Elsevier Editorial System(tm) for Expert Systems With Applications
Manuscript Draft

Manuscript Number:

Title: A Novel Associative Classification Model based on a Fuzzy Frequent Pattern Mining Algorithm

Article Type: Full Length Article

Keywords: Fuzzy association rule-based classifiers;
Fuzzy FP-Growth;
Fuzzy associative classifier;
Fuzzy association rules

Corresponding Author: Prof. Francesco Marcelloni, PhD

Corresponding Author's Institution: University of Pisa

First Author: Michela Antonelli, PhD

Order of Authors: Michela Antonelli, PhD; Pietro Ducange, PhD; Francesco Marcelloni, PhD; Armando Segatori, PhD student

A Novel Associative Classification Model based on a Fuzzy Frequent Pattern Mining Algorithm

Michela Antonelli, Pietro Ducange, Francesco Marcelloni*, Armando Segatori

Dipartimento di Ingegneria dell'Informazione

University of Pisa

56122, Pisa, Italy

Abstract

Associative classification models are based on two different data mining paradigms, namely pattern classification and association rule mining. These models are very popular for building highly accurate classifiers and have been employed in a number of real world applications.

During the last years, several studies and different algorithms have been proposed to integrate associative classification models with the fuzzy set theory, leading to the so called fuzzy associative classifiers.

In this paper, we propose a novel fuzzy associative classification approach based on a fuzzy frequent pattern mining algorithm. Fuzzy items are generated by first discretizing the input variables and building strong fuzzy partitions for each variable. Then, fuzzy associative classification rules are mined by using the fuzzy frequent pattern mining algorithm. Finally, a set of highly accurate classification rules is generated after a pruning stage.

*Corresponding Author: Tel: +39 050 2217678 Fax: +39 050 2217600
Email address: f.marcelloni@iet.unipi.it (Francesco Marcelloni)

We tested our approach on seventeen real-world datasets and compared the achieved results with the ones obtained by using both a non-fuzzy associative classifier, namely CMAR, and a well-known fuzzy classifier, namely FARC-HD, based on fuzzy association rules. Using non-parametric statistical tests, we show that our approach outperforms CMAR and achieves accuracies similar to FARC-HD.
Keywords: Fuzzy association rule-based classifiers, Fuzzy FP-Growth, Fuzzy associative classifier, Fuzzy association rules.

1. Introduction

Pattern classification and association rule mining are two of the most studied data mining paradigms [1]. Pattern classification deals with assigning a class label to an object described by a set of features. The classification task is carried out by using a specific model, namely the classifier, previously built by using a set of training examples. Association rule mining is the task of discovering correlation or other relationships among items in large database [2].

During the last years, association rule mining has become a very popular method to build highly accurate classification models. Such method is able to mine a set of high quality classification rules from huge amounts of data and to achieve a considerable performance in terms of classification accuracy. Associative classification, that is, classification based on association rules, has been extensively studied in the literature [3, 4, 5, 6, 7] and has been recently exploited in a number of real world applications such as detection phishing activities in websites [8], classification of XML documents [9], text analysis

[10] and classification of medical diseases [11, 12].

The hybridization of the two data mining paradigms can be summarized in the following steps which characterize the generation and the use of an associative classifier. First, a set of classification association rules (CARs) is mined from the training set. Second, a rule pruning step discards redundant or noisy information contained in the rule set and selects a subset of high quality CARs. The selected CARs are used to predict the class labels when the model is used for classifying unlabeled patterns.

Association rule mining is crucial to the success of the associative classification models. This mining process is generally performed in three steps. First, frequent itemsets are extracted from the training set. An itemset is frequent when its occurrence in the training set is higher than a prefixed threshold. Then, rules are mined from the frequent itemsets. Finally, rules are pruned by, for instance, considering their confidence and/or redundancy. The identification of fast and efficient algorithms for association rule mining still represents a challenge for the researchers [13, 14, 15, 16].

As stated in [17], even though learning based on association rule mining ensures high accuracy in pattern classification and generates rule-based models which are often "interpretable" by the user, this model suffers from some main weaknesses. First, when the number of training data objects is huge, the complexity of the learning process grows exponentially in terms of both time and memory. Second, association rule mining algorithms deal with binary or categorical itemsets. On the other hand, real data objects are often described by numerical continuous features. Thus, appropriate discretization algorithms have to be applied to transform continuous feature domains into a set of items.

In the literature a number of associative classification approaches, such as CBA [18], LB [19] and PCAR [14] extract itemsets and then mine CARS exploiting the well-known Apriori algorithm [20]. This algorithm uses a "bottom up" approach, where candidate itemsets are generated by extending frequent itemsets one item at a time (a step known as *candidate generation*), and are tested against the overall dataset for evaluating if they are frequent. The algorithm terminates when no further successful frequent extensions are possible. In many cases the Apriori candidate generate-and-test algorithm significantly reduces the size of candidate sets, leading to a good performance gain. However, the Apriori algorithm can suffer from two nontrivial costs [21]: it may still need to generate a huge number of candidate itemsets and to repeatedly scan the overall dataset for verifying whether they are frequent. Since datasets are often very large, scanning the dataset is very expensive, in particular when the dataset cannot be stored in the main memory.

An interesting approach, which mines the complete set of frequent itemsets without generating all the possible candidate itemsets has been proposed in [22]. The approach is called *frequent pattern growth*, or simply FP-Growth, and adopts a divide-and-conquer strategy. First, it compresses the dataset representing frequent items into a *frequent pattern tree*, or FP-tree, which retains the itemset association information. It then divides the compressed dataset into a set of conditional datasets (a special kind of projected datasets), each associated with a frequent item or a pattern fragment. For each pattern fragment, only its associated conditional dataset needs to be examined. Independently of the number of frequent items, FP-Growth scans the overall dataset only twice. On the contrary, Apriori can need several scans of the overall dataset. In [23] the authors have proposed an associative classifier, namely CMAR, just based on the FP-Growth algorithm. Similar to CMAR, we will adopt the FP-Growth algorithm with some modifications for dealing with fuzzy sets.

As regards the issue of managing continuous input variables, associative classification approaches adopt discretization algorithms for extracting a set of items and therefore allowing the rule mining algorithms to work properly. The discretization is accomplished by assigning each value to a bin. The data ranges (*bin boundaries*) and the number of bins are determined by the discretization algorithm. Bin boundaries are typically crisp. In the last years, however, a number of associative classification approaches have used fuzzy boundaries, thus generating fuzzy association rules [17, 24, 25, 26, 27]. Fuzzy boundaries allow modeling more accurately the real transitions between bins.

As an example, in [24], authors introduce a fuzzy associative classifier based on Apriori to mine all fuzzy CARs: notions of support, confidence, redundancy and rule conflict have been extended to the fuzzy context for selecting only the best CARs to build the classifier. Similarly, in [17], authors propose an associative classification model which generates fuzzy CARs by means of a fuzzy version of Apriori. Different methods for generating the initial fuzzy partitions and for classifying the patterns have been experimented.

Recent works, such as [25, 26], exploit the Apriori algorithm for mining fuzzy CARs. The Fuzzy Association Rule-based Classification model for High Dimensional datasets (FARC-HD), proposed in [25], is a fuzzy associative classification approach consisting of three steps. First, all possible fuzzy association rules are mined by applying the Apriori algorithm, limiting the cardinality of the itemsets, so as to generate fuzzy rules with a low number of conditions. Then, a pattern weighting scheme is employed to reduce the number of candidate rules, pre-selecting the most interesting. Finally, a single objective genetic algorithm is applied to select a compact set of fuzzy association rules and to tune the membership functions.

In [26], the authors introduce a fuzzy version of the well-known CBA algorithm [18] and discuss an example of how associative classification models can be used for building recommenders. Finally, in [27], authors propose a fuzzy CAR mining method, which exploits the Fuzzy C-Means clustering algorithm [28] and a multiple support Apriori algorithm [29]. In particular, the Fuzzy C-Means is used to generate the fuzzy partitions of the input attributes and the multiple support Apriori is employed for extracting the fuzzy CARs. The multiple support Apriori allows specifying multiple minimum supports for each item instead of using a single minimum support value for the overall database. The multiple support Apriori permits solving the *rare item dilemma* [29].

In this paper, we propose a new fuzzy association rule-based classification scheme. The proposed method mines fuzzy CARs by using a fuzzy version of the well-known FP-Growth algorithm. Even though some fuzzy versions of FP-Growth have been proposed in the literature [30, 31], to the best of our knowledge, our method represents the first attempt of using such algorithm for deriving fuzzy CARs. In particular, our method integrates a preliminary discretization step of the continuous attributes for creating fuzzy partitions. Then, a fuzzy version of the FP-Growth algorithm is used for mining a set of candidate CARs. Finally, a pruning method is applied for selecting a subset of high quality rules. These rules compose the rule base of a classical fuzzy rule-based classifier (FRBC) that can be used for classifying unlabeled patterns.

We compare the results achieved by the proposed approach on seventeen datasets with the ones achieved by CMAR. By using non-parametric statistical tests, we show that our approach outperforms CMAR in terms of accuracy. Further, we compare the proposed fuzzy associative classifier with FARC-HD. We show that our approach is statistically equivalent to FARC-HD, although it is more scalable.

This paper is organized as follows. Section 2 provides a basic description of the FRBCs and their inference models, and introduces some notations for the CARs. Section 3 describes each phase of the proposed approach and includes the details of the fuzzy FP-Growth. Section 4 presents the experimental setup and discusses the results that are obtained on seventeen real-world datasets. Finally, in Section 5, we draw some final conclusions.

2. Fuzzy Rule-based Classifiers

In this section, we first describe the structure of FRBCs and the inference model adopted for classifying patterns. Then, we introduce some notations for fuzzy association rules for classification.

2.1. Fuzzy Rule-based Classifiers

Pattern classification consists of assigning a class C_k from a predefined set $C = \{C_1, \ldots, C_K\}$ of classes to an unlabeled pattern. We consider a pattern as an F-dimensional point in a feature space $\Re^F$. Let $\mathbf{X} = \{X_1, \ldots, X_F\}$ be the set of input variables and U_f, $f = 1, \ldots, F$, be the universe of discourse of the f^{th} variable. Let $P_f = \{A_{f,1}, \ldots, A_{f,T_f}\}$ be a fuzzy partition of T_f fuzzy sets on variable X_f. The data base (DB) of an FRBC is the set of parameters which describe the partitions P_f of each input variable. The rule base (RB) contains a set of M rules usually expressed as:

$$\begin{aligned} R_m : \textbf{IF } X_1 \textbf{ is } A_{1,j_{m,1}} \textbf{ AND} \ldots \textbf{AND } X_F \textbf{ is } A_{F,j_{m,F}} \\ \textbf{THEN } Y \textbf{ is } C_{j_m} \; \textit{with } RW_m \end{aligned} \tag{1}$$

where Y is the classifier output, C_{j_m} is the class label associated with the m^{th} rule, $j_{m,f} \in [1, T_f]$, $f = 1, ..., F$, identifies the index of the fuzzy set (among the T_f linguistic terms of partition P_f), which has been selected for X_f in rule R_m. RW_m is the rule weight, i.e., a certainty degree of the classification in the class C_{j_m} for a pattern belonging to the fuzzy subspace delimited by the antecedent of rule R_m.

Let $(\mathbf{x}_n, y_n)$ be the n^{th} input-output pair, with $\mathbf{x}_n = [x_{n,1} \ldots, x_{n,F}] \in \Re^F$ and $y_n \in C$. The strength of activation (*matching degree* of the rule with the input) of the rule R_m is calculated as:

$$w_m(\mathbf{x_n}) = \prod_{f=1}^{F} A_{f,j_{m,f}}(x_{n,f}), \tag{2}$$

where $A_{f,j_{m,f}}(x)$ is the membership function (MF) associated with the fuzzy set $A_{f,j_{m,f}}$.

The *association degree* $h_m(\mathbf{x}_n)$ with the class C_{j_m} is calculated as

$$h_m(\mathbf{x}_n) = w_m(\mathbf{x}_n) \cdot RW_m \tag{3}$$

Different definitions have been proposed for the rule weight RW_m [32, 33]. As discussed in [34], the rule weight of each fuzzy rule R_m can improve the performance of FRBCs. In this paper, we adopt the fuzzy confidence value, or certainty factor (CF), defined as follows:

$$RW_m = CF_m = \frac{\sum_{\mathbf{x}_n \in C_{jm}} w_m(\mathbf{x}_n)}{\sum_{n=1}^{N} w_m(\mathbf{x}_n)} \tag{4}$$

where N is the number of input-output pairs contained in the training set T.

An FRBC is also characterized by its *reasoning method*, which uses the information from the RB to determine the class label for a specific input pattern. Two different approaches are often adopted in the literature:

1. *The maximum matching*: an input pattern is classified into the class corresponding to the rule with the maximum association degree calculated for the pattern.
2. *The weighed vote*: an input pattern is classified into the class corresponding to the maximum total strength of vote. In particular, for a new pattern $\widehat{\mathbf{x}}$, the total strength of vote for each class is computed as follows:

   $$V_{C_k}(\widehat{\mathbf{x}}) = \sum_{R_m \in RB; C_{jm} = C_k} h_m(\widehat{\mathbf{x}}) \tag{5}$$

   where $C_k = \{C_1, \ldots, C_K\}$. With this method, each fuzzy rule gives a vote for its consequent class. If no fuzzy rule matches the pattern $\widehat{\mathbf{x}}$, we classify $\widehat{\mathbf{x}}$ as *unknown*.

2.2. Fuzzy Association Rules for Classification

Association rules are rules in the form $Z \rightarrow Y$, where Z and Y are set of items. These rules describe relations among items in a dataset [21]. Association rules have been widely employed in the market basket analysis. Here, items identify products and the rules describe dependencies among the different products bought by customers [2]. Such relations can be used for decisions about marketing activities as promotional pricing or product placements.

In the associative classification context, the single item is defined as the couple $I_{f,j} = (X_f, v_{f,j})$, where $v_{f,j}$ is one of discrete values that the variable X_f, $f = 1, ..., F$, can assume. A generic classification association rule CAR_m is expressed as:

$$CAR_m : Ant_m \rightarrow C_{j_m} \tag{6}$$

where Ant_m is a conjunction of items, and C_{j_m} is the class label selected for the rule among the set $C = \{C_1, \ldots, C_K\}$ of possible classes. For each variable X_f, just one item is typically considered in Ant_m. Antecedent Ant_m can be represented more familiarly as

$$Ant_m : X_1 \textbf{ is } v_{1,j_{m,1}} \ \ldots \ \textbf{AND} \ \ldots \ X_F \textbf{ is } v_{F,j_{m,F}} \tag{7}$$

where $v_{f,j_{m,f}}$ is the value used for variable X_f in rule CAR_m.

Most of the association rule analysis techniques are focused on binary or discrete attributes. However, in the framework of pattern classification, input variables can be also continuous. For continuous variables, a discretization process is used to generate a finite set of Q_f atomic values $V_f = v_{f,1}, \ldots, v_{f,Q_f}$ associated with the specific variable X_f. In this context, fuzzy set theory can offer a very suitable tool for approaching the discretization problem, ensuring a high interpretability of the rules, thanks to the use of linguistic terms, and avoiding unnatural boundaries in the partitioning of the attribute domain [25].

In the fuzzy associative classification context, given a set of attributes $\mathbf{X} = \{X_1, \ldots, X_F\}$ and a fuzzy partition P_f defined for each attribute X_f, the single item is defined as the couple $I_{f,j} = (X_f, A_{f,j})$, where $A_{f,j}$ is one of the fuzzy values defined in the partition P_f of variable X_f, $f = 1, ..., F$. A generic fuzzy classification association rule (FCAR) for classification is expressed as:

$$FCAR_m : FAnt_m \rightarrow C_{j_m} \tag{8}$$

where C_{j_m} is the class label selected for the rule among the set $C = \{C_1, \ldots, C_K\}$ of possible classes and $FAnt_m$ is a conjunction of items. The antecedent $FAnt_m$ can be represented more familiarly as

$$FAnt_m : X_1 \textbf{ is } A_{1,j_{m,1}} \ \ldots \ \textbf{AND} \ \ldots \ X_F \textbf{ is } A_{F,j_{m,F}} \tag{9}$$

where $A_{f,j_{m,f}}$ is the fuzzy value used for variable X_f in rule $FCAR_m$.

In the association rule analysis, support and confidence are the most common measures to determine the strength of an association rule.

Support and confidence can be expressed for a fuzzy classification association rule $FCAR_m$ as follows:

$$fuzzySupp(FAnt_m \rightarrow C_{j_m}) = \frac{\sum_{\mathbf{x}_n \in C_{j_m}} w_m(\mathbf{x}_n)}{N} \tag{10}$$

$$fuzzyConf(FAnt_m \rightarrow C_{j_m}) = \frac{\sum_{x_n \in C_{j_m}} w_m(\mathbf{x}_n)}{\sum_{\mathbf{x}_n \in T} w_{Ant_m}(\mathbf{x}_n)} \tag{11}$$

where T is the training set, N is the number of objects in T, $w_m(\mathbf{x}_n)$ is the matching degree of rule $FCAR_m$ and $w_{FAnt_m}(\mathbf{x}_n)$ is the matching degree of all the rules which have the antecedent equal to $FAnt_m$.

3. The Proposed Approach

In this section, we present our Associative Classifier based on a Fuzzy Frequent Pattern (AC-FFP) mining algorithm. AC-FFP consists of the following three phases:

1. *Discretization*: a fuzzy partition is defined on each linguistic variable by using the multi-interval discretization approach based on entropy proposed by Fayyad and Irani in [35];
2. *FCAR Mining*: a fuzzy frequent pattern mining algorithm, which is an extension of the well known FP-Growth, is exploited to extract frequent fuzzy classification rules with confidence higher than a prefixed threshold;
3. *Pruning*: rule pruning based on redundancy and training set coverage is applied to generate the final RB.

At the end of the three phases, we obtain an FRBC which can be used for the classification task of unlabeled patterns.

In the following, we introduce in detail all the mentioned phases.

3.1. Discretization

The discretization of continous features is a critical aspect in the generation of association rule classifiers. In the last years, several different heuristic methods have been proposed [35, 36, 37, 38]. In this paper, we use the method proposed by Fayyad and Irani in [35]. This supervised method exploits the class information entropy of candidate partitions to select the bin boundaries for discretization.

Let $T_{f,0} = [x_{1,f}, ..., x_{N,f}]^T$ the projection of the training set T along variable X_f and $b_{f,r}$ a bin boundary for the same variable. Let $T_{f,1}$ and $T_{f,2}$ be the subsets of points of the set $T_{f,0}$ which lie in the two bins identified by $b_{f,r}$. The class information entropy of the discretization induced by $b_{f,r}$, denoted as $E(X_f, b_{f,r}; T_{f,0})$ is given by

$$E(X_f, b_{f,r}; T_{f,0}) = \frac{|T_{f,1}|}{|T_{f,0}|} \cdot Ent(T_{f,1}) + \frac{|T_{f,2}|}{|T_{f,0}|} \cdot Ent(T_{f,2}) \tag{12}$$

where $||$ denotes the cardinality and $Ent()$ is the entropy calculated for a set of points [35]. The boundary $b_{f,min}$ which minimizes the class information entropy over all possible partition boundaries of $T_{f,0}$ is selected as a binary discretization boundary. The method is then applied recursively to both the partitions induced by $b_{f,min}$ until the following stopping criterion based on the Minimal Description Length Principle is achieved. Recursive partitioning stops iff

$$Gain(X_f, b_{f,min}; T_{f,0}) < \frac{\log_2(|T_{f,0}| - 1)}{|T_{f,0}|} + \frac{\Delta(X_f, b_{f,min}; T_{f,0})}{|T_{f,0}|} \tag{13}$$

where

$$Gain(X_f, b_{f,min}; T_{f,0}) = Ent(T_{f,0}) - E(X_f, b_{f,min}; T_{f,0}), \tag{14}$$

$$\Delta(X_f, b_{f,min}; T_{f,0}) = \log_2(3^{k_0} - 2) - [k_0 \cdot Ent(T_{f,0}) - k_1 \cdot Ent(T_{f,1}) - k_2 \cdot Ent(T_{f,2})] \tag{15}$$

and k_i is the number of class labels represented in the set $T_{f,i}$.

The method outputs, for each variable, a set of bin boundaries. Let $U_f = [x_{f,l}, x_{f,u}]$ be the universe of variable X_f. Let $\{b_{f,1}, \ldots, b_{f,Q_f}\}$, with $\forall r \in [1, \ldots, Q_f - 1], b_{f,r} < b_{f,r+1}$, be the set of bin boundaries, where $b_{f,1} = x_{f,l}$ and $b_{f,Q_f} = x_{f,u}$. Then, the method identifies the set $\{[b_{f,1}, b_{f,2}], \ldots, [b_{f,Q_f-1}, b_{f,Q_f}]\}$ of contiguous intervals, which partition the universe of variable X_f.

To transform the crisp partition into a strong fuzzy partition, we adopt the following procedure. For each bin $[b_{f,r}, b_{f,r+1}]$, with $r \in [1, \ldots, Q_f - 1]$, we first compute the middle point $m_{f,r} = \frac{b_{f,r} + b_{f,r+1}}{2}$ and then generate three triangular fuzzy sets $A_{f,2r-1}$, $A_{f,2r}$ and $A_{f,2r+1}$ defined as $(m_{f,r-1}, b_{f,r}, m_{f,r})$, $(b_{f,r}, m_{f,r}, b_{f,r+1})$ and $(m_{f,r}, b_{f,r+1}, m_{f,r+1})$, respectively. We recall that a triangular fuzzy set is defined by three points (a, b, c), where b represents the core and a and c correspond to the lower and upper bounds of the support, respectively. The two fuzzy sets $A_{f,1}$ and $A_{f,2Q_f-1}$ at the lower and upper bounds of the universe of X_f are defined as $A_{f,1} = (-\infty, b_{f,1}, m_{f,1})$ and $A_{f,2Q_f-1} = (b_{f,Q_f-1}, m_{Q_f-1}, +\infty)$, respectively. The set $P_f = \{A_{f,1}, \ldots, A_{f,T_f}\}$, where $T_f = 2Q_f - 1$ is the number of fuzzy sets for each feature, defines the fuzzy partition of feature X_f. If no bin boundaries have been found by the algorithm for feature X_f, then no fuzzy value is generated for this feature and the feature is discarded. Figure 1 shows an example of strong fuzzy partition obtained by the fuzzification of the output of the Fayyad and Irani's discretizer.

As shown in Fig. 1 and discussed in the text, the cores of the triangular fuzzy sets are positioned in correspondence to both the middle points and the bin boundaries. We performed different experiments for determining the best number of fuzzy sets and also the best positioning of these fuzzy

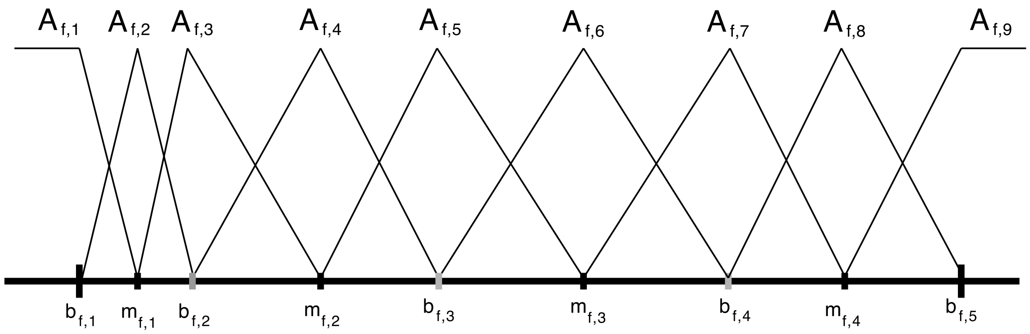


Figure 1: An example of strong fuzzy partition obtained by the fuzzification of the output of the Fayyad and Irani's discretizer.

sets. For instance, we generated strong fuzzy partitions by using only the middle points or only the bin boundaries. We verified that the best results in terms of accuracy are obtained by using fuzzy sets positioned on both middle points and bin boundaries. On the other hand, the fuzzy sets positioned on the middle points allow modeling accurately the instances in the bin and consequently the class connected to the bin. Further, the fuzzy sets on the bin boundaries permit to finely discriminate instances belonging to two different bins and possibly different classes.

3.2. FCAR Mining

To mine the FCARs from the dataset, we introduce a novel fuzzy frequent pattern (FFP) mining algorithm. This algorithm is based on the well known FP-Growth proposed by Han et al. in [22] for efficiently mining frequent patterns without generating candidate itemsets. The algorithm consists of two phases. The first phase of FP-Growth creates an FP-tree from the dataset and the second phase extracts frequent patterns from the FP-tree. The creation of the FP-tree is performed in three steps. First, the dataset is scanned to find the frequent items. Then, these items are sorted in descending frequency. Finally, the dataset is scanned again to construct the FP-tree according to the sorted order of frequent items.

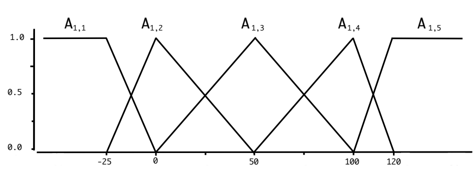
In the second phase, all frequent itemsets are mined from the FP-tree. For each item, a conditional FP-tree is generated and from this tree the frequent itemsets, including the processed item, are recursively mined.

Some papers have already proposed to integrate the fuzzy theory with the FP-Growth method. In [31] the authors choose only the most frequent linguistic value for each variable to build the FP-tree. For example, if the f^{th} partition contains T_f fuzzy sets, only one of these fuzzy sets is used to mine rules. Thus, only a limited subset of rules is generated and therefore useful information resulting from other fuzzy items might be removed. A similar approach is presented in [39]. Unlike these approaches, in AC-FFP we try to preserve information as much as possible.

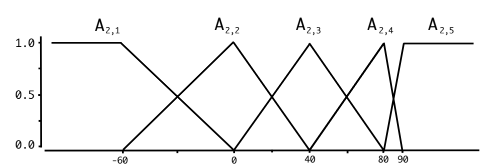
AC-FFP performs four scans of the dataset. The first two scans determine the fuzzy frequent values and build the FP-tree, respectively. The FP-tree is therefore used to mine fuzzy frequent patterns and then FCARs. The third and fourth scans are needed to compute fuzzy support and confidence, and the training set coverage, respectively, in the pruning phase. In the following, we will describe in detail the operations performed in the four scans with the help of an example of application. In the example, we adopt the training set shown in Table 1. Further, we assume that the discretization and the subsequent fuzzification process have partitioned the input variables as in Figure 2.

Table 1: A simple dataset characterized by four input features.

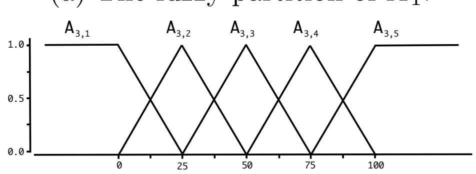
ID	X_1	X_2	X_3	X_4	**Class**
1	20	20	0	10	C_1
2	25	-60	10	80	C_3
3	-25	40	100	40	C_1
4	75	60	35	110	C_2
5	20	80	100	75	C_2
6	30	90	75	10	C_3
7	120	50	75	-25	C_1



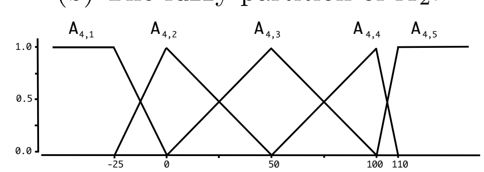
(a) The fuzzy partition of X_1.



(b) The fuzzy partition of X_2.



(c) The fuzzy partition of X_3.



(d) The fuzzy partition of X_4.

Figure 2: The fuzzy partitions of each variable in the example.

In the first scan, AC-FFP calculates the fuzzy support of each fuzzy value $A_{f,j}$. The fuzzy support is computed as:

$$fuzzySupp(A_{f,j}) = \frac{\sum_{n=1}^{N} A_{f,j}(x_{f,n})}{N} \tag{16}$$

Only the fuzzy values, called *frequent fuzzy values*, whose support is larger than the support threshold $minSup$ (0.2 in the example) are retained and organized in a list, called f_{list}, in support descending order. The other fuzzy values are pruned and therefore not considered in the FCAR mining. In Table 2, we show the fuzzy supports calculated for each fuzzy set considered in the example. From the analysis of Table 2, the following f_{list} is generated:

$f_{list} = \{A_{2,3}\ ,\ A_{1,3}\ ,\ A_{1,2}\ ,\ A_{4,3}\ ,\ A_{3,4}\ ,\ A_{3,5}\ ,\ A_{4,2}\ ,\ A_{2,4}\ ,\ A_{3,1}\ \}$.

Table 2: The fuzzy supports of each fuzzy set in the example.

Fuzzy Value	**Fuzzy Support**	**Fuzzy Value**	**Fuzzy Support**
$A_{1,1}$	0.14	$A_{3,1}$	0.23
$A_{1,2}$	0.30	$A_{3,2}$	0.14
$A_{1,3}$	0.34	$A_{3,3}$	0.06
$A_{1,4}$	0.07	$A_{3,4}$	0.29
$A_{1,5}$	0.14	$A_{3,5}$	0.29
$A_{2,1}$	0.14	$A_{4,1}$	0.14
$A_{2,2}$	0.07	$A_{4,2}$	0.26
$A_{2,3}$	0.38	$A_{4,3}$	0.30
$A_{2,4}$	0.25	$A_{4,4}$	0.16
$A_{2,5}$	0.14	$A_{4,5}$	0.14

In the second scan, AC-FFP builds the FP-tree in order to mine all the FCARs. The generation of the FP-tree is performed as in FP-Growth: the only difference is that here the items correspond to the fuzzy values. Actually, if we consider the example fuzzy partition in Figure 1, we can observe that each value on the universe belongs to two different fuzzy values with different membership grades. Thus, two fuzzy values should be associated with each value. However, if we associated two fuzzy values for each feature value, each object would generate 2^F patterns.

To limit the number of possible patterns, we assign each continuous value to the fuzzy set with the highest membership value (in case of tie, we randomly select one of the two fuzzy sets). Each object $\mathbf{x}_n$ is therefore transformed into a fuzzy object $\widetilde{\mathbf{x}}_n = \left\{A_{1_n,j_{1_n}}, \ldots, A_{Z_n,j_{Z_n}}\right\}$, where $A_{i_n,j_{i_n}}$, $i_n \in [1, \ldots, F]$, $j_{i_n} \in [1, \ldots T_{i_n}]$, indicates the frequent fuzzy value selected for feature i_n. The fuzzy values in $\widetilde{\mathbf{x}}_n$ are sorted in the same order as in the f_{list}, as required by the FP-Growth algorithm. Obviously, the number of features, which describe the fuzzy object, can be lower than F. Table 3 shows for each pattern of the example dataset, the fuzzy values associated with the highest membership degree and the corresponding fuzzy objects for each pattern in the example training set.

The fuzzy objects are used to build the FP-tree. Each branch from the root to a leaf node describes a fuzzy rule. When a fuzzy object of the training set is added to the FP-tree, the fuzzy values are considered as labels: if a node already exists, the corresponding counter is simply incremented by 1. Figure 3 shows the FP-tree generated after the FCAR mining process on the example dataset.

As in the CMAR algorithm, which is an associative classification model based on the classical version of FP-Growth [3], the rules are extracted from

Table 3: The fuzzy values associated with the highest membership degree and the corresponding fuzzy objects for each pattern in the example dataset.

ID	X_1	X_2	X_3	X_4	$\widetilde{\mathbf{x}}_\mathbf{n}$	**Class**
1	$A_{1,2}$	$A_{2,3}$	$A_{3,1}$	$A_{4,2}$	$(A_{2,3}\ ,\ A_{1,2}\ ,\ A_{4,2}\ ,\ A_{3,1})$	C_1
2	$A_{1,3}$	$A_{2,1}$	$A_{3,1}$	$A_{4,4}$	$(A_{1,3}\ ,\ A_{3,1})$	C_3
3	$A_{1,1}$	$A_{2,3}$	$A_{3,5}$	$A_{4,3}$	$(A_{2,3}\ ,\ A_{4,3}\ ,\ A_{3,5})$	C_1
4	$A_{1,4}$	$A_{2,4}$	$A_{3,2}$	$A_{4,5}$	$(A_{2,4})$	C_2
5	$A_{1,2}$	$A_{2,4}$	$A_{3,5}$	$A_{4,3}$	$(A_{1,2}\ ,\ A_{4,3}\ ,\ A_{3,5}\ ,\ A_{2,4})$	C_2
6	$A_{1,3}$	$A_{2,5}$	$A_{3,4}$	$A_{4,2}$	$(A_{1,3}\ ,\ A_{3,4}\ ,\ A_{4,2})$	C_3
7	$A_{1,5}$	$A_{2,3}$	$A_{3,4}$	$A_{4,1}$	$(A_{2,3}\ ,\ A_{3,4})$	C_1

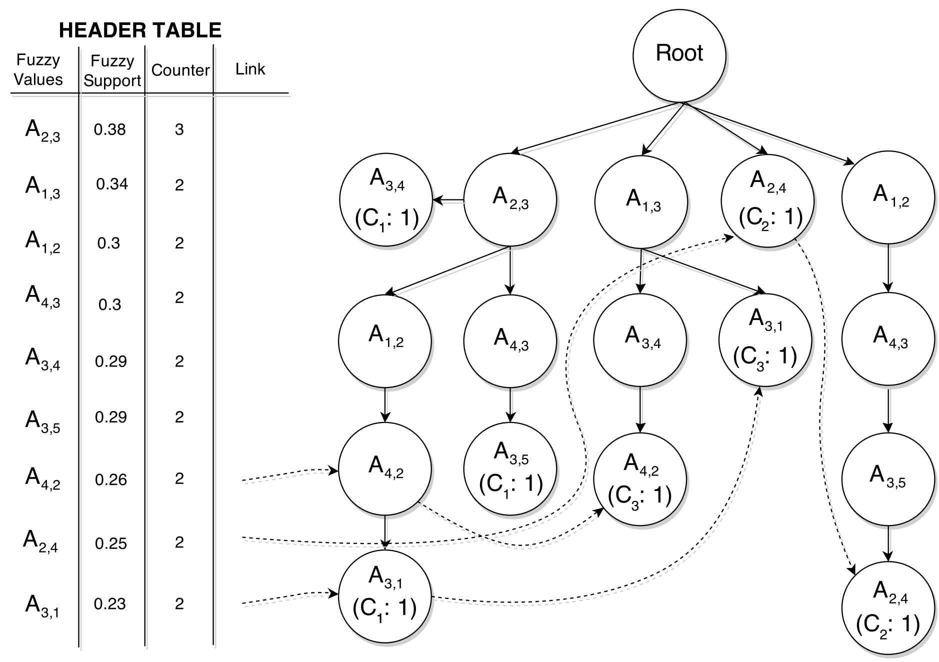


Figure 3: The FP-tree generated by using the example dataset.

the FP-tree by using $minSupp$ and $minConf$. In particular, the association rules which are not characterized by a support and a confidence higher than $minSupp$ and $minConf$, respectively, are first generated and therefore eliminated. We recall that support and confidence are here computed by only considering the frequency of the fuzzy values. Further, similarly to the pruning process discussed in [3] for CMAR, we test whether the antecedent of each rule is positively correlated with the consequent class by performing the χ^2 test. Only the rules with a χ^2 value higher than $min\chi^2$ are maintained. Figure 4 shows the pseudo code of FCAR mining process.

Since we consider only the fuzzy objects for generating the FP-tree, only a high quality subset of rules is stored in the $FCAR_{list}$. Indeed, each rule $FCAR_m$ mined from the FP-tree represents the rule with the highest matching degree for the specific object $\mathbf{x}_n \in T$. Other rules that could be mined from $\mathbf{x}_n$ would have had a lower matching degree and probably would have been pruned. At the end of the second scan, list $FCAR_{list}$ still contains a large amount of FCARs that are pruned in the subsequent phase.

3.3. Pruning

Rule pruning aims to discard slightly relevant rules so as to speed up the classification process. Pruning has to be applied carefully since an excessive elimination of rules may delete useful knowledge. Several approaches to rule pruning have been proposed in the last years, such as lazy pruning [40], database coverage [18] and pessimistic error estimation [1].

We perform three different types of pruning. In the first type, a rule $FCAR_m$ is pruned if its fuzzy support and confidence are not higher than *minFuzzySupp* and *minFuzzyConf*, respectively. These thresholds correspond

Data: $f_{list}, minSupp, minConf, min\chi^2$

Result: a list $FCAR_{list}$ of FCARs

$FP_{tree} \leftarrow$ clear;

$FCAR_{list} \leftarrow$ clear;

forall *data objects* $(\mathbf{x}_n, y_n)$ *in the training set* **do**

    forall *input feature values* $x_{n,f}$ **do**

        $\widetilde{x}_{n,f} \leftarrow$ fuzzy value with maximum membership grade for $x_{n,f}$;

        if *($\widetilde{x}_{n,f}$ is in the f_{list}, that is, $\widetilde{x}_{n,f}$ is frequent)* **then**

            $\widetilde{x}_n \leftarrow \widetilde{x}_{n,f}$;

        end

    end

    sort $\widetilde{\mathbf{x}}_n$ according to the f_{list};

    insert $(\widetilde{\mathbf{x}}_n, y_n)$ into the FP_{tree};

end

$FCAR_{list} \leftarrow$ mineAllRules$(FP_{tree}, minSupp, minConf, min\chi^2)$;

Figure 4: Pseudo-code of the FCAR mining process based on FP-Growth.

to *minSupp* and *minConf* adapted to the number of conditions and number of instances of each class, respectively, so as to take into account the effect of the t-norm used as conjunction operator and the imbalance of datasets.

Indeed, since we use the product as t-norm for implementing the conjunction operator, rules with a higher number of conditions in the antecedent will be characterized by a lower support value than rules with a lower number of conditions. Actually, this result is mainly due in general to the effect of the t-norm rather than to the activation of each condition. Indeed, each condition could be activated with a high matching degree, but for the behavior of the t-norm the matching degree of the rule, when the number of conditions is high, would result to be quite low. With the aim of reducing this effect and therefore avoiding to penalize more specific rules, we adapt the threshold *minFuzzySupp* on the fuzzy support to the rule length (RL), that is, the number of conditions in the antecedent of the rules, as follows:

$$minFuzzySupp_g = minSupp \cdot 0.5^{g-1} \tag{17}$$

where *minSupp* is the minimum support determined by the expert and $g \in [1..F]$ is the rule length.

For instance, for a rule with one condition in the antecedent, we have $minFuzzySupp_1 = minSupp$. For a rule with two conditions in the antecedent, we have $minFuzzySupp_2 = minSupp \cdot 0.5$, and so on.

To take into consideration also the imbalanced datasets, the confidence threshold is adapted by considering the imbalance ratio between each class and the majority class as follows:

$$minFuzzyConf_{C_j} = minConf \cdot \frac{N_{C_j}}{N_{MajorityClass}} \tag{18}$$

where $N_{MajorityClass}$ is the number of occurrences of the majority class label in the data set, N_{C_j} is the number of occurrences of the consequent class C_j in the training set and $minConf$ is the minimum confidence fixed by the expert. Formula 18 allows decreasing the $minConf$ threshold proportionally to the imbalance ratio between the class of the rule and the majority class. Thus, rules, which have a minority class in the consequent, are not pruned only because the number of instances of that class is very low in the training set.

Figure 5 shows the pseudo-code of the first type of pruning, which involves the third scan of the dataset.

With the adjustments performed by formulas 17 and 18, we are able to mine a higher number of fuzzy rules than the other approaches described in [31] and [39], without losing the advantages of the FP-Growth method, even if a third scan in the dataset is necessary.

In the second type of pruning, redundant rules are removed. First, the rules are sorted according to the fuzzy support, confidence and RL. In particular, rule $FCAR_l$ has higher rank than rule $FCAR_m$, if and only if:

1. $fuzzyConf(FCAR_l) > fuzzyConf(FCAR_m)$
2. $fuzzyConf(FCAR_l) = fuzzyConf(FCAR_m)$ **AND** $fuzzySupp(FCAR_l) > fuzzySupp(FCAR_m)$;
3. $fuzzyConf(FCAR_l) = fuzzyConf(FCAR_m)$ **AND** $fuzzySupp(FCAR_l) = fuzzySupp(FCAR_m)$ **AND** $RL(FCAR_l) < RL(FCAR_m)$.

Data: $FCAR_{list}, minSupp, minConf$

Result: Pruned $FCAR_{list}$

forall *data objects* $(\mathbf{x}_n, y_n)$ *in the training set* **do**

  forall *fuzzy rules* $FCAR_m$ *in* $FCAR_{list}$ **do**
    $w_m(\mathbf{x}_n) \leftarrow$ calculate matching degree of $\mathbf{x}_n$ with $FCAR_m$;

  end

end

forall *possible rule lengths g in [1..F]* **do**
  compute $minFuzzySupp_g$ by using formula 17;

end

forall *classes* C_j *in* $[C_1, ..., C_K]$ **do**
  compute $minFuzzyConf_{C_j}$ by using formula 18;

end

forall *fuzzy rules* $FCAR_m$ *in* $FCAR_{list}$ **do**
  $g_m \leftarrow$ rule length of $FCAR_m$

  $C_{j_m} \leftarrow$ class of $FCAR_m$;

  if $fuzzySupp(FCAR_m) < minFuzzySupp_{g_m}$ **OR**

  $fuzzyConf(FCAR_m) < minFuzzyConf_{C_{j_m}}$ **then**
    $FCAR_{list} \leftarrow$ remove $FCAR_m$ from $FCAR_{list}$;

  end

end

Figure 5: Pseudo-code of the first type of pruning.

A fuzzy rule $FCAR_m$ is pruned if and only if there exists a rule $FCAR_l$ with higher rank and more general than $FCAR_m$. A rule $FCAR_l : FAnt_l \rightarrow C_{j_l}$ is more *general* than a rule $FCAR_m : FAnt_m \rightarrow C_{j_m}$, if and only if, $FAnt_m \subseteq FAnt_l$. Our experimental results show that this second step can reduce significantly the number of FCARs in the $FCAR_{list}$.

In the third type of pruning, the *training set coverage* is exploited: only the fuzzy rules that are activated by at least one data object in the training set are retained. Each data object in the training set is associated with a counter initialized to 0. For each object, a scan over the sorted $FCAR_{list}$ is performed to find all the rules that match the object: we consider only those rules $FCAR_m$ with matching degree higher than the *fuzzy matching degree threshold* $\overline{w}_m = 0.5^{g_m - 1}$, where g_m is the rule length of $FCAR_m$. This threshold allows us to take into account only the most significant rules for a specific data object, without penalizing rules with high rule length. If $FCAR_m$ classifies correctly at least one data object, then it is inserted into the RB. Further, the counters associated with the objects, which activate $FCAR_m$, are incremented by 1. Whenever the counter of an object becomes larger than the *coverage threshold* δ, the data object is removed from the training set and no longer considered for subsequent rules. Since rules are sorted in descending ranks, it is very likely that these subsequent rules would have a very limited relevance for the object. The procedure ends when no more objects are in the training set or all the rules have been analyzed. Figure 6 shows the pseudo-code of the third type of pruning.

We would like to point out that the overall rule base is obtained by performing only 4 scans of the overall training set.

Data: sorted pruned $FCAR_{list}$, coverage threshold δ

Result: final RB

$RB \leftarrow$ clear();

forall *data objects* $(\mathbf{x}_n, y_n)$ *in the training set* **do**
    count $\leftarrow 0$;

    forall *rules* $FCAR_m$ *in* $FCAR_{list}$ **do**
        $w_m(x_n) \leftarrow$ calculate the matching degree;

        if $w_m(x_n) >= \overline{w}_m$ **then**
            count $\leftarrow$ count + 1;

            if $FCAR_m$ *correctly classifies* $\mathbf{x}_n$ **AND**

            $FCAR_m$ *is not in* RB **then**
                add $FCAR_m$ to RB;

            end

        end

        if *count* $> \delta$ **then**
            break;

        end

    end

end

Figure 6: Pseudo-code of the third type of pruning.

3.4. Classification

The set of rules survived after the pruning are used to classify unlabeled patterns. In this paper, we adopt the weighted vote [41] as *reasoning method*: an input pattern is classified into the class corresponding to the the maximum total strength of vote, calculated by using formula (5). Given an input pattern $\widehat{\mathbf{x}} = [\hat{x}_1 \ldots, \hat{x}_F]$, each fuzzy rule in the RB gives a vote for its consequent class. If $\widehat{\mathbf{x}}$ activates no rule, then $\widehat{\mathbf{x}}$ is classified as unknown.

Since we use the product t-norm as conjunction operator, rules with a higher number of conditions in the antecedent have generally a lower matching degree than rules with a lower number of conditions in the antecedent. Hence, more general rules are more influential than specific rules on the prediction phase. To re-balance the influence, we normalize formula (5) as follows:

$$V_{C_k}(\widehat{\mathbf{x}}) = \sum_{FCAR_m \in RB; C_{jm} = C_k} w_m(\widehat{\mathbf{x}}) \cdot 2^{g_m} \cdot CF_m \tag{19}$$

where $w_m(\widehat{\mathbf{x}})$ is the matching degree of $FCAR_m$ for the input $\widehat{\mathbf{x}}$, F is the number of features in the data set, g_m is the RL of $FCAR_m$ and CF_m is the certainty factor. For example, let us assume that three rules have, respectively, one condition, two conditions and three conditions, respectively. Let us suppose that each object is described by 3 features and each condition is activated by the unlabeled pattern $\widehat{\mathbf{x}}$ with membership degree equal to 0.5. The matching degrees for the three rules would be 0.5, 0.25 and 0.125, respectively. After re-balancing with formula 19, all the votes are equal to 1.

This normalization allows considering also the vote of rules with a high number of conditions. Indeed, the vote of these rules is strongly penalized by the number of conditions, even if all these conditions are activated with a high membership degree. In our experiments, we verified that this approach is more effective than for instance maximum matching.

4. Experimental Study

We tested our method on seventeen classification datasets extracted from the KEEL repository (available at http://sci2s.ugr.es/keel/datasets.php). As shown in Table 4, the datasets are characterized by different numbers of input variables (from 4 to 16), input/output instances (from 106 to 19020) and classes (from 2 to 11). For the datasets CLE and WIS, we removed the instances with missing values. The number of instances in the table refers to the datasets after the removing process.

We compare the results obtained by AC-FFP with the ones achieved by two different classification models, namely CMAR [3] and FARC-HD [25]. We chose these two algorithms because CMAR exploits as AC-FFP the FP-Growth algorithm for generating the association rules, and FARC-HD is, to the best of our knowledge, one of the most recent and effective fuzzy rule-based associative classifiers proposed in the literature.

Similar to our approach, CMAR first adopts the multi-interval discretization method presented in [35] to split the input domains into bins. Then, it builds a class distribution-associated frequent pattern tree to efficiently mine CARs. Finally, CARs are pruned based on the analysis of the: i) confidence, ii) correlation, iii) rule redundancy and iv) database coverage. The classification is performed based on a weighted χ^2 analysis enforced on multiple association rules. We implemented a JAVA version of CMAR following the description provided in [3].

Table 4: Datasets used in the experiments (sorted for increasing numbers of input variables).

Dataset	**# Instances**	**# Variables**	**# Classes**
Iris (IRI)	150	4	3
Phoneme (PHO)	5404	5	2
Newthyroid (NEW)	215	5	3
Monk-2 (MON)	432	6	2
Appendicitis (APP)	106	7	2
Ecoli (ECO)	336	7	8
Pima (PIM)	768	8	2
Yeast (YEA)	1484	8	10
Glass(GLA)	214	9	6
Wisconsin (WIS)	683	9	2
Page-Blocks (PAG)	5472	10	5
Magic (MAG)	19020	10	2
Heart (HEA)	270	13	2
Cleveland (CLE)	297	13	5
Wine (WIN)	178	13	3
Vowel (VOW)	990	13	11
Pen-Based (PEN)	10992	16	10

FARC-HD has been described in Section 1. In [25], the authors have shown that FARC-HD is very efficient since it outperforms a large number of classical classification algorithms, both based and not based on fuzzy rules and/or on CARs [25]. In the experiments, we have used the implementations of FARC-HD available in the KEEL package [42] .

Table 5 shows the parameters used for each algorithm in the experiments. The parameters have been chosen according to the guidelines provided by the authors in the papers in which each algorithm has been introduced. For FARC-HD, the description of the specific parameters can be found in [25]. Further, for each dataset and for each algorithm, we performed a ten-fold cross-validation by using the same folds for all the datasets.

Table 5: Values of the parameters for each algorithm used in the experiments.

Method	Parameters
CMAR	$MinSupp = 0.01, MinConf = 0.5, \delta = 4, min\chi^2 = 20\%$
FARC-HD	$MinSupp = 0.05, MaxConf = 0.80, Depth_{max} = 3, k_t = 2,$ $Pop = 50, Evaluations = 15.000, BITSGENE = 30, \delta = 2$
AC-FFP	$MinSupp = 0.01, MinConf = 0.5, \delta = 4, min\chi^2 = 20\%$

Table 6 shows, for each dataset and for each algorithm, the average values of the accuracy, both on the training (Acc_{Tr}) and test sets (Acc_{Ts}), associated with the ACs generated by the three algorithms. For each dataset, the values of the highest accuracies are shown in bold.

From Table 6, we can observe that, in most of the datasets, the AC-FFP algorithm generates classifiers more accurate than the ones generated by CMAR. In particular, in 12 out of 17 datasets, AC-FPP achieves the highest accuracies on the test set. As regards FARC-HD, we observe that AC-FFP and FARC-HD achieve similar average accuracies on the test set. Further, FARC-HD suffers from overtraining more than the other two approaches.

Table 6: Average results obtained by CMAR, FARC-HD and AC-FFP.

	CMAR		**FARC-HD**		**AC-FFP**	
Dataset	Acc_{Tr}	Acc_{Ts}	Acc_{Tr}	Acc_{Ts}	Acc_{Tr}	Acc_{Ts}
IRI	96.00	93.33	**98.59**	96.00	96.15	**98.00**
PHO	79.10	78.70	**83.50**	**82.14**	81.54	81.10
NEW	96.38	93.10	**98.98**	93.95	97.73	**95.87**
MON	77.78	77.56	**99.92**	**99.77**	97.22	97.27
APP	90.87	**86.00**	**93.82**	84.18	91.51	85.09
ECO	83.30	76.83	**92.33**	82.19	89.35	**83.39**
PIM	78.69	74.87	**82.90**	**75.66**	79.25	74.87
YEA	56.55	54.32	**63.81**	**58.50**	57.99	55.60
GLA	80.42	69.44	81.10	70.24	**83.59**	**74.18**
WIS	97.74	**96.80**	98.70	96.52	**98.85**	96.06
PAG	93.79	93.68	**95.62**	**95.01**	94.15	93.88
MAG	79.39	78.94	**85.36**	**84.51**	73.59	73.40
HEA	90.08	84.07	93.91	**84.44**	**94.77**	81.85
CLE	54.40	53.88	**88.18**	55.24	80.43	**56.91**
WIN	99.94	96.05	99.94	94.35	**100**	**97.12**
VOW	74.14	61.41	80.48	71.82	**98.95**	**91.52**
PEN	78.60	77.78	**97.04**	**96.04**	85.66	83.48
Mean	82.77	79.22	90.25	83.56	88.28	83.51

In order to verify if there exist statistical differences among the values of accuracy on the test set associated with the classifiers generated by the different algorithms, we perform a statistical analysis. As suggested in [43], we apply non-parametric statistical tests combining all the datasets: for each approach we generate a distribution consisting of the mean values of accuracy calculated on the test set. We compare both CMAR and FARC-HD with AC-FFP by using a pairwise comparison, namely the Wilcoxon signed-rank test [44], which detects signicant differences between two distributions. In all the tests, we use $\alpha = 0.05$ as *level of significance*.

Table 7 shows the results of the application of the Wilcoxon signed-rank test between AC-FFP and CMAR and between AC-FFP and FARC-HD. As regards the first comparison, since the *p-value* is lower than the significance level α, the null hypothesis of equivalence can be rejected. In conclusion, we can state that AC-FFP statistically outperforms CMAR in terms of classification accuracy on the test set.

Table 7: Results of the Wilcoxon signed-rank test with a significance level $\alpha = 0.05$.

Comparision	R^+	R^-	P*-value*	*Hypothesis*
AC-FFP vs CMAR	113	23	0.01825	**Rejected**
AC-FFP vs FARC-HD	76	77	1	**Not Rejected**

As regards the Wilcoxon signed-rank test between AC-FFP and FARC-HD, since the *p-value* is higher than the significance level α, the null hypothesis of equivalence is not rejected. Hence, AC-FFP and FARC-HD result to be statistically equivalent in terms of classification rate computed on the test set. In FARC-HD, however, the first two steps, which generate the initial set of candidate rules, are based on a fuzzy version of the Apriori algorithm. This algorithm suffers from the curse of dimensionality: the higher the number of input variables, the more difficult the generation of the set of candidate rules. In addition, the third step of FARC-HD requires the execution of a genetic algorithm for selecting a reduced set of rules. As stated in [45], the size of the search space grows with the increase of the number of input variables, thus leading to a slow and possibly difficult convergence of the genetic algorithm. Further, the computational cost of the fitness evaluation increases linearly with the increase of the number of instances in the dataset, thus obliging to limit the number of evaluations especially when the dataset is large. On the other hand, AC-FFP needs only four scans of the dataset.

5. Conclusions

In this paper, we have proposed a novel model of classification based on fuzzy association rules derived by using a fuzzy version of the well-known FP-Growth algorithm.

We have first discretized the input variables and generated strong fuzzy partitions by using a well-known discretization algorithm. Then, we have applied the FP-Growth for generating the fuzzy association rules. Finally, we have applied three different pruning types to select a subset of high quality fuzzy association rules for classification.

The proposed approach has been tested on seventeen classification benchmarks. We have compared the results achieved by the novel fuzzy model with the ones achieved by a well-know non-fuzzy associative classification model, namely the CMAR algorithm, and by a state-of-the-art algorithm for generating fuzzy rule-based association classifiers, namely FARC-HD.

By performing non parametric statistical tests, we have highlighted that the proposed approach outperforms the CMAR algorithm, in terms of classification accuracy on the test set, and achieves accuracies similar to FARC-HD.

References

[1] I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition, Morgan Kaufmann Series in Data Management Sys, Morgan Kaufmann, 2011.

[2] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, SIGMOD Rec. 22 (2) (1993) 207–216.

[3] W. Li, J. Han, J. Pei, CMAR: accurate and efficient classification based on multiple class-association rules, in: Proceedings of IEEE International Conference on Data Mining 2001, 2001, pp. 369–376.

[4] X. Yin, J. Han, CPAR: Classification based on predictive association rules, in: Proceedings of the third SIAM international conference on data mining, 2003, pp. 331–335.

[5] Y.-L. Chen, L. T.-H. Hung, Using decision trees to summarize associative classification rules, Expert Systems with Applications 36 (2, Part 1) (2009) 2338 – 2351.

[6] E. Baralis, P. Garza, I-prune: Item selection for associative classification, International Journal of Intelligent Systems 27 (3) (2012) 279–299.

[7] N. Abdelhamid, A. Ayesh, F. Thabtah, S. Ahmadi, W. Hadi, MAC: A multiclass associative classification algorithm, Journal of Information & Knowledge Management 11 (02).

[8] M. I. A. Ajlouni, W. Hadi, J. Alwedyan, Detecting phishing websites

using associative classification, European Journal of Business and Management 5 (15) (2013) 36–40.

[9] G. Costa, R. Ortale, E. Ritacco, X-class: Associative classification of XML documents by structure, ACM Transactions on Information Systems 31 (1) (2013) 3.

[10] Y. Yoon, G. G. Lee, Two scalable algorithms for associative text classification, Information Processing and Management 49 (2) (2013) 484 – 496.

[11] S. Dua, H. Singh, H. Thompson, Associative classification of mammograms using weighted rules, Expert Systems with Applications 36 (5) (2009) 9250 – 9259.

[12] M. Jabbar, B. Deekshatulu, P. Chandra, Knowledge discovery using associative classification for heart disease prediction, in: Intelligent Informatics, Springer, 2013, pp. 29–39.

[13] W.-C. Chen, C.-C. Hsu, Y.-C. Chu, Increasing the effectiveness of associative classification in terms of class imbalance by using a novel pruning algorithm, Expert Systems with Applications 39 (17) (2012) 12841 – 12850.

[14] W.-C. Chen, C.-C. Hsu, J.-N. Hsu, Adjusting and generalizing CBA algorithm to handling class imbalance, Expert Systems with Applications 39 (5) (2012) 5907 – 5919.

[15] A. Y. Rodrguez-Gonzlez, J. F. Martnez-Trinidad, J. A. Carrasco-Ochoa, J. Ruiz-Shulcloper, Mining frequent patterns and association rules using

similarities, Expert Systems with Applications 40 (17) (2013) 6823 – 6836.

[16] B. Vo, T.-P. Hong, B. Le, A lattice-based approach for mining most generalization association rules, Knowledge-Based Systems 45 (0) (2013) 20 – 30.

[17] F. P. Pach, A. Gyenesei, J. Abonyi, Compact fuzzy association rule-based classifier, Expert Systems with Applications 34 (4) (2008) 2406 – 2416.

[18] B. Liu, W. Hsu, Y. Ma, Integrating classification and association rule mining, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 1998, pp. 80–86.

[19] D. Meretakis, B. Wuthrich, Extending naive bayes classifiers using long itemsets, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99, ACM, New York, NY, USA, 1999, pp. 165–174.

[20] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, pp. 487–499.

[21] J. P. J. Han, M. Kamber, Data Mining: Concepts and Techniques, 3rd Edition, Data Management Systems, Morgan Kaufmann, 2012.

[22] J. Han, J. Pei, Y. Yin, R. Mao, Mining frequent patterns without candidate generation: A frequent-pattern tree approach, Data Min. Knowl. Discov. 8 (1) (2004) 53–87.

[23] B. Liu, Y. Ma, C.-K. Wong, Classification using association rules: weaknesses and enhancements, Data mining for scientific applications (2001) 1–11.

[24] Z. Chen, G. Chen, Building an associative classifier based on fuzzy association rules, International Journal of Computational Intelligence Systems 1 (3) (2008) 262–273.

[25] J. Alcala-Fdez, R. Alcala, F. Herrera, A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning, IEEE Transactions on Fuzzy Systems 19 (5) (2011) 857–872.

[26] J. P. Lucas, A. Laurent, M. N. Moreno, M. Teisseire, A fuzzy associative classification approach for recommender systems, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20 (04) (2012) 579–617.

[27] B. Sowan, K. Dahal, M. Hossain, L. Zhang, L. Spencer, Fuzzy association rule mining approaches for enhancing prediction performance, Expert Systems with Applications 40 (17) (2013) 6928 – 6937.

[28] J. C. Bezdek, R. Ehrlich, W. Full, Fcm: The fuzzy C-means clustering algorithm, Computers & Geosciences 10 (2) (1984) 191–203.

[29] B. Liu, W. Hsu, Y. Ma, Mining association rules with multiple minimum supports, in: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 1999, pp. 337–341.

[30] C.-W. Lin, T.-P. Hong, W.-H. Lu, Linguistic data mining with fuzzy FP-trees, Expert Systems with Applications 37 (6) (2010) 4560 – 4567.

[31] C.-T. P. Chien-Hua Wang, Wei-Hsuan Lee, Applying fuzzy FP-Growth to mine fuzzy association rules, Engineering and Technology, World Academy of Science, 2010, pp. 788–794.

[32] H. Ishibuchi, T. Nakashima, M. Nii, Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2004.

[33] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 13 (4) (2005) 428–435.

[34] H. Ishibuchi, T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 9 (4) (2001) 506–515.

[35] U. M. Fayyad, K. B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: Proceedings of IJCAI, 1993, pp. 1022–1029.

[36] E. J. Clarke, B. A. Barton, Entropy and mdl discretization of continuous variables for bayesian belief networks, International Journal of Intelligent Systems 15 (1) (2000) 61–92.

[37] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann, 1995, pp. 194–202.

[38] S. Kotsiantis, D. Kanellopoulos, Discretization techniques: A recent survey, GESTS International Transactions on Computer Science and Engineering 32 (1) (2006) 47–58.

[39] D. Mishra, S. Mishra, S. Satapathy, S. Patnaik, Genetic algorithm based fuzzy frequent pattern mining from gene expression data, in: S. Patnaik, Y.-M. Yang (Eds.), Soft Computing Techniques in Vision Science, Vol. 395 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2012, pp. 1–14.

[40] E. Baralis, P. Garza, A lazy approach to pruning classification rules, in: Proceedings of 2002 IEEE International Conference on Data Mining, 2002, pp. 35–42.

[41] O. Cordón, M. J. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning 20 (1) (1999) 21–45.

[42] J. Alcalá-Fdez, L. Sánchez, S. García, M. Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, F. Herrera,

Keel: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (3) (2009) 307–318.

[43] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3 – 18.

[44] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) pp. 80–83.

[45] M. Antonelli, P. Ducange, F. Marcelloni, An efficient multi-objective evolutionary fuzzy system for regression problems, International Journal of Approximate Reasoning 54 (9) (2013) 1434 – 1451.

***Highlights (for review)**

Highlights

- We propose a novel fuzzy associative classification approach
- We exploit a fuzzy version of the FP-Growth algorithm
- We perform an experimental analysis on 17 classification datasets
- We compare our approach with two well-known associative classifiers