

Automatic Construction of Fuzzy Rule Bases: a further Investigation into two Alternative Inductive Approaches

Marcos Evandro Cintra, Heloisa Arruda Camargo
Estevam R. Hruschka Jr., Maria do Carmo Nicoletti

(Federal University of São Carlos (UFSCar)
Computer Science Department-São Carlos, Brazil
{marcos_cintra, heloisa, estevam, carmo}@dc.ufscar.br)

Abstract: The definition of the Fuzzy Rule Base is one of the most important and difficult tasks when designing Fuzzy Systems. This paper discusses the results of two different hybrid methods, previously investigated, for the automatic generation of fuzzy rules from numerical data. One of the methods, named DoC-based, proposes the creation of Fuzzy Rule Bases using genetic algorithms in association with a heuristic for preselecting candidate rules based on the degree of coverage. The other, named BayesFuzzy, induces a Bayesian Classifier using a dataset previously granulated by fuzzy partitions and then translates it into a Fuzzy Rule Base. A comparative analysis between both approaches focusing on their main characteristics, strengths/weaknesses and easiness of use is carried out. The reliability of both methods is also compared by analyzing their results in a few knowledge domains.

Key Words: machine learning, fuzzy logics, Bayesian networks, Bayesian classification, genetic fuzzy systems

Category: I.2, I.2.6

1 Introduction

Systems based on Fuzzy Logic, generally called Fuzzy Systems (FS), have been successfully used for the solution of problems in many different areas, including pattern classification, optimization, and control of processes [Pedrycz, 1996].

The Fuzzy Systems of interest in this work are those known as Rule Based Fuzzy Systems (RBFS). Usually, a RBFS has two main components: a Knowledge Base (KB) and an Inference Mechanism (IM). The KB comprises the Fuzzy Rule Base (FRB), i.e., a set of fuzzy rules that represents a given problem, and the Fuzzy Data Base (FDB), which contains the definitions of the fuzzy sets related to the linguistic variables used in the FRB. The IM is responsible for carrying out the required computation that uses inferences to derive the output (or conclusion) of the system, based on both the KB and the input to the system.

Many approaches and methods can be used for the automatic generation of the KB from data, representing samples (or examples) of a problem. Clustering algorithms [Liao et al., 1997], neural networks [Jang et al., 1997], and Genetic Algorithms (GA) [Cordón et al., 2004] are among the most well-succeeded techniques. Recently there has been a considerable research effort focusing on the

use of GA [Goldberg, 1989] in the design of FS. This initiative coined the term Genetic Fuzzy Systems (GFS), which are, basically, FS with a learning process controlled by GA [Cordón et al., 2007, 2004].

As suggested in a few references, a very promising approach is the use of GA to generate FRBs based on previously defined and fixed fuzzy sets [Hoffmann, 2004; González and Pérez, 1999; Ishibuchi et al., 1999]. This approach was adopted by Castro & Camargo who proposed a method consisting of three consecutive steps: an attribute selection process, the use of a genetic algorithm to induce rules and, in sequence, the use of another GA to eliminate unnecessary rules [Castro and Camargo, 2005]. Depending on the number of variables and sets in the defined partition, however, the total number of possible rules can be extremely large, making it difficult to generate and codify the chromosomes and, consequently, the whole genetic learning process becomes overloaded.

As an alternative approach to deal with the dimensionality problem, Cintra & Camargo proposed the genetic generation of FRBs from a set of candidate rules preselected by a heuristic criteria based on the Degree of Coverage (DoC), named DoC-based method, as described in [Cintra and Camargo, 2007a,b]; the DoC-based method was used in the experiments described in this paper.

An entirely different proposal for the automatic generation of FRBs from data can be found in [Hruschka Jr et al., 2007], where the combination of the fuzzy granulation of datasets with a Bayesian Classifier (BC) learning process is investigated. The proposed method, named BayesFuzzy, aims at improving the comprehensibility of an induced BC by translating it into a FRB.

Besides presenting a description of the main characteristics of both, the DoC-based as well as the BayesFuzzy method, the objective of this paper is to compare the results of both methods in a few knowledge domains.

The remainder of the paper is organized as follows. In Section 2 the fundamental concepts of Fuzzy Classification Systems, Bayesian Networks, as well as Bayesian Classifiers are presented. Section 3 describes the genetic generation of FRBs, as well as the DoC-based method, and Section 4 describes the BayesFuzzy method. Section 5 discusses the experiments and comparisons concerning both approaches. Finally, the conclusions and perspectives are presented in Section 6.

2 A Brief Overview of Fuzzy Classification Systems and Bayesian Networks

The goal of this section is to highlight the main concepts of fuzzy classification systems and Bayesian Networks and Classifiers, in order to provide the necessary technical background for the sections that follow.

2.1 Fuzzy Classification Systems

Classification is an important task employed in many different areas such as pattern recognition, decision making and data mining. A classification task can be roughly described as: Given a set of objects $E = \{e_1, e_2, \dots, e_M\}$, also named *patterns*, which are described by n attributes, assign a class C_j from a set of classes $C = \{C_1, C_2, \dots, C_J\}$ to an object e_p , $e_p = (a_{p_1}, a_{p_2}, \dots, a_{p_n})$.

Fuzzy Classification Systems (FCS) are RBFS designed to perform a classification task that requires the attribute domains to be granulated by means of fuzzy partitions. The linguistic variables in the antecedent part of the rules represent attributes, and the consequent part represents a class. A typical classification fuzzy rule can be expressed by:

$$R_k : \mathbf{IF} X_1 \text{ is } A_{1l_1} \mathbf{AND} \dots \mathbf{AND} X_n \text{ is } A_{nl_n} \mathbf{THEN} \text{Class} = C_j \quad (1)$$

where R_k is the rule identifier, X_1, \dots, X_n are the attributes of the pattern considered in the problem (represented by linguistic variables), $A_{1l_1}, \dots, A_{nl_n}$ are the linguistic values used to represent the values of the attributes, and C_j ($j \in \{1, \dots, J\}$) is the class, fuzzy or crisp, the pattern belongs to.

An inference mechanism runs the pattern through the FRB aiming at determining the class it belongs to. Many FCS use the Classic Fuzzy Reasoning Method (CFRM) [González and Pérez, 1999] that classifies a pattern using the rule that has the highest compatibility degree with the pattern, as described next.

Let $e_p = (a_{p_1}, a_{p_2}, \dots, a_{p_n})$ be a pattern to be classified and $\{R_1, R_2, \dots, R_S\}$ the set of S rules of a classification system, each with n antecedents. Let $A_{il_i}(a_{p_i})$, $i = 1, \dots, n$, be the membership degree of attribute value a_{p_i} to the i -th fuzzy set of fuzzy rule R_k as defined in (1). The CFRM applies the three following steps to classify the pattern e_p :

1. Calculates the compatibility degree between the pattern e_p and each rule R_k , for $k = 1, \dots, S$, given by

$$\text{Compat}(R_k, e_p) = \mathbf{t}(A_{1l_1}(a_{p_1}), A_{2l_2}(a_{p_2}), \dots, A_{nl_n}(a_{p_n}))$$

where \mathbf{t} denotes a t -norm.

2. Finds the rule R_{kmax} with the highest compatibility degree with the pattern, i.e.,

$$\text{Compat}(R_{kmax}, e_p) = \max\{\text{Compat}(R_k, e_p)\}, k = 1, 2, \dots, S$$

3. Assigns the class C_j to the pattern e_p , where C_j is the class predicted by the rule R_{kmax} found in the previous step.

2.2 Bayesian Networks and Classifiers

A Bayesian Network (BN) [Pearl, 1988] has a directed acyclic graph structure. Each node in the graph corresponds to a discrete random variable in the domain. In the BN graph, an edge $Y \rightarrow X$ describes a parent child relation, where Y is the parent and X is the child. All parents of X constitute the parent set of X , denoted by $\pi(X)$. Each node X of the BN structure is associated to a Conditional Probability Table (CPT) specifying the probability of each possible state of X , given each possible combination of states of $\pi(X)$. If a node X has no parents, its CPT gives the marginal probabilities of X .

In a Bayesian network where $\lambda(X)$ is the set of children of X , the subset of nodes containing $\pi(X)$, $\lambda(X)$, and the parents of $\lambda(X)$ is called Markov Blanket (MB) of X (see Figure 1). As shown in [Pearl, 1988], the only nodes that have influence on the conditional distribution of a given node X (given the state of all remaining nodes) are the nodes that form the MB of X . Thus, after constructing the network structure from data, the MB of the class attribute can be used as a criterion for selecting a subset of relevant attributes for classification purposes.

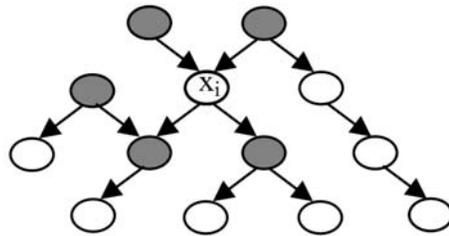


Figure 1: The Markov Blanket of X , represented by the shadowed nodes.

While a BN encodes a joint probability distribution over a set of random variables, a BC aims at correctly predicting the value of a designated discrete class variable, given a vector of attributes. Methods for inducing BNs can be used to induce BCs as well. The BC learning algorithm used in the BayesFuzzy method described in Section 4 is based on the BN learning algorithm known as K2 [Hruschka Jr et al., 2007; Cooper and Herskovits, 1992].

3 The Genetic Generation of Fuzzy Rules - The DoC-based Method

When Genetic Fuzzy Systems focus specifically on the generation or optimization of RBFS, they are named Rule Based Genetic Fuzzy Systems (RBGFS). In the

context of the RBGFS, the well-known methods that combine the genetic and fuzzy approaches for the generation of KBs can be divided into two main groups: methods that adjust KB components (Genetic Adaptation) and methods that build KB components (Genetic Construction).

Included in the group of Genetic Adaptation are the methods that initiate the process with an existing FRB (or FDB) and use GA to improve the performance of the system by adjusting or adapting one or more parts of the KB [Casillas et al., 2005; Gurocak, 1999; Ishibuchi et al., 1997; Bonissone et al., 1996; Herrera et al., 1995]. The group called here Genetic Construction includes the methods that use GA to effectively build or design one or more components of the KB, and comprises the approaches that produced the largest number of researches [Hoffmann, 2004; Cordón et al., 2001; Ishibuchi et al., 1999].

The DoC-based method described in this section, which has been previously introduced in [Cintra and Camargo, 2007a,b], can be included in the group of the Genetic Construction methods that require an additional preprocessing step. This method uses fuzzy partitions defined before the whole learning process starts and also some criteria based on heuristic knowledge for the preselection of candidate rules to be considered by the genetic algorithm when building the final FRB. The preselection of candidate rules aims at reducing the search space and simplifying the chromosomes codification.

A similar approach can be found in [Ishibuchi and Yamamoto, 2004] where a genetic algorithm is used to select the rules that will form the FRB from a set of candidate rules, generated from numerical datasets, based on the confidence and support measurements.

In the automatic generation of a FRB using GA, the search space is defined by the combinations of a certain number of rules from all the possible rules, considering the variables of the problem, as well as the defined fuzzy sets. As the number of variables increases, the set of possible rule combinations that will form the KB exponentially increases, interfering in both, the learning process and its output; in some situations, it makes the whole process unfeasible.

The DoC-based method deals with this problem by using the DoC (Degree of Coverage) of rules in order to reduce the number of rules. Although the DoC value alone is not a selection parameter that tells which rules should be part of the FRB, it allows to discard a large number of possible rules, without any quality loss for the generated FRB. The rules to be discarded are the ones with low or null DoC value. The calculation of the DoC is presented next.

Let $E = \{e_1, e_2, \dots, e_M\}$ be a set of examples. The DoC of the rule R with relation to E (DoC_R) is defined as:

$$DoC_R = \sum_{i=1}^M DoC_{(R, e_i)}$$

where $DoC_{(R, e_i)}$ is the DoC of rule R with respect to example e_i , obtained by

aggregating the membership degrees of the attribute values of e_i in the corresponding fuzzy sets that are in the antecedent part of rule R .

In the DoC-based method, once the fuzzy partitions of the attribute domains are defined, the DoC values are calculated for all possible rules; the rules are then decreasingly ordered by their DoC values. This step allows the use of very simple criteria to select candidate rules as well as a simple representation of rules in the chromosomes. Two criteria were independently used to preselect the candidate rules. One of them uses the Wang & Mendel (WM) method [Wang, 2003; Wang and Mendel, 1992] as a reference. The two criteria are:

1. Select the rules from the ordered set until all the rules present in the FRB generated by the WM method have been selected;
2. Select the rules from the ordered set with non-null DoC values.

The two criteria originated two different versions of the method concerning the preselection phase. In both cases, the set of candidate rules is then used as a reduced search space for the generation of the FRB using GA.

The preselection of candidate rules and their ordering allow the identification of each rule by its position in the ordered list and induce a simple binary codification. The size of each chromosome was set as the total number of preselected rules with a direct correspondence between the rule position in the ordered list and the gene position in the chromosome, so that 0 represents an inactive rule, and 1 represents an active rule. Figure 2 presents a binary chromosome with 10 positions representing 10 rules, with rules 1, 4, 5, 6 and 9 active and all the others inactive.



Figure 2: Binary chromosome representing a complete Rule Base with 5 active and 5 inactive rules.

The WM method was used as a reference for the definition of additional parameters in the DoC-based method. For instance, in the initial population, each chromosome was created with a percentage of active rules based on the number of rules generated by the WM method for a specific dataset. The chromosomes were randomly generated and conflicting rules were eliminated.

To avoid the occurrence of redundancies and to improve the clearness and understanding of the generated FRB, the fitness value was defined based on the Correct Classification Rate (CCR) as well as on the number of rules in the base using the self-adaptive algorithm presented in [Cintra and Camargo, 2007a]. This

way, chromosomes with high CCR and small number of rules were awarded a higher CCR, leading the whole genetic process to a faster convergence. Thus, for each chromosome, the fitness function evaluates its CCR and then divides this value by a penalty constant. This constant is based on the relation between the number of active rules in the chromosome and a reference value found as the best (smallest) number of rules ($Best_NR$) for the best (highest) CCR found so far ($Best_CCR$). The constant values used for the penalization, shown in Table 1, were empirically defined.

Table 1: Penalization rates for the fitness of the chromosomes according to the number of active rules they codify.

	Number of Rules	Fitness Value Becomes
If	$\leq Best_NR$	CCR
Otherwise If	$\leq Best_NR * 1.5$	$CCR/1.25$
Otherwise If	$\leq Best_NR * 2$	$CCR/1.5$
Otherwise If	$\leq Best_NR * 3$	$CCR/2$
Otherwise		$CCR/3$

The number of rules and CCR of the FRB generated by the WM method were used as initial values for the reference parameters $Best_NR$ and $Best_CCR$ respectively. For each generation, the two parameters were automatically updated.

4 The BayesFuzzy Method

The BayesFuzzy method proposed in [Hruschka Jr et al., 2007] allows translating the knowledge represented by a Bayesian Classifier (BC) into a Fuzzy Rule Base (FRB). Considering D a domain described by N attributes, the method applies a fuzzyfication process in the attribute values of D , producing a fuzzyfied dataset D' . BayesFuzzy then induces a BC from D' and extracts, from the BC, a set of classification rules that can be used as the FRB of a fuzzy system. In the conducted experiments described in Section 5, the $K2$ algorithm [Cooper and Herskovits, 1992] was used to induce the BCs.

Algorithm 1 presents the BayesFuzzy procedure used by BayesFuzzy to extract rules from a given BC [Hruschka Jr et al., 2007].

Lines 1 to 9 in Algorithm 1 are the initialization steps. In lines 10 up to 21 the rules are extracted from the BC and inserted into the FRB (called RSR in the algorithm) using the MAP approach; this part of the pseudocode is based on the intuition that the best explanation for a piece of evidence is the most probable state of the world, given the evidence. Each rule corresponds to a specific

instantiation of each variable (antecedents) and the most probable inferred class (consequent). Line 22 consists of removing from the set of rules those containing superfluous conditions; this can be seen as a naive pruning step.

Algorithm 1 BayesFuzzy

input BC : Bayesian classifier with N nodes
 X_1 : Class variable
output: RSR {Reduced set of rules}
begin
 1: $RSR \leftarrow \emptyset$ {reduced set of rules is empty}
 2: $CMB \leftarrow MB(X_1)$ {Markov Blanket of X_1 (class variable)}
 3: $M \leftarrow |CMB|$
 4: Rename the variables in CMB as X_2, X_3, \dots, X_{M+1}
 5: **for** $i \leftarrow 2$ to $M + 1$ **do**
 6: $V_i \leftarrow$ the possible values of variable X_i
 7: $J_i \leftarrow |V_i|$
 8: **end for**
 9: $RI \leftarrow 1$ {rule index}
 10: **for** $k_2 \leftarrow 1$ to j_2 **do**
 11: **for** $k_3 \leftarrow 1$ to j_3 **do**
 12:
 13: **for** $k_{M+1} \leftarrow 1$ to j_{M+1} **do**
 14: Rule_antecedent $\leftarrow X_2 = v_{2,k_2}$ and $X_3 = v_{3,k_3}$ and ... and $X_{M+1} = v_{M+1,k_{M+1}}$
 15: • propagate Rule_antecedent throughout BC and determine the class value Val_Class
 16: • define rule R_{RI} as: *If* Rule_antecedent *then* $X_1 = Val_Class$
 17: $RSR \leftarrow RSR \cup \{R_{RI}\}$
 18: $RI \leftarrow RI + 1$
 19: **end for**
 20: **end for**
 21: **end for**
 22: $RSR \leftarrow remove_irrelevant_rules$ (RSR)
end

The rule extraction process is guided by the MAP (maximum a posteriori) approach. Thus, based on a BC, one rule is created for each possible value of the classifier variables and the class identification is carried out following the most probable state of the class attribute. This is a computationally expensive

procedure mainly because the presence of hundreds or thousands of variables in probabilistic models [Druzdzet, 1996] is very common. In most cases, however, many variables may only be relevant for some types of reasoning; very rarely all of them will be relevant in the reasoning process associated to one single query. Therefore, focusing only on the relevant part of a BC is fundamental when translating it into a set of rules. In this sense, BayesFuzzy explores the Markov Blanket concept, described in Section 2, to select the attributes to be used in the antecedent part of the rules. Thus, the number and the complexity of rules are minimized along with the rule extraction process. The attribute selection strategy, however, does not guarantee a minimal rule set. Therefore, a pruning step may be conducted after the rule set generation.

An interesting issue about the BayesFuzzy method is that the BC structure provides a simple and efficient mechanism (Markov Blanket) to reduce the number and the complexity of the rule set. Another interesting characteristic is that the BC built by BayesFuzzy can be used for predicting the value of any variable (i.e., each variable can be seen as a class variable). This allows a reduction in the time needed to build models when more than one variable can play the role of the class variable.

5 Experiments and Analysis of Results

In this section, the data domains and the results of a few experiments using the two previously described methods are presented and discussed. The results of the Friedman test [Demšar, 2006], suitable for comparing the performance of multiple algorithms in multiple datasets, are also presented and discussed.

All experiments with GA were performed with 250 iterations, elitism rate of 5%, crossover rate of 70% and mutation rate of 5%. For the BayesFuzzy method, the experiments were performed using a 10-fold cross validation strategy. The experiments with GA and WM were performed with a 5-fold cross validation strategy due to time restrictions.

The four domains used are available at the UCI Machine Learning repository [Asuncion and Newman, 2007]. The choice of each dataset was based on their attribute type (numerical-valued ones). Only four randomly selected attributes for each domain were used in an attempt to reduce the computational time of the process. Table 2 summarizes the domain characteristics giving the total number of instances, the percentage of instances for the majority class, and the selected attributes.

For each domain three distinct partitions were defined, with three, five and seven fuzzy sets for each input attribute, totalizing 12 different experiment setups. For each of the 12 setups, four distinct approaches were evaluated: GA with preselection of the best rules (GA I) (selecting all rules in the FRB generated

by the WM algorithm), GA with preselection of rules with non-null DoC (GA II), the BayesFuzzy method, and the WM method.

Table 2: Domain characteristics.

Domain	# Instances	Majority Class	Selected Attributes
Diabetes	724	65.60%	glucose concentration, body mass, insulin level, blood pressure.
MPeG	392	38.52%	acceleration, number of cylinders, weight, power.
Iris	150	33.33%	sepal length, sepal width, petal length, petal width.
Machine	209	57.89%	processor speed, max. memory, cache memory, # of channels.

Table 3 shows the total number of possible rules for each created partition (column Total) and the number of rules in the FRBs produced by each of the four approaches. The numerical suffix added to each domain name represents the number of fuzzy sets for each input variable. To allow further comparisons, the Average number of active rules Per Domain (APD) in the FRBs is presented.

The preselection of candidate rules implemented by the DoC-based method has proved to be a promising approach to reduce the search space and, thus, accelerate the learning process as a whole. The possible refinements of this method can be explored as an efficient tool to cope with the dimensionality problem.

As the rule set reduction carried out by BayesFuzzy is based on the identification of the most relevant attributes for the classification task, the Bayesian and fuzzy techniques collaboration, proposed by this method, is suitable mainly in applications having irrelevant attributes. In such domains, the Markov Blanket attribute selection principle, embedded in BayesFuzzy, enables the reduction of the number of rules as well as their complexity (number of variable in the antecedent part of each rule). Considering also that BayesFuzzy uses a Bayesian classification approach to perform the class prediction, domains in which traditional BCs perform well also favor the BayesFuzzy classification rates.

Table 4 presents the CCRs for the FRBs generated in each experiment, as well as the ranking of the algorithms according to the Friedman test ¹(in parentheses). Notice the average rank of the algorithms in the last line of the table, which themselves, provide a fair comparison of the algorithms. Also, to allow further

¹ The Friedman test is the nonparametric equivalent of the repeated-measures ANOVA. See [Demšar, 2006] for a thorough discussion regarding statistical tests in machine learning research.

comparisons, the Average CCR Per Domain (APD) is presented.

Table 3: Number of rules and average per domain (APD).

Domain	Total	GA I	GA II	BayesFuzzy	WM
Diabetes 3	162.0	14.2	28.4	8.0	24.0
Diabetes 5	1250.0	51.0	67.4	16.3	84.4
Diabetes 7	4802.0	38.6	75.4	37.1	159.2
APD	2071.3	34.6	57.1	20.5	89.2
MPeG 3	243.0	13.0	16.4	4.6	20.0
MPeG 5	3125.0	36.8	46.0	13.6	46.2
MPeG 7	16807.0	34.4	51.2	5.2	76.6
APD	6725.0	28.1	37.6	7.8	47.6
Iris 3	243.0	7.6	13.8	7.7	15.0
Iris 5	1875.0	15.0	41.2	21.4	44.8
Iris 7	7203.0	54.4	61.4	42.0	67.2
APD	3107.0	25.7	38.8	23.7	42.3
Machine 3	243.0	6.2	12.8	9.5	13.8
Machine 5	3125.0	21.6	25.4	50.5	29.0
Machine 7	16807.0	26.0	27.8	109.0	33.0
APD	6725.0	17.9	22.0	56.3	25.3
General Average	4657.1	26.6	38.9	27.1	51.1

The results in Tables 3 and 4 show that the preselection of the best rules, implemented by the GA I scheme produces FRBs with higher CCRs in almost all experiment setups. Compared to GA II, the GA I scheme needed a lesser number of iterations to reach convergence.

The main characteristics of a good FRB are: i) to be described by a small set of high quality rules and ii) to produce high CCR. Thus, the results presented in Tables 3 and 4 should not be analysed independently. In this sense, it is possible to notice that concerning the Diabetes domain, the most accurate FRB is the one having only 29.9 rules on average. Such a FRB was produced by GA I and can be considered the best combination of good CCR and small number of rules in this domain since all other methods produced either FRBs with lower CCR, or a bigger number of rules, or even both.

Concerning the MPeG domain, the best CCR was produced by the FRB generated by GA I. Also for this domain, this method can be considered better than the others because of its combination of the best CCR and a considerable small number of rules.

Table 4: Correct Classification Rates (CCR), Ranks for the Friedman test (RFT) and Average Per Domain (APD).

	GA I	GA II	BayesFuzzy	WM
Domain	CCR (RFT)	CCR (RFT)	CCR (RFT)	CCR (RFT)
Diabetes 3	1.000 (1.0)	0.998 (2.0)	0.711 (4.0)	0.911 (3.0)
Diabetes 5	1.000 (1.0)	0.994 (2.0)	0.743 (4.0)	0.890 (3.0)
Diabetes 7	0.929 (2.0)	0.949 (1.0)	0.765 (4.0)	0.869 (3.0)
APD	0.976 (1.3)	0.975 (1.6)	0.739 (4.0)	0.890 (3.0)
MPeG 3	0.875 (1.0)	0.863 (2.0)	0.839 (3.0)	0.793 (4.0)
MPeG 5	0.787 (1.0)	0.741 (3.0)	0.637 (4.0)	0.768 (2.0)
MPeG 7	0.648 (1.0)	0.518 (3.0)	0.509 (4.0)	0.623 (2.0)
APD	0.777 (1.0)	0.702 (2.7)	0.661 (3.7)	0.728 (2.7)
Iris 3	0.997 (2.0)	0.987 (3.0)	0.957 (4.0)	1.000 (1.0)
Iris 5	1.000 (2.0)	1.000 (2.0)	0.947 (4.0)	1.000 (2.0)
Iris 7	0.983 (1.0)	0.973 (2.0)	0.947 (3.5)	0.947 (3.5)
APD	0.993 (1.7)	0.987 (2.3)	0.950 (3.5)	0.982 (2.2)
Machine 3	0.942 (2.0)	0.951 (1.0)	0.932 (4.0)	0.937 (3.0)
Machine 5	0.943 (2.0)	0.920 (4.0)	0.932 (3.0)	0.956 (1.0)
Machine 7	0.914 (2.0)	0.869 (3.0)	0.864 (4.0)	0.928 (1.0)
APD	0.933 (2.0)	0.913 (2.7)	0.909 (3.7)	0.900 (1.7)
General Average	(1.500)	(2.333)	(3.792)	(2.375)

For the Iris domain GA I produced the best CCR, and its number of rules is very close to the best (smallest) number of rules produced by BayesFuzzy, which, although having the best number of rules, produced the lowest CCR.

For the Machine domain, it is easy to verify that GA I produced the best results since it produced the best CCR and the best number of rules.

To verify if there are statistically significant differences in performance among the four methods, the Friedman test was run under the null-hypothesis, which states that the algorithms are equivalent, so their ranks should be equal. Based on the results, the null-hypothesis can be rejected with 95% confidence level.

6 Conclusion

This work has presented a comparative analysis of two different approaches for the automatic generation of Fuzzy Rule Bases from datasets, namely a genetic-based approach (DoC-based) and a Bayesian-based approach (BayesFuzzy).

The main idea of the genetic approach is the use of an easily obtained knowledge about the dataset to reduce the number of possible rules and thus reduce

the search space. The reduction is accomplished in the initial phase by the selection of candidate rules; potential rules that cannot cooperate in the classification process are discarded before the genetic process begins. Two different criteria were used to direct the choice of the rules; experiments have showed that the criterion based on WM was more successful as far as the CCR values and the number of rules are concerned.

It is important to emphasize that both GA approaches proposed are dependent on the number of variables of the domain as well as on the initial fuzzy partition of each variable domain to generate the initial search space. Particularly, the influence of the initial partition on the results can be observed in Tables 3 and 4 because, for most of the experiments, the ones with smaller number of fuzzy sets defined in their partitions were the ones producing the best FRBs, i. e., they produced the FRBs with the best CCR associated with the smallest number of rules.

In the Bayesian approach, a Bayesian Classifier is used to generate rules from a dataset that has been previously granulated, meaning that the attribute domains were previously organized into categories represented by fuzzy partitions. In the sequel, the classifier is translated into a Fuzzy Rule Base.

A domain characteristic that favors the use of BayesFuzzy is the presence of more than one class attribute. It can be illustrated considering a domain D having M class attributes. In such a domain, BayesFuzzy will build a single BC to extract M classification Rule Sets. Thus, the computational effort to induce the BC is not proportional to the number of class attributes. For this reason as future work it is intended to explore data domains which allow a better understanding of this characteristic.

The paper has discussed the very relevant topic of fuzzy rules generation, focusing on two hybrid approaches, since it has become clear that the automatic generation of fuzzy knowledge bases can substantially profit from using learning techniques based on different methodologies.

Acknowledgements

This research was partially supported by CAPES, CNPq and FAPESP. This paper is an extended version of an earlier conference paper [Cintra et al., 2007].

References

- [Asuncion and Newman, 2007] Asuncion, A., Newman, D.: “UCI machine learning repository - <http://www.ics.uci.edu/~mllearn/MLRepository.html>”; (2007).

- [Bonissone et al., 1996] Bonissone, P., Khedkar, P. S., Chen, Y. T.: “Genetic algorithms for automated tuning of fuzzy controllers, a transportation application”; IEEE International Conference on Fuzzy Systems; (1996), 674–680; new Orleans - EUA.
- [Casillas et al., 2005] Casillas, J., Cordón, O., Jesus, M., Herrera, F.: “Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction”; IEEE Transactions on Fuzzy Systems; 13 (2005), 13–29.
- [Castro and Camargo, 2005] Castro, P. A. D., Camargo, H. A.: “Focusing on interpretability and accuracy of genetic fuzzy systems”; IEEE International Conference on Fuzzy Systems; (2005), 696–701; reno-Nevada - EUA.
- [Cintra and Camargo, 2007a] Cintra, M. E., Camargo, H. A.: “Fuzzy rules generation using genetic algorithms with self-adaptive selection”; IEEE International Conference on Information Reuse and Integration - IRI; (2007a), 261–266; las Vegas, Nevada.
- [Cintra and Camargo, 2007b] Cintra, M. E., Camargo, H. A.: “Genetic generation of fuzzy rules with preselection of candidate rules (in Portuguese)”; VI Encontro Nacional de Inteligência Artificial; (2007b), 1341–1350; rio de Janeiro, Brazil.
- [Cintra et al., 2007] Cintra, M. E., Camargo, H. A., R. Hruschka Jr, E., Nicoletti, M. C.: “Fuzzy rule base generation through genetic algorithms and Bayesian classifiers - a comparative approach”; International Conference on Intelligent Systems Design and Application - ISDA; 1 (2007), 315–320; rio de Janeiro, Brazil.
- [Cooper and Herskovits, 1992] Cooper, G. F., Herskovits, E.: “A Bayesian method for the induction of probabilistic networks from data”; Machine Learning; 9 (1992), 309–347.
- [Cordón et al., 2007] Cordón, O., Alcalá, R., Alcalá-Fdez, J., Rojas, I.: “Special section on genetic fuzzy systems”; IEEE Transactions on Fuzzy Systems; 15 (2007), 533–592.
- [Cordón et al., 2004] Cordón, O., Gomide, F. A. C., Herrera, F., Hoffmann, F., Magdalena, L.: “Special issue on genetic fuzzy systems”; Fuzzy Sets and Systems; 141 (2004).
- [Cordón et al., 2001] Cordón, O., Herrera, F., Villar, P.: “Generating the knowledge base of a fuzzy-based system by the genetic learning of the data base”; IEEE Transactions on Fuzzy Systems; 9 (2001), 4, 667–674.
- [Demšar, 2006] Demšar, J.: “Statistical comparison of classifiers over multiple data sets”; Journal of Machine Learning Research; 7 (2006), 1, 1–30.
- [Druzdzel, 1996] Druzdzel, M. J.: “Qualitative verbal explanations in Bayesian belief networks”; Artificial Intelligence and Simulation of Behavior Quarterly; 94 (1996), 43–54.

- [Goldberg, 1989] Goldberg, D. E.: Genetic algorithms in search, optimization and machine learning; Addison-Wesley, 1989.
- [González and Pérez, 1999] González, A., Pérez, R.: “SLAVE: A genetic learning system based on an iterative approach”; IEEE International Conference on Fuzzy Systems; 7 (1999), 176–191.
- [Gurocak, 1999] Gurocak, H. B.: “A genetic algorithm-based method for tuning fuzzy logic controllers”; Fuzzy Sets and Systems; 108-1 (1999), 39–47.
- [Herrera et al., 1995] Herrera, F., Lozano, M., Verdegay, J. L.: “Tuning fuzzy logic controllers by genetic algorithms”; International Journal of Approximate Reasoning; 12 (1995), 299–315.
- [Hoffmann, 2004] Hoffmann, F.: “Combining boosting and evolutionary algorithms for learning of fuzzy classification rules”; Fuzzy Sets and Systems; 141 (2004), 47–58.
- [Hruschka Jr et al., 2007] Hruschka Jr, E. R., Camargo, H. A., Cintra, M. E., Nicoletti, M. C.: “BayesFuzzy: using a Bayesian classifier to induce a fuzzy rule base”; IEEE International Conference on Fuzzy Systems; (2007), 1788–1793; London, UK.
- [Ishibuchi et al., 1997] Ishibuchi, H., Murata, T., Turksen, I. B.: “Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems”; Fuzzy Sets and Systems; 89 (1997), 134–150.
- [Ishibuchi et al., 1999] Ishibuchi, H., Nakashima, T., Murata, T.: “Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems”; IEEE Transaction on Fuzzy Systems, Man and Cybernetics; 29 (1999), 5, 601–618.
- [Ishibuchi and Yamamoto, 2004] Ishibuchi, H., Yamamoto, T.: “Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining”; Fuzzy Sets and Systems; 141 (2004), 59–88.
- [Jang et al., 1997] Jang, S. R., Sun, C. T., Mizutani, E.: Neuro-fuzzy and soft computing; Prentice Hall, 1997.
- [Liao et al., 1997] Liao, T. W., Celmins, A. K., Hammell, R. J.: “A fuzzy C-Means variant for the generation of fuzzy term sets”; Fuzzy Sets and Fuzzy Systems; 135 (1997), 241–257.
- [Pearl, 1988] Pearl, J.: Probabilistic reasoning in intelligent systems.; Morgan Kaufmann, 1988.
- [Pedrycz, 1996] Pedrycz, W.: Fuzzy modelling: paradigms and practice; Kluwer Academic Press, 1996.
- [Wang, 2003] Wang, L.: “The WM method completed: a flexible fuzzy system approach to data mining”; IEEE International Conference on Fuzzy Systems; 11 (2003), 768–782.
- [Wang and Mendel, 1992] Wang, L., Mendel, J.: “Generating fuzzy rules by learning from examples”; IEEE Trans. on SMC; 22 (1992), 414–427.