European Journal of Operational Research 256 (2017) 924–934

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Computational Intelligence & Inform. Management

An integer programming approach for fuzzy rule-based classification systems

Shahab Derhami, Alice E. Smith*

Department of Industrial and Systems Engineering, Auburn University, Auburn, AL 36849, USA

ARTICLE INFO

Article history:
Received 13 September 2015
Accepted 28 June 2016
Available online 1 July 2016

Keywords:
Fuzzy sets
Integer programming
Classification
Rule learning
Data mining

ABSTRACT

Fuzzy rule-based classification systems (FRBCSs) have been successfully employed as a data mining technique where the goal is to discover the hidden knowledge in a data set in the form of interpretable rules and develop an accurate classification model. In this paper, we propose an exact approach to learn fuzzy rules from a data set for a FRBCS. First, we propose a mixed integer programming model that extracts optimal fuzzy rules from a data set. The model's embedded feature selection allows absence of insignificant features in a fuzzy rule in order to enhance its accuracy and coverage. In order to build a comprehensive Rule Base (RB), we use this model in an iterative procedure that finds multiple rules by converting the obtained optimal solutions into a set of taboo constraints that prevents the model from re-finding the previously obtained rules. Furthermore, it changes the search direction by temporarily removing the correctly predicted patterns from the training set aiming to find the optimal rules that predict uncovered patterns in the training set. This procedure ensures that most of the patterns in the training set are covered by the RB. Next, another mixed integer programming model is developed to maximize predictive accuracy of the classifier by pruning the RB and removing redundant rules. The predictive accuracy of the proposed model is tested on the benchmark data sets and compared with the state-of-the-art algorithms from the literature by non-parametric statistical tests.

1. Introduction

Fuzzy rule-based classification systems (FRBCSs) have two advantages as a data mining technique. First, their capability in encompassing nonlinear and complex relations in a data set enables them to establish precise classification models, especially when a system deals with ambiguous and imprecise data (Al-Ebbini, Oztekin, & Chen, 2016; de Andrés, Landajo, & Lorca, 2005; Nakandala, Samaranayake, & Lau, 2013; Yuan, Feldhamer, Gafni, Fyfe, & Ludwin, 2002). Second, the rule learning procedure used to extract the fuzzy rules for a FRBCS can be used to discover the latent knowledge in large data sets in the form of a set of interpretable fuzzy rules compatible with human logic. The main advantage of FRBCSs is that their interpretable linguistic rules are easily understood by the users. In a typical fuzzy rule-based system (FRBS) used in control problems or other similar areas, fuzzy rules are provided to the FRBS by an expert person (Feng, 2006). However, when a FRBCS is used as a data mining tool, these rules are the hidden knowledge in a data set and the main goal is to extract them and establish a classification model (Amo, Montero, Biging, & Cutello, 2004; Baykasoğlu & Özbakir, 2007; Martens, Baesens, Gestel, & Vanthienen, 2007). Various heuristic approaches have been proposed to extract fuzzy rules from a data set such as neural networks (Bekiros, 2010; Hu, Hu, Chen, & Tzeng, 2004; Kulluk, Özbakır, & Baykasoğlu, 2013; Li & Wang, 2004), support vector machines (Castro, Flores-Hidalgo, Mantas, & Puche, 2007; Chiang & Hao, 2004; Ren, Liu, & Cao, 2011), genetic algorithms (GA) (Alcalá-Fdez, Alcalá, & Herrera, 2011; Hoffmann, Baesens, Mues, Gestel, & Vanthienen, 2007a; Hong, Lee, & Wu, 2014; Ishibuchi, Mihara, & Nojima, 2013), and other heuristic approaches (Belacel, Raval, & Punnen, 2007; Mansoori, 2011; Ravi & Zimmermann, 2000; Wang, Liu, Pedrycz, Zhu, & Hu, 2012). Several studies developed multi-objective approaches to take the accuracy-interpretability trade-off into account while learning fuzzy rules for FRBCSs (Fazzolari, Giglio, Alcalá, Marcelloni, & Herrera, 2013; Ishibuchi & Nojima, 2007; Khalili-Damghani, Sadi-Nezhad, Lotfi, & Tavana, 2013). A complete review of the evolutionary algorithms developed to learn fuzzy rules for FRBCSs can be found in Cordón (2011).

Despite various heuristic approaches that have been proposed for learning fuzzy rules, no exact optimization approach, to the best of our knowledge, has been developed for this problem. In this paper, we propose an Integer Programming approach to

* Corresponding author.
E-mail address: smithae@auburn.edu (A.E. Smith).

http://dx.doi.org/10.1016/j.ejor.2016.06.065
0377-2217/

Extract and Prune fuzzy rules (IPEP). The proposed method composes of two procedures. First, a Mixed Integer Programming model is proposed to Extract Fuzzy rules (MIPEF). It is capable of learning one optimal fuzzy rule; therefore, it iterates to collect more fuzzy rules and generate the Rule Base (RB). Once the RB is built, another Mixed Integer Programming model is employed to Prune the RB (MIPP) and select the set of rules that maximizes total predictive accuracy. A preliminary version of MIPEF appeared in Derhami and Smith (2014). The improvements herein include adding additional constraints to enhance the accuracy and reliability of the model and taking the interpretability of the rules into account. The other difference between this paper and the model proposed in Derhami and Smith (2014) is that it supports learning multiple fuzzy rules and considers the interactions and conflicts among the rules in rule selection.

Different types of fuzzy linguistic rules are employed in FRBCSs which are mainly different in the consequent or antecedent structure (Cordón, 2011). In terms of the antecedent structure, fuzzy rules are classified into Disjunctive Normal Form (DNF) and non-DNF (canonical) fuzzy rules. The antecedent of a DNF fuzzy rule allows the disjunction of linguistic labels for each variable (i.e., linguistic labels of a feature joined by a disjunctive operator, 'OR' operator) while the antecedent of the latter one allows at most one linguistic label per feature (variable). Various studies in the literature have proposed different approaches to learn DNF (Berlanga, Rivera, del Jesus, & Herrera, 2010; Casillas, Martínez, & Benítez, 2009) and non-DNF (Derhami & Smith, 2014; Ravi, Reddy, & Zimmermann, 2000; Tsakonas, 2006) fuzzy rules for FRBCSs. In this paper, we consider non-DNF fuzzy rules with one class label and a certainty degree.

Fuzzy sets and linguistic labels are defined by two different approaches in FRBCSs, pre-specified membership functions (Derhami & Smith, 2014; Mansoori, Zolghadri, & Katebi, 2008) and embedded feature selection (Hoffmann, Baesens, Mues, Gestel, & Vanthienen, 2007b; del Jesus, Hoffmann, Navascués, & Sánchez, 2004; Li & Wang, 2009; Sanz, Fernández, Bustince, & Herrera, 2011). In the former approach, the fuzzy sets (linguistic labels) and their membership functions are determined before the algorithm starts learning the fuzzy rules. In this method, the definition of membership functions for fuzzy sets are identical for all fuzzy rules and the search selects the best set of linguistic labels for the antecedents of the fuzzy rules aiming to maximize accuracy, coverage or interpretability. In the latter approach, fuzzy rules have their own definition of membership functions. In this approach, the search simultaneously optimizes both fuzzy membership functions and fuzzy rules. The main advantage of the former approach is that the rules are more interpretable than the ones obtained by the latter approach, and therefore they are more appropriate for the purpose of data mining. For this reason, our algorithm follows the first approach.

We used the most common membership function shape in the literature – symmetric triangular – with five linguistic labels. Fig. 1 presents this partitioning for a typical feature, where min_f and max_f represent the minimum and maximum values for feature f in the data set, respectively. However, the proposed model works with any type of membership function because the membership function is not part of the optimization process. Derhami and Smith (2014) experimented with different numbers of linguistic labels for symmetric triangular membership functions and showed that using five linguistic labels leads to slightly higher predictive accuracy. This partitioning works for all types of numerical and categorical features except nominal features as they do not have any intrinsic order.

This paper is organized as follows. First, we briefly introduce the FRBCSs and define new components used in the proposed model in Section 2. Next, we explain our proposed IPEP algorithm in Section 3. Then, the computational and comparative results are reported in Section 4. Finally, we conclude this paper in Section 5.

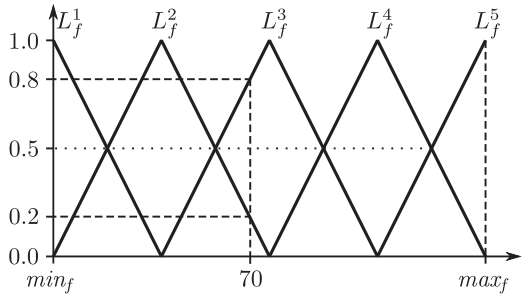

Fig. 1. Partitioning a feature into five linguistic labels with symmetric triangular membership function.

2. Fuzzy rule-based classification systems

A FRBCS is comprised of two parts, knowledge discovery and fuzzy reasoning. Fuzzy rules are extracted from a data set in the knowledge discovery phase and then used in the fuzzy reasoning inference to predict classifications of new patterns. In this section, we briefly introduce these two systems.

2.1. Knowledge discovery

Consider a classification problem consisting of n_p training patterns, n_f features and n_c class labels. Pattern i of the data set can be represented as an n-tuple $p_i = (p_{i1}, \ldots, p_{in_f}, c)$, where p_{ij} is the value of the jth feature ($j = 1, \ldots, n_f$) in the pattern i ($i = 1, \ldots, n_p$) and $c \in \{1, \ldots, n_c\}$ is the class label for this pattern. In this paper, we employ the following type of fuzzy rule:

$$\text{Rule } R_k: \text{ if } p_1 \text{ is } \hat{A}_1 \text{ and } \ldots \text{ and } p_{n_f} \text{ is } \hat{A}_{n_f} \text{ then class is } C_k \text{ with } CD_k \quad (1)$$

where R_k is the label of the kth rule, $p = (p_1, \ldots, p_{n_f})$ is an n_f-dimensional pattern vector, $\hat{A}_j \in \{L_j^1, \ldots, L_j^{l_j}\}$ is an antecedent fuzzy set of the jth feature that consists of l_j fuzzy linguistic labels, and $C_k \in \{1, \ldots, n_c\}$ and $CD_k \in [0, 1]$ are the class label and the certainty degree of the rule, respectively. We use five linguistic labels (i.e., $l_j = 5$) with symmetric triangular membership functions as shown in Fig. 1. The certainty degree, CD_k, indicates the classification confidence (accuracy) for rule k and is calculated as

$$CD_k = \frac{\sum_{i \in P_{C_k}} m_k(p_i)}{\sum_{i \in P} m_k(p_i)} \quad (2)$$

where P is the set of all patterns, P_{C_k} is the set of the patterns whose class labels are C_k, and $m_k(p_i)$ is the degree of compatibility between pattern p_i and fuzzy rule k. It is also termed as matching degree in the literature and calculated as

$$m_k(p_i) = \prod_{j=1}^{n_f} \mu_{\hat{A}_{kj}}(p_{ij}) \quad (3)$$

where $\mu_{\hat{A}_{kj}}(p_{ij})$ is the degree of membership for the value of the jth feature of pattern i to the corresponding fuzzy antecedent (condition) of rule k. The matching degree cannot be used in the MIPEF as it was defined by (3) because it would have been introduced to the model as

$$m_k(p_i) = \prod_{j=1}^{n_f} \mu_{\hat{A}_{kj}}(p_{ij}) x_j \quad (4)$$

where x_j is a binary decision variable, such that it is equal to 1 if feature j is in the antecedent of the optimal fuzzy rule. Expression (4) is a nonlinear term and using it in a mixed integer programming model makes it extremely hard to solve. To prevent this complexity, we approximate (4) by the following linear term

$$m_k(p_i) = \begin{cases} \frac{\sum_{j=1}^{n_f} \mu_{\hat{A}_{kj}}(p_{ij}) x_j}{n_{\hat{A}_k}} & \text{if } x_j > 0 \Rightarrow \mu_{\hat{A}_{kj}}(p_{ij}) > 0, \\ & \forall x_j, (j \in 1, \ldots, n_f) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $n_{\hat{A}_k}$ is the number of conditions (labels) in the rule k. One shall notice that since we consider just non-DNF fuzzy rules, the maximum value of $n_{\hat{A}_k}$ is n_f.

In addition to the certainty degree that determines the accuracy of a rule, there is another factor that evaluates coverage capability of a rule. It is termed *Support* and calculated as

$$S_k = \frac{\sum_{\substack{i=1 \\ p_i \in C_k}}^{n_p} m_k(p_i)}{N_{C_k}} \quad (6)$$

where N_{C_k} is the number of patterns for which the class label is the same as that of the rule k (i.e., C_k).

2.2. Fuzzy reasoning method

A fuzzy reasoning method (FRM) is an inference procedure to predict the class label for a pattern using a set of fuzzy rules. Various FRMs have been developed for FRBCSs (Cordón, del Jesus, & Herrera, 1999; Ishibuchi & Yamamoto, 2005; Mesiarová-Zemánková, 2014). The most commonly used FRMs are maximum matching and normalized sum FRMs. The maximum matching FRM classifies a pattern using the class label of the rule that achieves the highest matching degree with that pattern. In this approach, the information provided by the other rules is discarded. The normalized sum FRM aggregates the matching degrees between a pattern and all rules that have the same class label, obtains the normalized sum for each class label, and classifies the pattern to the class label that obtains the highest normalized sum. This procedure aims to utilize all information provided by the rules.

The proposed rule pruning model (MIPP) applies the same concept to classify patterns and distinguish redundant rules. However, it does not normalize the aggregated matching degrees in order to avoid having nonlinear constraints in the model. Instead, it computes the weighted sum of matching degrees between a pattern and all rules that have the same class label using the rule accuracies as the weights. The pattern is then classified to the class label with the highest weighted sum. This approach is also used for fuzzy reasoning. The procedure works as follows. First, the degree of compatibility between a pattern and all class labels is calculated by

$$m'_{ic} = \sum_{\substack{r \in R \\ C_r = c}} A_r m_r(p_i) \quad (7)$$

where m'_{ic} is the degree of compatibility between pattern i and class label c, A_r is the accuracy of rule r and $m_r(p_i)$ is the degree of compatibility between pattern p_i and fuzzy rule r. Then, the pattern is classified to the class label with the highest compatibility degree. That is,

$$C_i = \arg\max_{c \in C} \{m'_{ic}\} \quad (8)$$

where C_i is the assigned class label to pattern i.

3. IPEP algorithm

IPEP is an exact algorithm that consists of two steps: learning fuzzy rules and constructing the RB, and then pruning the RB to select the most accurate rules. The learning process is carried out by a mixed integer programming model (MIPEF) that learns fuzzy rules one by one through an iterative procedure. Once the RB is built, the pruning process is performed by another mixed integer programming model (MIPP) and the best rules are selected. These two procedures are described in the following sections.

3.1. Learning fuzzy rules

MIPEF is a mixed integer programming model that learns an optimal fuzzy rule for a pre-determined certainty degree and class label. It allows the absence of insignificant features in the optimal fuzzy rule (feature selection). This means that not all the features have to be included in the antecedent of the fuzzy rule and only those that are beneficial to the accuracy or coverage of the rule are selected. Since MIPEF extracts only one optimal fuzzy rule at a time, it is run iteratively for all class labels in the data set to generate the RB. The procedure finds new optimal rules by converting the previously obtained rules into a set of taboo constraints that preclude the MIPEF from re-finding previously obtained solutions. Moreover, the search direction of the MIPEF is changed in favor of unclassified patterns by temporarily removing the correctly predicted patterns from the data set. This forces the model to learn fuzzy rules that predict uncovered patterns. The definition of the sets, parameters and decision variables used in MIPEF is described in the following.

P	set of patterns, $P = \{1, \ldots, n_p\}$
F	set of features, $F = \{1, \ldots, n_f\}$
C	set of classes, $C = \{1, \ldots, n_c\}$
L	set of linguistic labels, $L = \{1, \ldots, 5\}$
R	set of fuzzy rules, $R = \{1, \ldots, n_r\}$
n_p	number of patterns in the training set
n_f	number of features in the data set
n_c	number of class labels in the data set
N_{c^*}	number of patterns for which the class label is the same as the target class label, $(c^* \in C)$
N_r	number of rules that have to be learned before the correctly predicted patterns are removed from the training set
M_i	arbitrary large enough number, $(i = \{0, \ldots, 10\})$
ϵ	arbitrary small enough number
CD_{min}	minimum level of the certainty degree for the optimal rule
S_{min}	minimum level of *Support* (coverage) for the optimal rule
K_{pc}	binary parameter, $K_{pc} = 1$ if the class label for pattern p is c, $(p \in P, c \in C)$
h_c	binary parameter, $h_c = 1$ for the target class label, $(c \in C)$
μ_{pfl}	the degree of membership for the value of the feature f of the pattern p to the fuzzy linguistic label L^l_f, $(p \in P, f \in F, l \in L)$
w_i	objective weights, $(i = \{1, 2\})$
x_{fl}	binary decision variable, $x_{fl} = 1$ if the linguistic label corresponding to the feature f in the antecedent of the optimal fuzzy rule is L^l_f, (i.e., $\hat{A}_f = L^l_f$), $(f \in F, l \in L)$
y_p	binary decision variable, $y_p = 1$ if the degree of compatibility (matching degree) between pattern p and the optimal rule is equal to zero, $(p \in P)$
m_p	continuous decision variable, the degree of compatibility (matching degree) between pattern p and the optimal rule (matching degree), $(p \in P)$

m'_{pc} continuous decision variable, the degree of compatibility (matching degree) between pattern p and the optimal rule if their class labels are c (i.e., correct prediction), otherwise it is equal to zero, $(p \in P, c \in C)$

t_{pc} binary decision variable, $t_{pc} = 1$ if $m'_{pc} > 0$ (i.e., the optimal rule classifies pattern p correctly to the class c), $(p \in P, c \in C)$

The data preparation step and MIPEF formulation are described in the following sections.

3.1.1. Data preparation

In this step, the parameters of the model are initialized, and fuzzy sets and their membership functions are defined. The degrees of membership to the fuzzy linguistic labels (μ_{pfl}) are calculated for all patterns by applying the following rule. If the degree of membership for the value of the feature f of pattern p to the fuzzy linguistic label L_f^l is equal to zero, then μ_{pfl} is set equal to $-M_0$ instead of zero, where M_0 is an arbitrary large enough number that is bigger than n_f. The reason for using this adjustment and the appropriate value for M_0 is discussed in the next section. Furthermore, for the sake of simplicity we rounded down any membership value less than 0.01 to zero.

For example, assume that Fig. 1 presents fuzzy partitions for the first feature of a data set and the second pattern in the data set has the value of 70 for this feature. Then, based on the partitions, this value has a membership value of 0.2 in L_1^2, 0.8 in L_1^3, and zero in the other fuzzy linguistic labels. Thus, μ_{pfl} are given to MIPEF as: $\mu_{211} = -M_0$, $\mu_{212} = 0.2$, $\mu_{213} = 0.8$, $\mu_{214} = -M_0$, and $\mu_{215} = -M_0$.

3.1.2. MIPEF formulation

MIPEF aims to extract an accurate and interpretable rule. Hence, the objective function consists of two terms that are to be maximized simultaneously. The first term determines the percentage of the patterns that are classified correctly by the optimal rule. It aims to maximize coverage and accuracy of the optimal rule. The second term describes the average number of conditions (labels) in the optimal rule. It aims to find more interpretable rules by minimizing the complexity that is caused as a result of abundant conditions.

$$\text{Maximize} \quad w_1 \left(\frac{\sum_{p \in P} \sum_{c \in C} t_{pc}}{N_{c^*}} \right) + w_2 \left(1 - \frac{\sum_{f \in F} \sum_{l \in L} x_{fl}}{n_f} \right) \tag{9}$$

The first constraint ensures that at most one linguistic label is presented per feature in the optimal fuzzy rule. It allows the absence of non-beneficial features in the optimal rule.

$$\sum_{l \in L} x_{fl} \leq 1 \quad \forall f \in F \tag{10}$$

The next set of constraints works as described in the following to determine the degree of compatibility (matching degree) between the optimal fuzzy rule and the patterns.

$$\sum_{f \in F} \sum_{l \in L} \mu_{pfl} x_{fl} + M_1 y_p \geq 0 \qquad \forall p \in P \tag{11}$$

$$m_p \leq M_2 (1 - y_p) \qquad \forall p \in P \tag{12}$$

$$\sum_{f \in F} \sum_{l \in L} \mu_{pfl} x_{fl} - M_3 (1 - y_p) \leq -\epsilon \qquad \forall p \in P \tag{13}$$

$$m_p \leq \sum_{f \in F} \sum_{l \in L} \mu_{pfl} x_{fl} + M_4 y_p \qquad \forall p \in P \tag{14}$$

$$m_p \geq \sum_{f \in F} \sum_{l \in L} \mu_{pfl} x_{fl} \qquad \forall p \in P \tag{15}$$

Notice that μ_{pfl} is calculated by applying the adjustment described in the data preparation step. This adjustment helps constraints (11)–(14) distinguish when a feature is absent in the optimal fuzzy rule from when a pattern is not compatible with the rule. Without the adjustment, both cases would end up with $\mu_{pfl} x_{fl} = 0$, and $\Sigma_{f \in F} \Sigma_{l \in L} \mu_{pfl} x_{fl}$ might be larger than zero which causes the matching degree between the pattern and the optimal rule to be non-zero; however, the matching degree should be equal to zero in the latter case. After the adjustment, $\Sigma_{f \in F} \Sigma_{l \in L} \mu_{pfl} x_{fl}$ will be less than zero if one of the features of a pattern is not compatible with the optimal rule and, alternatively $\mu_{pfl} x_{fl} = 0$ if the corresponding feature is absent in the antecedent of the optimal rule. A large enough number must be assigned to M_0 in the preparation step such that $\Sigma_{f \in F} \Sigma_{l \in L} \mu_{pfl} x_{fl}$ becomes less than zero even if one of the features of the pattern is not compatible with the optimal fuzzy rule and the rest of them are compatible at the highest possible value. Since the optimal fuzzy rule has at most one label per feature and the maximum value of a degree of membership is one, any number bigger than n_f satisfies the required condition for M_0.

Therefore, if one of the features of a pattern is not compatible with the corresponding label in the optimal fuzzy rule (i.e., $\mu_{pfl} < 0$), $\Sigma_{f \in F} \Sigma_{l \in L} \mu_{pfl} x_{fl}$ becomes negative; hence, y_p must be equal to one to keep constraint (11) feasible. This makes the right hand side of constraint (12) equal to zero, enforces $m_p = 0$ and constraints (13)–(15) become unbinding. If all features of a pattern are compatible with the optimal fuzzy rule (i.e., $\Sigma_{f \in F} \Sigma_{l \in L} \mu_{pfl} x_{fl} > 0$) then y_p must be equal to zero to keep constraint (13) feasible. In this case, constraints (11) and (12) become unbinding and constraints (14) and (15) become binding, and work as an equality constraint to determine the matching degree between the pattern and the optimal fuzzy rule.

The next set of constraints determines the degree of compatibility between the optimal rule and the patterns whose class labels are the same as the one of the optimal fuzzy rule (i.e., correct predictions).

$$m'_{pc} \leq m_p \qquad \forall p \in P, c \in C \tag{16}$$

$$m'_{pc} \geq m_p - M_5 (1 - K_{pc} h_c) \qquad \forall p \in P, c \in C \tag{17}$$

$$m'_{pc} \leq M_6 K_{pc} h_c \qquad \forall p \in P, c \in C \tag{18}$$

If the class label of a pattern is identical with that of the optimal fuzzy rule, then $K_{pc} h_c$ is equal to one and constraints (16) and (17) become binding, and work as an equality constraint and set $m'_{pc} = m_p$. Constraint (18) becomes unbinding in this case. Otherwise, constraints (16) and (17) become unbinding and constraint (18) becomes binding and sets $m'_{pc} = 0$.

Constraints (19) and (20) determine if a pattern is predicted correctly by the optimal fuzzy rule (i.e., the matching degree is greater than zero and the class labels of the pattern and the optimal fuzzy rule are identical). They relate the binary decision variable t_{pc} (indicator for correct classifications) and the continuous decision variable m'_{pc} (matching degree for correct classifications) such that $t_{pc} = 1$ if the optimal fuzzy rule correctly classifies a pattern (i.e., $m'_{pc} > 0$), and $t_{pc} = 0$ otherwise.

$$t_{pc} \leq M_7 m'_{pc} \qquad \forall p \in P, c \in C \tag{19}$$

$$M_8 t_{pc} \geq m'_{pc} \qquad \forall p \in P, c \in C \tag{20}$$

The next constraint calculates the certainty degree of the optimal fuzzy rule and ensures that it is higher than the minimum

acceptable level (CD_{min}).

$$\sum_{p\in P}\sum_{c\in C} m'_{pc} \geq CD_{min}\sum_{p\in P} m_p \tag{21}$$

The next constraint calculates the coverage (*Support*) of the optimal fuzzy rule and ensures that it is higher than the minimum acceptable level, S_{min}.

$$\sum_{p\in P}\sum_{c\in C}\frac{t_{pc}}{N_{c^*}} \geq S_{min} \tag{22}$$

where, N_{c^*} is the number of the patterns whose class labels are the same as the target class label (c^*), which is specified by h_c at each iteration.

MIPEF finds a rule that has the highest coverage and satisfies CD_{min}. However, during the final iterations of the algorithm, few patterns are left and rules typically only apply to a few patterns. To avoid this, constraint (22) sets a lower bound on rule coverage. If the coverage is less than S_{min}, the constraint is violated. This will force the decrement of CD_{min} which should lead to rules with higher coverage.

3.1.3. Iterative procedure

MIPEF is an exact approach that finds one optimal rule for the pre-determined class label, c^*, and pre-specified level of certainty degree, CD_{min}, if an optimal solution exists. To extract multiple rules, it is run through the following iterative procedure for different CD_{min}.

The procedure starts with the data preparation step in which the model parameters are initialized. Then, the first class label is selected as the target class label (i.e., $c^* = 1$) by setting h_1 equal to one and the remaining h_cs to zero. After that, CD_{min} is initialized to its maximum level, which is equal to one. In the next step, MIPEF is solved for the given parameters. If an optimal solution exists, it is added to the RB and converted to a set of two taboo constraints which are added to the original model to prevent the model from re-obtaining this solution in the next steps. If an optimal solution does not exist, the CD_{min} is decremented by 0.05 and MIPEF is re-solved with the new parameters. This process continues until N_r fuzzy rules are extracted. Afterwards, all the patterns that have been predicted correctly by the extracted rules are temporarily removed from the training set. Then, N_{c^*} is recalculated and N_r is set equal to zero. This enforces MIPEF to search for the rules that predict those patterns in the training set that have not been covered yet. The algorithm aims to build a RB that covers almost all patterns in the training set.

This process continues until an acceptable percent of the patterns whose class labels are the same as the target class label are removed from the training set (i.e., $N_{c^*} \leq N_{min}$). Afterwards, the removed patterns are restored to the training set and the next iteration starts by changing the target class label to the next class label and re-setting CD_{min} to its maximum level. The algorithm stops once all class labels are covered.

In each iteration of the algorithm, the obtained solution is converted to a set of taboo constraints as described in the following. Assume binary parameters a^r_{fl} and b^r_c are defined as:

a^r_{fl}: Binary parameter, $a^r_{fl} = 1$ if $x^r_{fl} = 1$ where x^r_{fl} is the x_{fl} of the rth fuzzy rule, $(r \in R, f \in F, l \in L)$

b^r_c: Binary parameter, $b^r_c = 1$ if $c = C_r$ where C_r is the class label of rule r, $(r \in R, c, C_r \in C)$

and parameters $\bar{a}^r_{fl}$ and $\bar{b}^r_c$ are calculated as follows:

$$\bar{a}^r_{fl} = 1 - a^r_{fl}$$
$$\bar{b}^r_c = 1 - b^r_c$$

The fuzzy rule that was extracted in the rth iteration is converted to the following taboo constraints in the subsequent iterations:

$$\sum_{f\in F}\sum_{l\in L} a^r_{fl}x_{fl} + \sum_{c\in C} b^r_c h_c \leq n_{\hat{A}_r} + v_r \quad \forall r \in R \tag{23}$$

$$\sum_{f\in F}\sum_{l\in L} \bar{a}^r_{fl}x_{fl} + \sum_{c\in C} \bar{b}^r_c h_c \geq v_r \quad \forall r \in R \tag{24}$$

where v_r is a binary decision variable and is equal to one if all conditions in the antecedent of the fuzzy rule r are repeated in the optimal fuzzy rule, and $n_{\hat{A}_r}$ is the number of conditions (labels) in the rule r. If antecedents and class labels of the optimal fuzzy rule and previously extracted rule r are identical (i.e., $\sum_{f\in F}\sum_{l\in L} a^r_{fl}x_{fl} + \sum_{c\in C} b^r_c h_c = n_{\hat{A}_r} + 1$), then v_r must be equal to one. This forces the left hand side of constraint (24) to be greater than one. Thus the optimal rule must have at least one additional antecedent if it has all active antecedents of the rule r and both have the same class label, otherwise both constraints become unbinding.

For all class labels, MIPEF is solved first without any taboo constraints. Once the first optimal fuzzy rule is found, the taboo constraints associated with that rule are added to the MIPEF and kept until all rules for that class label are extracted. Then, all taboo constraints are removed from MIPEF and the procedure continues for the next class label.

One should notice that in MIPEF, the certainty degree of the optimal rule is calculated in constraint (21) according to (2) and using (5) to calculate the degree of compatibility between the patterns and the optimal rule. However, we did not use (5) as written to calculate the degree of compatibility between the patterns and the optimal rule in the constraints. The denominator of (5) was removed because $n_{\hat{A}_k}$ equals to $\Sigma_{f\in F}\Sigma_{l\in L}x_{fl}$ making the associated constraints nonlinear. However, this adjustment does not affect the results of the model because the purpose of dividing the aggregated membership values by $n_{\hat{A}_k}$ in (5) is to normalize the compatibility degree of various rules which can be ignored in MIPEF without loss of generality as it extracts one fuzzy rule per time.

Also, constraint (22) does not use (6) as written to calculate *Support* of the optimal rule. It approximates *Support* by replacing the numerator of (6) with $\Sigma_{p\in P}\Sigma_{c\in C}t_{pc}$. It means that this constraint restricts the percentage of the patterns correctly covered by the optimal rule. This change is made to restrict the percentage of the correctly covered patterns instead of the average matching degrees of the optimal rule with the correct predictions. This facilitates determining S_{min} because it is now interpreted as the acceptable coverage percentage of the optimal rule and can be determined regardless of the matching degrees that can be different for different patterns. That is, $S_{min} = 0.3$ means that the optimal rule is required to cover at least 30 percent of the patterns correctly.

Does the temporary removal from the training set impact the certainty degree of new rules? Since the certainty degree is the sum of matching degrees of correct predictions divided by the sum of correct and incorrect predictions and the class label of removed patterns is the same as the target class label, there is no misclassification. The removed patterns do not decrease the certainty degree of the new rule. It is possible that some removed patterns have non-zero matching degrees with the new rule which will slightly increase the denominator and numerator. However, its lower bound used by constraint (21) remains valid.

3.2. Pruning the RB

Once all class labels are covered by MIPEF, the RB contains more rules than is required to cover the patterns in the training set. Albeit, this surplus depends on the N_r and higher values of N_r

result in keeping more fuzzy rules in the RB. In addition, there are other deficiencies that justify a rule pruning process. Some rules may be redundant as the patterns that they cover can be predicted by the other rules that may have higher certainty degrees or *Support*. Furthermore, MIPEF extracts fuzzy rules one by one and does not take interactions and conflicts among the rules into consideration. Therefore, some rules may conflict with the others. In other words, IPEP learns more fuzzy rules than required aiming to provide more flexibility to MIPP in finding the best combination of rules with minimum conflicts.

MIPP is the corrected formulation of the model developed by Aydogan, Karaoglan, and Pardalos (2012). The corrections herein include adding a set of constraints and removing two sets of redundant constraints to make the model more accurate and efficient (Derhami & Smith, 2016). Their paper is not considered in our experimental study because the reported results are flawed (Derhami & Smith, 2016). The definition of the new sets, parameter and decision variables used in addition to the ones described for MIPEF is presented in the following.

m''_{pr}	the degree of compatibility (matching degree) between the pattern p and rule r, $(p \in P, r \in R)$
C_r	class label of the rule r
A_r	accuracy of the rule r
z_{pc}	binary decision variable, $z_{pc} = 1$ if pattern p is classified to the class label c, $(p \in P, c \in C)$
w_r	binary decision variable, $w_r = 1$ if rule r is selected, $(r \in R)$
u_p	continuous decision variable, the maximum matching degree for the pattern p, $(p \in P)$

The MIPP formulation is described in the following.

3.2.1. MIPP formulation

$$\text{Maximize} \quad \sum_{p\in P}\sum_{c\in C} K_{pc} z_{pc} \tag{25}$$

Subject to

$$\sum_{c\in C} z_{pc} \le 1 \qquad \forall p \in P \tag{26}$$

$$u_p \ge \sum_{\substack{r\in R \\ C_r=c}} A_r m''_{pr} w_r \qquad \forall p \in P, c \in C \tag{27}$$

$$z_{pc} \le 1 - \frac{1}{M_9}\left(u_p - \sum_{\substack{r\in R \\ C_r=c}} A_r m''_{pr} w_r\right) \qquad \forall p \in P, c \in C \tag{28}$$

$$z_{pc} \le M_{10} \sum_{\substack{r\in R \\ C_r=c}} A_r m''_{pr} w_r \qquad \forall p \in P, c \in C \tag{29}$$

$$z_{pc} \in \{0, 1\} \qquad \forall p \in P, c \in C \tag{30}$$

$$w_r \in \{0, 1\} \qquad \forall r \in R \tag{31}$$

$$u_p \ge 0 \qquad \forall p \in P \tag{32}$$

The objective function (25) maximizes correct classifications. Constraint (26) guarantees that patterns are not classified into more than one class label. Constraint (27) determines the highest matching degree between a pattern and all selected rules. The model employs the FRM described in Section 2.2. For all class labels, the matching degrees between a pattern and all rules with the same class label are aggregated and the class label associated to the highest weighted sum is selected to classify that pattern. Constraint (28) assigns class labels to the patterns. For the class label associated with the highest sum of matching degrees, $u_p - \sum_{\substack{r\in R \\ C_r=c}} A_r m''_{pr} w_r$ is equal to zero. So, z_{pc} would be equal to one as consistent with the objective function. For the remaining class labels $u_p - \sum_{\substack{r\in R \\ C_r=c}} A_r m''_{pr} w_r$ is greater than zero so the right hand side of constraint (28) becomes less than one and therefore z_{pc} is enforced to be equal to zero for these class labels. Constraint (29) ensures that if the matching degrees of a pattern with all selected rules are zero then this pattern is not assigned to any class label. In other words, if none of the selected rules are compatible with a pattern, then that pattern is considered classified incorrectly.

3.3. Parameter setting

The pseudo-code of the IPEP algorithm is shown in Algorithm 1. The analysis of the parameters used in MIPEF and MIPP and the values used in the experiments are described in Table 1. Setting appropriate values for the big *M*s significantly decreases complexity of a mixed integer programming model. As Table 1 presents, the lower bounds for M_1 to M_9 (except M_7) depend on n_f and M_0.

Algorithm 1 IPEP pseudo-code.

```
Initialize parameters
for all (c ∈ C) do
    h_c = 1, h_{i:i≠c} = 0
    CD_min = 1.0
    N_min = 0.02 * N_c*
    while N_c* > N_min do
        while (Count ≤ N_r) do
            Solve MIPEF
            if MIPEF is feasible then
                Convert solution to a set of taboo constraints
                Add the new taboo constraints to the MIPEF
                Add solution to the RB
                Count = Count + 1
            else
                CD_min = CD_min − 0.05
        Remove all correctly predicted patterns
        Update N_c*
        Count = 0
    Restore removed patterns
    Remove taboo constraints from MIPEF
Solve MIPP
return Selected rules
```

M_7 enlarges small matching degrees (less than one) to prevent violation of constraint (19). As explained in the data preparation step, any membership values less than 0.01 are rounded down to zero, thus the smallest value that m'_{pc} can have is 0.01. Hence, M_7 has to be larger than 100 to prevent violation of this constraint for such small matching degrees.

The value of M_{10} depends on the minimum acceptable level for the weighted sum of matching degrees when the weighted sum is very small (close to zero). Smaller values of M_{10} result in a large weighted sum to be treated as zero while the reverse is true for larger values. Setting it to 10,000 forces the MIPP to consider any weighted sum larger than 0.0001.

ϵ in constraint (13) has to be a small number to force y_p to become zero when $\Sigma_{f\in F}\Sigma_{l\in L}\mu_{pfl}x_{fl}$ is positive (i.e., the pattern is compatible with the optimal rule).

N_{min} impacts both the predictive accuracy and computational time. Smaller values of N_{min} force MIPEF to cover more patterns in the training set and, therefore, the predictive accuracy of the classifier and computational time increase as the result of extracting

Table 1
Parameters used in the IPEP.

Parameter	Minimum value	Value used
M_0	n_f	$n_f + 5$
M_1	$n_f \times M_0$	$(n_f \times M_0) + 5$
M_2	M_0	$M_0 + 5$
M_3	M_0	$M_0 + 5$
M_4	$n_f \times M_0$	$(n_f \times M_0) + 5$
M_5	M_0	$M_0 + 5$
M_6	M_0	$M_0 + 5$
M_7	100	100
M_8	M_0	$M_0 + 5$
M_9	n_f	$n_f + 5$
M_{10}	10, 000	10, 000
ϵ	0.001	0.00001
N_r	1	1, 5 and 10
N_{min}	–	$0.02N_{c^*}$
S_{min}	–	0.3
w_1	–	0.8
w_2	–	0.2

more information. However, a very small value forces the model to extract rules that cover only a few patterns (e.g., one or two patterns). The reverse is true for larger values of N_{min}. In this case, some significant information in the training set may be unused. We set N_{min} to $0.02N_{c^*}$, where N_{c^*} is the initial number of patterns in the training set that belong to the target class label. This means MIPEF keeps learning rules until 98 percent of the patterns belonging to the target class label are covered by the extracted rules.

S_{min} determines the minimum acceptable coverage for the optimal rule. There is a trade-off between S_{min} and CD_{min}. A large value of S_{min} results in obtaining a rule that has small accuracy and high coverage while the reverse is true for a small S_{min}. The primary objective of the model is to find accurate rules, therefore we set S_{min} to 0.3 to prioritize accuracy while rules with low coverage are avoided.

w_1 and w_2 determine the accuracy-interpretability trade-off. We tested different quantities for these two parameters and found that setting w_1 to 0.8 and w_2 to 0.2 results in a more accurate classifier.

N_r significantly impacts both the predictive accuracy and computational cost. The complexity and computational time of MIPEF increase as N_r increases. Moreover, the predictive accuracy of the classifier is likely to increase as the result of considering more information. We studied this parameter in the experimental analysis section for the values of 1, 5 and 10.

4. Experimental framework

The performance of the proposed model is evaluated using 22 classification data sets obtained from the UCI repository of machine learning databases[1] and KEEL data set repository[2]. The characteristics of these data sets are summarized in Table 2. A non-stratified 10 fold cross-validation as described in the following, was used to evaluate performance of the proposed algorithm. The data sets were randomly partitioned into 10 subproblems. Then from the subproblems, one of them was chosen as the test set to validate the model and the remaining ones were used as the training set. This process was repeated 10 times, each time with a different subproblem as the test set. At the end, all subproblems used exactly once as the validation data.

The experimental framework is designed as follows: first, we analyze the effects of two main components of IPEP to determine the best structure of IPEP in terms of computational efficiency and predictive accuracy. These two components are the rule pruning model (MIPP) and the parameter N_r, the number of fuzzy rules learned before the pattern reduction starts. To evaluate the significance of the MIPP model in the predictive accuracy of IPEP, we tested IPEP with and without the rule pruning model for N_r equal to five and termed them $IPEP_{5R}$ and IPE_{5R}, respectively. Furthermore, we tested IPE and IPEP with N_r equal to one and ten, respectively, to study the effect of N_r on predictive accuracy. Finally, we compared the results of IPEP with the ones obtained by other rule classifier learning algorithms from the literature.

We utilize non-parametric statistical tests to evaluate significant differences among the results (Demšar, 2006; García, Molina, Lozano, & Herrera, 2009). As recommended by Demšar (2006), we use Wilcoxon's Signed-Rank test (Wilcoxon, 1945) for pair-wise comparisons, and Friedman's test (Friedman, 1937) and Iman and Davenport's test (Iman & Davenport, 1980) for multiple comparisons. Finally, we use Holm's method (Holm, 1979) as a post hoc test.

IPEP was coded in ILOG CPLEX 12.6 Java API and ran on a workstation equipped with an Intel Xeon CPU E5-1650 v2 (3.5 GHz) and 64GB of RAM memory.

4.1. Analysis of the components of the IPEP

The experimental analysis in this section aims to demonstrate the effectiveness of the rule pruning (MIPP) model on predictive accuracy and to determine a suitable value for N_r, in terms of balancing computational efficiency and predictive accuracy. The following are considered:

- *IPE_{1R}:* The IPEP algorithm without rule pruning. This algorithm is an improved version of the model proposed in Derhami and Smith (2014). In this model, N_r is set equal to one and rule pruning (MIPP) is not performed on the RB; therefore, all fuzzy rules generated by MIPEF are used to classify the patterns.
- *IPE_{5R}:* The IPEP algorithm without the rule pruning procedure (MIPP). In this model, N_r is set equal to five.
- *$IPEP_{5R}$:* The complete version of IPEP algorithm described in Algorithm 1. N_r is set equal to five in this model.
- *$IPEP_{10R}$:* The complete version of IPEP algorithm with N_r equal to 10.

The average accuracy of the proposed models in the training (Tra.) and test (Tst.) data sets, and the average number of rules (#Rul) are presented in Table 3. Due to computational complexity, a termination criterion is set for the optimization process of MIPP in large data sets. Specifically, the optimization process is prematurely terminated for the MIPP if it does not find the optimal solution in three hours search. In such a case, IPEP uses the best integer solution MIPP obtained. The results obtained by invoking this condition are marked in Table 3 for the corresponding data sets.

We compared $IPEP_{10r}$ with IPE_{1r}, IPE_{5r} and $IPEP_{5r}$ using Wilcoxon's Signed-Rank test. Table 4 shows the results of these statistical comparisons. Wilcoxon's test detects significant differences between $IPEP_{10r}$ and all remaining models. Using rule pruning (MIPP) along with N_r equal to 10 significantly enhances the predictive accuracy. However, considering the small gap in the total average accuracy on the test sets between $IPEP_{5r}$ and $IPEP_{10r}$, and the computational resources required for higher values of N_r, we can conclude that increasing N_r to more than 10 would not make a significant impact on predictive accuracy. Since $IPEP_{10r}$ statistically obtained higher predictive accuracy, we select it to compare performance of IPEP with the other classifiers from the literature.

Table 5 shows the average computational time of the proposed algorithms for one fold. The two algorithms that did not utilize rule pruning (IPE_{1r} and IPE_{5r}) required much less computational time than the ones that employed rule pruning ($IPEP_{1r}$ and

[1] http://archive.ics.uci.edu/ml/datasets.html
[2] http://www.keel.es

Table 2
Characteristics of data sets used for the experimental study.

Data set	# patt.	# feat.	# class.	Data set	# patt.	# feat.	# class.
Cleveland	297	13	5	New-thyroid	215	5	3
Contraceptive	1473	9	3	Parkinsons	195	22	2
Dermatology	358	34	6	Pima	768	8	2
Ecoli	336	7	8	Saheart	462	9	2
Glass	214	9	6	Segment	2310	19	7
Haberman	306	3	2	Sonar	208	60	2
Hayes-Roth	160	4	3	Tae	151	5	3
HillValley1	1212	100	2	Vehicle	846	18	4
HillValley2	1212	100	2	Wdbc	569	30	2
Iris	150	4	3	Wine	178	13	3
Libras Mov.	360	90	15	Wisconsin	683	9	2

Table 3
Average accuracy (percentage) and number of rules obtained by different components of IPEP.

Data set	IPE_{1r}			IPE_{5r}			$IPEP_{5r}$			$IPEP_{10r}$		
	Tra.	Tst.	#Rul	Tra.	Tst.	#Rul	Tra.	Tst.	#Rul	Tra.	Tst.	#Rul
Cleveland	82.15	56.57	32.1	81.18	56.57	111.1	88.74	53.87	61.7	90.46	**57.91**	81.8
Contraceptive	53.24	53.36	16.4	53.35	50.24	56.5	57.77*	51.32	30.3	58.49	**54.31**	32.0
Dermatology	99.41	91.90	16.8	99.50	91.90	79.5	99.47	92.74	79.0	99.47	**93.02**	158.0
Ecoli	88.62	84.23	26.9	87.80	**84.82**	106.3	93.22	82.44	50.8	94.31	83.93	60.1
Glass	80.01	62.62	28.8	79.70	62.15	116.6	89.36	**71.03**	54.0	90.08	70.56	70.2
Haberman	76.33	74.51	11.6	76.40	72.55	24.0	79.88	**75.16**	13.5	79.70	74.84	14.5
Hayes-Roth	88.26	78.12	15.4	77.08	70.63	16.9	86.11	80.00	10.9	89.86	**82.50**	11.6
HillValley1	53.03	51.49	4.0	52.62	50.25	20.0	53.47*	50.74	9.1	55.36	**52.06**	18.6
HillValley2	51.83	50.33	4.3	52.10	51.16	20.5	54.82*	51.98	9.8	56.69*	**53.88**	14.8
Iris	97.11	**96.67**	8.1	95.19	92.67	21.0	98.37	94.67	15.3	99.26	**96.67**	25.4
Libras	97.72	69.72	39.9	97.62	70.83	194.5	97.78	70.28	187.6	97.87	**72.22**	378.9
New-thyroid	94.57	93.02	10.4	95.04	93.95	40.6	97.88	**95.35**	21.8	98.60	**95.35**	27.8
Parkinsons	98.63	90.26	14.4	98.92	**91.28**	63.5	99.43	89.23	37.8	99.26	89.23	64.4
Pima	76.07	**74.74**	13.8	76.06	72.66	49.3	80.51*	74.61	26.7	81.47*	74.48	32.3
Saheart	78.04	**71.21**	16.8	77.92	69.70	56.4	83.36*	69.70	32.3	85.11*	70.56	43.8
Segment	92.62	91.30	36.4	93.02	91.90	176.0	95.76	93.25	96.1	96.05*	**93.85**	152.6
Sonar	99.63	75.48	12.2	99.68	80.77	48.5	99.73	**83.17**	49.5	99.79	79.81	98.0
Tae	62.62	54.30	17.9	61.59	45.70	51.3	73.22	55.63	25.2	73.80	**57.62**	29.0
Vehicle	75.93	67.26	34.1	76.71	68.79	148.4	82.81*	**69.62**	63.7	83.31*	68.44	82.4
Wdbc	98.85	95.43	14.8	98.98	**96.31**	68.5	99.36	95.08	47.3	99.30	**96.31**	87.8
Wine	99.06	**96.07**	9.3	99.56	94.94	38.5	99.63	93.82	35.4	99.56	95.51	61.0
Wisconsin	98.15	95.17	12.1	97.56	95.61	32.5	98.21	96.49	15.1	98.42	**96.63**	16.4
Average	83.72	76.08	18.02	83.07	75.24	70.02	86.77	76.83	44.22	87.56	**77.71**	70.97

*Optimization prematurely terminated for the MIPP once the elapsed time reached three hours.

Table 4
Wilcoxon's test to analyze $IPEP_{10r}$, $\alpha = 0.05$.

$IPEP_{10r}$ vs.	W^+	W^-	p-value	Hypothesis
IPE_{1r}	209	22	0.0012	Rejected
IPE_{5r}	207	24	0.0015	Rejected
$IPEP_{5r}$	173	37	0.0111	Rejected

Table 5
Computational time comparison (hour:minute:second).

Data set	IPE_{1r}	IPE_{5r}	$IPEP_{5r}$	$IPEP_{10r}$
Cleveland	0:02:07	0:04:15	0:04:23	0:13:20
Contraceptive	0:12:05	0:14:34	3:18:00	3:27:02
Dermatology	0:00:19	0:01:04	0:01:09	0:02:08
Ecoli	0:09:35	0:17:28	0:17:10	0:25:59
Glass	0:00:40	0:01:16	0:01:18	0:01:26
Haberman	0:00:09	0:00:10	0:00:10	0:00:23
Hayes-Roth	0:00:04	0:00:03	0:00:03	0:00:05
HillValley1	0:20:18	0:21:48	3:58:04	3:26:45
HillValley2	0:12:02	0:18:29	3:37:09	3:55:33
Iris	0:00:02	0:00:04	0:00:04	0:00:05
Libras Mov.	0:12:34	0:28:16	0:30:59	0:20:03
New-thyroid	0:00:06	0:00:09	0:00:09	0:00:14
Parkinsons	0:00:25	0:00:29	0:00:47	0:00:46
Pima	0:03:37	0:06:51	3:00:35	3:17:34
Saheart	0:03:06	0:07:04	0:22:00	3:07:17
Segment	3:49:42	3:07:30	6:32:58	4:57:09
Sonar	0:07:48	0:35:42	0:33:27	1:03:08
Tae	0:00:13	0:00:15	0:00:19	0:00:39
Vehicle	3:05:11	5:44:23	8:57:18	1:56:54
Wdbc	0:10:32	0:26:54	0:25:23	0:48:27
Wine	0:00:03	0:00:10	0:00:10	0:00:20
Wisconsin	0:03:07	0:05:24	0:05:25	0:06:15
Average	0:23:21	0:32:50	1:26:41	1:14:10

$IPEP_{10r}$); however, they achieved lower predictive accuracy. IPE_{1r} required the lowest computational time for most data sets and $IPEP_{10r}$ required the highest computational time while it achieved the highest predictive accuracy. This shows that the rule pruning model (MIPP) enhances the predictive accuracy of the classifier but needs additional computation.

IPEP comprises two NP-hard mixed integer programming models that become computationally hard to solve when the size of the problem grows. One may ask what is the size of the largest data set that IPEP can handle? It is hard to define a relationship between the size of the problem and computational time because it depends on both problem size and structure. However, this question can be roughly answered by estimating the sizes of the mixed integer programming models that are solved. MIPEF generates $n_p(n_c+1)+5n_f$ binary decision variables and $n_p(n_c+1)$ continuous decision variables. MIPP generates $n_p n_c + n_r$ binary

Table 6
Average accuracies (percentage) and computational times (hour:minute:second) of the IPEP and other classifiers.

Data set	FURIA			FARCHD			IVTURS-FARC			$IPEP_{10r}$		
	Tra.	Tst.	Time	Tra.	Tst.	Time	Tra.	Tst.	Time	Tra.	Tst.	Time
Cleveland	60.83	56.21	0:00:01	87.54	56.20	0:00:28	83.69	**57.91**	0:15:50	90.46	**57.91**	0:13:20
Contraceptive	55.74	53.77	0:00:01	62.54	52.81	0:01:22	60.21	53.43	1:08:52	58.49	**54.31**	3:27:02
Dermatology	98.76	**95.26**	0:00:01	100.00	92.17	0:01:00	99.84	94.98	0:24:01	99.47	93.02	0:02:08
Ecoli	91.77	83.70	0:00:01	91.90	83.38	0:00:09	89.85	**84.81**	0:03:37	94.31	83.93	0:25:59
Glass	84.00	69.16	0:00:00	81.41	68.27	0:00:09	78.92	65.91	0:02:58	90.08	**70.56**	0:01:26
Haberman	76.69	72.24	0:00:00	80.47	73.90	0:00:02	80.18	**75.88**	0:00:48	79.70	74.84	0:00:23
Hayes-Roth	87.50	80.00	0:00:00	91.67	75.00	0:00:03	91.81	80.00	0:01:05	89.86	**82.50**	0:00:05
HillValley1	55.02	50.99	0:00:02	54.35	50.75	0:26:00	55.21	51.57	1:02:09	55.36	**52.06**	3:26:45
HillValley2	54.99	51.15	0:00:02	52.15	50.57	0:36:12	52.26	50.33	4:59:53	56.69	**53.88**	3:55:33
Iris	98.00	96.00	0:00:00	98.37	94.67	0:00:01	98.22	95.33	0:00:23	99.26	**96.67**	0:00:05
Libras Mov.	92.38	62.22	0:00:02	95.43	**75.00**	4:50:18	85.68	67.78	5:23:44	97.87	72.22	0:20:03
New-thyroid	99.28	93.48	0:00:00	98.76	93.98	0:00:02	98.35	94.03	0:00:40	98.60	**95.35**	0:00:14
Parkinsons	98.29	88.71	0:00:00	96.81	**92.34**	0:00:10	94.87	87.71	0:01:11	99.26	89.23	0:00:46
Pima	79.08	73.58	0:00:01	83.13	74.22	0:00:19	80.51	73.83	0:13:02	81.47	**74.48**	3:17:34
Saheart	74.80	**71.19**	0:00:01	82.25	68.60	0:00:14	79.89	70.35	0:03:35	85.11	70.56	3:07:17
Segment	99.33	**97.27**	0:00:03	94.82	93.55	0:02:12	91.45	90.39	0:40:28	96.05	93.85	4:57:09
Sonar	97.92	78.88	0:00:01	99.04	80.67	0:18:50	96.10	**84.17**	0:22:52	99.79	79.81	1:03:08
Tae	56.89	47.13	0:00:00	73.66	56.42	0:00:03	70.64	**61.04**	0:00:47	73.80	57.62	0:00:39
Vehicle	80.04	**71.99**	0:00:01	79.80	67.96	0:01:17	72.76	67.37	0:21:00	83.31	68.44	1:56:54
Wdbc	99.26	96.14	0:00:01	98.75	95.08	0:00:32	98.36	96.13	0:05:47	99.30	**96.31**	0:48:27
Wine	99.38	94.41	0:00:00	100.00	95.52	0:00:08	99.69	**96.63**	0:00:55	99.56	95.51	0:00:20
Wisconsin	98.98	96.35	0:00:01	98.58	96.20	0:00:08	98.42	96.34	0:02:33	98.42	**96.63**	0:06:15
Average	83.59	76.36	0:00:01	86.43	76.69	0:17:15	84.41	77.09	0:41:39	87.56	**77.71**	1:14:10

decision variables and n_p continuous decision variables. That means MIPEF and MIPP together generate $2n_p(n_c+1)+5n_f$ and $n_p(n_c+1)+n_r$ decision variables. Hence, n_p, n_c, and n_f influence the size of the models.

The largest model in the tested data sets belongs to the Segment data set for which MIPEF contains more than 37,000 decision variables. The computational time for this data set is the second highest time. The highest computational time belongs to the Vehicle data set where MIPEF contains more than 8500 decision variables (the fourth largest model).

Considering that n_r depends on the number of patterns in the training set (the more patterns that exist in the training set, the more rules required to cover them), one can infer that n_p and n_c are the most influential factors on the size (complexity) of the models. From the experimental results, data sets with more than 3000 patterns and five or more classes may exceed reasonable computational times. The heuristic algorithms from the literature are options in these very large cases.

4.2. Comparison with the other models

The predictive accuracy of IPEP was statistically analyzed and compared with the predictive accuracy of the following state-of-the-art algorithms from the literature:

- *FURIA* (Hühn & Hüllermeier, 2009): An extension of the RIPPER rule learning algorithm (Cohen, 1995) that identifies unordered rule set in two phases: building and optimization. It uses a trapezoidal membership function for fuzzy sets, applies a new rule stretching technique to manage uncovered patterns, and utilizes a greedy heuristic approach to fuzzify the crisp rules obtained by the RIPPER algorithm.
- *FARC-HD* (Alcalá-Fdez et al., 2011): A GA-based algorithm that employs fuzzy association rules for classification. It learns fuzzy association rules using a search tree to list all possible frequent fuzzy item sets for all classes, and a pattern weighting scheme to preselect the best rules. The preselection process aims to decrease the computational cost in the rule selection process. Finally, a GA tunes the lateral position of the membership functions and selects the set of fuzzy association rules with highest classification accuracy.
- *IVTURS-FARC* (Sanz, Fernández, Bustince, & Herrera, 2013): A fuzzy association rule-based classification algorithm which utilizes an interval-valued FRM. It learns fuzzy rules using FARC-HD and a modified FRM that computes the matching degrees using Interval-Valued Restricted Equivalence Functions (IV-REF). This FRM aims to manage the ignorance that the interval-valued fuzzy sets represent. Finally, a GA tunes the parameters used in constructing the IV-REFs and selects the best rules.

Table 7
Statistical analysis with Friedman's and Iman–Davenport's tests, $\alpha = 0.05$.

Test	Statistics	Critical value	p-value	Hypothesis
Friedman	15.0409	7.8147	0.0018	Rejected
Iman and Davenport	6.1983	2.7505	0.0009	Rejected

These three algorithms are included in the KEEL software which is publicly available on the referenced website[3]. We used the default configurations and parameters for these algorithms in the KEEL software. To develop a fair comparison, we tested all algorithms on the same cross-validation partitions that we used for $IPEP_{10r}$. Table 6 compares accuracies and computational times of these models with the proposed model. The computational times show the average runtime for one fold. $IPEP_{10r}$ obtained the highest predictive accuracy in the test sets of 11 data sets and obtained the second highest accuracy on the most of the remaining data sets. Moreover, it achieved the highest average accuracy in the test and training sets among all the investigated approaches.

Fig. 2 compares the average ranking of each classifier using Friedman's method. Friedman's and Iman–Davenport's tests explored the significant differences among the approaches considered. Table 7 shows that both statistical tests reject equivalence of results between the investigated algorithms. In the next step, we used Holm's method as the post hoc test to distinguish the significant difference between the $IPEP_{10r}$ as the control algorithm and

[3] http://www.keel.es

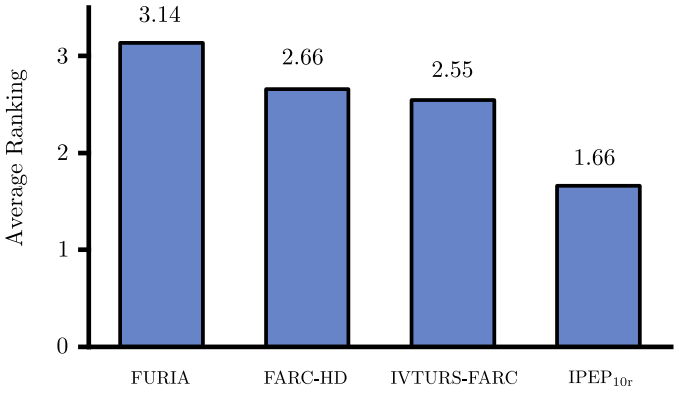

Fig. 2. Friedman rankings on the test sets.

Table 8
Holm's statistical analysis, $\alpha = 0.05$.

i	Algorithm	z	p-value	α/i	Hypothesis
3	FARC-HD	3.7952	0.0001	0.0167	Rejected
2	FURIA	2.5690	0.0102	0.025	Rejected
1	IVTURS-FARC	2.2771	0.0228	0.05	Rejected

the other algorithms. Table 8 shows that Holm's method rejects all hypotheses and that $IPEP_{10r}$ significantly outperforms all considered algorithms.

From a computational point of view, FURIA (Hühn & Hüllermeier, 2009) is the fastest algorithm among those analyzed. It obtained the lowest computational time in all data sets. FARCHD (Alcalá-Fdez et al., 2011) is the second fastest algorithm. It obtained the second lowest runtime in all data sets except the Libras Mov. in which $IPEP_{10r}$ performed faster. However, these two quick approaches both obtained the lowest average predictive accuracies and Friedman's rankings. In most of the high-dimensional data sets, our proposed approach obtained the first or the second highest predictive accuracy. This shows that our approach obtains more accurate results in high-dimensional datasets. $IPEP_{10r}$ performed faster than IVTURS-FARC (Sanz et al., 2013) in 12 datasets. Although the computational times of $IPEP_{10r}$ are reasonable, it is not the fastest approach. This is because it uses exact optimization algorithms while the others use heuristic approaches.

5. Conclusion

In this paper, we develop an integer programming approach to learn fuzzy rules for fuzzy rule-based classification systems. The proposed approach is a hybrid algorithm consisting of two integer programming models. The first is a mixed integer programming model that learns optimal fuzzy rules. We applied this model in an iterative procedure to find fuzzy rules to cover almost all patterns in the training set. New rules are obtained by converting the previous rules into a set of taboo constraints to prevent the re-finding of old solutions. The second prunes the RB by removing redundant rules and maximizing the total accuracy and coverage of the RB.

We carried out an experimental study on 22 benchmark data sets and calculated statistical analysis to compare predictive accuracy of the proposed algorithm with the most recent and accurate algorithms from the literature. The comparative study shows that the proposed algorithm significantly outperforms the other algorithms investigated in this paper in terms of the predictive accuracy.

It is important to note that our algorithm is an exact algorithm that consists of two NP-hard mixed integer programming models. While they always yield the optimal answer (set of rules) the computational effort grows with problem size. Modifying our approach to consider problem structure is a future challenge that could alleviate computational effort for large data sets. Another opportunity for further work is to consider the effects in the final results of the shape and number of fuzzy sets defining the membership values.

References

Al-Ebbini, L., Oztekin, A., & Chen, Y. (2016). FLAS: Fuzzy lung allocation system for US-based transplantations. *European Journal of Operational Research, 248*(3), 1051–1065.

Alcalá-Fdez, J., Alcalá, R., & Herrera, F. (2011). 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), 857–872.

Amo, A., Montero, J., Biging, G., & Cutello, V. (2004). Fuzzy classification systems. *European Journal of Operational Research, 156*(2), 495–507.

de Andrés, J., Landajo, M., & Lorca, P. (2005). Forecasting business profitability by using classification techniques: A comparative analysis based on a Spanish case. *European Journal of Operational Research, 167*(2), 518–542.

Aydogan, E. K., Karaoglan, I., & Pardalos, P. M. (2012). hGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems. *Applied Soft Computing, 12*(2), 800–806.

Baykasoğlu, A., & Özbakir, L. (2007). Mepar-miner: Multi-expression programming for classification rule mining. *European Journal of Operational Research, 183*(2), 767–784.

Bekiros, S. D. (2010). Fuzzy adaptive decision-making for boundedly rational traders in speculative stock markets. *European Journal of Operational Research, 202*(1), 285–293.

Belacel, N., Raval, H. B., & Punnen, A. P. (2007). Learning multicriteria fuzzy classification method PROAFTN from data. *Computers & Operations Research, 34*(7), 1885–1898.

Berlanga, F., Rivera, A., del Jesus, M., & Herrera, F. (2010). GP-COACH: Genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems. *Information Sciences, 180*(8), 1183–1200.

Casillas, J., Martínez, P., & Benítez, A. (2009). Learning consistent, complete and compact sets of fuzzy rules in conjunctive normal form for regression problems. *Soft Computing, 13*(5), 451–465.

Castro, J., Flores-Hidalgo, L., Mantas, C., & Puche, J. (2007). Extraction of fuzzy rules from support vector machines. *Fuzzy Sets and Systems, 158*(18), 2057–2077.

Chiang, J.-H., & Hao, P.-Y. (2004). Support vector learning mechanism for fuzzy rule-based modeling: A new approach. *IEEE Transactions on Fuzzy Systems, 12*(1), 1–12.

Cohen, W. W. (1995). Fast effective rule induction. In A. P. Russell (Ed.), *Machine learning proceedings 1995* (pp. 115–123). San Francisco (CA): Morgan Kaufmann.

Cordón, O. (2011). A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning, 52*(6), 894–913.

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

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research, 7*, 1–30.

Derhami, S., & Smith, A. (2014). Iterative mixed integer programming model for fuzzy rule-based classification systems. In *Proceedings of the 2014 IEEE international conference on fuzzy systems (FUZZ-IEEE)* (pp. 2079–2084).

Derhami, S., & Smith, A. E. (2016). A technical note on the paper "hGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems". *Applied Soft Computing, 41*, 91–93.

Fazzolari, M., Giglio, B., Alcalá, R., Marcelloni, F., & Herrera, F. (2013). A study on the application of instance selection techniques in genetic fuzzy rule-based classification systems: Accuracy-complexity trade-off. *Knowledge-Based Systems, 54*, 32–41.

Feng, G. (2006). A survey on analysis and design of model-based fuzzy control systems. *IEEE Transactions on Fuzzy Systems, 14*(5), 676–697.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association, 32*(200), 675–701.

García, S., Molina, D., Lozano, M., & Herrera, F. (2009). A study on the use of non–parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 Special Session on Real parameter Optimization. *Journal of Heuristics, 15*(6), 617–644.

Hoffmann, F., Baesens, B., Mues, C., Gestel, T. V., & Vanthienen, J. (2007a). Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms. *European Journal of Operational Research, 177*(1), 540–555.

Hoffmann, F., Baesens, B., Mues, C., Gestel, T. V., & Vanthienen, J. (2007b). Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms. *European Journal of Operational Research, 177*(1), 540–555.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics, 6*, 65–70.

Hong, T.-P., Lee, Y.-C., & Wu, M.-T. (2014). An effective parallel approach for genetic-fuzzy data mining. *Expert Systems with Applications, 41*(2), 655–662.

Hu, Y.-C., Hu, J.-S., Chen, R.-S., & Tzeng, G.-H. (2004). Assessing weights of product attributes from fuzzy knowledge in a dynamic environment. *European Journal of Operational Research, 154*(1), 125–143.
Hühn, J., & Hüllermeier, E. (2009). FURIA: An algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery, 19*(3), 293–319.
Iman, R. L., & Davenport, J. M. (1980). Approximations of the critical region of the friedman statistic. *Communications in Statistics-Theory and Methods, 9*(6), 571–595.
Ishibuchi, H., Mihara, S., & Nojima, Y. (2013). Parallel distributed hybrid fuzzy GBML models with rule set migration and training data rotation. *IEEE Transactions on Fuzzy Systems, 21*(2), 355–368.
Ishibuchi, H., & Nojima, Y. (2007). Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning, 44*(1), 4–31. Genetic Fuzzy Systems and the Interpretability Accuracy Trade-off
Ishibuchi, H., & Yamamoto, T. (2005). Rule weight specification in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems, 13*(4), 428–435.
del Jesus, M., Hoffmann, F., Navascués, L., & Sánchez, L. (2004). Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms. *IEEE Transactions on Fuzzy Systems, 12*(3), 296–308.
Khalili-Damghani, K., Sadi-Nezhad, S., Lotfi, F. H., & Tavana, M. (2013). A hybrid fuzzy rule-based multi-criteria framework for sustainable project portfolio selection. *Information Sciences, 220*(0), 442–462.
Kulluk, S., Özbakır, L., & Baykasoğlu, A. (2013). Fuzzy DIFACONN-miner: A novel approach for fuzzy rule extraction from neural networks. *Expert Systems with Applications, 40*(3), 938–946. FUZZYSS11: 2nd International Fuzzy Systems Symposium 17–18 November 2011, Ankara, Turkey
Li, M., & Wang, Z. (2009). A hybrid coevolutionary algorithm for designing fuzzy classifiers. *Information Sciences, 179*(12), 1970–1983.
Li, R., & Wang, Z. (2004). Mining classification rules using rough sets and neural networks. *European Journal of Operational Research, 157*(2), 439–448.
Mansoori, E. (2011). FRBC: A fuzzy rule-based clustering algorithm. *IEEE Transactions on Fuzzy Systems, 19*(5), 960–971.
Mansoori, E., Zolghadri, M., & Katebi, S. (2008). SGERD: A steady-state genetic algorithm for extracting fuzzy classification rules from data. *IEEE Transactions on Fuzzy Systems, 16*(4), 1061–1071.
Martens, D., Baesens, B., Gestel, T. V., & Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research, 183*(3), 1466–1476.
Mesiarová-Zemánková, A. (2014). Multipolar aggregation operators in reasoning methods for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems, 22*(6), 1569–1584.
Nakandala, D., Samaranayake, P., & Lau, H. (2013). A fuzzy-based decision support model for monitoring on-time delivery performance: A textile industry case study. *European Journal of Operational Research, 225*(3), 507–517.
Ravi, V., Reddy, P., & Zimmermann, H.-J. (2000). Pattern classification with principal component analysis and fuzzy rule bases. *European Journal of Operational Research, 126*(3), 526–533.
Ravi, V., & Zimmermann, H.-J. (2000). Fuzzy rule based classification with *Feature Selector* and modified threshold accepting. *European Journal of Operational Research, 123*(1), 16–28.
Ren, Y., Liu, X., & Cao, J. (2011). A parsimony fuzzy rule-based classifier using axiomatic fuzzy set theory and support vector machines. *Information Sciences, 181*(23), 5180–5193.
Sanz, J., Fernández, A., Bustince, H., & Herrera, F. (2011). A genetic tuning to improve the performance of Fuzzy Rule-Based Classification Systems with Interval-Valued Fuzzy Sets: Degree of ignorance and lateral position. *International Journal of Approximate Reasoning, 52*(6), 751–766.
Sanz, J., Fernández, A., Bustince, H., & Herrera, F. (2013). IVTURS: A linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection. *IEEE Transactions on Fuzzy Systems, 21*(3), 399–411.
Tsakonas, A. (2006). A comparison of classification accuracy of four genetic programming-evolved intelligent structures. *Information Sciences, 176*(6), 691–724.
Wang, X., Liu, X., Pedrycz, W., Zhu, X., & Hu, G. (2012). Mining axiomatic fuzzy set association rules for classification problems. *European Journal of Operational Research, 218*(1), 202–210.
Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*(6), pp. 80–83.
Yuan, Y., Feldhamer, S., Gafni, A., Fyfe, F., & Ludwin, D. (2002). The development and evaluation of a fuzzy logic expert system for renal transplantation assignment: Is this a useful tool? *European Journal of Operational Research, 142*(1), 152–173.