefulness of adding the fuzzy predicate of the target node to the fuzzy predicate of the source node, and it will take into account two aspects: 1) the mutual information between both fuzzy predicates, and 2) the redundancy between the target predicate and the set of predicates already selected.

In order to calculate the mutual information and the information gain between two fuzzy predicates we are going to calculate the entropy many times in every iteration of the algorithm. We can consider each fuzzy predicate as a variable with values on [0, 1] and calculate some classical entropy or fuzzy entropy measure between variables, but then the complexity of the algorithm would be very high. Instead, we use a more simple entropy measure for these fuzzy predicates. It consists in applying the classical entropy formula, considering as probability the proportion between fuzzy sets, since it can be considered as a generalization of Laplace probability. It is not the aim of this paper to propose or discuss a new entropy measure, neither new mutual information nor information gain measures, but only to consider a simple and efficient alternative useful enough for the propose of our algorithm.

Let X be a non empty finite set, and let $\mathcal{P}_f(X)$ be the set of all fuzzy subsets of X. For every A in $\mathcal{P}_d(X)$, A can be identified by its membership function $A : X \to [0, 1]$. In this sense, $A(x) \in [0, 1]$ denotes the degree in which x belongs to the fuzzy set A.

It will be defined $Fcard(A) = \sum_{x\in X} A(x)$ as the fuzzy cardinality of the fuzzy subset A. It is obvious that if A is crisp then $Fcard(A) = |A|$.

Definition *(Proportion of elements in a fuzzy subset).* For every A in $\mathcal{P}_f(X)$, the proportion of elements in A is defined as

$$pr_X(A) = \frac{Fcard(A)}{|X|} = \frac{\sum_{x\in X} A(x)}{|X|}. \tag{3}$$

Propositions. *Given $A, A' \in \mathcal{P}_f(X)$,*

i) $pr_X(\neg A) = 1 - pr_X(A)$
ii) *if* $A \subseteq A'$ *then* $pr_X(A) \leq pr_X(A')$
iii) *if the product t-norm* $a \cdot a'$ *is used for* $\cap$ *and its dual t-conorm* $a + a' - a \cdot a'$ *is used for* $\cup$, *then* $pr_X(A \cup A') = pr_X(A) + pr_X(A') - pr_X(A \cap A')$
iv) $pr_X(X) = 1$
v) $pr_X(\emptyset) = 0$
vi) *if A is a crisp set, then* $pr_X(A) = prob(x \in A/x \in X)$

Proof. i) $pr_X(\neg A) = \frac{\sum_{x\in X} \neg A(x)}{|X|} = \frac{\sum_{x\in X}(1-A(x))}{|X|} = \frac{\sum_{x\in X} 1-\sum_{x\in X} A(x)}{|X|} =$
$= \frac{|X|-\sum_{x\in X} A(x)}{|X|} = 1 - pr_X(A),$
ii)-vi) *are obvious.* □

Similarly, if A and B are fuzzy subsets of X, the proportion of elements of A that belong to B can be defined.

Definition *(Proportion of elements of a fuzzy subset that are in other one).* For every A, B in $\mathcal{P}_f(X)$, the proportion of elements of A in B is defined as

$$pr(A/B) = \frac{Fcard(A \cap B)}{Fcard(B)}. \tag{4}$$

This is equivalent to the conditional probability in $\mathcal{P}_f(X)$, that is, if A and B are crisp sets then $pr(A/B) = p(A/B)$.

Propositions. *Given $A, A', B \in \mathcal{P}_f(X)$,*

vii) *if the product t-norm is used for $\cap$, $pr(\neg A/B) = 1 - pr(A/B)$*
viii) *if $A \subseteq A'$ then $pr(A/B) \leq pr(A'/B)$*
ix) *if the product t-norm $a \cdot a'$ is used for $\cap$ and its dual t-conorm $a + a' - a \cdot a'$ is used for $\cup$, then $pr(A \cup A'/B) = pr(A/B) + pr(A'/B) - pr(A \cap A'/B)$*
x) $pr(X/B) = 1$
xi) $pr(\emptyset/B) = 0$

Proof. vii) $pr(\neg A/B) = \frac{Fcard(\neg A \cap B)}{Fcard(B)} = \frac{\sum_{x \in X}(1-A(x)) \cdot B(x)}{\sum_{x \in X} B(x)} = \frac{\sum_{x \in X} B(x) - A(x) \cdot B(x)}{\sum_{x \in X} B(x)} =$
$= \frac{\sum_{x \in X} B(x) - \sum_{x \in X} A(x) \cdot B(x)}{\sum_{x \in X} B(x)} = 1 - pr(A/B),$
viii)-xi) *are obvious.* □

In this way, the proportion is an extension of the Laplace's probability and conditional probability from $\mathcal{P}(X)$ to $\mathcal{P}_f(X)$. It is not a probability because $\mathcal{P}_f(X)$ is not a Boolean algebra, but it has very similar properties. This measure will be used for extending entropy, information gain, and mutual information classical measures.

The decision problem consists in classifying each example into one of the d classes, based on the values of its features. Given a set of examples $\mathcal{E}$, the distribution of every class in $\mathcal{E}$ induces a probability on $\mathcal{C}$. Then, the entropy of $\mathcal{E}$ is defined as the entropy of this probability distribution, that is:

$$H(\mathcal{E}) = -\sum_{c \in \mathcal{C}} p(c/\mathcal{E}) \cdot \log_2 p(c/\mathcal{E}), \tag{5}$$

where $p(c/\mathcal{E})$ is the conditional probability of class c given the set $\mathcal{E}$:

$$p(c/\mathcal{E}) = \frac{|\mathcal{E}_c|}{|\mathcal{E}|}, \tag{6}$$

being $\mathcal{E}_c$ the subset of examples within class c. That is, $p(c/\mathcal{E})$ is the proportion of examples of $\mathcal{E}$ within class c.

Given a binary predicate P, the entropy of $\mathcal{E}$ considering this predicate is defined as

$$H(\mathcal{E}, P) = p(P/\mathcal{E}) \cdot H(\mathcal{E}^P) + p(\neg P/\mathcal{E}) \cdot H(\mathcal{E}^{\neg P}), \tag{7}$$

where $\mathcal{E}^P$ is the set of examples of $\mathcal{E}$ satisfying P, $\mathcal{E}^{\neg P}$ is the set of examples of $\mathcal{E}$ not satisfying P, $p(P/\mathcal{E}) = |\mathcal{E}^P|/|\mathcal{E}|$ and $p(\neg P/\mathcal{E}) = |\mathcal{E}^{\neg P}|/|\mathcal{E}|$.

Now the concept of entropy and information gain will be extended to the case of fuzzy features using the proportion of a fuzzy subset defined previously. Given a fuzzy predicate about some feature, that is, a fuzzy predicate $P_A : X_i$ is A, where A is a fuzzy subset on the domain of feature f_i, then it induces the fuzzy subset of $\mathcal{E}$ defined as the examples of $\mathcal{E}$ that verify P_A, $\widetilde{\mathcal{E}}^{P_A}(e_k) = A(x_i^k)$. The fuzzy subset of $\mathcal{E}$ that does not verify $P_A : X_i$ is A will be $\widetilde{\mathcal{E}}^{\neg P_A}(e_k) = 1 - A(x_i^k)$.

Definition *(Entropy of a fuzzy subset of examples).* Given a fuzzy subset of examples, $\widetilde{\mathcal{E}} : \mathcal{E} \to [0, 1]$, its entropy is defined as

$$H_f(\widetilde{\mathcal{E}}) = -\sum_{c \in \mathcal{C}} pr(c/\widetilde{\mathcal{E}}) \cdot \log_2 pr(c/\widetilde{\mathcal{E}}), \tag{8}$$

where $pr(c/\widetilde{\mathcal{E}}) = pr(\widetilde{\mathcal{E}}_c/\widetilde{\mathcal{E}})$.

In this way, if $\mathcal{E}$ is a crisp subset of examples then $H_f(\mathcal{E}) = H(\mathcal{E})$.

Definition *(Entropy of a fuzzy subset of examples considering a fuzzy predicate).* Given a fuzzy subset of examples $\widetilde{\mathcal{E}}$ and a fuzzy predicate P, the entropy of $\widetilde{\mathcal{E}}$ considering P is defined as

$$H_f(\widetilde{\mathcal{E}}, P) = pr(P/\widetilde{\mathcal{E}}) \cdot H_f(\widetilde{\mathcal{E}}^P) + pr(\neg P/\widetilde{\mathcal{E}}) \cdot H_f(\widetilde{\mathcal{E}}^{\neg P}), \tag{9}$$

where $\widetilde{\mathcal{E}}^P(e) = \widetilde{\mathcal{E}}(e) \cdot P(e)$ is the fuzzy subset of examples of $\widetilde{\mathcal{E}}$ satisfying P, and $\widetilde{\mathcal{E}}^{\neg P}(e) = \widetilde{\mathcal{E}}(e) \cdot (1 - P(e))$ is the fuzzy subset of examples of $\widetilde{\mathcal{E}}$ not satisfying P.

If $\mathcal{E}$ is a crisp set of examples and P is a binary predicate then $H_f(\mathcal{E}, P) = H(\mathcal{E}, P)$.

Now, this concept will be used to define the information gain obtained when fuzzy predicates are considered.

Definition *(Entropy associated with a single fuzzy predicate).* Given a fuzzy predicate $P_A : X_i$ is A, we consider the entropy associated with a single predicate

$$H_f(\mathcal{E}, P_A) = pr_{\mathcal{E}}(\widetilde{\mathcal{E}}^{P_A}) \cdot H_f(\widetilde{\mathcal{E}}^{P_A}) + pr_{\mathcal{E}}(\widetilde{\mathcal{E}}^{\neg P_A}) \cdot H_f(\widetilde{\mathcal{E}}^{\neg P_A}) \tag{10}$$

as a measure of the uncertainty associated to that predicate.

Definition *(Entropy associated considering a second fuzzy predicate).* Given a second fuzzy predicate $P_B : X_j$ is B, the entropy after the new fuzzy predicate is defined as

$$H_f(\mathcal{E}, P_A, P_B) = pr_{\mathcal{E}}(\widetilde{\mathcal{E}}^{P_A}) \cdot H_f(\widetilde{\mathcal{E}}^{P_A}, P_B) + pr_{\mathcal{E}}(\widetilde{\mathcal{E}}^{\neg P_A}) \cdot H_f(\widetilde{\mathcal{E}}^{\neg P_A}, P_B). \tag{11}$$

3.1.1. *Information gain by adding a second fuzzy predicate*

In the same way that it is defined the information gain in decision tree algorithms, the information gain by adding a second fuzzy predicate is now defined as

$$IG(\mathcal{E}, P_A, P_B) = H_f(\mathcal{E}, P_A) - H_f(\mathcal{E}, P_A, P_B). \tag{12}$$

3.1.2. *Mutual information of fuzzy predicates*

In the crisp case, the mutual information between two variables is defined as

$$MI(X, Y) = \sum_x \sum_y p(X = x, Y = y) \cdot \log_2 \frac{p(X = x, Y = y)}{p(X = x) \cdot p(Y = y)}. \tag{13}$$

In particular, given two binary predicates P_A and P_B, the mutual information of these two predicates is defined as

$$MI(P_A, P_B) = p(P_A, P_B/\mathcal{E}) \cdot log_2 \frac{p(P_A, P_B/\mathcal{E})}{p(P_A/\mathcal{E}) \cdot p(P_B/\mathcal{E})}. \tag{14}$$

Thus, the mutual information between two fuzzy predicates can be defined by extending (14) using proportion instead of probability.

Definition *(Mutual information between two fuzzy predicates).* Given two fuzzy predicates $P_A : X_i$ is A and $P_B : X_j$ is B, the mutual information between them is defined as

$$MI(P_A, P_B) = pr_{\mathcal{E}}(\widetilde{\mathcal{E}}^{P_A \cap P_B}) \cdot \log_2 \frac{pr_{\mathcal{E}}(\widetilde{\mathcal{E}}^{P_A \cap P_B})}{pr_{\mathcal{E}}(\widetilde{\mathcal{E}}^{P_A}) \cdot pr_{\mathcal{E}}(\widetilde{\mathcal{E}}^{P_B})}. \tag{15}$$

3.2. *Adaptation of TextRank to our fuzzy feature ranking method*

In order to adapt the TextRank method described in Section 2 to the context of fuzzy feature ranking, we have to define the nodes of the graph and the weights between nodes.

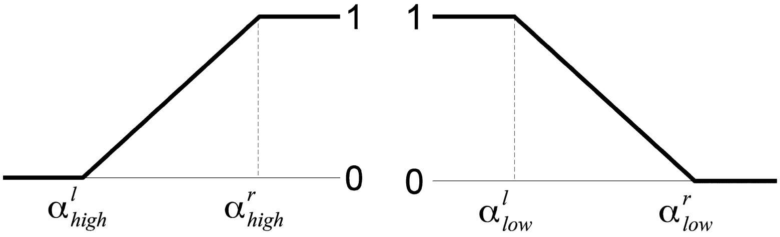
Regarding the nodes, the graph contains a node for each possible fuzzy predicate $P_i^j : X_i$ is LX_i^j, obtaining a graph with p_i nodes for each feature f_i. Therefore, the graph will contain a total of $\sum_{i=1}^n p_i$ nodes.

Besides, initially the graph is fully connected, although after the calculation of the weights described in the next section and according to (2), all edges with a weight equal to 0 are as if they did not exist.

Regarding the weights, a weight is attached to each edge that measures the usefulness of adding the fuzzy expression contained in the target node to the fuzzy expression contained in the source node in the light of the training set.

Algorithm 1 The FuzzyFeatureRank feature reduction algorithm.

```
1:  $C \leftarrow \{P_i^j\}, i = \{1, \ldots, n\}, j = \{1, \ldots, p_i\}$
2:  $S \leftarrow \emptyset$
3:  Initialize $W$: $w(\mathcal{E}, P_A, P_B) = high(IG(\mathcal{E}, P_A, P_B))$
4:  Delete irrelevant nodes from $C$
5:  **while** $|S| < n_{best}$ **and** $C \neq \emptyset$ **do**
6:      Rank $C$ using the adapted TextRank method
7:      $best$ = best predicate in $C$
8:      $worsts = n_{worst}$ worst predicates in $C$
9:      $S \leftarrow S \cup \{best\}$
10:     $C \leftarrow C \setminus \{best\} \setminus \{worsts\}$
11:     Update $W$: $w(\mathcal{E}, P_A, P_B) = high(IG(\mathcal{E}, P_A, P_B)) \times [\rho(P_B) \cdot low(MI(P_B, S))]$
12:     Delete irrelevant nodes from $C$
13: **end while**
14: **return** $S$
```



(a) Fuzzy set *high*.

(b) Fuzzy set *low*.

Fig. 1. Fuzzy sets used to normalize the information gain and the mutual information.

Concretely, a weight w_{AB} is assigned to the edge from node A to node B that measures the usefulness of adding the fuzzy predicate $P_B = P_{i_B}^{j_B}$ to the fuzzy predicate $P_A = P_{i_A}^{j_A}$.

This usefulness is measured in terms of the information gain achieved when adding the fuzzy predicate P_B to the fuzzy predicate P_A and the redundancy between the fuzzy predicate P_B and the set of already selected predicates in terms of mutual information.

Additionally, in order to grasp the individual potential of each fuzzy expression, an autoreference is added to each node with a weight that measures the usefulness of adding the predicate P_i^j it represents to the *void* predicate $P_i^{\emptyset} : True$.

3.3. The FuzzyFeatureRank algorithm

Algorithm 1 outlines the proposed FuzzyFeatureRank method. It takes as inputs the training set and the fuzzy domains and returns a set of selected fuzzy predicates as output. It can be described as follows:

- *Line 1*: The nodes of the graph are initialized with the set of all possible candidate fuzzy predicates, C.
- *Line 2*: The set of selected fuzzy predicates, S, is initially empty.
- *Line 3*: The weights between nodes are initialized based on how much *high* the information gain is when adding the predicate in the target node to the predicate in the source node (where the concept *high* is defined by the fuzzy set shown in Fig. 1a). The parameters α^l_{high} and α^r_{high} are set to the minimum and maximum values contained in the information gain matrix, respectively, thus being the weights 0 for the minimum value(s) contained in the information gain matrix, 1 for the maximum value(s) contained in the information gain matrix and some value between 0 and 1 for the remaining information gain values.
- *Line 4*: After the initialization of the weights, all the irrelevant nodes are deleted from C. An irrelevant node is the one that only receives zero weights, since the second addend is always zero in (2).
- *Line 5*: Iteratively, fuzzy predicates are added to S (one per iteration) until the desired number of selected predicates n_{best} is reached or the set of candidate predicates C becomes empty.

Table 1
Main information of the datasets.

Dataset	Features	Classes	Instances
IRIS	4	3	150
Wine	13	3	178
Vehicle	18	4	846
WDBC	30	2	569
Ionosphere	34	2	351
Sonar	60	2	208
Musk	166	2	476

- *Line 6*: The TextRank based ranking method is applied over the weighted graph, obtaining a ranked C set.
- *Lines 7–10*: The best fuzzy predicate is added to S and this node and the worst n_{worst} fuzzy predicates are deleted from C. The n_{worst} parameter is obtained as a percentage of the initial number of candidate fuzzy predicates (i.e., as a percentage of $\sum_{i=1}^{n} p_i$).
- *Line 11*: The weights of the remaining nodes are updated now considering, besides the information gain, the redundancy between the fuzzy predicate in the target node and the fuzzy predicates that S already contains. The redundancy is based on how much *low* the mutual information between the target fuzzy predicate and S is (where the concept *low* is defined by a fuzzy set shown in Fig. 1b), weighted by the ratio of examples satisfying the target fuzzy predicate (see Equation (3)). The mutual information between P_B and S is defined as $\max_{P_C \in S} MI(P_B, P_C)$, and the parameters α_{low}^{l} and α_{low}^{r} are set to 0 and 0.3, respectively.
- *Line 12*: After the update of the weights, irrelevant nodes are deleted from C.

4. Experimental results

4.1. Experimental setup

The experiments where applied to several classification datasets from UCI repository [19]: Iris, Wine, Vehicle, Wisconsin Diagnostic Breast Cancer (WDBC), Ionosphere, Sonar, and Musk (Version 1). Table 1 shows the main information for those datasets.

We used the same fuzzy partition with 5 triangular membership functions for every feature. Regarding the FuzzyFeatureRank parameters, n_{best} was set to the maximum number of possible predicates (predicates will be selected until the set of candidate predicates C was empty) and n_{worst} was obtained using a percentage of 4% of the initial size of C. Regarding the adapted TextRank method, the parameter d in (2) was set to the usual value 0.85 [16], and the threshold τ related with the convergence condition was set to 10^{-5}.

For each run of the classifier building, the dataset was randomly split in training and test subsets with a distribution of 80% and 20%, respectively, using a stratified sampling to maintain in each subset the same proportion of examples from each class existing in the whole dataset. Then, each method was applied to the training set and the classification performance was analyzed when applying the obtained fuzzy classifier to the test set. For each dataset, each method was run 5 times and the averaged results were obtained.

All methods were implemented in Python 3.6.8 and were run on a Core i7, 3.60-GHz CPU with 12 GB of memory in Windows 10.

4.2. Design of the fuzzy classifier

In order to evaluate the performance of our proposal in terms of accuracy, interpretability of the obtained fuzzy rules and computational cost, some fuzzy classifier design method must be selected. We selected the method proposed by Ishibuchi and Yamamoto in [5,6], on one hand, due to its good performance results in terms of accuracy and interpretability of the obtained fuzzy classifiers and, on the other hand, because it already tries to reduce the search space by restricting the size of the candidate fuzzy rule set. Therefore, our goal in this experimental stage is to support the benefits of our proposal in terms of accuracy, computational cost and/or interpretability even when the fuzzy classifier already has good results in those aspects.

Algorithm 2 Ishibuchi fuzzy classifier design method (according to [5,6]).

```
 1:                                                   ▷ Generate the set of candidate fuzzy rules
 2: $CRS \leftarrow \emptyset$
 3: for all antecedent $\mathbf{A}$ with up to $k$ predicates do
 4:     Select the rule $\mathbf{A} \rightarrow C$ where $C = \arg\max_{\{c_1 \ldots c_d\}} conf(\mathbf{A} \rightarrow c_i)$
 5:     if $CF(\mathbf{A} \rightarrow C) > 0$ then
 6:         Assign to the rule a weight equals to $CF(\mathbf{A} \rightarrow C)$
 7:         Add the rule $\mathbf{A} \rightarrow C$ to $CRS$
 8:     end if
 9: end for
10:                                                   ▷ Generate the set of final fuzzy rules
11: $FRS \leftarrow \emptyset$
12: for all class $C$ do
13:     Rank the rules in $CRS$ with consequent $C$ using $eval(\mathbf{A} \rightarrow C)$
14:     Add to $FRS$ the best up to $N$ rules
15: end for
16: return $FRS$
```

The method is outlined in Algorithm 2. Firstly, lines 2–9 generate a set of candidate fuzzy rules with all the possible combinations of up to k predicates in the antecedent (the remaining at least $n-k$ input variables are associated with a special *don't care* fuzzy value whose membership degree is 1 for all the universe of discourse). The consequent of each candidate rule is obtained using the class that entails the maximum confidence of the rule, defined as:

$$conf(\mathbf{A} \rightarrow C) = \frac{\displaystyle\sum_{e_i \in \mathcal{E}_C} \mathbf{A}(e_i)}{\displaystyle\sum_{e_i \in \mathcal{E}} \mathbf{A}(e_i)}, \tag{16}$$

where $\mathcal{E}_C$ is the set of examples in class C and $\mathbf{A}(e) = \prod_i LX_i(x_i)$ is the compatibility degree of the example e with the antecedent $\mathbf{A}$, being LX_i the fuzzy set associated with each fuzzy predicate in $\mathbf{A}$. Moreover, a weight is attached to each rule obtained as:

$$CF(\mathbf{A} \rightarrow C) = conf(\mathbf{A} \rightarrow C) - \sum_{c \neq C} conf(\mathbf{A} \rightarrow c). \tag{17}$$

The rules with no covering from the training set or with negative weights are discarded from the CRS.

Secondly, lines 11–16 generate the set of final fuzzy rules selecting the best up to N rules for each class. In order to rank the rules, we apply one of the several evaluation measures used in [5], proposed initially in [20,21] and defined as:

$$eval(\mathbf{A} \rightarrow C) = supp(\mathbf{A} \rightarrow C) - \sum_{c \neq C} supp(\mathbf{A} \rightarrow c) \tag{18}$$

where $supp(\mathbf{A} \rightarrow C)$ is the support of the rule $\mathbf{A} \rightarrow C$ defined as:

$$supp(\mathbf{A} \rightarrow C) = \frac{\displaystyle\sum_{e_i \in \mathcal{E}_C} \mathbf{A}(e_i)}{|\mathcal{E}|}. \tag{19}$$

In order to classify a new example, the weighted vote strategy applied in [6] is adopted. In this strategy, each fuzzy rule compatible with the example casts a weighted vote for its consequent class, where the strength of the vote is the product of the compatibility degree and the weight of the rule. Therefore, the resulting class for a new example e is:

$$C = \arg\max_{c} \left(\sum_{\mathbf{A} \rightarrow c} \mathbf{A}(e) \times CF(\mathbf{A} \rightarrow c) \right) \tag{20}$$

It must be noted that, although the parameter k tries to reduce the curse of dimensionality by restricting the number of fuzzy predicates in the antecedent of the rules, the number of rules in the CRS still quickly increases. Concretely,

Table 2
Comparison of FFR+Ish versus Ishibuchi methods.

Dataset	Accuracy (%)		Time (in secs.)		Rule# [length]	
	Ishibuchi	FFR+Ish	Ishibuchi	FFR+Ish	Ishibuchi	FFR+Ish
Iris	96.7	**97.3**	**1**	2	**6.0** [1.1]	6.6 [1.3]
Wine	95.0	95.0	12	12	12.0 [1.5]	**10.2** [1.5]
Vehicle	**54.1**	51.8	65	**30**	**16.0** [2.2]	18.4 [2.6]
WDBC	93.3	**94.0**	237	**63**	7.2 [1.0]	**6.8** [1.1]
Ionosphere	81.7	**82.8**	181	**45**	7.2 [2.1]	**6.8** [1.3]
Sonar	78.6	**79.5**	1050	**180**	**4.8** [1.1]	6.8 [1.1]
Musk	71.7	**73.5**	**240**	1444	**4.0** [1.7]	6.4 [2.4]

in the simplified case where all the n fuzzy domains have the same number of fuzzy sets p, the size of a CRS with up to k predicates in the antecedent of the rules (excluding the *don't care* predicates) is

$$size(CRS) = \sum_{i=1}^{k} p^i \times \binom{n}{i}. \tag{21}$$

Therefore, the reduction on the number of fuzzy predicates considered to generate the CRS selecting the most promising ones suppose a chance to overcome this curse of dimensionality when dealing with high-dimensional datasets.

4.3. *Analysis of the results obtained with the proposed method*

The analysis of the results was divided in two sections. In the first one, our goal is to analyze the benefit obtained when applying our FuzzyFeatureRank feature reduction method as a preprocessing step in the design of a fuzzy classifier. With this aim, we analyze the results of the original Ishibuchi method with the results obtained when including FuzzyFeatureRank as a preprocessing step (we name this combination FFR+Ish). In the second one, we compare the results of FuzzyFeatureRank feature reduction method with some classical feature selection methods.

4.3.1. *Analysis of FuzzyFeatureRank as a preprocessing step in building fuzzy classifiers*

Regarding the parameters of the Ishibuchi method k (maximum number of fuzzy predicates per antecedent), and N (maximum number of rules per class), in the absence of a criterion for fixing them, we used a range of values in our experiments. Concretely, due to the high difference in computational cost between the original Ishibuchi method and our FFR+Ish method for the same k, we used $k \in [2, 3]$ for Ishibuchi and $k \in [3, 4]$ for FFR+Ish. Moreover, in the case of the dataset Musk, the Ishibuchi method becomes intractable for $k = 3$ (according (21) the size of the CRS is about 94,000,000 rules) and, due to this, we used only $k = 2$ for the original Ishibuchi method when it was applied to this dataset. Regarding N, we used $N \in [1, 5]$ for both methods in all datasets.

Table 2 shows the performance of both methods in terms of accuracy, computational cost and interpretability of the obtained fuzzy rules. Each value corresponds with the averaged result over the 5 best values obtained in each of the 5 runs among the different FRS generated for the range of values for k and N considered in each case.

Regarding the accuracy, the winner method for each dataset (if any) is shown in boldface. The results show that, despite the expected computational cost savings, in most of the cases the accuracy also improved when reducing the search space of candidate fuzzy rules. The only exception was the dataset Vehicle, which, as we could see in the next section, seems to be averse to the feature reduction.

Regarding the computational cost, it is clear the benefit obtained when using FuzzyFeatureRank, specially in the case of high-dimensional datasets. A special mention must be made respecting Musk dataset, where the higher computational cost of FFR+Ish is due to the fact that the restriction on the antecedent of fuzzy rules was relaxed, allowing to search for fuzzy rules with up to 4 fuzzy predicates in the antecedent. This relaxation can not even considered in the case of Ishibuchi method, as explained before, and it can only deal with fuzzy rules with up to 2 fuzzy predicates in the antecedent. The increase in the search space gained by FuzzyFeatureRank was translated into a better accuracy.

Table 3
Accuracy of FuzzyFeatureRank versus other feature selection methods.

Dataset	Sel.feat.	ReliefF	MRMR	FFR
Iris	2	96.0%	96.0%	**97.3%**
Wine	4	91.1%	85.0%	**95.0%**
Vehicle	4	38.5%	36.4%	**51.8%**
WDBC	5	93.9%	92.3%	**94.0%**
Ionosphere	6	77.5%	82.3%	**82.8%**
Sonar	8	**80.0%**	71.0%	79.5%
Musk	13	64.0%	69.4%	**73.5%**
Average		77.3%	76.0%	**82.0%**

Finally, respecting the interpretability of the final fuzzy classifier, in all the cases the number and length of rules were quite low, considering the dimensionality of the datasets. When comparing both methods, except in the Vehicle dataset, FFR+Ish either sacrificed interpretability in exchange for accuracy (Iris, Sonar and Musk) or improved the interpretability with the same accuracy results (Wine) or even improved both performance criteria (Ionosphere and WDBC).

Therefore, in general, the results supports the benefits expected with our proposal.

4.3.2. Comparison of FuzzyFeatureRank with other feature selection methods

In this section, our proposal was compared with the results obtained with two feature selection methods: ReliefF [22] and Minimum Redundancy Maximum Relevance (MRMR) [23]. For these two methods the implementation from Scikit-feature feature selection repository [2] was used.

In order to compare the results in terms of accuracy, the features selected by these methods were the only ones considered as the input of the Ishibuchi method. That way, the search space was reduced by discarding some of the original features instead of discarding a set of fuzzy predicates. With the aim to consider a reasonable number of features to be selected in each dataset, we selected $\sqrt{n}$ features, where n is the original number of features.

In this experiment, the range of values for k and N was set to $[3, 4]$ and $[1, 5]$ for all the methods in all datasets.

The results shown in Table 3 are quite illustrative. Only in the Sonar dataset the ReliefF method achieved a better accuracy than FuzzyFeatureRank. Moreover, in Vehicle –the dataset where the original Ishibuchi method achieved the best accuracy– the results obtained from the other feature selection methods were much worse.

5. Conclusions

This work proposes FuzzyFeatureRank, a novel feature reduction method whose goal is to reduce the dimensionality of the search space when building fuzzy classifiers. It is presented an approach consisting in splitting the original features into a set of fuzzy expressions in the form of fuzzy predicates and then ranking them in order to select the best ones. The approach can be regarded as falling in a category halfway between feature selection and feature extraction, with the advantage of maintaining the physical meaning of the original features while allowing to translate them into a new feature space focused on improving the classification performance.

In order to validate the suitability of the proposal, the method was applied using several well known datasets to the design of a fuzzy classifier that, in addition to obtaining good results in terms of interpretability and accuracy, introduces by itself a mechanism for reducing the search space by restricting the size of the fuzzy rules being considered. Moreover, the proposal is also compared with other classical feature selection methods from the literature.

The results endorse the appropriateness of the proposed feature reduction method.

On one hand, the reduction of the search space entailed by its use as a preprocessing step in the design of a fuzzy classifier not only suppose a saving in computational time, but it also increases in general the accuracy of the classifier. This can be achieved due to the mitigation of the curse of dimensionality that involves the reduction of the search space for candidate fuzzy rules, which in turn allows to explore this search space in a more efficient manner.

On the other hand, the accuracy obtained with our feature reduction method improves in general the ones obtained with other classical feature selection methods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been financially supported by the Spanish Ministry of Economy and Competitiveness (MINECO), project FFI2016-79748-R, and cofinancied by the European Social Fund (ESF). Besides, this work has been partially supported by FEDER and Junta de Extremadura (GR18135) and FEDER NanoSen-AQM Project (SOE2/P1/E0569). Finally, authors want to thank Junta de Extremadura (GR18135) and FEDER (Fondo Europeo de Desarrollo Regional "Una manera de hacer Europa") for supporting the IB16042 project.

References

[1] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective, Neurocomputing 300 (2018) 70–79.
[2] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection, ACM Comput. Surv. 50 (6) (2017) 1–45.
[3] S. Khalid, T. Khalil, S. Nasreen, A survey of feature selection and feature extraction techniques in machine learning, in: 2014 Science and Information Conference, IEEE, 2014, pp. 372–378.
[4] J. Marin-Blazquez, Q. Shen, From approximative to descriptive fuzzy classifiers, IEEE Trans. Fuzzy Syst. 10 (4) (2002) 484–497.
[5] H. Ishibuchi, T. Yamamoto, Comparison of heuristic criteria for fuzzy rule selection in classification problems, Fuzzy Optim. Decis. Mak. 3 (2) (2004) 119–139, https://doi.org/10.1023/b:fodm.0000022041.98349.12.
[6] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, IEEE Trans. Fuzzy Syst. 13 (4) (2005) 428–435, https://doi.org/10.1109/tfuzz.2004.841738.
[7] D. Nauck, R. Kruse, A neuro-fuzzy method to learn fuzzy classification rules from data, Fuzzy Sets Syst. 89 (3) (1997) 277–288, https://doi.org/10.1016/s0165-0114(97)00009-2.
[8] N. Mitrakis, J. Theocharis, V. Petridis, A multilayered neuro-fuzzy classifier with self-organizing properties, Fuzzy Sets Syst. 159 (23) (2008) 3132–3159, https://doi.org/10.1016/j.fss.2008.01.032.
[9] J. Alcala-Fdez, R. Alcala, F. Herrera, A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning, IEEE Trans. Fuzzy Syst. 19 (5) (2011) 857–872, https://doi.org/10.1109/tfuzz.2011.2147794.
[10] E. Zhou, A. Khotanzad, Fuzzy classifier design using genetic algorithms, Pattern Recognit. 40 (12) (2007) 3401–3414, https://doi.org/10.1016/j.patcog.2007.03.028.
[11] J.A. Iglesias, P. Angelov, A. Ledezma, A. Sanchis, Evolving classification of agents' behaviors: a general approach, Evol. Syst. 1 (3) (2010) 161–171, https://doi.org/10.1007/s12530-010-9008-8.
[12] Y. Chen, J. Wang, Support vector learning for fuzzy rule-based classification systems, IEEE Trans. Fuzzy Syst. 11 (6) (2003) 716–728, https://doi.org/10.1109/tfuzz.2003.819843.
[13] S. Halgamuge, M. Glesner, Neural networks in designing fuzzy systems for real world applications, Fuzzy Sets Syst. 65 (1) (1994) 1–12, https://doi.org/10.1016/0165-0114(94)90242-9.
[14] M. Setnes, R. Babuska, U. Kaymak, H.v.N. Lemke, Similarity measures in fuzzy rule base simplification, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 28 (3) (1998) 376–386.
[15] H. Roubos, M. Setnes, Compact fuzzy models through complexity reduction and evolutionary optimization, in: Ninth IEEE International Conference on Fuzzy Systems. FUZZ- IEEE 2000 (Cat. No. 00CH37063), IEEE, 2000, pp. 762–767.
[16] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: bringing order to the web, Technical Report 1999-66, Stanford InfoLab, previous number = SIDL-WP-1999-0120, November 1999, http://ilpubs.stanford.edu:8090/422/.
[17] R. Mihalcea, P. Tarau, TextRank: bringing order into texts, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, 2004, pp. 404–414.
[18] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, R. Zadeh, Wtf: the who to follow service at Twitter, in: Proceedings of the 22nd International Conference on World Wide Web - WWW'13, ACM Press, 2013, pp. 505–514.
[19] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2017.
[20] A. González, R. Pérez, SLAVE: a genetic learning system based on the iterative approach, IEEE Trans. Fuzzy Syst. 7 (2) (1999) 176–191.
[21] L. Castillo, A. González, R. Pérez, Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm, Fuzzy Sets Syst. 120 (2) (2001) 309–321, https://doi.org/10.1016/s0165-0114(99)00095-0.
[22] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of relieff and rrelieff, Mach. Learn. 53 (1) (2003) 23–69.
[23] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.