

# Intelligent Data Analysis, Soft Computing and Imperfect Data

Jose M. Cadenas and M. Carmen Garrido

**Abstract** In different real problems the available information is not as precise or as accurate as we would like. Due to possible imperfection in the data (understanding that these contain data where not all the attributes are precisely known, such as missing, imprecise, uncertain, ambiguous, etc. values), tools provided by Soft Computing are quite adequate, and the hybridization of these tools with the Intelligent Data Analysis is a field that is gaining more importance. In this paper, first we present a brief overview of the different stages of Intelligent Data Analysis, focusing on two core stages: data preprocessing and data mining. Second, we perform an analysis of different hybridization approaches of the Intelligent Data Analysis with the Soft Computing for these two stages. The analysis is performed from two levels: If elements of Soft Computing are incorporated in the design of the method/model, or if they are also incorporated to be able to deal with imperfect information. Finally, in a third section, we present in more detail several methods which allow the use of imperfect data both for their learning phase and for the prediction.

## 1 A Brief Overview of Intelligent Data Analysis

Intelligent data analysis (IDA) or knowledge discovery in databases is defined in [23] as the “non-trivial process of identifying valid, novel, potentially useful and understandable (if not immediately, with some kind of further processing) patterns from the data”. As it follows from this definition, in the IDA process, the data are the most important part of the discipline [23] and it is a complex process that includes the obtaining of the models and also other stages.

---

J.M. Cadenas (✉) · M.C. Garrido  
Department of Information and Communications Engineering,  
University of Murcia, Murcia, Spain  
e-mail: jcadenas@um.es

M.C. Garrido  
e-mail: carmengarrido@um.es

IDA is divided into the following stages, [23]:

- The “Data Integration and Collection” (selection) stage.
- The “Data Preprocessing” stage, related to the treatment of the data and the strategies that would be used to handle the available information.
- The “Data Mining” stage, related to the selection and application of the appropriate methods for the modeling from the available data and the obtaining of understandable models and high accuracy.
- The “Evaluation (interpretation) and Diffusion” stage.

Although all stages are fundamental to the development of the IDA process, its core is in the data preprocessing and data mining stages.

## 1.1 Data Preprocessing Stage

Data preprocessing can have a great impact on the performance of the data mining methods, [27]. One of the problems that must be faced in this stage is to understand and analyze the nature of the data avoiding the loss of useful information during the process. This stage includes, among others, the cleaning of data (such as the elimination of inconsistent data, treatment of missing values, etc.), data integration (multiple sources), data transformation (discretization, etc.) and reduction of data (attribute/instance selection) [27].

Specifically, the “discretization of continuous attributes” plays a critical role in IDA and has been studied in depth. Discretization consists in dividing the values of a numerical (continuous) attribute into a set of intervals. By means of the discretization, a numerical attribute can be more concise and easier to understand. In the general description of the discretization process, we can do the following taxonomy (there are other taxonomies for the different discretization methods such as that presented in [51]):

- Top-down methods: The attribute domains are progressively cut to construct a set of intervals.
- Bottom-up methods: They start with the individual values in the dataset that are fused progressively until constructing a set of intervals.

Among the top-down methods we find the ones proposed in [15, 33, 34, 46, 81]. Besides, the decision trees construction methods, such as, ID3 [71] and C4.5 [72], can be interpreted as top-down discretization methods. Among the bottom-up methods we find methods such as those proposed in [9, 44, 52]. All these methods generate classical discretization, i.e., crisp intervals.

Also, the “attribute selection” plays an important role in the IDA process and more specifically in the classification task. On the one hand the computational cost is reduced and on the other hand, a model is constructed from the simplified data and this improves the general abilities of classifiers. The first motivation is clear, since the computation time to build models is lower with a smaller number of attributes.

The second reason indicates that when the dimension is small, the risk of “overfitting” is reduced. Removing insignificant attributes of datasets can make the model more transparent and more comprehensible providing a better explanation of the system [53]. Therefore, the attribute selection addresses the problem of reducing dataset dimensionality by identifying an available attributes subset. Researchers have studied various aspects of attribute selection. One of the key aspects is to measure the goodness of an attribute subset determining an optimal one. Depending on evaluation criteria, attribute selection methods can be divided into the following categories, [29, 75]:

- Filter methods: These methods select subsets of attributes as a preprocessing step, independently of the chosen classifier.
- Wrapper methods: These methods use a method of data mining as a black-box to score attribute subsets according to their predictive power.
- Embedded methods: These methods select attributes in the training process and are usually specific to the given modeling method.
- Hybrid methods: These methods are a combination of filter and wrapper methods. Hybrid methods use the ranking information obtained using filter methods to guide the search in the optimization algorithms used by wrapper methods.

In literature we can find a variety of methods to carry out attribute selection, such as the proposed in [3, 42, 52, 75].

## 1.2 Data Mining Stage

The data mining (DM) stage is the more characteristic stage in the IDA process. The purpose of DM is the construction of models based on the data to produce new knowledge that can be used by the user. The model is a description of patterns and relationships in the data, which can be used to make predictions in a particular area, better understand the domain, improve performance or explain past situations. In practice, there are two types of models: Predictive (identify patterns to estimate future values using predictor attributes) and Descriptive (identify patterns that explain the data). In addition, different types of tasks are distinguished in DM. Each task has its own requirements and obtains a type of knowledge different from the obtained one by other tasks. Among the aimed tasks that obtain predictive models, we can find both the classification and the regression tasks; while clustering and association are tasks aimed at obtaining descriptive models. This stage includes the choice of the most appropriate task for the problem, the choice of the DM method, and finally the use and adaptation to the problem of the selected method, [27, 85].

We group these methods according to the type of model obtained. Without being exhaustive, we find models represented by discriminant functions, decision trees, neural networks, based on rules or based on instances.

- One of the most useful ways of representing a model is through a set of discriminant functions. The model in this case can be seen as a machine which computes  $c$  discriminant functions  $g_i(x)$  and which selects for  $x$  the class  $\omega_i$  with the highest value for the discriminant function [22, 26]. In this way the model is expressed as  $g_i(x) = P(\omega_i/x)$ , such that the maximum discriminant function is the maximum a posteriori probability. When the discriminant functions are linear functions we find methods such as descending gradient, Newton's algorithm, the Perceptron criterion [32, 36]. When the discriminant functions are complex density functions, these can be approximated by a mixture of simpler density functions.
- The models based on instances approximate an unknown density function using an averaged version of the density based on the probability of a specific vector's falling within a certain region of the attribute space [22]. The methods based on these models have no learning phase since the model is formed by the dataset instances. There are two common methods based on these models: Parzen method and k neighbors method [21, 31, 58].
- The methods which model the problem through decision trees are useful for finding structures in high dimensionality spaces or when the conditional densities of the classes are unknown or are multimodal. Some basic and well-known methods to generate decision/regression trees are ID3 [71], C4.5 [72] and CART [10].
- Rules based methods model a system through a base of rules (if-then) constructed from the instances. Some methods for obtaining rules (association rules) are based on the concept of frequent items sets and use counting and minimum support methods [79]. Other methods obtain rules covering the instances (cover methods) such as those based on CN2 [17] and AQ algorithms [57]. Genetic algorithms/programming [51, 86] have also been used to generate rules.
- Other type of model is the neural network. Neural networks are a very powerful computation paradigm allowing complex problems with possible non linear interactions between the attributes. Among the most important neural networks we can find the multilayer Perceptron which generates more than one boundary of separating in the attributes space [32, 36, 74].
- There is a further group of methods whose aim is to generate groupings of data and these are known as clustering. The aim of cluster analysis is to find a valid and convenient organization for the data and an underlying structure. Within these methods we can include Kohonen's self-organizing maps [45], those based on the  $K$ -means algorithms which obtain partitional cluster [62], in contrast to the hierarchical methods which do not establish a priori the groups number [25].

## 2 Intelligent Data Analysis and Soft Computing

In [56] several paradigms introduced with the data analysis are identified. Among them, the management and processing of data respecting the true nature of them (imperfect data) are included. Therefore, by focusing on the data, and before applying any stage of the IDA process, we must take into account the nature of these data

to ensure the success of the process. This means that depending on the nature and precision of these data, we must apply different methods depending on their degree of tolerance to them. A clear example to illustrate the problem of the different nature of the data and the importance of tolerance to different types of imperfect data is the problem of parking a car [91], where most of the population is able to do it easily. Therefore, we need methods that can extract knowledge and handle imperfect data, in order to provide quality information and generate useful knowledge.

Generally, the IDA process uses and combines different methods and tools from a variety of disciplines [5]. Due to possible imperfection in the data, tools provided by the Fuzzy Sets theory [90] and, in general, Soft Computing (SC) [7, 82, 91] are quite adequate. In this way, the hybridization of the SC methods with IDA is a field that is gaining more importance. The methods proposed by SC and their applications have been very important in recent years, and in particular, the advances in the hybridization of SC with IDA are aimed at obtaining more flexible methods with results more efficient compared to the classical methods [30, 61]. In this framework, we comment on different methods proposed from two levels: In a first level, if the SC elements are incorporated in the design of methods/models; and, in a second level, if they are incorporated for the treatment of imperfect information, additionally.

## 2.1 Data Preprocessing in Soft Computing Framework

In the data preprocessing stage, SC has generally been applied to the design of flexible methods for the different tasks of this stage. Although most of them use SC in their development, to our knowledge, the methods that allow and management imperfect data are seldom studied.

In particular, in the discretization of numerical attribute we find methods that allow the use of membership degree to intervals (denoted by fuzzy discretization methods). These methods are grouped according to the used algorithm.

- Decision tree based methods: In [40, 43, 63] different approaches for the fuzzy discretization of numerical attributes are proposed. All of them use a fuzzy decision tree combined with some basic strategy.
- Clustering based methods: These methods are based on dividing a numerical attribute domain into fuzzy partitions by using fuzzy clustering. In particular, several methods using the fuzzy c-means method are proposed in [59, 64, 80].
- Genetic algorithm based methods: The genetic algorithms (GA) are combined with existing specialized methods to create hybrid algorithm that improve the overall results. In particular, we can find several methods, [16, 18], using strategies of classical/fuzzy discretization together a genetic algorithm to optimize the number of partitions, interval limits and the degree of overlaps of these limits.
- Hybrid methods: In the literature we can also find methods based on combinations of two or more methods. In [88] a cluster and a neural network (NN) are used. In

[76] the combination of the FCM clustering algorithm and a GA are used, and in [48, 73, 84] a kd-tree and a minimum spanning tree are used.

In attribute selection, there are a lot of methods using SC in their development but they perform the selection from crisp data.

- Attribute selection methods using SC for their design can be find in [3, 16, 42] where a neural network, a GA or an ant colony (AC) are used, respectively. There are other methods that also use elements of the fuzzy set theory as in [53, 83] where a fuzzy criteria or fuzzy entropy are used, or in [87] where the attribute selection is performed using the fuzzy evidence theory.
- To perform the attribute selection from imperfect data we can find several proposals: in [41] a method taking into account the uncertainty in the data through fuzzy-rough sets is presented. This method employs fuzzy-rough sets to provide a means by which discrete or real-valued noisy data (or a mixture of both) can be effectively reduced without the need for user-supplied information. In [77, 78] a fuzzy mutual information measure between two fuzzified numerical attributes to handle imprecise data is used (they define a new extended version of Battiti’s filter attribute selection method). This measure is used in combination with a genetic optimization to define the method proposed.

Table 1 shows the summary of papers discussed.

**Table 1** Hybridization of data preprocessing with Soft Computing: summary of papers

Method based on ...		SC at method level	SC at minable view level	
				Allowed data
Fuzzy discretization	Fuzzy decision trees	[40, 43, 63]	–	–
	Fuzzy clustering	[59, 64, 80]	–	–
	GA to optimize	[16, 18]	–	–
	kd tree—spanning tree	[48, 73, 84]	–	–
	Cluster—GA	[76]	–	–
	Cluster—NN	[88]	–	–
Attribute selection	NN, GA, AC	[3, 16, 42]	[77, 78]	Fuzzy sets
	Fuzzy criteria/entropy	[53, 83]	–	–
	Fuzzy evidence theory	[87]	–	–
	Fuzzy-rough metric	–	[41]	Fuzzy-rough sets

## 2.2 Data Mining in Soft Computing Framework

SC has also been applied in the DM stage, and, to our knowledge, the methods that allow and management imperfect data are seldom studied. From this, we can consider the DM methods hybridized with SC in two levels:

- At the level of generated models: Methods that generate models described in the framework of SC. These models are more interpretable and we can find elements of SC in rule-based systems, methods based on k-nearest neighbors, decision trees, clustering and support vector machines.

In 1971 Zadeh proposed the design of rules if-then using linguistic variables that can be provided by a group of experts or obtained through DM methods. So, among others, in [4] a set of fuzzy rules is obtained using a method based on genetic programming, in [24] a set of fuzzy rules is obtained in unbalanced problems using a genetic selection process of rules, in [37] different weights are assigned to a set of fuzzy rules using heuristic methods, and, in [65] an initial set of fuzzy rules is constructed by clustering and then are optimized using a neuro-fuzzy learning algorithm.

Among the fuzzy versions of the k-nearest neighbors rule we can highlight works that assign memberships degree of each instance to each class, use fuzzy distance measures, use different ways of combining the votes of neighbors, etc. A complete review of these methods is carried out in [20].

Also, fuzzy decision trees have been designed as the proposed in [66] that obtains the best fuzzy partition of the best attribute in each node to split. Using fuzzy decision trees, fuzzy ensembles are proposed as in [19] where an ensemble is constructed from a non-fuzzy tree construction algorithm that subsequently is transformed to fuzzy.

With the aim to construct data partitions that allow an instance belongs to more than one partition, fuzzy clustering algorithms have been developed such as the fuzzy C-means proposed in [6]. Different versions of this algorithm are found in [35] to extend it to nominal data, in [49] to deal with missing values through intervals or in [80] to deal with fuzzy values.

Also, fuzzy versions of support vector machines have been designed. So, in [50] a membership degree to each class is assigned to each instance, allowing that each one contributes in a different way in the learning of the decision surface. In [1] a method for multilabel classification is generalized. For each multilabel class, a region with the associated membership function is defined and an instance is classified into a multilabel class whose membership function is the largest.

- At minable view level: Methods that besides incorporating the SC elements, support input imperfect data. In this case, the methods allow us that the data are composed of attributes described by imperfect values. This generates the following advantages: (1) methods can interpret the imprecision/uncertainty expressed in the data and generate robust models to these types of information without transforming the true nature of them; (2) data preprocessing is simplified by not carrying out these transformations (replacement, deleting data, ...); and (3) the minable view

contains a greater number of instances because the imprecise and uncertain data are not discarded. In general, significant efforts are being carried out to incorporate the treatment of imperfect data into DM methods using SC.

Thus we can find works that incorporate the treatment of fuzzy values. There are fuzzy decision trees based on a fuzzy partition of numerical attributes. This partition is used in the test of nodes as in [38, 47]. Fuzzy partitions of numerical attributes are also use in the construction of fuzzy ensembles to incorporate fuzzy values. This approach is used in [39] where to select the test of each node, the set of the best attributes for partitioning that node is used or in [55] where a fuzzy ensemble for each class value of the problem is constructed. In [60] a fuzzy version of multilayer perceptron is presented which performs the learning from fuzzy values. In [68] a genetic classifier based on fuzzy rules is obtained from data described with fuzzy values. In [69, 70] Adaboost and FURIA algorithms are extended in order to obtain fuzzy rules from this type of values. In [67] an algorithm to obtain a set of fuzzy association rules from a fuzzy partition is proposed. As particular cases of fuzzy values, some works deal with values expressed by intervals as in [47, 67–70].

On the other hand, the set of methods that allow the existence of missing values is considerable. We highlight only a few that allow the treatment of some other type of imperfect information as [38, 39] or as in [47], where missing values are only allowed in the classification phase.

Finally, there is a considerable set of methods that have considered the possibility that an instance has more than one associated class value (multi-valued class), but few extend this possibility to other nominal attributes of a problem (multi-valued attributes). So, among the first we can find works as [68] where class may be defined by a crisp set, or [89] where a fuzzy k-nearest neighbor method is used to allow that an instance can belong to more than one class with several degrees. In [54] we can find a comparison of this kind of methods.

Table 2 shows the summary of papers discussed.

**Table 2** Hybridization of data mining with Soft Computing: summary of papers

Method based on ...	SC at method level	SC at minable view level	
			Allowed data
Fuzzy rules	[4, 24, 37, 65]	[67–70]	Fuzzy sets, intervals
	–	[68]	Fuzzy sets, intervals, multivalued class
k-nearest neighbors	[20]	[89]	Multivalued class
Fuzzy decision trees	[19, 66]	[55]	Fuzzy sets
		[38, 39]	Fuzzy sets, missing
		[47]	Fuzzy sets, intervals, missing
Fuzzy clustering	[6, 35, 49, 80]	[35, 49, 80]	Nominal, fuzzy sets, intervals
Support vector m	[1, 50]	[1]	Multivalued class
Neural network	–	[60]	Fuzzy sets

### 3 Hybridization on the Two Level of Soft Computing and Data Preprocessing/Mining Methods

In this section we describe the characteristic elements of two methods in the data preprocessing stage and three methods in DM stage that use SC in the two levels commented: at model/technique level and at minable view level. Due to the high flexibility in the design of these methods, they can easily be extended to support new types of imperfect data.

A more detailed analysis of these methods can be found in papers [11, 13] for the preprocessing methods and papers [8, 12, 14, 28] for the DM ones.

#### 3.1 Notation, Types and Representation of Imperfect Values

Let us consider a set of instances  $E$ , where each instance  $\mathbf{x}$  is characterized by  $n$  attributes in a vector  $(x_1, x_2, \dots, x_n)$  (the  $n$ -th attribute represents the class). The domains of each attribute,  $\Omega_{x_1}, \Omega_{x_2}, \dots, \Omega_{x_{n-1}}$ , can be numerical or nominal, while the domain of the class  $\Omega_{x_n}$  (nominal attribute) can take the values  $\{\omega_1, \omega_2, \dots, \omega_l\}$ .

The numerical attributes are represented by fuzzy sets with a trapezoidal fuzzy membership function [2]  $\mu(x)$  defined by a quadruple  $(a, b, c, d)$ :

$$\mu(x) = \begin{cases} 0 & x < a \text{ or } x \geq d \\ \frac{x-a}{b-a} & a \leq x < b \\ 1 & b \leq x < c \\ \frac{d-x}{d-c} & c \leq x < d \end{cases}$$

With this representation, the methods use the following values:

- Crisp values are represented by the quadruple  $(a, a, a, a)$ .
- Interval values  $[a, b]$  are represented by the quadruple  $(a, a, b, b)$ .
- Fuzzy values are represented by trapezoidal fuzzy membership functions.
- Missing values include pieces of information that are unknown. These values are represented by the quadruple  $(\min_i, \min_i, \max_i, \max_i)$ , where  $\min_i$  and  $\max_i$  are, respectively, the minimum and maximum values of  $\Omega_{x_i}$  included in the dataset.

The nominal attributes (including the class attribute) are represented by fuzzy subsets  $\{\mu(h_1)/h_1, \dots, \mu(h_s)/h_s\}$ , where  $h_j$  is a value into attribute domain and  $\exists h_k \in \Omega_i : \mu(h_k) = 1$ . With this representation, the methods use the following values:

- Crisp values are represented by the fuzzy subset  $\{1/h_j\}$ .
- Crisp subset values consider more than a possible nominal value. They are represented as  $\{1/h_1, \dots, 1/h_s\}$ .

- Fuzzy subset values consider more than one nominal value with a membership value  $\mu \in [0, 1]$ . They are represented using the notation introduced above.
- Missing nominal values are represented using a fuzzy subset that contains all possible values with membership degree equals to 1.

### 3.2 OFP\_CLASS: A Hybrid Method for Attribute Discretization

In [13], OFP\_CLASS method is proposed to data preprocessing. It is a hybrid method for discretizing numerical (continuous) attributes by means of fuzzy sets, which constitute a fuzzy partition of the domains of these attributes. The aim of this method is to find an attribute partition so that the fuzzy classification methods obtain better results. The OFP\_CLASS method can deal with datasets with imperfect values and it is labeled as supervised, local, top down, and incremental, using the entropy as measure to obtain the partition.

The OFP\_CLASS method is composed of two stages (Fig. 1): (a) In the first stage, crisp intervals are defined for each attribute using a fuzzy decision tree (FDT); and (b) in the second stage, these intervals are used as the starting point to form an optimal fuzzy partition for classification. In this second stage, a genetic algorithm is used to determine the cardinality and fuzzy boundary of these intervals.

The partition obtained for each attribute guarantees:

- Completeness (no point in the domain is outside the fuzzy partition), and
- Strong fuzzy partition (it verifies that  $\forall x \in \Omega_i, \sum_{f=1}^{F_i} \mu_{B_f}(x) = 1$ , where  $B_1, \dots, B_{F_i}$  are the  $F_i$  fuzzy sets for the partition corresponding to the  $i$ -th numerical attribute with  $\Omega_i$  domain).

The FDT used in the first stage allows the dealing of imperfect data, and for this, uses a specific information gain,  $G_i$ , for each attribute  $i$  in order to choose the best attribute to divide a node. Function  $G_i$  uses the standard information associated with the node (taking into account the membership degree of an instance to the node and

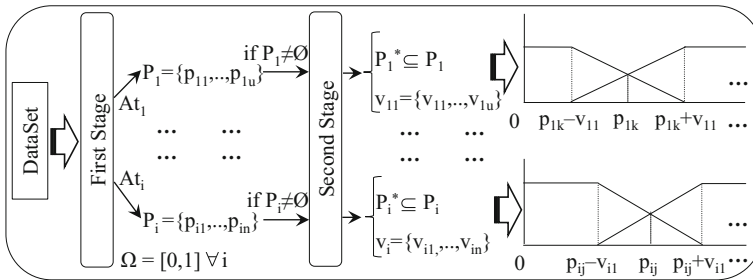


Fig. 1 Scheme for the discretization of numerical attributes using the OFP\_CLASS method

the membership degree of example  $e_j$  to each class) and a factor which represents the standard information obtained by dividing the node using attribute  $i$  adjusted to the existence of missing values. We must highlight that, in the second stage, the fitness function of the genetic algorithm is defined by  $\frac{\sum_{i=1}^n I_i}{\sum_{i=1}^n H_i}$  where  $I_i$  and  $H_i$  are the information gain and entropy of attribute  $i$  respectively, taking into account the crisp intervals obtained in the first stage.

OFP\_CLASS method is an effective strategy and it obtains very good results when is compared with other methods of the literature. These results have been validated by applying statistical techniques to analyze the behavior of different methods in each experiment.

### 3.3 FRF\_fs: A Filter-Wrapper Method for Attribute Selection

In [11] is proposed the FRF\_fs method of attribute selection to data preprocessing which can handle imperfect data. This method is based on a Fuzzy Random Forest ensemble (a method that supports imperfect data, [8, 12]) and is classified as a Filter-Wrapper method with sequential forward selection on the subset of attributes obtained by the Filter method and using a ranking obtained with these attributes. This method consists of the following main steps (Fig. 2): (1) Scaling and discretization process of the attribute set; and attribute pre-selection using the discretization process (Filter); (2) Ranking process of the attribute pre-selection; and (3) Wrapper attribute selection based on cross-validation.

Note that in each step the approach obtains information useful to the user (pre-selected attribute subset, pre-selected attribute subset ranking and optimal attribute subset). Some details of these steps are discussed below.

- Filter method for attribute pre-selection

Initially, the method carry out a scaling and discretization (in [13], a hybrid method for the fuzzy discretization of numerical attributes is presented), and as in the discretization process some attributes may be discretized into a single interval,

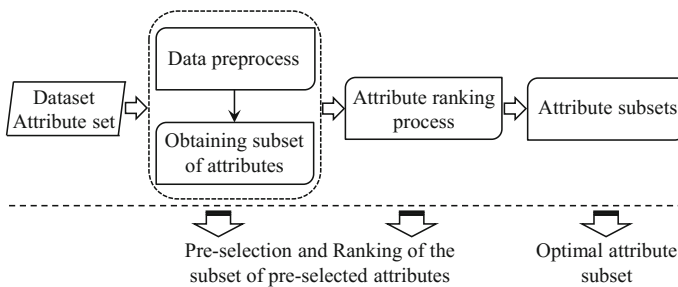


Fig. 2 Framework for the attribute selection using the FRF\_fs method

these latter attributes can be removed. Thus, the method obtain a pre-selection of the attribute set.

- Attribute importance (Ranking process)

From the pre-selected attribute subset and through a fuzzy random forest ensemble, the method obtains a vector  $RANK$  ordered, in descending order, of this attribute subset. This vector is obtained from the value of each attribute  $x_i$  as  $RANK = \sum_{t=1}^T W \cdot IMP_t$ , where the information provided by the  $T$  trees of the fuzzy random forest ensemble is aggregated using an OWA operator. Values  $IMP_t$  are obtained from the information gain of nodes in the FDT  $t$  to each attribute  $x_i$ , and from the accuracy of FDT  $t$  classifying the *OOB* dataset.

- Wrapper for attribute final selection

Once the ranking of the pre-selected attribute subset,  $RANK$ , is obtained, the method find an optimal subset of attributes. The process adds a single attribute at a time following the  $RANK$  vector. The several attribute subsets obtained by this process are evaluated by a method that supports imperfect data using a cross-validation. In particular, and using a fuzzy random forest ensemble, an ascending sequence of fuzzy random forest models is constructed, by invoking and testing the stepwise attributes.

The efficiency and effectiveness of the FRF\_fs method is proved through several experiments using both high dimensional and imperfect datasets. The method shows a good performance (not only classification accuracy, but also with respect to the number of selected attributes) and good behavior both with high dimensional datasets and with imperfect datasets.

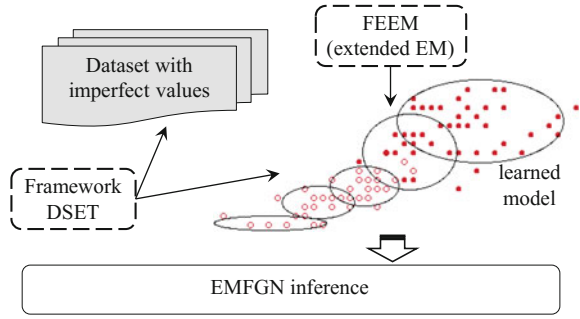
### 3.4 EMFGN: A Method Based on Gaussian Mixture Models

Extended Mixture of Factorized Generalized Normal (EMFGN) method [28] is a predictive DM method for performing learning and inference from imperfect data. The method obtains an explicit expression of the model-observation joint function of the attributes, where both the model expression and the input information are interpreted and represented in a common framework. The Dempster-Shafer Evidence Theory (DSET) is the framework that allows its interpretation as mass functions defined on the domains of the single attributes.

In Fig. 3, the general scheme of the process followed by EMFGN method is shown. From the dataset with imperfect information, the method provides a model reflecting the joint dependence of the attributes by means of a mixture of factorized normals. This model and the input available information are interpreted and represented in the DSET in order to combine them (using the Dempster-Shafer's combination rule). The model provided by EMFGN method is the following:

$$p(z) = \sum_{ir} P\{C_i\} \pi_r \prod_{j=1}^n F_{irj}(m_{rj}(\Theta_j) \oplus m_{ij}(\Theta_j))$$

**Fig. 3** A general scheme of the EMFGN method



where:

- $\Theta_j \in \mathcal{P}(\Omega_{z_j})$  and  $\mathcal{P}(\Omega_{z_j})$  is the set of parts of  $\Omega_{z_j}$ .
- $m_{irj}(\Theta_j)$  is the likelihood function of the  $r$ -th component of the input information expressed through a mass function.
- $m_{ij}(\Theta_j)$  is the mass function corresponding to the  $i$ -th component of the model.
- $F_{irj}$  is a necessary normalization factor in the combination of two mass functions.
- $P\{C_i\}\pi_r$  is the product of the likelihood function of input information in its  $r$ -th component and the expression of the model in the  $i$ -th component.

In this framework, the EMFGN method uses the FEEM algorithm in the learning phase. This algorithm is an extended EM algorithm to allow both the imperfect information and the model represented in DSET.

From the learned model, EMFGN method can infer both nominal and numerical attributes. To numerical attributes, the method infers the value  $z_j = \sum_{ir} \alpha_{ir} \bar{m}_{irj}$ , and to nominal attributes, the method infers the value  $z_j = \operatorname{argmax}_w \sum_{ir} \alpha_{ir} m_{irj}(\omega)$ , with  $\omega \in \Omega_{z_j}$ . The value  $\alpha_{ir}$  indicates the likelihood of the  $r$ -th component of the input information having been generated by the  $i$ -th component of the mixture.  $m_{irj}(\cdot)$  is a mass function combining the input information and the model to the attribute  $j$ , and the value  $\bar{m}_{irj}$  is the average value of  $m_{irj}(\Theta_j)$ .

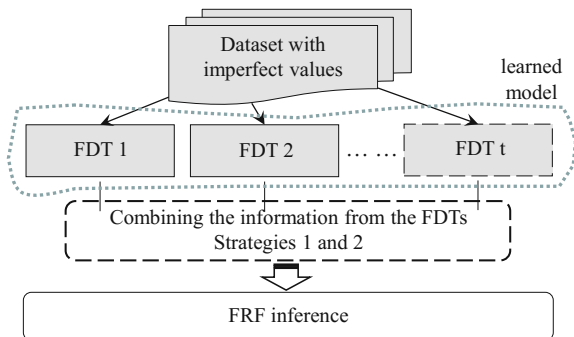
The results obtained are very satisfactory with the advantage of having a global model to be able to perform inference on any attribute of an instance.

### 3.5 FRF: A Method Based on an Ensemble of Fuzzy Decision Trees

Fuzzy Random Forest (FRF) method [8, 12] is a multiple classifier system (ensemble) to DM. FRF is a predictive method for classification and show us its ability to handle imperfect data both in the model learning and in the inference process.

In Fig. 4, the general scheme of the process followed by FRF method is shown. FRF obtains a model with the structure of an ensemble based on FDTs. The learning

**Fig. 4** A general scheme of the FRF method



phase generates FDTs with the following characteristics: (a) each FDT is constructed from a dataset obtained by bagging, (b) the FDTs are constructed without considering all the attributes to split the nodes (a random subset of the set of attributes available at each node is selected), (c) the numerical attributes are discretized by fuzzy partitions, (d) each FDT is constructed to the maximum size and without pruning, (e) a function ( $\chi_{t,N}(\cdot)$ ) is used to indicate the degree with which an instance satisfies the conditions that lead to node  $N$  of tree  $t$ , and (f) FDTs support instances with imperfect values (a function  $\mu_{simil}(\cdot)$  is used to measure the membership degree of these types of values to the fuzzy sets forming the partition of the numerical attributes).

From the obtained model, FRF method uses two strategies to combine the information of several FDTs and to obtain the final decision for a target instance. Strategy 1 combines the information from the different leaves reached in each FDT to obtain the decision of each individual FDT and then applying the same or another combination method to generate the global decision of the FRF model. Strategy 2 combines the information from all reached leaves from all FDTs to generate the global decision of the FRF model.

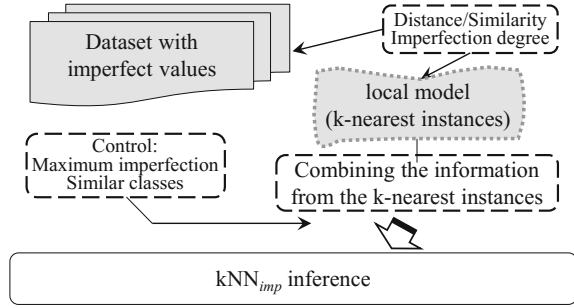
The method assigns class  $\omega_M$  to a new instance such that  $\omega_M = \operatorname{argmax}_i \{D\_FRF_i\}$  where  $D\_FRF$  is a vector with size  $I$  that indicates the confidence assigned by the method to each class  $i$ . The vector elements are obtained from the support for each class in the leaves reached when applying the several strategies and combination methods.

The results obtained by FRF method are promising concluding that by using imperfect values instead of crisp, we capture better the nature of the underlying information.

### 3.6 $KNN_{imp}$ : A Method Based on Instances

The  $kNN_{imp}$  method [14] is a k-nearest neighbors classifier from datasets with imperfect values to DM. Figure 5 shows the general scheme followed by  $kNN_{imp}$  method. This method belongs to the methods with lazy learning, that is, the method does not

**Fig. 5** A general scheme of the  $kNN_{imp}$  method



need of an explicit learning phase. Therefore, this method requires that all dataset instances are stored.

To classify a instance, the  $kNN_{imp}$  method computes its “ $k$ ” nearest instances and generates a class value from them (a local model dependent on the new instance has been constructed). By containing imperfect values the dataset, the importance of each instance (neighbor) in the output decision is based on relative distance/similarity  $d_{imp}(\cdot, \cdot)$  (distance/similarity measures that support imperfect data) and its degree of imperfection. Specifically, for each instance, two weights are calculated depending on its degree of imperfection  $p(\cdot)$  and its distance/similarity  $q(\cdot)$ .

Furthermore, the overall degree of imperfection in “ $k$ ” nearest instances is measured, if it is too high, the classification is not performed. To establish the maximum degree of imperfection,  $kNN_{imp}$  method uses the parameter  $U_I$ .

Once the local model is obtained ( $k$  nearest instances),  $kNN_{imp}$  method combines the information provided for each neighbor instance (weights  $p(\cdot)$  and  $q(\cdot)$ ) to obtain the set of possible weighted classes. The class with the highest score is chosen as output, together with other classes whose score is similar to the highest. To assess if a class should be included in the final output, this method uses the threshold  $U_D$ .

The method obtains a fuzzy subset  $\{\mu(\omega_i)/\omega_i\}$  as possible values to the class attribute of the new instance where  $\mu(\omega_i) = \frac{\sum_j^k \mu^j(\omega_i)p(x^j)q(x^j)}{\sum_j^k \mu^j(\omega_i)p(x^j)q(x^j)}$  and  $\mu^j(\omega_i)$  is the membership degree of the  $j$ -th neighbor to the class value  $\omega_i$ . Therefore, the method assigns to the new instance the class  $\omega_M = \text{argmax}_i \{\mu(\omega_i)\}$  or the fuzzy subset  $\{\omega_M, \omega_i\}$ , with  $\frac{\omega_M - \omega_i}{\omega_M} > U_D$ .

The  $kNN_{imp}$  classifier is robust when working with imperfect data and maintains a good performance when is compared with other methods in the literature, applied to datasets with or without imperfection.

## 4 Conclusions

In data-driven application domains, the suitable use of available information is very important. Because of this, it becomes increasingly necessary to design methods that support different types of information (imperfect or not) and obtain more flexible

models with an appropriate behavior. In this framework, the hybridization of the tools provided by Soft Computing and Intelligent Data Analysis methods is a field that is gaining more importance. In this work, some proposals that carry out this hybridization obtaining quite satisfactory results are commented and analyzed. For this reason we consider that it is a field in which new proposals must be made with the objective of approaching the Intelligent Data Analysis process from datasets that express the true nature of the information.

**Acknowledgements** Supported by the projects TIN2014-52099-R (EDISON) and TIN2014-56381-REDT (LODISCO) granted by the Ministry of Economy and Competitiveness of Spain (including ERDF support).

## References

1. Abe, S.: Fuzzy support vector machines for multilabel classification. *Pattern Recogn.* **48**(6), 2110–2117 (2015)
2. Barua, A., Mudunuri, L.S., Kosheleva, O.: Why trapezoidal and triangular membership functions work so well: Towards a theoretical explanation. *J. Uncertain Syst.* **8**(3), 164–168 (2014)
3. Battiti, R.: Using mutual information for selection features in supervised neural net learning. *IEEE Trans. Neural Netw.* **5**, 537–550 (1994)
4. Berlanga, F.J., Rivera, A., del Jesús, M.J., Herrera, F.: GP-COACH: genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems. *Inf. Sci.* **180**(8), 1183–1200 (2010)
5. Berthold, M.R., Borgelt, C., Höppner, F., Klawonn, F.: *Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data*. Springer (2010)
6. Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press (1981)
7. Bonissone, P.: Soft computing: the convergence of emerging reasoning technologies. *Soft Comput.* **1**(1), 6–18 (1997)
8. Bonissone, P., Cadenas, J.M., Garrido, M.C., Díaz-Valladares, R.A.: A fuzzy random forest. *Int. J. Approximate Reasoning* **51**(7), 729–747 (2010)
9. Boulle, M.: Khiops: a statistical discretization method of continuous attributes. *Mach. Learn.* **55**(1), 53–69 (2004)
10. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: *Classification and Regression Trees*. CRC Press (1984)
11. Cadenas, J.M., Garrido, M.C., Martínez, R.: Feature subset selection filter-wrapper based on low quality data. *Expert Syst. Appl.* **40**(16), 6241–6252 (2013)
12. Cadenas, J.M., Garrido, M.C., Martínez, R., Bonissone, P.P.: Extending information processing in a fuzzy random forest ensemble. *Soft Comput.* **16**(5), 845–861 (2012)
13. Cadenas, J.M., Garrido, M.C., Martínez, R., Bonissone, P.P.: OFP\_CLASS: a hybrid method to generate optimized fuzzy partitions for classification. *Soft Comput.* **16**(4), 667–682 (2012)
14. Cadenas, J.M., Garrido, M.C., Martínez, R., Muñoz, E., Bonissone, P.P.: A fuzzy k-nearest neighbor classifier to deal with imperfect data. *Soft Comput.* (2017). doi:[10.1007/s00500-017-2567-x](https://doi.org/10.1007/s00500-017-2567-x)
15. Chan, C.C., Batur, C., Srinivasan, A.: Determination of quantization intervals in rule based model for dynamic systems. In: *Proceedings of the International Conference on Systems, Man, and Cybernetics*, pp. 1719–1723 (1991)
16. Choi, Y.S., Moon, B.R.: Feature selection in genetic fuzzy discretization for the pattern classification problems. *IEICE Trans. Inf. Syst.* **90**(7), 1047–1054 (2007)

17. Clark, P., Boswell, R.: Rule induction with CN2: Some recent improvements. In: Y. Kodratoff (ed.) *Lecture Notes in Artificial Intelligence. Machine Learning EWSL-91*, pp. 151–163. Springer (1991)
18. Cox, E., Taber, R., O'Hagen, M.: *The Fuzzy Systems Handbook*, 2nd edn. Academic Press (1998)
19. Crockett, K., Bandar, Z., Mclean, D.: Growing a fuzzy decision forest. In: *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 614–617 (2001)
20. Derrac, J., García, S., Herrera, F.: Fuzzy nearest neighbor algorithms: taxonomy, experimental analysis and prospects. *Inf. Sci.* **260**, 98–119 (2014)
21. Djouadi, A., Bouktache, E.: A fast algorithm for the nearest-neighbor classifier. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(3), 277–282 (1997)
22. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley (2001)
23. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: an overview. In: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds.) *Advances in Knowledge Discovery and Data Mining*, pp. 1–34. American Association for Artificial Intelligence (1996)
24. Fernández, A., del Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. *Int. J. Approximate Reasoning* **50**(3), 561–577 (2009)
25. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.* **2**(2), 139–172 (1987)
26. Fukunaga, K.: *Introduction to statistical pattern recognition*. Academic Press (1990)
27. García, S., Luengo, J., Herrera, F.: *Data Preprocessing in Data Mining*. Springer (2015)
28. Garrido, M.C., Cadenas, J.M., Bonissone, P.P.: A classification and regression technique to handle heterogeneous and imperfect information. *Soft Comput.* **14**(11), 1165–1185 (2010)
29. George, G., Raj, V.C.: Review on feature selection techniques and the impact of svm for cancer classification using gene expression profile. *Int. J. Comput. Sci. Eng. Surv.* **2**(3), 16–27 (2011)
30. Ghosh, A., Pal, S.K.: *Soft computing approach to pattern recognition and image processing*. World Scientific (2002)
31. Hamamoto, Y., Uchimura, S., Tomita, S.: A bootstrap technique for nearest neighbor classifier design. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(1), 73–79 (1997)
32. Hecht-Nielsen, R.: *Neurocomputing*. Addison-Wesley (1990)
33. Ho, K.M., Scott, P.D.: Zeta: a global method for discretization of continuous variables. In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pp. 191–194 (1997)
34. Holte, R.C.: Very simple classification rules perform well on most on most commonly used datasets. *Mach. Learn.* **11**, 63–90 (1993)
35. Huang, Z., Ng, M.K.: A fuzzy k-modes algorithm for clustering categorical data. *IEEE Trans. Fuzzy Syst.* **7**(4), 446–452 (1999)
36. Isasi, P., Galván, I.: *Las redes neuronales artificiales - Un enfoque práctico*. Pearson - Prentice Hall (2004)
37. Ishibuchi, H., Yamamoto, T.: Rule weight specification in fuzzy rule-based classification systems. *IEEE Trans. Fuzzy Syst.* **13**(4), 428–435 (2005)
38. Janikow, C.Z.: Fuzzy decision trees: issues and methods. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **28**(1), 1–14 (1998)
39. Janikow, C.Z.: Fuzzy decision forest. In: *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society*, pp. 480–483 (2003)
40. Janikow, C.Z., Fajfer, M.: Fuzzy partitioning with fid3. 1. In: *Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society*, pp. 467–471 (1999)
41. Jensen, R., Shen, Q.: Fuzzy-rough sets assisted attribute selection. *IEEE Trans. Fuzzy Syst.* **15**(1), 73–89 (2007)
42. Kabir, M., Shahjahan, M., Murase, K.: A new hybrid ant colony optimization algorithm for feature selection. *Expert Syst. Appl.* **39**, 3747–3763 (2012)

43. Kbir, M.A., Benkirane, H., Maalmi, K., Benslimane, R.: Hierarchical fuzzy partition for pattern classification with fuzzy if-then rules. *Pattern Recogn. Lett.* **21**(6), 503–509 (2000)
44. Kerber, R.: ChiMerge: discretization of numeric attributes. In: *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 123–128 (1992)
45. Kohonen, T.: *Self-Organising Maps*, 3rd edn. Springer (2001)
46. Kurgan, L.A., Cios, K.J.: Caim discretization algorithm. *IEEE Trans. Knowl. Data. Eng.* **16**(2), 145–153 (2004)
47. Lee, K.M., Lee, K.M., Lee, J.H., Lee-Kwang, H.: A fuzzy decision tree induction method for fuzzy data. *Proceedings of the IEEE International Conference on Fuzzy Systems* **1**, 16–21 (1999)
48. Li, C.: A combination scheme for fuzzy partitions based on fuzzy majority voting rule. In: *Proc. of the International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, pp. 675–678 (2009)
49. Li, D., Gu, H., Zhang, L.: A fuzzy c-means clustering algorithm based on nearest-neighbor intervals for incomplete data. *Expert Syst. Appl.* **37**(10), 6942–6947 (2010)
50. Lin, C.F., Wang, S.D.: Fuzzy support vector machines. *IEEE Trans. Neural Netw.* **13**(2), 464–471 (2002)
51. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: an enabling technique. *Data Min. Knowl. Discov.* **6**(4), 393–423 (2002)
52. Liu, H., Setiono, R.: Feature selection via discretization. *IEEE Trans. Knowl. Data. Eng.* **9**(4), 642–645 (1997)
53. Luukka, P.: Feature selection using fuzzy entropy measures with similarity classifier. *Expert Syst. Appl.* **38**(4), 4600–4607 (2011)
54. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. *Pattern Recogn.* **45**(9), 3084–3104 (2012)
55. Marsala, C.: Data mining with ensembles of fuzzy decision trees. In: *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pp. 348–354 (2009)
56. Mayer-Schönberger, V., Cukier, K.: *Big Data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt (2013)
57. Michalski, R.S., Moztet, I., Hong, J., Lavrac, N.: The AQ15 inductive learning system: an overview and experiments. In: *Proceedings of the International Meeting on Advances in Learning* (1986)
58. Micó, M.L., Oncina, J., Vidal, E.: A new version of the nearest-neighbour approximating and eliminating search algorithm with linear preprocessing time and memory requirements. *Pattern Recogn. Lett.* **15**(1), 9–17 (1994)
59. Mirkin, B.: *Mathematical Classification and Clustering*. Kluwer Academic Publishers (1996)
60. Mitra, S., Pal, S.K.: Fuzzy multi-layer perceptron, inferencing and rule generation. *IEEE Trans. Neural Netw.* **6**(1), 51–63 (1995)
61. Mitra, S., Pal, S.K., Mitra, P.: Data mining in soft computing framework: a survey. *IEEE Trans. Neural Netw.* **13**(1), 3–14 (2002)
62. Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. *Neural Comput.* **1**(2), 281–294 (1989)
63. Myles, A.J., Brown, S.D.: Induction of decision trees using fuzzy partitions. *J. Chemom.* **17**(10), 531–536 (2003)
64. Nascimento, S., Mirkin, B., Moura-Pires, F.: A fuzzy clustering model of data and fuzzy c-means. *Proceedings of the Ninth IEEE International Conference on Fuzzy Systems* **1**, 302–307 (2000)
65. Nauck, D., Kruse, R.: A neuro-fuzzy method to learn fuzzy classification rules from data. *Fuzzy Sets Syst.* **89**(3), 277–288 (1997)
66. Olaru, C., Wehenkel, L.: A complete fuzzy decision tree technique. *Fuzzy Sets Syst.* **138**(2), 221–254 (2003)
67. Palacios, A.M., Palacios, J.L., Sánchez, L., Alcalá-Fdez, J.: Genetic learning of the membership functions for mining fuzzy association rules from low quality data. *Inf. Sci.* **295**, 358–378 (2015)

68. Palacios, A.M., Sánchez, L., Couso, I.: Extending a simple genetic cooperative-competitive learning fuzzy classifier to low quality datasets. *Evol. Intell.* **2**(1–2), 73–84 (2009)
69. Palacios, A.M., Sanchez, L., Couso, I.: Boosting of fuzzy rules with low quality data. *Multiple-Valued Logic Soft Comput.* **19**(5–6), 591–619 (2012)
70. Palacios, A.M., Sanchez, L., Couso, I.: An extension of the FURIA classification algorithm to low quality data. In: *Proceedings of the International Conference on Hybrid Artificial Intelligence Systems*, pp. 679–688 (2013)
71. Quinlan, J.R.: Induction of decision trees. *Mach. Learn.* **1**, 81–106 (1986)
72. Quinlan, J.R.: *C4.5 Programs for Machine Learning*. Morgan Kaufmann (1993)
73. Redmond, S.J., Heneghan, C.: A method for initialising the k-means clustering algorithm using kd-trees. *Pattern Recogn. Lett.* **28**(8), 965–973 (2007)
74. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., Suter, B.W.: The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Trans. Neural Netw.* **1**(4), 296–298 (1990)
75. Saeyns, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. *Bioinformatics* **23**(19), 2507–2517 (2007)
76. Saha, S., Bandyopadhyay, S.: A fuzzy genetic clustering technique using a new symmetry based distance for automatic evolution of clusters. In: *Proceedings of the International Conference on Computing: Theory and Applications*, pp. 309–314 (2007)
77. Sánchez, L., Suárez, M.R., Villar, J.R., Couso, I.: Mutual information-based feature selection and partition design in fuzzy rule-based classifiers from vague data. *Int. J. Approximate Reasoning* **49**, 607–622 (2008)
78. Suárez, M.R., Villar, J.R., Grande, J.: A feature selection method using a fuzzy mutual information measure. *Int. J. Reasoning-Based Intell. Syst.* **2**, 133–141 (2010)
79. Toivonen, H.: Sampling large databases for association rules. In: *Proceedings of the 22nd Conference on Very Large Databases*, vol. 96, pp. 134–145 (1996)
80. Torra, V.: Fuzzy c-means for fuzzy hierarchical clustering. In: *Proceedings of the 14th IEEE International Conference on Fuzzy Systems*, pp. 646–651 (2005)
81. Tsai, C.J., Lee, C.I., Yang, W.P.: A discretization algorithm based on class-attribute contingency coefficient. *Inf. Sci.* **178**, 714–731 (2008)
82. Verdegay, J.L., Yager, R.R., Bonissone, P.: On heuristic as a fundamental constituent of soft computing. *Fuzzy Sets Syst.* **159**(7), 846–855 (2008)
83. Vieira, S.M., Sousa, J.M.C., Kaymak, U.: Fuzzy criteria for feature selection. *Fuzzy Set Syst.* **189**, 1–18 (2012)
84. Wang, Y., Li, C., Zuo, Y.: A selection model for optimal fuzzy clustering algorithm and number of clusters based on competitive comprehensive fuzzy evaluation. *IEEE Trans. Fuzzy Syst.* **17**(3), 568–577 (2009)
85. Witten, I.H., Frank, E., Hall, M.: *Data Mining—Practical Machine Learning Tools and Techniques*. Morgan Kaufmann (2011)
86. Wong, M.L., Leung, K.S.: *Data Mining Using Grammar Based Genetic Programming and Applications*. Kluwer Academic Publishers (2002)
87. Yan-Qing, Y., Ju-Sheng, M., Zhou-Jun, L.: Attribute reduction based on generalized fuzzy evidence theory in fuzzy decision systems. *Fuzzy Sets Syst.* **170**, 64–75 (2011)
88. Yang, Y., Jia, Z., Chang, C., Qin, X., Li, T., Wang, H., Zhao, J.: An efficient fuzzy kohonen clustering network algorithm. *Proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery* **1**, 510–513 (2008)
89. Younes, Z., Abdallah, F., Dencœux, T.: Fuzzy multi-label learning under veristic variables. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1–8 (2010)
90. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**(3), 338–353 (1965)
91. Zadeh, L.A.: Soft computing and fuzzy logic. *IEEE Softw.* **11**(6), 48–56 (1994)

Soft Computing Based Optimization and Decision  
Models

To Commemorate the 65th Birthday of Professor José  
Luis "Curro" Verdegay

Pelta, D.A.; Cruz, C. (Eds.)

2018, XV, 308 p. 52 illus., 34 illus. in color., Hardcover

ISBN: 978-3-319-64285-7