Expert Systems With Applications 80 (2017) 58–74

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A new nearest neighbor classification method based on fuzzy set theory and aggregation operators

Soufiane Ezghari*, Azeddine Zahi, Khalid Zenkouar

Intelligent System and Application Laboratory (SIA), Faculty of Science and Technology, Fez, 2202, Morocco

ARTICLE INFO

Article history:
Received 12 November 2015
Revised 6 March 2017
Accepted 7 March 2017
Available online 8 March 2017

Keywords:
Nearest neighbor classification
Fuzzy set theory
Fuzzy analogy based classification
OWA operators
Quasi-arithmetic mean operators
Management of uncertainty and impreciseness

基于模糊类比分类
模糊语言学
赋予
结合

ABSTRACT

The Fuzzy Nearest Neighbor Classification (FuzzyNNC) has been successfully used, as a tool to deal with supervised classification problems. It has significantly increased the classification accuracy by considering the uncertainty associated with the class labels of the training patterns. Nevertheless, FuzzyNNC's limited methods fail to efficiently handle the imprecision in features measurement and the uncertainty induced by the choice of the distance measure and the number of neighbors in the decision rule. In this paper, we propose a new method called Fuzzy Analogy-based Classification (FABC) to tackle the FuzzyNNC limitations. In this work, we exploit the fuzzy linguistic modeling and approximate reasoning materials in order to endow FABC with intelligent capabilities, like imprecision tolerance, optimization, adaptability and trade-off. Hence, our approach is composed of two main steps. Firstly, we describe the domain features using fuzzy linguistic variables. Secondly, we define the classification process using two intelligent aggregation operators. The first one allows the optimization of the similarity evaluation, by defining the adequate features to be considered. The second one integrates a trade-off strategy within the decision rule, by using a global voting approach with compensation property. The integration of such mechanisms will increase the classification accuracy and make the FuzzyNNC approach more useful for classification problems where imprecision and uncertainty are unavoidable. The proposed FABC is validated on 验证 the most known datasets, representing various classification difficulties and compared to the many extensions of the FuzzyNNC approach. The results obtained show that our proposed FABC method can be adapted to different classification problems and improve the classification accuracy. Thus, the FABC has the best rank value against the comparison methods with high significant level. Moreover, we conclude that our optimized similarity and global voting rule are more robust to handle the uncertainty in the classification process than those used by the comparison methods.

1. Introduction

Classification is one of the most well-known tasks in supervised learning and data mining (Alpaydin, 2014; Witten & Frank, 2005). This task is fundamental to many application domains like computer vision, decision making, information retrieval, natural language processing, Bioinformatics, pattern recognition, etc. Indeed, knowing the class labels of the objects in each domain allows us to predict many of its properties and so act appropriately. Subsequently, classification problems aim to assign class labels to new patterns (objects) from a collection of correctly classified patterns, called the training set (TR). These problems are usually addressed by the supervised learning techniques, which may be grouped in two main categories according to the ways of representing the classification knowledge extracted from the TR:

- Abstraction-based techniques (AB) – these techniques extract the classification knowledge from the TR by building a structured and generalized model. The resulting model is then stored and used during the prediction step to characterize new patterns and to classify them into the predefined classes. The class labels for the new patterns are derived by using an inference mechanism related to the generated model. This approach appears in many models, such as Rule-Based Systems (RBS) (Gadaras & Mikhailov, 2009; Ishibuchi, Nakashima, & Morisawa, 1997), Decision Trees (DT) (Costa & La Neve, 2012; Suarez & Lutsko, 1999), Artificial Neural Networks (ANN) (Jang & Sun, 1995; Keles, Samet Hasiloglu, Keles, & Aksoy, 2007), and Support Vector Machines (SVM) (Lin & Wang, 2002).
- Instance-based techniques (IB) – unlike the AB approach, the main distinctive characteristic of these techniques is to store,

* Corresponding author.
E-mail addresses: soufiane.ezghari@usmba.ac.ma (S. Ezghari), azeddine.zahi@usmba.ac.ma (A. Zahi), khalid.zenkouar@usmba.ac.ma (K. Zenkouar).

http://dx.doi.org/10.1016/j.eswa.2017.03.019
0957-4174/

during the prediction phase, a set of instances selected from the TR (Aha, Kibler, & Albert, 1991). The classification process relies on the similarity between the new pattern and the stored instances. So, the estimated class label is derived based on the assumption that “similar patterns have similar class labels”. Hence, the new pattern is, firstly, compared to the stored instances so as to enable the selection of the most similar patterns; then the class label is selected from the correct class labels of its similar patterns. Several classifiers have been designed based on this approach such as, case-based reasoning classifiers (CBR) (Rezvan, Zeinal Hamadani, & Shalbafzadeh, 2013), and the Nearest Neighbors classifiers (NNC) (Cover & Hart, 1967; Hastie, Tibshirani, & Friedman, 2009), and Nearest Prototype Classifiers (NPC) (Triguero, Derrac, Garcia, & Herrera, 2012).

K Nearest Neighbors (K-NN) (Cover & Hart, 1967) is the most well-known classifier of the NNC approach. Due to its simplicity, popularity and usefulness for many domains, a wide variety of extensions have been developed to improve its performances and capabilities. The most effective improvement is the Fuzzy Nearest Neighbor Classification approach (FuzzyNNC) (Derrac, García, & Herrera, 2014), which is based on Fuzzy Set Theory (FST) (Zadeh, 1965). In fact, the FuzzyNNC methods use Fuzzy sets to model the degree of membership of each pattern to the all class labels in the problem. The obtained fuzzy membership degrees represent the uncertainty associated with the true class label to which every pattern belongs. Thus, the Fuzzy membership degrees allow FuzzyNNC to increase the classification accuracy in most application domains (Chen et al., 2013, 2011; Sim, Kim, & Lee, 2005; Torteeka & Chundi, 2014).

However, FuzzyNNC still suffers from some weaknesses. Indeed, it assumes that the patterns in TR are points in n-dimensional space. Thus, the features are represented by numerical values or intervals (in the case of categorical features) and the similarity between the patterns is measured by distances defined on a suitable metric space. Unfortunately, this is not always the case; especially in practical applications such as diagnostic classification problems where the uncertainty and the imprecision are unavoidable (Arsene, Dumitrache, & Mihu, 2015; Chattopadhyay, 2014; Gil-Aluja, 2004). In fact, in these applications the feature values usually come from human judgments that are always vague, imprecise and uncertain; definitely, the features are measured by linguistic terms such as very small, small, medium, old, large, etc.
感知
Moreover, the similarity is usually perceived, by humans, according simultaneously to the common and distinctive features of the compared items. So, the similarity is measured by terms like more similar, dissimilar, better than, close to, etc., according to common and distinctive features; for this reason, similarity measures should
揭示对比
reveal the contrast between the similarity and the difference between the unlabeled and training patterns (Tversky & Gati, 1978). On the other hand, the choice of the number, *K*, of neighbors to be considered in the decision rule is still problematic and the behavior of the existing K-NN enhancements according to this parameter is unclear and ambiguous.

赋予人性
To overcome this shortcoming, we propose in this work to humanize the classification process in the FuzzyNNC approach by introducing intelligent capabilities, like imprecision tolerance, optimization, adaptability and trade-off. So, we propose a new classification technique, called Fuzzy Analogy-Based Classification (FABC). This extends the FuzzyNNC approach by including specific mechanisms: from the representation of the classification knowledge extracted from TR; to the assessment of the similarity between patterns; and to the way in which the decision for the classification of a new pattern is taken. Our aim is to increase the accuracy of classification and enhance the reliability of the FuzzyNNC approach
可靠性
in complex application domains, where the imprecision and uncertainty are unavoidable. The FABC can be considered as an adaptation of the Fuzzy Analogy-Based software cost estimation model (FABSC) proposed in (Idri, Abran, & Khoshgoftaar, 2001). The above model was designed to predict the software cost, which is a numerical value, for new software projects. We note that, in contrast with the FABSC, the FABC provides a class label instead of a numerical value for new inputs. Hence, the proposed method is based, first, on the Fuzzy linguistic modeling approach (Herrera, Alonso, Chiclana, & Herrera-Viedma, 2009; Zadeh, 1975, 1965), which suggests representing the features describing the patterns by linguistic variables; this allows avoidance of the imprecision induced by the measurement of the feature values and obtaining understood classifiers. Here, each linguistic variable is described by a set of linguistic terms such as *small, medium, old, large, etc.*, where a linguistic term is represented by a fuzzy set associated with its membership function. Second, we use the aggregation operators (Grabisch, 聚合运
Orlovski, & Yager, 1998; Yager, 1988; Zimmermann & Zysno, 1980) to guide the classification process. Such operators are used with the aim to correctly handle the uncertainty, which arises during the similarity assessment and when voting to predict the class label for the new pattern. So, the similarity between the candidate pattern and each pattern in the TR is evaluated by the measure proposed in Idri and Abran (2001), which is based on the Ordered Weight Averaging (OWA) operator and RIM linguistic quantifiers (Yager, 1996). These RIM quantifiers, defined by linguistic terms like *most, some, all, etc.*, indicate the percentage of common and distinctive features that can be considered in the similarity evaluation. Subsequently, the known class labels, the membership degrees and the obtained similarities are used to generate membership degrees to each class by the means of a global voting rule. This rule is based on compensatory aggregation operators that enable the FABC model to overcome the issues of choosing the number of neighbors. Indeed, the compensatory operators hold a trade-off strategy, which allow the high voting values to compensate for the low voting values according to a compensation degree. This behavior, contrary to the neighbors voting rule, always ensures that the class label with the highest voting values is the one that has a high membership degree. Finally, the crisp class label for the candidate pattern can also be derived by aggregating those membership degrees. Furthermore, the proposed design of FABC provides a user-friendly model defined by linguistic terms, which allows the experts to easily initialize the parameter set.

In this paper, we describe the proposed FABC approach, focusing on the ways in which the fuzzy logic and aggregation operators are introduced in the NN classification process. Moreover, our FABC method is evaluated on different classification difficulties by using various datasets obtained from the keel repository (Alcalá et al., 2010). The obtained results are then compared to the most known Fuzzy Nearest Neighbor algorithms presented in Derrac et al. (2014). We also analyze the behavior of our classifier according to the aggregation parameters.

The remainder of the paper is organized as follows: Section 2 presents the related work, which describes the most popular Fuzzy Nearest Neighbor algorithms. Section 3 shows the background of our proposal. Section 4 describes the proposed classifier. Section 5 is devoted to the experimental framework. Section 6 presents and discusses the experiment results obtained when validating our classifier on various datasets and comparing it to the Fuzzy Nearest Neighbor methods. Finally, conclusions and future works are given in Section 7.

2. Related works

K-NN has gained considerable popularity and has proven useful for researchers in the supervised classification domain over many years. The main characteristic is the voting rule, which assigns, to the unlabeled pattern, the class label represented by a majority of its nearest K neighbors in the TR set. Many drawbacks of the K-NN classification rule have been identified and studied. The most critical are, first, the inefficiency because the size of the problem increases, regarding both the number of patterns in the TR set and the number of features that will be used in the evaluation of the similarity. Second, the low tolerance to the imprecision and the uncertainty, which appear because the TR set defines the decision boundaries among the classes as abrupt; this is not always the case. Third, the noise induced by the decision rule when inappropriately choosing the K parameter, and the majority voting technique, which is very inconvenient because it does not take in consideration the similarity between the new pattern and the K nearest patterns. Consequently, several extensions of the K-NN have been proposed in the literature in order to tackle these drawbacks. The improvements concern the evaluation of the similarity (Chen, Garcia, Gupta, Rahimi, & Cazzanti, 2009; Luukka & Kurama, 2013; Tolias, Panas, & Tsoukalas, 2001), the choice of the optimum parameter K (Ghosh, 2006), the techniques of weighting patterns and features (Jahromi, Parvinnia, & John, 2009; Wettschereck, Aha, & Mohri, 1997), the reduction of the training data (Triguero et al., 2012), and the handling of the uncertainty and the imprecision of knowledge (Jóźwik, 1983; Keller, Gray, & Givens, 1985). The later weaknesses were addressed by Fuzzy Set Theory (FST). From this combination between FST and K-NN emerges a new and distinctive area in the field of NN classification, called Fuzzy Nearest Neighbor Classification (FuzzyNNC) (Derrac et al., 2014). The main idea behind this approach is the use of fuzzy sets to represent the knowledge of the TR set and the decision rule to classify a new pattern. So, the patterns in the TR set belong to every class with different degrees of membership within the [0,1] interval, instead of belonging completely to one class. Jóźwik presented in (Jóźwik, 1983) the first proposal of FuzzyNNC approach; it introduced a learning scheme of class memberships, replacing the class label of each training pattern by a fuzzy membership to each class. After the learning phase, the final classification is performed similarly to K-NN, but every neighbor uses its membership to each class for the voting rule, instead of just giving one vote as in the crisp K-NN. This algorithm is considered as the baseline of several advanced methods of hybridization between the FST and K-NN rule. Thus, many extensions have been developed in the same sense. In Derrac et al. (2014) the authors present an interesting survey of fuzzy nearest neighbor algorithms and characterize theirs behaviors by the following:

- Membership degree to a class, which allows for modeling the uncertainty of the knowledge about the true class of each pattern. This characteristic defines the degree at which the pattern can be considered as a typical instance of the class. Therefore, the classification method must dictate the way in which the fuzzy membership degree of each training pattern in each class is computed.
- Similarity measure, which evaluates the degree of similarity between two patterns. It defines the degree of contribution of each pattern in the TR set to the voting rule, when classifying a new instance.
- Decision rule, which attempts to use the maximum relevant information by introducing membership degrees or similarity measures in the voting process and so deriving the estimates of membership degrees to each class for a new instance.

According to the above characteristics, we present the most known algorithms of the FuzzyNNC approach. The first and the most used one is FuzzyKNN, which was defined by Keller et al. (1985). FuzzyKNN introduced two principal modifications to the original K-NN:

- A fuzzy membership function that assigns membership degrees to each pattern in the TR set. It attributes a high membership degree to the instances close to the class center and low membership degrees to the instances close to the boundaries. In Keller et al. (1985) authors have proposed three ways for computing the membership degrees; the best performing one is defined by:

$$\mu_c(x_i) = \begin{cases} 0.51 + \left(\frac{v_c}{K}\right) * 0.49 \ if \ c = \omega \\ \left(\frac{v_c}{K}\right) * 0.49 \ otherwise \end{cases} \tag{1}$$

Where v_c is the number of the same class neighbors, and ω is the class label of x_i, also the two coefficients 0.51 and 0.49 ensure that the largest membership degree will be assigned to the class label ω.

- A new voting rule that includes additional information related to the reliability of the neighbors. The neighbors vote for each class, by using the computed membership values. In addition, the votes can be weighted using the inverse of the distance. Finally, all votes are aggregated and the new pattern takes the class with the greatest combined vote.

Subsequently, many other extensions of the FuzzyKNN are introduced. Most of them try to deal with the uncertainty induced by the similarity measure and the choice of the number of neighbors K by fitting, either the decision rule or the function that measures the membership degree to a class in the TR. The FuzzyKNN extensions can be categorized in four approaches:

- Generalized FuzzyKNN (Bezdek, Chuah, & Leep, 1986): this approach tries to avoid the uncertainty induced by the choice of the number of neighbors K in the measure of the membership degree to a class measure. To complete this task, the generalized FuzzyKNN first generates a set of identity patterns from the TS for each class by the means of clustering algorithms. Thus, identity patterns are generated in the way that is the most representative of all the patterns in the TS, and then allow for the measure of an accurate membership degree to a class.
- Weighted FuzzyKNN (Han & Kim, 1999; Pham, 2005): this approach introduces a computational schema, based on the analysis of a discriminant function, to derive optimal weights for each pattern in the TS. The obtained weights are combined with the membership degrees in the decision rule in order to adjust the voting values. This approach allows for the substitution of the inverse of distance measure by a set of optimal weights. in order to avoid the uncertainty induced by the distance measure.
- Evolutionary FuzzyKNN (Derrac, Chiclana, García, & Herrera, 2016; Hu & Xie, 2005): this approach extends the FuzzyKNN method by introducing a new parameter m that adjusts the weights in the decision rule. In order to handle the uncertainty involved in the selection of K and m, an evolutionary optimization algorithm is dedicated to properly initialize those parameters. Accordingly, the selection of the most optimal parameters K and m, by using an evolutionary optimization algorithm, can respectively increase the relevance of the membership degree to a class and improve the decision rule accuracy.
- Alternative uncertainty representation in FuzzyKNN (Denoeux, 1995; Rhee & Hwang, 2003): this approach extends the membership degree of a class to another uncertainty representation concept. For instance, in Rhee and Hwang (2003) the authors use the Interval Type-2 Fuzzy Sets in order to manage the uncertainty of the choice of the number of neighbors K. Here, the idea is to use a range of K values rather than just one. In this way, the uncertainty induced by choosing an undesirable K value can be decreased. On the other hand,

the authors in Denoeux (1995) use the Dempster-Shafer theory against the Fuzzy Set theory to represent the uncertainty. This method allows for an overall treatment of several issues such as ambiguity, distance rejection and imperfect knowledge regarding the class membership of training patterns.

Although the reviewed extensions show remarkable improvement in classification accuracy against the FuzzyKNN method, some drawbacks should be considered, such as:

- All approaches ignore the imprecision in feature measurement. Indeed, the measurement of features is usually subject to error. Certainly when it comes from human judgements that are always vague, imprecise and uncertain.
- Most approaches still use distance functions as a similarity measure. Indeed, the choice of a distance measure is not based on any principle of optimality. Thus, it's not always evident for representing the features in Euclidian space.
- The approaches that can choose just one number of neighbors K still suffer from the uncertainty induced by this parameter, even if it is selected by using optimization algorithms (see Section 4.3).

In this work, we propose a new FABC method to more efficiently handle the imprecision and uncertainty in the FuzzyNNC approach. In our method, we suggest the representation of features using linguistic variables, instead of the provided feature values. This allows us to capture and to avoid the errors arising from features measurement. Also, we dedicate a powerful component to deal with the uncertainty induced by both the similarity measure and the choice of the number of neighbors. Indeed, likewise in the weighted FuzzyKNN approach, we calculate a set of weights by means of similarity relation. The latter can be optimized in each classification problem by selecting the appropriate RIM quantifier. Moreover, instead of using several numbers of neighbors K, we use an efficient global voting rule guided by a parametrized compensation aggregation operator. Thus, the voting rule is fitted by finding the optimal compensation level. In the next sections, we present a detailed description of our proposed FABC method.

3. Background concepts

This section covers the background concepts necessary to describe our proposal. Section 3.1 gives background information about fuzzy linguistic modeling and Section 3.2 describes aggregation operators.

3.1. Fuzzy linguistic approach

定性知识

Fuzzy linguistic modeling (FLM) has emerged as a powerful approach for modeling the qualitative knowledge of data mining problems (Fernández & Herrera, 2012; Herrera et al., 2009). It has been satisfactorily used in various applications, especially when the noise, the uncertainty and the imprecision are unavoidable; the fuzzy rule-based systems are the best known models that use the FLM approach to design fuzzy classifiers. FLM uses the concept of linguistic variables (Zadeh, 1975) to represent the variables describing the problem under modeling. According to Zadeh, a linguistic variable is a variable with values that are not numbers but
语言变量
linguistic terms such as *small, tall, hot, old, young, etc*. The main purpose of using linguistic values instead of numbers is that linguistic characterizations are, in general, less specific than numerical ones, but much closer to the way that humans express and use their knowledge. For example, in medical diagnostic applications, the patient's age can be described by qualitative values that can be modeled by the linguistic variable *Age*, which takes its values in the linguistic term-set: *child, young, adult and old*. Undoubtedly,

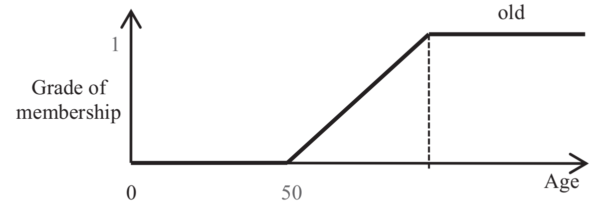

Fig. 1. The membership function associated with the fuzzy set of the linguistic value old.

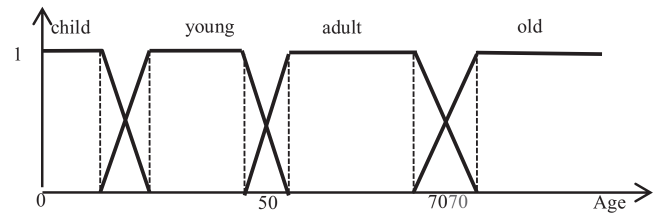

Fig. 2. Representation of the linguistic variable age with four fuzzy terms.

in order to derive some decisions about a patient, it is not necessary to know his exact age. So, we can use expressions such as "if a patient is young or old, then he needs a hospitalization".

Formally, a linguistic variable is characterized by a quintuple $(X, (X), U, S, M(X))$, where X is the name of the variable, $T(X)$ is the term-set containing linguistic values taken by X, U is the universe of discourse (domain of X), S is the syntactic rule which 句法规
generates the linguistic terms in $T(X)$, and M is the semantic rule 语义规
which assigns the meaning of each linguistic term. The semantic rule is usually designed with the aim of correctly modeling the gradual nature of linguistic terms defined in $T(X)$. Fuzzy Set Theory, founded by Zadeh (1965) seems to be suitable for this aim. Indeed, FST introduced the concept of graded membership to a set where each element of the universe of discourse belongs to a fuzzy set with a certain degree. Effectively, a fuzzy set is associated with a gradual membership function that assigns a degree of membership, in the real interval [0,1], to each element of the universe of discourse. Various forms such as triangular, trapezoidal or Gaussian can represent membership functions. For that, each linguistic term in $T(X)$ is described by a fuzzy set associated with its membership function. Fig. 1 shows a fuzzy set representation of the linguistic term "old" by a semi trapezoidal function.

Consequently, we can say that a linguistic variable is described by a fuzzy partition composed of fuzzy subsets associated with their membership functions. The underlying fuzzy sets overlap as a natural consequence of their boundaries. Each fuzzy set covers, in a unique way, a part of the variable domain and shares another part with adjacent fuzzy sets. Such an overlap implements a realistic and practical semantic rule, which captures the gradual nature of a linguistic variable and therefore provides a smooth and coherent transition from one state to another. To illustrate this purpose, Fig. 2 shows a fuzzy partition representing the linguistic variable *Age*.

3.2. Aggregation operators

In the last few decades, aggregation techniques have received considerable attention, with many results in term of properties and potential applications. They are widely used in decision-making tasks and pattern recognition contexts to guide some parameters of the underlying classifiers (Cordon, Herrera, del Jesus, Villar, & Zwir, 2000). In a rather informal way, an aggregation technique consists of summarized information, usually contained in an n-tuple of input values into a single representative value; so that the final result of aggregation takes into account, in a given manner, all the individual values. In this scope, an aggregation operator is simply a function, F, that assigns a value, $y = F(x_1, x_2 \ldots x_n)$, to any n-tuple $(x_1, x_2 \ldots x_n)$. Several aggregation operators have been developed and studied in the literature (Detyniecki, Bouchon-meunier, Yager, & Prade, 2000; Mitra, Murthy, & Pal, 2002). The conjunctive and the disjunctive operators called, respectively, t-norms and t-conorms (Cordon et al., 2000) are prototypical examples of such operators. The t-norm aggregation provides a way of implementing the type of "anding" aggregation; which is required when the aggregated elements should be connected by the logical "and". The t-conorm aggregation provides a way of implementing the type of "oring", which is required when the aggregated elements should be connected by the logical "or". In contrast, averaging aggregation operators (Grabisch, Marichal, Mesiar, & Pap, 2011; Yager, 1988) are a kind of aggregation that allows for the easy adjustment of the degree of "anding" and " oring" by generating OR operator (Max), AND operator (Min) and any other aggregation operator between them. A formal definition of mean aggregation operator can be defined as follows:

Definition 1. Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of values to aggregate in [0,1], a mean operator is a mapping $F : \mathbb{R}^n \to \mathbb{R}$ that has the following properties:

幂等
- Idempotence: $F(x, x, \ldots, x) = x$, if all $x_i = x$.
- Boundness: $F(0, 0, \ldots, 0) = 0$ and $F(1, 1, \ldots, 1) = 1$.

单调性
- Monotonicity, also called non-decreasing; which characterizes the fact that if an argument increases the aggregation increases too:

  $$F(x_1, x_2, \ldots, x_n) \geq F(y_1, y_2, \ldots, y_n),$$
  $$if\ (x_1, x_2, \ldots, x_n) \geq (y_1, y_2, \ldots, y_n)$$

- Compensation that allows the operator to compensate the bad (resp. good) score on one element by a good (resp. bad) one on another element:

  $$Min(x_i) < F(x_1, x_2, \ldots, x_n) < Max(x_i)$$

In the following, we describe the well-known mean type aggregation methods, namely the quasi-arithmetic mean operators (Grabisch et al., 1998) and the Ordered Weighted Averaging operators (Yager, 1988), commonly called OWA operators.

Definition 2. (Quasi-arithmetic mean operator) — Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of values to aggregate. A quasi-arithmetic operator is simply a function $M_f : \mathbb{R}^n \to \mathbb{R}$ that assigns a value for A by the following formulas:

$$M_f(a_1, \ldots, a_n) = f^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} f(a_i)\right) \tag{2}$$

Where f is any continuous strictly monotonic function, called generator of the operator M_f.

The quasi-arithmetic means operators define a family of arithmetic mean operators based on the generator function. This provides different behaviors, with a wide spectrum of mean operators, including arithmetic, quadratic, geometric, harmonic, root-power and exponential means (Table 1).

Yager introduced the OWA operators in Yager (1988). They are weighted mean type aggregation operators used when the weights of importance and the degree of satisfaction on aggregated elements are required. Formally, an OWA operator is defined as follows:

Definition 3. Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of values to aggregate in [0,1]. An OWA operator of dimension n is defined by

Table 1
Examples of quasi-arithmetic means.

f(x)	Name
X	Arithmetic mean
x^2	Quadratic mean
$Logx$	Geometric mean
x^{-1}	Harmonic mean
$x^{\alpha}(\alpha \in \mathbb{R}\backslash\{0\})$	Root-mean-power
$e^{\alpha}(\alpha \in \mathbb{R}\backslash\{0\})$	Exponential mean

the mapping $F : \mathbb{R}^n \to \mathbb{R}$ that has an associated weighting vector $W = (w_1, w_2, \ldots, w_n)$, Such as:

$$w_i \in [0,\ 1], 0 \leq i \leq n, \text{ and}$$

$$\sum_{i=1}^{i=n} w_i = 1$$

Furthermore, the aggregated value of A is obtained by the following formulas:

$$F(a_1, \ldots, a_n) = \sum_{i=1}^{n} w_i b_i \tag{3}$$

Where b_i being the i^{th}largest element in the aggregated values $a_1, \ldots, a_n$.

The most important issue when using OWA operators is the determination of the associated weight vector. Yager (1996) shows that linguistic quantifiers introduced by Zadeh (1983) can easily interpret OWA operator behaviors. He suggested the use of Regular Increasing Monotone Quantifiers (RIM) which provide information aggregation by verbal descriptions such as *al*', *most, many, at most* α, or *there exists*. A RIM quantifier can be represented by a fuzzy set, Q, of the unit interval [0,1]. In this representation for any $r \in [0, 1]$, $Q(r)$ is the degree to which the proportion r satisfies the concept represented by Q. Furthermore, the fuzzy set Q is presenting the following proprieties:

$$Q(0) = 0$$

$$Q(1) = 1$$

$$Q(r_1) > Q(r_2),\ r_1 \geq r_2$$

Finally, the weights are obtained by the following formulas:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right) i = 1, \ldots, n \tag{4}$$

Where n is the number of aggregated values and Q is the linguistic quantifier.

4. The proposal: Fuzzy Analogy-Based Classification (FABC)

A classification problem in a given domain can be defined by a triplet $D = \{F, C, TR\}$, where $F = \{f_1, f_2, \ldots, f_t\}$ is the set of features describing the domain, $C = \{C_1, \ldots, C_M\}$ is the set of class labels previously defined by the expert domain, $TR = \{e_1, e_2, \ldots, e_n\}$ is the training set with the number of correctly labeled and classified patterns. In this paper, we propose a new classifier, FABC, designed as an extension of the FuzzyNNC approach. Therefore, we introduce enhancement techniques in the FuzzyNNC process as the following:

- Fuzzy Linguistic Modeling (FLM) is used to describe the features selected for the classification problem.

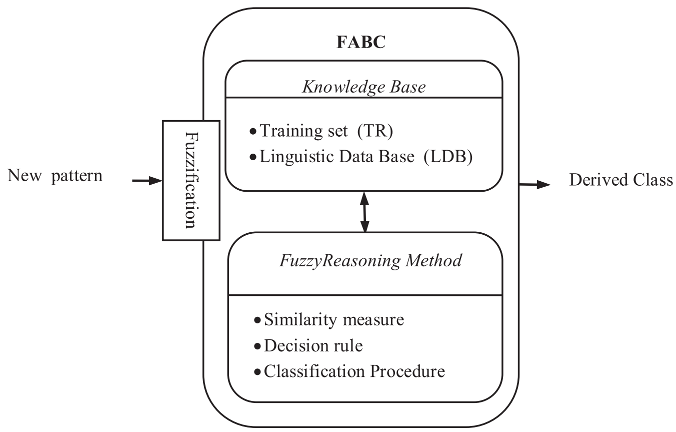

Fig. 3. The structure of the FABC.

- A new function is reviewed from the one defined by the Eq. (1) in order to determine the membership degrees of the training patterns of the TR to each class.
- We propose the use of a new similarity measure, which is based on fuzzy sets and guided by OWA operators.
- A new decision rule is also proposed. It uses quasi-arithmetic mean operators to select the most similar patterns and to derive a class label for a new input.

The structure of the FABC is composed of two components (Fig. 3):

- Knowledge base (KB) contains the knowledge of the training dataset, with the TR and the linguistic variables describing the selected features. The KB is determined in the training phase of the classification process.
- Fuzzy Reasoning Method (FRM) performs the predicting phase to derive the class label for a new input pattern. It includes a fuzzy similarity measure, a decision rule to aggregate the votes of the similar patterns and a classification procedure.

In the following, we describe the components of the proposed classifier. Section 4.1 describes the Knowledge base. Section 4.2 presents the similarity measure. Section 4.3 shows the classification procedure.

4.1. Knowledge base

经验技巧

The Knowledge base (KB) is composed of two parts:

- Training set (TR) which contains a set of correctly classified patterns, where each pattern, e_j, is defined by:
  - The values of the features, says $\{e_j^{f_1}, e_j^{f_2}, \ldots, e_j^{f_t}\}$; in this work, we consider just the numerical features.
  - The correct class label e_j^C.
  - The membership degrees of e_j to each class C_i of C; this can make out which pattern is close to the center of C_i and which pattern is close to the boundaries. To compute these degrees we extend the reviewed Eq. (1) as the following:

$$\mu_{C_i}(e_j) = \begin{cases} 0.51 + \left(\dfrac{\sum_{e_k \in lb_i} d_{jk}}{\sum_{k=1}^{N} d_{jk}} \right) * 0.49 \ \textit{if} \ e_j^C = c_i \\ \left(\dfrac{\sum_{e_k \in lb_i} d_{jk}}{\sum_{k=1}^{N} d_{jk}} \right) * 0.49 \ \textit{otherwise} \end{cases} \tag{5}$$

Where:

$\mu_{C_i}(e_j)$ defines the membership degree of e_j to the class label C_i.
N is the total number of patterns in the TR.
lb_i is the set of patterns labeled with the class C_i.
$d_{jk} = Sim(e_j, e_k)$ is the similarity, between e_j and the patterns, e_k, of the TR. It is calculated by the equation 7.

语义

- Linguistic Data Base (LDB), which contains the semantics of the linguistic variables and models the features of the classification problem. Thus, each feature is represented by a Strong Fuzzy Partition (SFP) of an adequate number of fuzzy sets defined by their membership functions; triangular shape is retained 三角形 for intermediate fuzzy sets and semi-trapezoidal shape of the 中间模 first and the last ones. Fig. 4 shows an example of SFP with k linguistic terms. The estimation of membership functions associated with linguistic variables is fundamental to the performance of fuzzy systems. In the literature, many techniques have been proposed; they may be grouped into two major categories: (1) empirical techniques which construct membership functions from expert knowledge (Idri, Abran, & Kjiri, 2000; Sicilia, Cuadrado-Gallego, Crespo, & García-Bariocanal, 2005), and (2) automatic techniques, which construct membership functions, from training data. An overview of these techniques is presented in Medasani, Kim, and Krishnapuram (1998); the most popular of them are based on clustering methods (Au, Chan, & Wong, 2006; Guillaume & Charnomordic, 2004; Idri, Zahi, & Abran, 2006). Here, we use the technique proposed in Idri et al. (2006). It consists of two main steps sketched by the algorithm presented in Fig. 5. In the first step, the numerical data describing the feature, is partitioned into an adequate number of clusters (fuzzy sets) using, the well-known clustering algorithm (Fuzzy C-Means — FCM) with a clustering validity index such as those proposed in Kim, Lee, and Lee (2004), Su, Chen, and Yih (2006), Xie and Beni (1991). In fact, we apply FCM with various numbers of clusters; then we select the adequate number of fuzzy sets by means of optimizing the validity index. In the second step, we build membership functions from the previously generated cluster centers. Fig. 4 shows an exam-

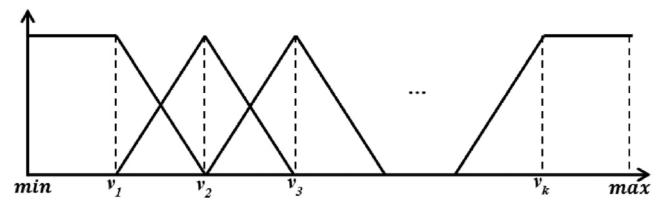

Fig. 4. Membership functions built from the clusters center v_j.

Procedure StrongFuzzyPartition(f_i)
CL : *empty cluster centers of a condidate partition*
FP : *optimal fuzzy partition*
VL : *optimal validity index*
$[m, n]$: *number of clusters interval*
for $k \leftarrow m$ to n
$CL \leftarrow FCM(f_i)$
$TVL \leftarrow validityIndexof(CL)$
if $isOptimal(TVL, VL)$
$VL \leftarrow TVL$
$FP \leftarrow CL$
end if
end for
$SFP \leftarrow constructing\ membership\ funcions\ from\ FP$
Return SFP

Fig. 5. Algorithm of constructing the strong fuzzy partition for a feature f_i.

ple of a linguistic variable described by k fuzzy sets generated from the k clusters center v_j.

4.2. Fuzzy similarity measure

Similarity is a key concept in the context of classification when using an analogy-based classification approach. The choice of a similarity measure is very important since it will influence the selection of analogies that will contribute to the classification step (Wettschereck et al., 1997). The most frequently used functions in classical and fuzzy NNC approaches are those based on conventional distances like Euclidean, Manhattan, Chebyshev or Minkowski (Witten & Frank, 2005). These measures may usually lead to erroneous similarities and, consequently, to deterioration in classification performances when redundant, irrelevant and high correlated features describe the patterns. To overcome this problem many authors propose to weight each feature in the assessment of the similarity. The weights assigned to features are determined according to their importance for the classification (Fernández & Isasi, 2008; Luukka & Kurama, 2013; Paredes & Vidal, 2006; Sáez, Derrac, Luengo, & Herrera, 2014). Therefore, a weighing similarity function assigns a higher weight to the features that are helpful to the classification and a low weight to the harmful features. The weighting methods are also insufficient because they favor the minority at the expense of the majority, or the other way around. In this respect, the authors of Luukka and Kurama (2013) proposed to use ordered weighted averaging operators (OWA) in the evaluation of the similarity in order to capture the tradeoff between bad and good features.

The similarity functions referred to above present several advantages, in terms of simplicity of description, efficiency of computation, and improvements of the classification performances. In spite of these promising benefits, these similarity functions are unable to model the similarity in many real-world classification problems such as medical diagnostics, biological management systems, and remote sensing. Indeed, the experts in such domains usually measure the features by using human terms such as *low, high, few, all, etc.,* which can be modeled by linguistic variables. Moreover, the principles of the overhead similarity measures do not correspond to the human perception of the similarity; humans perceive the similarity between two patterns according to their common and distinctive features in the same time (Tversky & Gati, 1978). To avoid those restrictions, we propose the use of a variant of the measure proposed in Idri et al. (2001). It is a general measure based on a fuzzy logic framework (Idri & Abran, 2000) and the Ordered Weighted Averaging operator (OWA). To define this measure, we consider two patterns $e_1 = \{e_1^{f_1}, e_1^{f_2}, \ldots, e_1^{f_t}\}$ and $e_2 = \{e_2^{f_1}, e_2^{f_2}, \ldots, e_2^{f_t}\}$, described by t features $\{f_1, f_2, \ldots, f_t\}$ modeled by linguistic variables. The similarity between e_1 and e_2 is evaluated by the following steps:

- *Individual similarity*: evaluates the similarity between e_1 and e_2 according to one feature, say f_i. Many similarity measures have been proposed in Idri et al. (2001) such as sum-product, max-min, etc., we have retained the sum-min measure defined by the following formulas:

$$Sind_{f_i}(e_1, e_2) = \sum_{j=1}^{k} \min\left(\mu_j\left(e_1^{f_i}\right), \mu_j\left(e_2^{f_i}\right)\right) \quad (6)$$

Where:
$Sind_{f_i}$ is the individual similarity according to the feature f_i,
μ_j is the membership function associated with the jth fuzzy set of the feature f_i; $j = 1, \ldots, k$,
k is the number of fuzzy sets defined for the feature f_i,

We can distinguish that when the individual similarity, $Sind_{f_i}(e_1, e_2)$, is high we say that the feature f_i is common to the samples e_1, e_2. In the contrast, when the individual similarity $Sind_{f_i}(e_1, e_2)$ is low, we say that the samples e_1, e_2 are distinct according to the feature f_i.

- *Overall similarity*: evaluates the overall similarity between e_1 and e_2 according to all features by combining the individual similarities, $Sind_{f_i}(e_1, e_2)$, using an OWA operator. So the overall similarity, $Sim(e_1, e_2)$, is associated with a weighting vector, $W = (w_1, w_2, \ldots, w_n)$, such as:

$$Sim(e_1, e_2) = \sum_{i=1}^{t} w_i Sind_{f_i}(e_1, e_2) \quad (7)$$

Where
$Sind_{f_i}(e_1, e_2)$ is the i^{th} largest individual similarity calculated by equation 6.

$$\sum_{i=1}^{i=n} w_i = 1, \ \ 0 \le w_j \le 1$$

We obtain a high overall similarity when we consider just the common features and a low overall similarity when we consider just the distinctive features. Therefore, by using a weighting vector, we can define the proportion of the contribution of each feature in the similarity assessment. So, when we choose $w_1 = 1$, we use only the most common feature and when we select $w_t = 1$, we consider just the most distinctive feature and when we pick different values for each w_i, we employ fairly distinctive and common features.

Hence, the similarity measure presented above seems to be well suitable for the context of classification, especially when irrelevant features and linguistic terms describe patterns. Indeed, by appropriately selecting the weighting vector W, we can adapt the similarity measure in order to obtain correct analogies and accurate classifications.

4.3. Decision rule

In our FABC model, we propose a new decision rule to derive a classification for a new pattern. It is an adaptive and global voting technique, which measures, for an input pattern, the membership degrees to each class label of C. The proposed technique suggests the use of the quasi-arithmetic mean operators to aggregate the overall voting values associated with each input pattern neighbor. The rule enables the tradeoff between the voting values of the different neighbors in the sense that the high voting values compensate the low voting values. 补偿运算符

The use of compensatory operator, in the decision rule, can resolve the dilemma of choosing the number, K, of neighbors. Indeed, 困境 the K-NN decision rule does not always tend to the class label that has the highest voting values. So, there is no uniform behavior of the decision rule according to the number of neighbors. More precisely, the small values of K do notalways take into account some situations, such as where additional neighbors are needed to make a correct decision. On the other hand, high values of K induce typical noises by means of neighbors with low similarities; the accumulation of these low similarities can degrade the final classification decision. This is due to the lack a knowledge of other interesting voting values, sensitivity to the noise and the absence of tradeoff mechanisms in the voting values accumulation. Hence, the introduction of an adaptive and global behavior, by means of compensatory operators, leads to a powerful decision rule acting as follows:

- The decision is guided by high similarities. The more the voting values include high similarities, the more the associated

Table 2
The illustrative example.

	S_1		S_2		S_3	
	C_1	C_2	C_1	C_2	C_1	C_2
e_1	0.88	0.9	0.91	0.89	0.9	0.92
e_2	0.85	0.88	0.9	0.86	0.85	0.91
e_3	0.8	0.6	0.8	0.85	0.85	0.3
e_4	0.8	0.4	0.71	0.74	0.8	0.2
e_5	0.7	0.4	0.7	0.71	0.75	0.1
e_6	0.7	0.3	0.7	0.3	0.5	0.1
e_7	0.6	0.25	0.68	0.2	0.4	0.1
e_8	0.6	0.23	0.65	0.2	0.3	0.1
e_9	0.6	0.2	0.65	0.1	0.2	0.1
e_{10}	0.6	0.1	0.5	0.1	0.1	0.1

相反

membership degrees increase. Inversely, the membership degrees decrease when low similarities are considered. In fact, our decision rule is helpful in confusing situations when the voting values of two candidate class labels have almost the same high similarities. Thus, the subsequent similarities help to remove the ambiguity and therefore lead to a correct decision.

- The decision is parameterized by the compensation level, which defines the level with which the high voting values compensate the low voting values. Implicitly, the aggregation result tends toward the high voting values when increasing the compensation level. Moreover, the compensation level is measured in the scale *low, medium* or *high*. So, it can be chosen gradually in an environment of uncertainty. Here, the increase of compensation level can change the decision result if only the target class label has a lot of high voting values. On the other hand, the decrease of compensation level can affect the decision result if the target class label has a lot of low voting values.

Formally, suppose that we want to classify a new pattern e, which has obtained the voting values $T = \{V_1, V_2, \ldots, V_M\}$ for each class label of C. The membership degree of the pattern e to the class C_i is obtained by the following formula:

$$\mu_{C_i}(e) = f^{-1}\left(\frac{1}{n_{c_i}}\sum_{j=1}^{n_{c_i}} f\left(v_{ij}\right)\right) \tag{8}$$

Where

$\mu_{C_i}(e)$ represents the membership degree of e to the class label C_i,

f is the generator function of the compensatory operator,

n_{c_i} is the number of patterns previously classified in the class C_i,

v_{ij} is a voting value of V_i performed by the pattern e_j according to the class label C_i. It is calculated as follows:

$$v_{\text{ij}} = Sim\left(e, e_j\right) * \mu_{C_i}\left(e_j\right) \tag{9}$$

Additionally, to determine the crisp class label of the new pattern e, say $Cl(e)$, we choose the class label that has the highest membership degree:

$$Cl(e) = max_{\text{i}=1,2,\ldots,\text{M}}\left(\mu_{C_i}(e)\right) \tag{10}$$

In the following, we present an illustrative classification example to better highlight the proposed decision rule behavior next to the K-NN decision rule. So, we consider a classification problem defined by a training set $TR = \{e_1, e_2, \ldots, e_{10}\}$, and two predefined class labels C_1 and C_2. We also assume that we want to classify three new patterns S_1, S_2 and S_3, which are associated with their respective voting values (see Table 2) and the true class label C_1. We suggest the use of Minkowski mean operator defined by the generator function $f(x) = x^p$, in the decision rule.

We applied our decision rule and the K-NN decision rule by varying the parameters p and K in the interval [1, 10], accordingly by increasing the value of the parameter p the compensation level increase too. The obtained results representing the membership degrees of the patterns S_1, S_2 and S_3 to each class label are presented in the Figs. 6 to 8. These figures highlight the dilemma of choosing the adequate K parameter in the K-NN decision rule and how it is resolved. We show in Fig. 6(a) that the correct classification is obtained with high values of K; C_1 is advantageous, regarding the voting values, next to C_2 for $K \geq 4$. In contrast, Fig. 7(a) shows that there is no stable value for K; C_1 is advantageous, regarding the voting values, next to C_2 for $K \leq 4$. In Fig. 8(a), the decision rule tends towards uniform behavior based on the number of neighbors; the correct decision is obtained with larger values of $K \geq 4$. However, the Figs. 6(b) to 8(b) show the capability of the proposed decision rule to adapt the decision to different situations and to resolve the above dilemma. Indeed, the proposed rule acts adequately in the three cases, in the sense that it considers all high similarities and is empowered by the compensation mechanism when including low similarities.

4.4. Classification procedure

The classification procedure defines the computation process leading to the class label of a new unlabeled pattern. Given a classification problem defined by $D = \{F, C, TR\}$, to classify a new pattern e, the FABC performs four steps:

1. Evaluation of the similarity between the new pattern e and each classified pattern in the TR, the result of this step is the vector $SV = \{Sim(e, e_1), Sim(e, e_2), \ldots, Sim(e, e_N)\}$.
2. Computation of the voting values, the result of this step is $T = \{V_1, V_2, \ldots, V_M\}$,where $V_i = \{v_{i1}, v_{i2}, \ldots, v_{iN}\}$ describes the voting value $v_{ij}$for the class C_i.
3. Fuzzy classification: Computation of the membership degrees of the new pattern to each class label using the voting values and equation 8.
4. Crisp classification: assign a class label to the new pattern using equation 10.

The algorithms of different steps are sketched in Fig. 9.

5. Experiment framework

This section presents the experimental framework used to analyze our proposed classification approach (FABC). We present useful materials and tools for evaluating the accuracy, analyzing the behavior of the resulting classifier and facilitating the comparison of their performances with other classification methods belonging to the same approach. This framework includes the datasets, the experimental setup, the performance criteria, and the comparison algorithms.

5.1. Datasets

A set of well-known datasets, taken from the KEEL repository[1] (Alcalá et al., 2010), is used in all experiments. The selected datasets represent various classification difficulties and are described by a set of numerical features and a number of predefined class labels. Table 3 summarizes the main characteristics of the datasets. For each dataset the name (name), the number of features (#Features), the number of patterns (#Patterns) and the number of class labels (#Class) are given.

To conduct the experiments, we used the 10-fold cross-validation procedure (Stone, 1974). So, the selected datasets are partitioned into ten subsets with the same size and preserving the

[1] http://www.keel.es/datasets.php

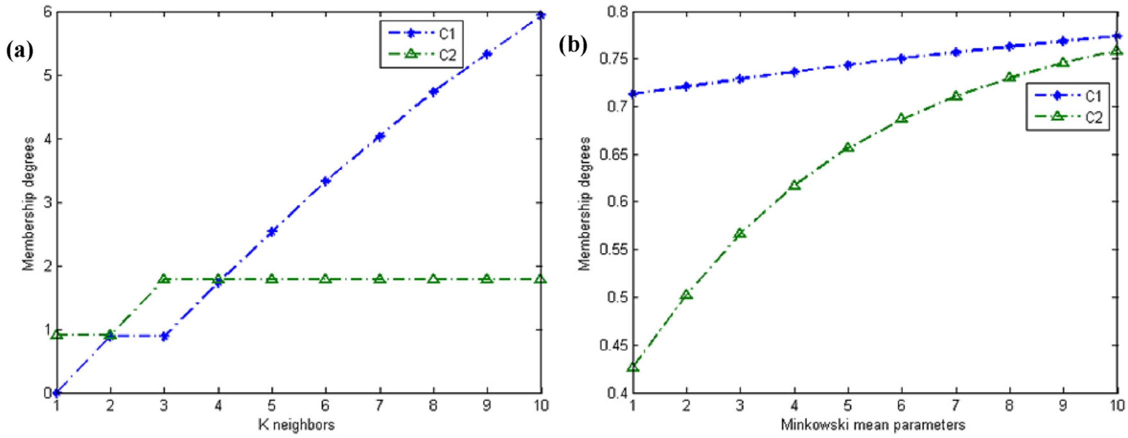

Fig. 6. Membership degrees of the pattern S_1 according to the K neighbors (a) and compensation level (b).

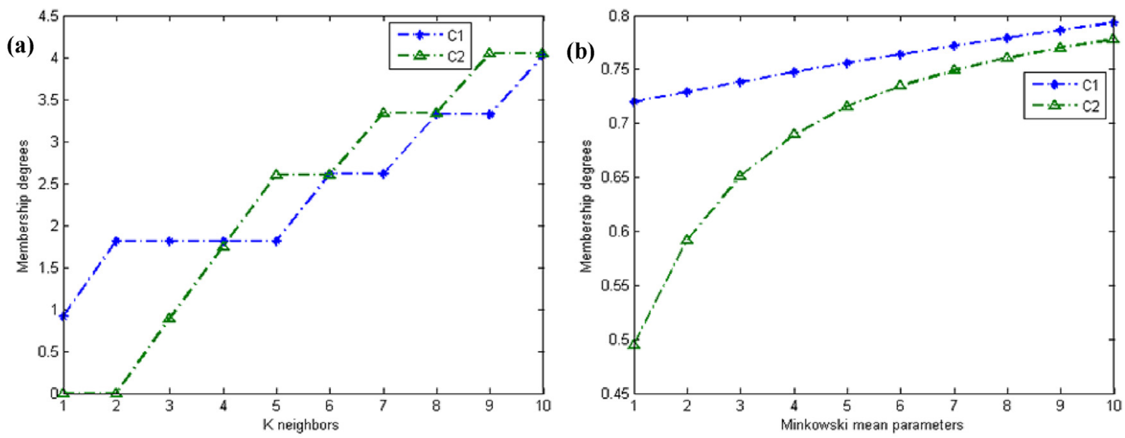

Fig. 7. Membership degrees of the pattern S_2 according to the K neighbors (a) and compensation level (b).

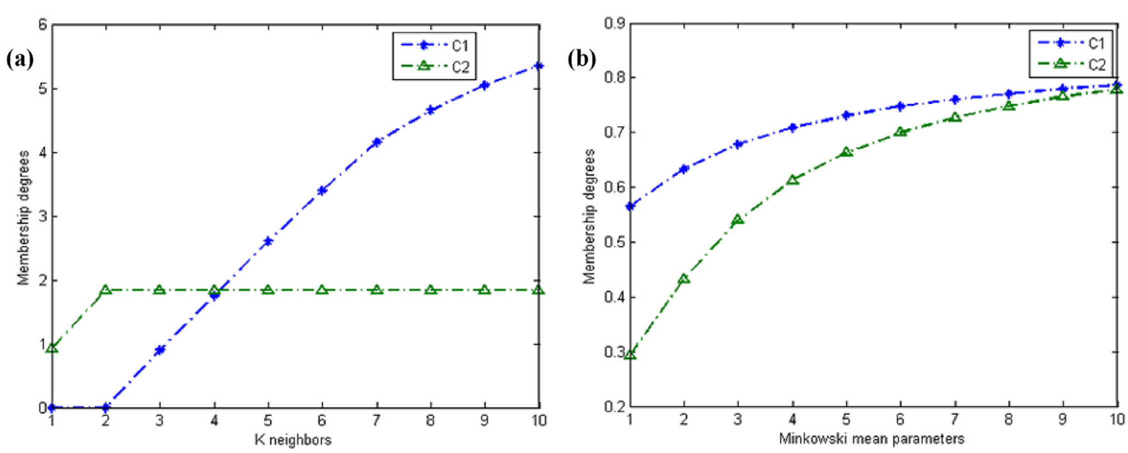

Fig. 8. Membership degrees of the pattern S_3 according to the K neighbors (a) and compensation level (b).

same class distribution between partitions. The classifier is then applied to the different partitions where one partition is selected as the test set and the training set is composed of the rest. The final results per dataset are obtained by averaging the results obtained over the ten partitions.

5.2. Performance measures

To evaluate the performances of a given classifier on a test dataset described by N patterns and M class labels, we assume that the classification results are performed in a confusion matrix defined by $CM = (n_{ij})_{1\leq i,j\leq M}$, where:

n_{ii} represents the number of patterns of the class label C_i correctly classified;

n_{ij}represents the number of patterns, belonging to the class label C_i and classified in the class label C_j.

So, several performance measures can be built from the confusion matrix (CM). In this study, we used the accuracy and Kappa measures to evaluate the precision of our proposed classifier and the classifiers considered in the comparison.

Procedure SimilarityEvaluation(e_1,e_2,t,w)
1. **for** $i \leftarrow 1$ to t
2. $\boldsymbol{s_i} \leftarrow \sum_{j=1}^{k} \min\left(\mu_j\left(e_1^{f_i}\right), \mu_j\left(e_2^{f_i}\right)\right)$
3. **end for**
4. $S \leftarrow \sum_{i=1}^{t} w_i \times s_i$
5. **Return** $\boldsymbol{S}$

Procedure VotingValues(TR, N, M, e, t, w)
1. $T:\ emptyN \times M\ voting\ values\ Matrix$
2. $V_j:\ j^{th} empty\ column\ of\ theMatrix\ T$
3. $v_{ij}:\ empty\ cell\ in\ the\ i^{th} linej^{th} culumn$
4. **for** $i \leftarrow 1$ to N
5. $S \leftarrow$ SimilarityEvaluation(e,e_i,t,w)
6. **for** $j \leftarrow 1$ to$M$
7. $v_{ij} \leftarrow S \times \mu_{C_j}(e_i)$
8. **end for**
9. **end for**
10. **Return**$T \leftarrow \{V_1, V_2, \dots, V_M\}$

Procedure ClassificationProcess(TR, N, M, e, t, w)
1. $T \leftarrow$ VotingValues(TR, N, M, e, t, w)
2. **for** $j \leftarrow 1$ to M
3. $\mu_{C_j}(e) \leftarrow f^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} f(v_{ij})\right)$
4. **end for**
5. **Return** C_j: the class label has high membership degree

Fig. 9. Algorithms of the different steps in the classification process.

Table 3
List of selected datasets in experimental framework.

Name	#Features	#Patterns	#Class
Appendicitis	7	106	2
Banana	2	5300	2
Cleveland	13	297	5
Ecoli	7	336	8
Glass	9	214	7
Iris	4	150	3
movement_libras	90	360	15
Pima	8	768	2
Segment	19	2310	7
Sonar	60	208	2
Titanic	3	2201	2
twonorm	20	7400	2
wdbc	30	569	2
Wine	13	178	3
winequality-red	11	1599	11
Yeast	8	1484	10

The accuracy is determined by using the correct classification rate, which measures the number of the successful class estimations relative to the total number of patterns (Eq. 11).

$$acc = \frac{\sum_{i=1}^{M} n_{ii}}{N} \tag{11}$$

The Cohen's Kappa accuracy (Cohen, 1960) is another classification rate measure, which evaluates the agreement between the correct classifications and those provided by the classifier. In fact, Kappa determines the portion of the results that can be attributed to the classifier itself (i.e. not to mere chance), relative to all the classifications that cannot be attributed to chance alone. For example, Kappa of 0.85 means there is 85% better agreement with the real classification than by chance alone. The Kappa is computed by the following expression:

$$kappa = \frac{N\sum_{i=1}^{M} n_{ii} - \sum_{i=1}^{M} n_{ir}n_{ri}}{N^2 - \sum_{i=1}^{M} n_{ir}n_{ri}} \tag{12}$$

Where n_{ir} and n_{ri} are the total counts, respectively, of the column and row i. Kappa ranges between −1 (total disagreement) through 0 (random classifications) up to 1 (total agreement).

5.3. Comparison: algorithms and statistical procedures

The comparative study aims to compare our proposed classifier with some advanced implementations of the FuzzyNNC algorithms presented in the review[2] (Derrac et al., 2014). To perform this comparative study (see Table 4), we selected six algorithms according to the taxonomy and algorithm ranking provided by the results presented in the review. FuzzyKnn, GAFuzzyKnn, VW-FuzzyKnn, FCMKnn, which are based on fuzzy set theory. IT2FKnn, which is based on Interval Type-2 Fuzzy sets, and D-SKnn, which is based on Dempster-Shafer theory, are selected due to their performances and ranking scores among the algorithms of the FuzzyNNC classification approach.

Furthermore, and as recommended by the authors of Derrac et al. (2014), we included the non-parametric statistical tests (García, Fernández, Luengo, & Herrera, 2010) in order to contrast and confirm the comparative results obtained in the experiment. Therefore, we used the Freidman test followed by the Shaffer post-hoc test. The Freidman test (Friedman, 1937) purposes to test the hypothesis of equality between the results obtained by the compared classifiers; the rejection of this test implies that significant differences arise among the compared classifiers. Assume that we want to compare k classifiers over n datasets; the Freidman test is performed as follows:

[2] http:/sci2s.ugr.es/fuzzyKNN/survey

Table 4
The list of FuzzyNNC algorithms used in the comparative study.

Acronym	Name	Year	Ref.
IT2FKnn	Interval Type-2 Fuzzy Knn	2003	(Rhee & Hwang, 2003)
FuzzyKnn	Fuzzy Knn	1985	(Keller et al., 1985)
GAfuzzyKnn	Genetic algorithms for Fuzzy Knn	2005	(Hu & Xie, 2005)
D-SKnn	Dempster-Shafer theory based Knn	1995	(Denoeux, 1995)
VWFuzzyKnn	Variance weighted Fuzzy Knn	1999	(Han & Kim, 1999)
FCMKnn	Fuzzy C-means Knn	1986	(Bezdek et al., 1986)

- First, the k classifiers are ranked according to their accuracies for each dataset; the classifier with highest accuracy value gets a rank of 1, the second highest gets a rank of 2 and so on, $r_i^j,\ 1 \le j \le k$ denotes the rank of the classifier j in the dataset i.
- Second, the final rank of each classifier is calculated as the average of the ranks over all the datasets, we denote $R_j = \frac{1}{n}\sum r_i^j,\ 1 \le j \le k$ the final rank of the classifier j.
- Finally, the test statistic, denoted F, is computed by the following formula:

$$F = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{13}$$

F determines whether there are significant differences among the classifiers considered in the comparison or not. Significant differences occur when a low p-value is obtained through the adjustment of F.

The differences can be evaluated by using the Shaffer post-hoc test (Shaffer, 1986). The latter performs a pairwise comparison and provides a set of p-values that represent which pairwise comparison, as well as which pairs of compared algorithms have significantly different performance.

5.4. Experiment setup

According to the design of the FABC, presented in Section 4, we specify three parameters that govern its behavior: the KB containing the fuzzy representation of the features, the RIM quantifier guiding the similarity assessment, and the quasi-arithmetic mean operator controlling the compensation level in the decision rule.

5.4.1. Knowledge base generation

The main parameter of the KB generating process, described in Section 4, is the validity index used in the clustering step. In our experiment, we used the index proposed in Su et al. (2006). This index evaluates the quality of a fuzzy partition by using the degree of consistency of each of its fuzzy sets, where the consistency is measured by an index called *H-Index*. The *H-Index* is based on the assumption that two patterns belonging to the same class label must have similar feature values. Thus, the *H-Index* of a fuzzy set A_j is defined as follows:

$$H - Index(A_j) = \max_{C_i,\ i=1,\dots,M} \frac{\sum_{x_k \in C_i} \mu_j(x_k)}{\sum_{k=1}^{N} \mu_j(x_k)} \tag{14}$$

Where:

- μ_j is the membership function describing the fuzzy set A_j;
- $C_i,\ i = 1, .., M$is the set of class labels;
- $x_k,\ i = 1, .., N$ is the set of training patterns.

Subsequently, a fuzzy partition $(A_j)_{j=1,\dots,k}$ is evaluated by the following index:

$$v_i = \min_{j=1,\dots,k} H - Index(A_j) \tag{15}$$

This index is maximized by the best partitions. Hence, to obtain the adequate number of fuzzy sets for a given linguistic variable, we perform the algorithm sketched in Fig. 5 with the number of clusters interval ($[m, n]$) defined as [2, (9 + *#classes*)], where *#classes* is the number of output class labels in the target dataset. Then, the optimal partition is one that maximizes the validity index v_i. Table 5 shows the number of fuzzy sets obtained for each linguistic variable in each dataset. Finally, the membership functions are built from the obtained cluster centers.

5.4.2. FRM configuration

The FRM of our proposed classifier is configured as follows:

- For the RIM quantifier, we used the parameterized alpha-RIM quantifier defined by the fuzzy set associated with the membership function: $Q(x) = x^r$, where $r \in [0, \infty[$. Various possible interpretations of the linguistic quantifiers such as *there exist, some, most, at most, all*, and *for all*, can be obtained by assigning different values to the parameter r. Then, if $r = 0$, we get the existential quantifier; when $r = \infty$, we get the quantifier *for all* and when $r = 1$, we get the quantifier *some*. In addition, for the case in which$r = 2$, we get one of the possible interpretations of the quantifier *most*. In our experiment, we used different values of the parameter r taken in the interval[0.1, 10].
- For the quasi-arithmetic mean operator, we used the parameterized-mean operator defined by the root-power generator function: $f(x) = x^p$, where $p \in R$. By assigning different values to the parameter p, we can obtain various mean operators such as min (p=-∞), harmonic (p=−1), arithmetic (p=1), quadratic (p=2) and max (p=∞). In our experiment, we used different values of the parameter p taken in the interval[1, 10].

6. Experiment results and discussion

In this section, we aim to highlight the properties of our proposed classifier, such as adaptability, efficiency and stability. For that, we have conducted several studies, which we summarize in two stages:

- Stage 1 consists of evaluating the performance of our classifier according to its two parameters alpha-RIM and p-Mean.
- Stage 2 achieves comparison between the results obtained by our classifier and those obtained by the selected FuzzyNNC algorithms (see Section 5.3).

6.1. Stage 1: FABC performance évaluation

The aim of this stage is to evaluate the performance of our classifier according to the parameters alpha-RIM and p-Mean. We also evaluate its capability to manage the uncertainty usually existing in the datasets. In this respect, we provide three types of results:

- First, we fix the value of the alpha-RIM parameter, and we compute the mean of the accuracy and the Kappa accuracy, according to all values of the parameter p-Mean (Figs. 10 and 12).

Table 5
Number of fuzzy sets for each feature in each dataset.

Name	#Number of fuzzy sets
Appendicitis	5, 4, 5, 2, 2, 4, 6
Banana	4, 6
Cleveland	2, 2, 2, 2, 4, 2, 2, 11, 2, 2, 2, 3, 7
Ecoli	8, 16, 12, 8, 13, 13, 9
Glass	16, 11, 9, 7, 8, 10, 14, 7, 8
Iris	2, 2, 3, 6
movement_libras	15, 15, 19, 15, 16, 19, 17, 15, 20, 24, 23, 23, 22, 17, 20, 15, 19, 15, 16, 24, 22, 15, 21, 23, 24, 21, 23, 22, 23, 23, 15, 22, 24, 21, 24, 19, 24, 17, 21, 24, 23, 23, 24, 17, 19, 22, 20, 15, 24, 19, 24, 24, 24, 21, 24, 18, 15, 22, 22, 16, 22, 22, 20, 21, 24, 24, 24, 23, 24, 21, 20, 18, 20, 24, 24, 24, 20, 20, 24, 16, 21, 18, 24, 21, 20, 21, 24, 20, 20, 22
Pima	10, 3, 2, 2, 2, 6, 4, 7
Segment	10, 12, 7, 7, 7, 10, 12, 10, 16, 15, 8, 8, 7, 12, 15, 8, 8, 7, 14,
Sonar	3, 4, 5, 3, 4, 3, 3, 6, 4, 3, 3, 3, 3, 4, 4, 7, 10, 2, 4, 2, 2, 2, 6, 5, 9, 8, 3, 3, 6, 2, 9, 8, 3, 3,2, 2, 2, 3, 5, 3, 5, 7, 10, 3, 5, 7, 3, 3, 3, 4, 3, 4, 4, 9, 4, 9, 9, 4, 11, 8
Titanic	6, 2, 2
twonorm	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
Wdbc	2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 4, 2, 2, 2, 3, 2, 3, 3, 2, 2, 2, 5, 9
Wine	9, 4, 6, 12, 3, 5, 3, 6, 5, 3, 5, 9, 4
winequality-red	17, 12, 11, 18, 18, 11, 12, 11, 17, 16, 11
Yeast	10, 18, 10, 10, 17, 14, 19, 11

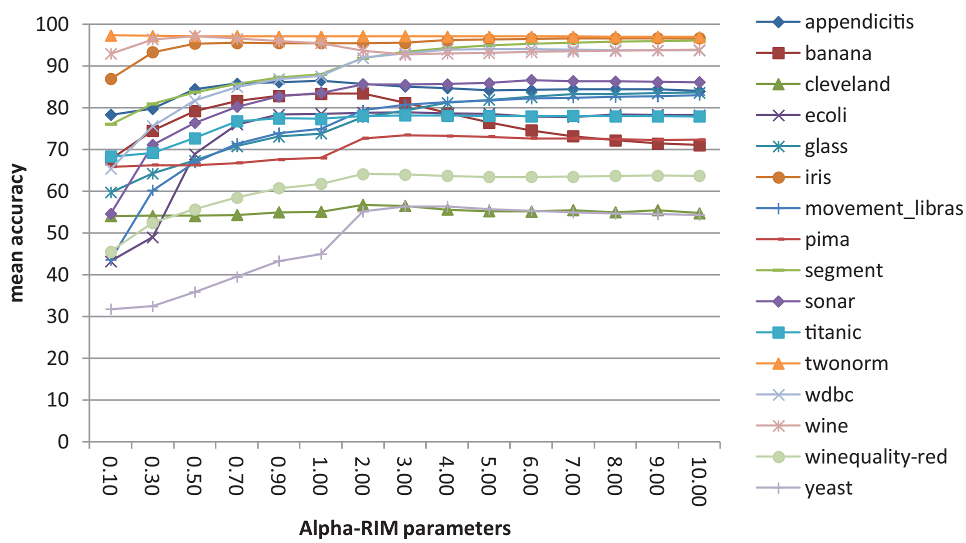

Fig. 10. Relation between the accuracy of FABC and the alpha-RIM.

- Second, we fix the value of the p-Mean parameter, and we compute the mean of the accuracy and the Kappa accuracy, according to all values of the alpha-RIM parameter (Figs. 11 and 13).
- Third, we describe the accuracy results, obtained previously, by the dispersion parameters, namely minimum, maximum, variance and standard deviation.

By analyzing the obtained results, we can remark that our proposed classifier provides accurate classifications for most datasets. The best average of the accuracy is more than 75.5% for all datasets, except for the Cleveland and Yeast datasets; we obtained, respectively, 56.77% and 56.41%. We also obtained a promising nonrandom classification rate; the best average of the kappa accuracy is greater than 0.30 for all datasets. Besides, we observe that the accuracy (respectively, the kappa accuracy) depends on the alpha-RIM and the p-Mean parameters. The Figs. 10 and 11 (respectively, Figs. 12 and 13) show that the average of the accuracy (respectively the average of kappa accuracy) increases considerably for all datasets whether by changing the alpha-RIM or p-Mean values. Table 6 (respectively, Table 7), confirms this conclusion. Indeed, considerable differences between the minimum and the maximum of the average of the accuracy (respectively, the average of the kappa accuracy) obtained according to those two parameters, are perceived in these tables. These remarks confirm that alpha-RIM and p-Mean contribute significantly to the improvement of the classification accuracy. Therefore, the choice of the adequate parameters will lead to better classification rates. On another side, the results obtained for kappa accuracy approve that the classifications obtained by our proposal are in agreement with the training datasets classifications, and confirm that the random classification rate is low when using our classifier.

Moreover, the obtained results show that our classifier seems to maintain the same behavior relative to the alpha-RIM and p-mean parameters over all datasets. They also confirm its ability to manage the uncertainty correctly, in the evaluation of the similarity and the decision rule. More precisely, we conclude that the p-Mean is very effective in uncertainty management compared to the alpha-RIM. Indeed, as we can see in Figs. 10 and 12, the best classification rates obtained according to alpha-RIM is almost concentrated in a small interval; the best results with alpha-RIM are reached with alpha-RIM values equal to 2 or 3. In contrast, the Figs. 11 and 13, show that the best classification rates obtained according to p-Mean is stabilized in a wide interval for all datasets; the best results with p-Mean are reached with p-Mean values all up to 6. The above remarks are approved by the results presented in Tables 6 and 7; the variance and the standard deviation ob-

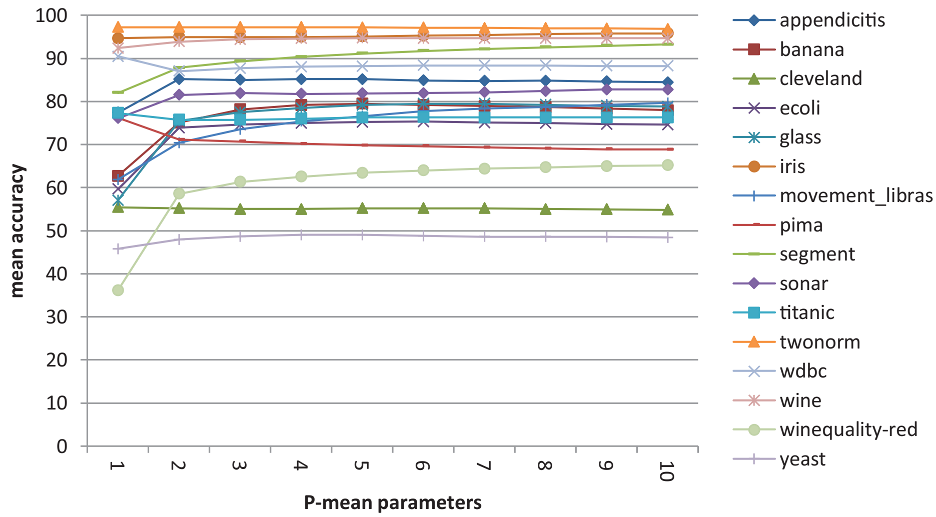

Fig. 11. Relation between the accuracy of FABC and the p-Mean.

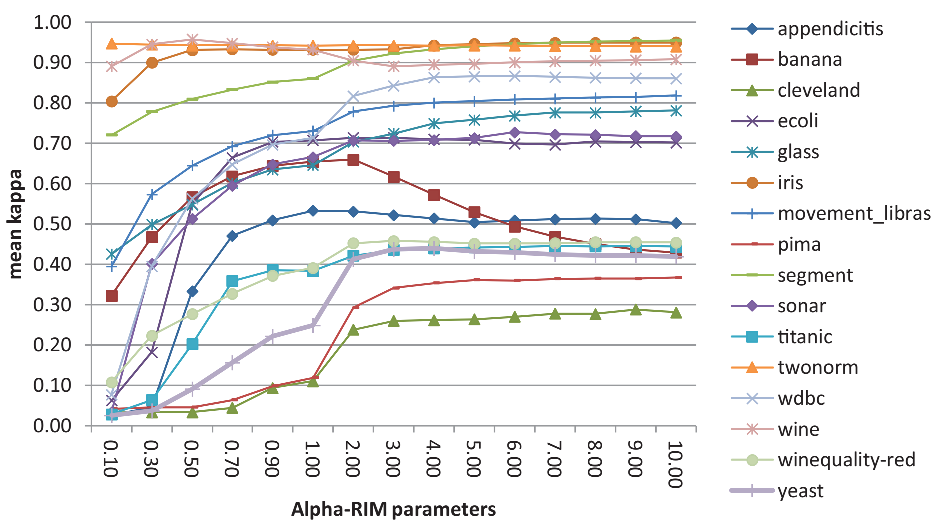

Fig. 12. Relation between the Kappa of FABC and the alpha-RIM.

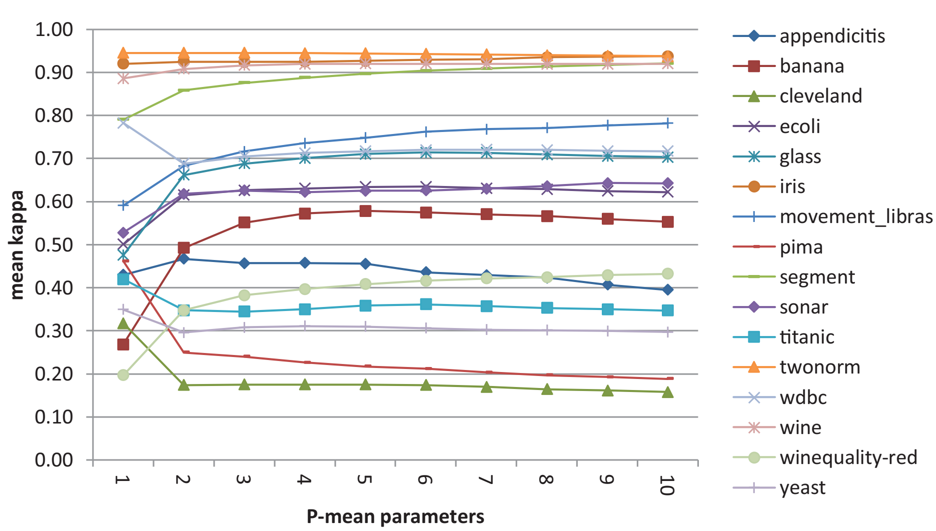

Fig. 13. Relation between the Kappa of FABC and the p-Mean.

Table 6
Descriptive results for the obtained mean accuracy.

	alpha-Rim descriptive result				p-mean descriptive result			
	Min	Max	Var	Std	Min	Max	Var	Std
Appendicitis	78.29	86.51	4.96	2.23	77.44	85.21	5.64	2.38
Banana	67.70	83.47	26.14	5.11	62.71	79.45	25.96	5.10
Cleveland	54.09	56.77	0.64	0.80	54.85	55.43	0.03	0.16
ecoli	43.26	78.88	129.02	11.36	59.64	75.32	23.24	4.82
Glass	59.73	83.61	60.98	7.81	57.08	79.49	47.44	6.89
Iris	86.93	96.67	6.05	2.46	94.67	95.82	0.15	0.39
movement_libras	43.56	83.08	122.54	11.07	61.87	79.67	30.03	5.48
Pima	65.88	73.44	9.56	3.09	68.83	76.26	4.84	2.20
Segment	76.02	96.15	40.85	6.39	82.03	93.21	11.35	3.37
Sonar	54.58	86.64	75.27	8.68	76.13	82.80	3.78	1.95
Titanic	68.32	78.20	11.17	3.34	75.77	77.35	0.19	0.44
twonorm	97.00	97.37	0.01	0.09	96.86	97.27	0.02	0.15
wdbc	65.34	94.07	71.38	8.45	87.03	90.44	0.74	0.86
Wine	92.74	97.19	2.28	1.51	92.40	94.71	0.53	0.73
winequality-red	45.55	64.14	29.30	5.41	36.17	65.19	77.43	8.80
Yeast	31.73	56.41	88.14	9.39	45.78	49.02	0.90	0.95
Average	**64.42**	**82.04**	**42.39**	**5.45**	**70.58**	**79.79**	**14.52**	**2.79**

Table 7
Descriptive results for the obtained mean Kappa.

	Alpha-Rim descriptive result				p-mean descriptive result			
	Min	Max	Var	Std	Min	Max	Var	Std
Appendicitis	0.03	0.53	0.03	0.17	0.40	0.47	0.00	0.02
Banana	0.32	0.66	0.01	0.10	0.27	0.58	0.01	0.09
Cleveland	0.03	0.29	0.01	0.11	0.16	0.32	0.00	0.05
Ecoli	0.06	0.71	0.04	0.21	0.50	0.63	0.00	0.04
Glass	0.43	0.78	0.01	0.11	0.48	0.71	0.01	0.07
Iris	0.80	0.95	0.00	0.04	0.92	0.94	0.00	0.01
movement_libras	0.39	0.82	0.01	0.12	0.59	0.78	0.00	0.06
Pima	0.04	0.37	0.02	0.15	0.19	0.46	0.01	0.08
Segment	0.72	0.96	0.01	0.07	0.79	0.92	0.00	0.04
Sonar	0.03	0.73	0.04	0.19	0.53	0.64	0.00	0.03
Titanic	0.03	0.45	0.02	0.14	0.34	0.42	0.00	0.02
twonorm	0.94	0.95	0.00	0.00	0.94	0.95	0.00	0.00
wdbc	0.08	0.87	0.05	0.23	0.69	0.78	0.00	0.02
Wine	0.89	0.96	0.00	0.02	0.89	0.92	0.00	0.01
winequality-red	0.11	0.46	0.01	0.11	0.20	0.43	0.01	0.07
Yeast	0.03	0.44	0.03	0.16	0.30	0.35	0.00	0.02
Average	**0.31**	**0.68**	**0.02**	**0.12**	**0.51**	**0.64**	**0.00**	**0.04**

tained according to alpha-RIM, are high next to those obtained according to p-Mean. Therefore, we conclude that the impact of changing a voting value on the overall classification provided by the decision rule depends strongly on its frequency among the voting values.

Finally, we can highlight the above conclusions according to three principal factors:

- *Fuzzy modeling of the knowledge*. Indeed, by representing the features describing the datasets by linguistic variables, the data is raised to a high-level representation. This provides a normalized and homogeneous behavior, when handling the features in the whole of the classification process, independently to the datasets. More precisely, the features are measured on the same scale, in all datasets, by their membership degrees to fuzzy sets. Besides, the variation between two feature values is evaluated according to those membership degrees.
- *The quality of the clustering*. In fact, the H-index based validity index, used in the KB generation, ensures that the defined fuzzy sets for each linguistic variable have an optimal consistency degree. In other words, the large part of the support of the fuzzy set defines the same class label. In this respect, it is very probable that two feature values that have a higher individual similarity belong to the same class label, relative to the consistency degree. This assertion leads to the fact that alpha-RIM parameter has almost the same behavior in all datasets.
- *The compensatory behavior of the decision rule*. Indeed, the mean of the accuracy increases, according to the growth of compensation level (p-Mean parameter) and stabilizes, relatively after a sufficiently high value of the p-Mean. Thus, we can conclude that our classifier performs the best accuracy with a high compensation level. This means that the patterns with high global similarities have the highest weights in the compensation process. This behavior confirms the previous assertion that, if two patterns are very similar, then there is a high probability that they belong to the same class label, with respect to the consistency degree.

6.2. Stage 2: comparative study

At this stage, we provide the results of the comparative study, performed between our proposed classifier and the FuzzyNNC algorithms described in Table 4. In order to achieve, this study, we used the results extracted from the experimental framework available in Derrac et al. (2014). Table 8 shows the average of the accuracy and the Kappa accuracy obtained by our classifier and each comparative algorithm over all the selected datasets. We provide two types of results:

Table 8
Comparison between our proposed algorithms and the six selected FuzzyNNC algorithms.

Accuracy (Fixed)		#drp	Accuracy (best)		Kappa (Fixed)		#drp	kappa (best)	
Proposed	83.19	7	Proposed	83.98	Proposed	0.70	5	Proposed	0.71
GAFuzzyKNN	82.31	7	IT2FKNN	82.81	GAFuzzyKNN	0.68	7	IT2FKNN	0.69
IT2FKNN	82.22	7	GAFuzzyKNN	82.74	IT2FKNN	0.68	7	GAFuzzyKNN	0.69
FuzzyKNN	82.07	7	FuzzyKNN	82.67	FuzzyKNN	0.68	7	FuzzyKNN	0.69
D-SKNN	80.57	5	D-SKNN	81.72	D-SKNN	0.66	5	D-SKNN	0.67
VWFuzzyKNN	79.37	3	VWFuzzyKNN	80.35	VWFuzzyKNN	0.64	3	VWFuzzyKNN	0.65
FCMKNN	66.92	5	FCMKNN	67.55	FCMKNN	0.50	9	FCMKNN	0.51

Table 9
Freidman–Shaffer statistical test results for the selected FuzzyNN methods and the proposed method.

		$\alpha = 0.05$		$\alpha = 0.1$	
Algorithms	Rank	+	±	+	±
Proposed	2.5625	3	6	4	6
GAFuzzyKNN	2.7812	3	5	3	5
IT2FKNN	2.8438	2	4	2	4
FuzzyKNN	3.25	1	3	2	3
D-SKNN	4.7188	2	3	2	2
VWFuzzyKNN	5.2812	1	5	1	4
FCMKNN	6.5625	0	4	0	4

- The accuracy and the Kappa accuracy, which are computed according to a fixed decision rule parameter (#drp) of the classifier (we fixed the number of neighbors for FuzzyNNC algorithms and the p-Mean parameter for our proposed classifier). The obtained results are presented in the columns denoted by Accuracy/Kappa (fixed).
- The accuracy and the Kappa accuracy obtained by selecting the best average result performed for each dataset. The obtained results are presented in the columns denoted by Accuracy/Kappa (best).

According to the obtained results in Table 8 we can see that our proposed method performs better than the reviewed algorithms. This implies that our proposed classifier has a great adaptation capability to various classification domains next to the revised classifiers. We can also conclude from the results obtained according to the kappa accuracy that our proposed classifier performs less random classifications compared to other classifiers.

So, to highlight the obtained results presented in Table 8, we used the Freidman-Shaffer tests. We first conducted the Freidman test, over the results obtained regarding the accuracy with fixed parameterizations, in order to rank the comparison algorithms. After analyzing the obtained results, we observed that the Freidman test returns a p-value of 8.99×10^{-9}, this proves that significant differences arise between the comparison algorithms. To quantify the differences between each two classifiers, we resorted to the Shaffer post-hoc procedure. Out of 21 pairwise comparisons, we found 12 differences in the significant level $\alpha = 0.1$ and 14 differences in the significant level $\alpha = 0.05$. Table 9 summarizes the obtained results; we included, for each algorithm the rank obtained in the Freidman test and the number of classifier for which it is statistically better (+) or equal or better (±) at the two significance levels $\alpha = 0.1$ and $\alpha = 0.05$ considering the adjusted p-values computed by the Shaffer test.

By analyzing the results of Table 9, we remark that the statistical tests Rank confirm the ranking of each classifier obtained in Table 8 using the fixed parameterizations. Indeed, regarding the pairwise comparisons, in the significant level $\alpha = 0.\ 05$, our proposed classifier and GAFuzzyKNN are the best classifiers by significant difference equal to a 3 out of the rest. However, in the significant level $\alpha = 0.\ 1$ just the proposed classifier has been the best showing significant difference, with 4 out of the rest. On the other hand, the results show that just the proposed method is equal or better to the rest (±6) which means that the proposed method has never been worse compared to other algorithms; in contrary the other methods were the worst, at least in one comparison, for example the GAuzzyKNN is significantly worse one time and the IT2FKNN is significantly worse 2 times.

6.3. Discussion

The experimental study highlights several important findings about the FuzzyNNC improvement. In the first place, from the statistical test results obtained when comparing FuzzyNCC methods (Table 9), we remark that the optimization of the decision rule considerably increases the classification accuracy. For instance, the GAFuzzyKNN and IT2FKNN have a good Rank, and both of them enhance the decision rule. The first one by searching for the optimal value of the number of neighbors K and adjusting the parameter m using evolutionary algorithms. The second one by using several values of K instead of one. Moreover, the distance is an important information that must be included in the decision rule, since the FuzzyKNN have good Rank next to VWFuzzyKNN. The latter use calculated weights in the decision rule instead of the inverse of distance.

Accordingly, our FABC model appears very effective in enhancing the FuzzyNNC decision rule, and gain the best Rank in statistical tests. Indeed, the compensation property in our global voting rule is shown to be more convenient next to the number of neighbors K in increasing the classification precision. This is obviously caused by the trade-off performed between the high and low voting rule in the decision process, which tackles the uncertainty induced by the choice of the optimal value of K. Furthermore, the decision rule weighted using a vector of optimized similarities may be very adequate next to an adjusted inverse of distance used in GAFuzzyKNN.

On the other hand, the experimental study in stage 1 shows a high adaptability of the FABC model to various classification problems. In fact, by representing the domain features with linguistic variables, the relevance of features is increased and provides a unified features measurement using linguistic terms. Then, the classification process is adapted with respect to the quality degree of the linguistic variable. This adaptation optimizes the similarity measure to handle the uncertainty induced by the irrelevant features and finds the optimal compensation level to manage the uncertainty induced by the similarity measure. Nevertheless, this schema can be very disadvantageous when the generation process of the linguistic terms for features fail to derive consistent linguistic variables, especially when the feature values have high overlapping between different class labels in data.

From the above analysis, it can be seen that the FABC model is a robust extension of the FuzzyNNC approach. Using the FABC model enables the FuzzyNNC to be more effective in the domain where the imprecision and the uncertainty is unavoidable. Especially, in the domain where the data comes from human judgement, like the expert systems.

7. Conclusion

In this work, we present a new FABC method, which is an extension of the FuzzyNNC approach. Our method looks to tolerate the imprecision in features measurement and handle efficiently the uncertainty induced in similarity measure and the choice of the number of neighbors *K*. To fill this task three main components are introduced:

- A features representation using linguistic variables to tolerate the imprecision in features measurement.
- A similarity measure based on OWA operator, which can be optimized by selecting the percentage of features to be considered using RIM quantifier.
- A decision rule using global voting rule guided by a compensation operator. The latter enables a trade-off strategy during the decision process.

We have also validated and compared our method with a set of FuzzyNNC extensions by using different datasets from various domains. The experimental study performed has shown that our classifier has overcome all comparison methods and manifests several promising properties and general capabilities:

(1) The proposed global voting rule guided by a compensation operator is more convenient for the nearest neighbor decision rule next to *K* nearest neighbor voting rule.
(2) FABC provides a classification model with uniform behavior for all datasets in the sense that an unexpected change in classifier behavior happens as little as possible.
(3) FABC Classifier presents high interpretability in the sense that we can easily explain the descriptions and the classification procedure to the users of the given domain. This capability is due to the nearby human descriptions and the analogical reasoning process which is used in classifying a new pattern.
(4) The favorable context of the use of the FABC next to other FuzzyNNC extensions, is when the features are measured linguistically, the expert judgement is considered and the interpretability of the classification procedure is required.

Our perspective in this work is to build our model in several domains where the imprecision and the uncertainty are unavoidable, such as medical diagnosis or text analysis. Also, extending our FABC model to the expert system applications will be very useful. Especially as our model can deal with expert judgements and the parameters set are expressed linguistically. Also, some improvement can be addressed, like: FABC parameters set learning and extending our model to the Nearest Prototype Classification.

Acknowledgment

The authors wish to thank the Laboratory of Intelligent Systems and Applications for his support to achieve this work.

References

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning, 6*, 37–66. doi:10.1007/BF00153759.
Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., et al. (2010). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multi-Valued Logic Soft Comput., 17*, 255–287.
Alpaydin, E. (2014). *Introduction to machine learning* (3rd edition, pp. 21–48). MIT Press.
Arsene, O., Dumitrache, I., & Mihu, I. (2015). Expert system for medicine diagnosis using software agents. *Expert System with Applications, 42*, 1825–1834.
Au, W.-H., Chan, K. C., & Wong, A. K. (2006). A fuzzy approach to partitioning continuous attributes for classification. *Knowledge Data Engineering IEEE Transactions On, 18*, 715–719.
Bezdek, J. C., Chuah, S. K., & Leep, D. (1986). Generalized k-nearest neighbor rules. *Fuzzy Sets System, 18*, 237–256.
Chattopadhyay, S. (2014). A neuro-fuzzy approach for the diagnosis of depression. *Applications on Computer Informtion, 13*, 10–18.
Chen, H.-L., Huang, C.-C., Yu, X.-G., Xu, X., Sun, X., Wang, G., et al. (2013). An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach. *Expert System with Applications, 40*, 263–271.
Chen, H.-L., Yang, B., Wang, G., Liu, J., Xu, X., Wang, S.-J., et al. (2011). A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method. *Knowledge-Based Systems, 24*, 1348–1359.
Chen, Y., Garcia, E. K., Gupta, M. R., Rahimi, A., & Cazzanti, L. (2009). Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research, 10*, 747–776.
Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*, 37–46.
Cordon, O., Herrera, F., del Jesus, M. J., Villar, P., & Zwir, I. (2000). Different proposals to improve the accuracy of fuzzy linguistic modeling. In *Fuzzy if-then rules in computational intelligence* (pp. 189–221). Springer.
Costa, H. R. D. N., & La Neve, A. (2012). Fuzzy decision tree applied to defects classification of glass manufacturing using data from a glass furnace model. In *Presented at the fuzzy information processing society (NAFIPS), 2012 annual meeting of the North American* (pp. 1–6). doi:10.1109/NAFIPS.2012.6291000.
Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*, 21–27. doi:10.1109/TIT.1967.1053964.
Denoeux, T. (1995). A k-nearest neighbor classification rule based on Dempster–Shafer theory. *Systems Man Cybernetics IEEE Transaction On, 25*, 804–813.
Derrac, J., Chiclana, F., García, S., & Herrera, F. (2016). Evolutionary fuzzy k-nearest neighbors algorithm using interval-valued fuzzy sets. *Information Sciences, 329*, 144–163.
Derrac, J., García, S., & Herrera, F. (2014). Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects. *Information Sciences, 260*, 98–119.
Detyniecki, M., Bouchon-meunier, D. B., Yager, D. R., & Prade, R. H. (2000). Mathematical aggregation operators and their application to video querying.
Fernández, A., & Herrera, F. (2012). Linguistic fuzzy rules in data mining: Follow-up Mamdani fuzzy modeling principle. In *Combining experimentation and theory* (pp. 103–122). Berlin Heidelberg: Springer.
Fernández, F., & Isasi, P. (2008). Local feature weighting in nearest prototype classification. *Neural Network IEEE Transactions on, 19*, 40–53.
Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of American Statistical Association, 32*, 675–701.
Gadaras, I., & Mikhailov, L. (2009). An interpretable fuzzy rule-based classification methodology for medical diagnosis. *Artificial Intelligence in Medicine, 47*, 25–41. doi:10.1016/j.artmed.2009.05.003.
García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences, Special Issue on Intelligent Distributed Information Systems, 180*, 2044–2064. doi:10.1016/j.ins.2009.12.010.
Ghosh, A. K. (2006). On optimum choice of k in nearest neighbor classification. *Computational Statistics and Data Analysis, 50*, 3113–3123.
Gil-Aluja, J. (2004). *Fuzzy sets in the management of uncertainty*. Berlin Heidelberg: Springer Science & Business Media.
Grabisch, M., Marichal, J.-L., Mesiar, R., & Pap, E. (2011). Aggregation functions: Means. *Information Sciences, 181*, 1–22.
Grabisch, M., Orlovski, S. A., & Yager, R. R. (1998). Fuzzy aggregation of numerical preferences, in: Fuzzy sets in decision analysis. In *Operations research and statistics* (pp. 31–68). New York: Springer.
Guillaume, S., & Charnomordic, B. (2004). Generating an interpretable family of fuzzy partitions from data. *Fuzzy Systems IEEE Transactions on, 12*, 324–335.
Han, J. H., & Kim, Y. K. (1999). A fuzzy k-NN algorithm using weights from the variance of membership values, in: Computer Vision and Pattern Recognition, 1999. *IEEE Computer Society Conference on. IEEE, 2*, 394–399.
Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning, springer series in statistics*. New York: Springer.
Herrera, F., Alonso, S., Chiclana, F., & Herrera-Viedma, E. (2009). Computing with words in decision making: Foundations, trends and prospects. *Fuzzy Optimization and Decision Making, 8*, 337–364.
Hu, X., & Xie, C. (2005). Improving fuzzy k-NN by using genetic algorithm. *Journal of Computer Information Systems, 1*, 203–213.
Idri, A., & Abran, A. (2001). Evaluating software project similarity by using linguistic quantifier guided aggregations. In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th. IEEE* (pp. 470–475).
Idri, A., & Abran, A. (2000). Towards a fuzzy logic based measures for software projects similarity. In *Proceeding of Sixth Maghrebian Conference on Computer Sciences* (pp. 1–12).
Idri, A., Abran, A., & Khoshgoftaar, T. M. (2001). Fuzzy analogy: A new approach for software cost estimation. In *Proceeding of International Workshop on Software Measurement (IWSM'01) Montréal, Québec, Canada - August 28–29* (pp. 93–101).
Idri, A., Abran, A., & Kjiri, L. (2000). COCOMO cost model using fuzzy logic. *7th international conference on fuzzy theory & techniques*.

Idri, A., Zahi, A., & Abran, A. (2006). Generating fuzzy term sets for software project attributes using fuzzy c-means and real coded genetic algorithms. In *proceeding of International Conference on Information & Communication Technology for the Muslim World (ICT4M 2006), Kuala Lumpur, Malaysia from 21st to 23rd November 2006*.
Ishibuchi, H., Nakashima, T., & Morisawa, T. (1997). Simple fuzzy rule-based classification systems perform well on commonly used real-world data sets. In *Fuzzy information processing society, 1997. NAFIPS '97., 1997 annual meeting of the North American* (pp. 251–256). doi:10.1109/NAFIPS.1997.624046.
Jahromi, M. Z., Parvinnia, E., & John, R. (2009). A method of learning weighted similarity function to improve the performance of nearest neighbor. *Information Sciences, 179*, 2964–2973.
Jang, J.-S. R., & Sun, C.-T. (1995). Neuro-fuzzy modeling and control. *Proceedings of IEEE, 83*, 378–406. doi:10.1109/5.364486.
Jóźwik, A. (1983). A learning scheme for a fuzzy k-NN rule. *Pattern Recognition and Letters, 1*, 287–289.
Keles, A., Samet Hasiloglu, A., Keles, A., & Aksoy, Y. (2007). Neuro-fuzzy classification of prostate cancer using NEFCLASS-J. *Computers in Biology and Medicine, 37*, 1617–1628. doi:10.1016/j.compbiomed.2007.03.006.
Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *Systems Man Cybernetics IEEE Transactions on, SMC-15*, 580–585.
Kim, D.-W., Lee, K. H., & Lee, D. (2004). On cluster validity index for estimation of the optimal number of fuzzy clusters. *Pattern Recognition, 37*, 2009–2025.
Lin, C.-F., & Wang, S.-D. (2002). Fuzzy support vector machines. *IEEE Transactions on Neural Network, 13*, 464–471. doi:10.1109/72.991432.
Luukka, P., & Kurama, O. (2013). Similarity classifier with ordered weighted averaging operators. *Expert System with Applications, 40*, 995–1002.
Medasani, S., Kim, J., & Krishnapuram, R. (1998). An overview of membership function generation techniques for pattern recognition. *International Journal of Approximate Reasoning, 19*, 391–417.
Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 301–312.
Paredes, R., & Vidal, E. (2006). Learning weighted metrics to minimize nearest-neighbor classification error. *Pattern Analysis and Machine Intelligence IEEE Transactions on, 28*, 1100–1110.
Pham, T. D. (2005). An optimally weighted fuzzy k-NN algorithm. In *Pattern recognition and data mining* (pp. 239–247). Berlin Heidelberg: Springer.
Rezvan, M. T., Zeinal Hamadani, A., & Shalbafzadeh, A. (2013). Case-based reasoning for classification in the mixed data sets employing the compound distance methods. *Engineering Applications of Artificial Intelligence, 26*, 2001–2009. doi:10.1016/j.engappai.2013.07.014.
Rhee, F. C.-H., & Hwang, C. (2003). An interval type-2 fuzzy K-nearest neighbor. In *Fuzzy systems, 2003. FUZZ'03. The 12th IEEE international conference on IEEE* (pp. 802–807).
Sáez, J. A., Derrac, J., Luengo, J., & Herrera, F. (2014). Statistical computation of feature weighting schemes through data estimation for nearest neighbor classifiers. *Pattern Recognition, 47*, 3941–3948.
Shaffer, J. P. (1986). Modified sequentially rejective multiple test procedures. *Journal of American Statistical Association, 81*, 826–831.
Sicilia, M.-Á., Cuadrado-Gallego, J.-J., Crespo, J., & García-Bariocanal, E. (2005). Software cost estimation with fuzzy inputs: Fuzzy modelling and aggregation of cost drivers. *Kybernetika, 41*, 249–264.
Sim, J., Kim, S.-Y., & Lee, J. (2005). Prediction of protein solvent accessibility using fuzzy k-nearest neighbor method. *Bioinformatics, 21*, 2844–2849.
Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of Royal Statistical Society: Series B Methodology, 36*, 111–147.
Su, C.-T., Chen, L.-S., & Yih, Y. (2006). Knowledge acquisition through information granulation for imbalanced data. *Expert System with Applications, 31*, 531–541.
Suarez, A., & Lutsko, J. F. (1999). Globally optimal fuzzy decision trees for classification and regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*, 1297–1311. doi:10.1109/34.817409.
Tolias, Y. A., Panas, S. M., & Tsoukalas, L. H. (2001). Generalized fuzzy indices for similarity matching. *Fuzzy Sets Systems, 120*, 255–270.
Torteeka, P., & Chundi, X. (2014). Indoor positioning based on wi-fi fingerprint technique using fuzzy k-nearest neighbor. In *Applied sciences and technology (IBCAST), 2014 11th International bhurban conference on. IEEE* (pp. 461–465).
Triguero, I., Derrac, J., Garcia, S., & Herrera, F. (2012). A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C Applications and Reviews, 42*, 86–100. doi:10.1109/TSMCC.2010.2103939.
Tversky, A., & Gati, I. (1978). Studies of similarity. *Cognitive Categorizes, 1*, 79–98.
Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review, 11*, 273–314.
Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd Edition, pp. 1–362). Morgan Kaufmann.
Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 841–847.
Yager, R. R. (1996). Quantifier guided aggregation using OWA operators. *International Journal of Intelligent Systems, 11*, 49–73.
Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *Systems Man Cybernetics IEEE Transactions on, 18*, 183–190.
Zadeh, L. A. (1983). A computational approach to fuzzy quantifiers in natural languages. *Computers and Mathematics with Applications, 9*, 149–184.
Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences, 8*, 199–249.
Zadeh, L. A. (1965). Fuzzy sets. *Information Control, 8*, 338–353.
Zimmermann, H.-J., & Zysno, P. (1980). Latent connectives in human decision making. *Fuzzy Sets Systems, 4*, 37–51.