Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 161 (2010) 3 – 19
FUZZY sets and systems
www.elsevier.com/locate/fss

Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method☆

Julián Luengo*, Francisco Herrera

Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

Available online 15 April 2009

Abstract

The analysis of data complexity is a proper framework to characterize the tackled classification problem and to identify domains of competence of classifiers. As a practical outcome of this framework, the proposed data complexity measures may facilitate the choice of a classifier for a given problem. The aim of this paper is to study the behaviour of a fuzzy rule based classification system and its relationship to data complexity. We use as a case of study the fuzzy hybrid genetic based machine learning method presented in [H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification problems, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 35 (2) (2005) 359–365]. We examine several metrics of data complexity over a wide range of data sets built from real data and try to extract behaviour patterns from the results. We obtain rules which describe both good or bad behaviours of the fuzzy rule based classification system. These rules use values of data complexity metrics in their antecedents, so we try to predict the behaviour of the method from the data set complexity metrics prior to its application. Therefore, we can establish the domains of competence of this fuzzy rule based classification system.

Keywords: Classification; Data complexity; Fuzzy rule based systems; Genetic fuzzy systems

1. Introduction

Fuzzy rule based classification systems (FRBCSs) [17,19] are a very useful tool in the ambit of machine learning since they are capable of building a linguistic model clearly interpretable by human beings. There is a vast literature in the field of FRBCSs [19], which is very active at this time.

The prediction capabilities of classifiers are strongly dependent on the problem's characteristics. An emergent field has recently arisen, that uses a set of complexity measures applied to the problem to describe its difficulty. These measures quantify particular aspects of the problem which are considered complicated to the classification task [15]. Studies of data complexity metrics applied to particular classification algorithms can be found in [15,3,2,22].

☆ Supported by the Spanish Ministry of Science and Technology under Project TIN2008-06681-C06-01. J. Luengo holds an FPU scholarship from Spanish Ministry of Education and Science.
* Corresponding author. Tel.: +34 958240598.
E-mail addresses: julianlm@decsai.ugr.es (J. Luengo), herrera@decsai.ugr.es (F. Herrera).

0165-0114/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2009.04.001

The complexity in the data can be used for characterizing FRBCS performance and it can be considered a new trend in the use of FRBCSs in pattern recognition. We understand that no data complexity metrics have been analysed with FRBCSs up to now.

In this work we are interested in analysing the relationship between FRBCSs and the complexity measures, considering a case of study using the fuzzy hybrid genetic based machine learning (FH-GBML) method proposed by Ishibuchi and Yamamoto [18]. In particular we consider three types of data complexity measures based on the overlaps in feature values from different classes; separability of classes; and measures of geometry, topology, and density of manifolds.

To perform this study, we have created 438 binary classification data sets from real world problems, and computed the value of eight metrics proposed by Ho and Basu [14]. We have analysed the intervals of the complexity measure values related to the created data sets, in which FH-GBML method performs well or badly, and then formulated a rule for such intervals. The rules try to describe the ranges where some information and conclusions about the behaviour of the FH-GBML method can be stated.

The paper is organized as follows. In Section 2 we describe the FRBCS we have used. In Section 3 the considered complexity measures are introduced as well as the most recent literature on the topic. In Section 4 we show the process used to build up the bunch of data sets used and the validation scheme. In Section 5 we include the experimental set-up and the results obtained and rules extracted, along with their analysis. Finally, in Section 6 some concluding remarks are pointed out.

2. Preliminaries: fuzzy rule based classification systems

Any classification problem consists of m training patterns $x_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ from M classes where x_{pi} is the ith attribute value ($i = 1, 2, \ldots, n$) of the p-th training pattern.

In this work we use fuzzy rules of the following form:

$$\text{Rule } R_j: \text{ If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then Class} = C_j \text{ with } RW_j, \tag{1}$$

where R_j is the label of the jth rule, $x = (x_1, \ldots, x_n)$ is an n-dimensional pattern vector, A_{ji} is an antecedent fuzzy set, C_j is a class label, and RW_j is the rule weight. We use triangular membership functions as antecedent fuzzy sets.

As learning method we use FH-GBML [18], which belongs to the genetic fuzzy system (GFS) [8]. In the following three subsections we include the fuzzy reasoning model, a complete description of the algorithm, and a short review on the GFSs topic.

2.1. Fuzzy reasoning model

Considering a new pattern $x_p = (x_{p1}, \ldots, x_{pn})$ and a rule base (RB) composed of L fuzzy rules, the steps followed by the reasoning model are the following [7]:

1. *Matching degree*. To calculate the *strength of activation of the if-part for all rules in the RB with the pattern* x_p, using a conjunction operator (usually a T-norm):

$$\mu_{A_j}(x_p) = T(\mu_{A_{j1}}(x_{p1}), \ldots, \mu_{A_{jn}}(x_{pn})), \quad j = 1, \ldots, L. \tag{2}$$

   In this work we will use the product T-norm.
2. *Association degree*. To compute the *association degree of the pattern* x_p *with the M classes according to each rule in the RB*. When using rules with the form of (1), this association degree only refers to the consequent class of the rule (i.e., $k = C_j$):

$$b_j^k = h(\mu_{A_j}(x_p),\ RW_j^k), \quad k = 1, \ldots, M,\ \ j = 1, \ldots, L. \tag{3}$$

   We model function h as the product T-norm in every case.
3. *Pattern classification soundness degree for all classes*. We use an aggregation function that combines the positive degrees of association calculated in the previous step:

$$Y_k = f(b_j^k,\ j = 1, \ldots, L \text{ and } b_j^k > 0), \quad k = 1, \ldots, M. \tag{4}$$

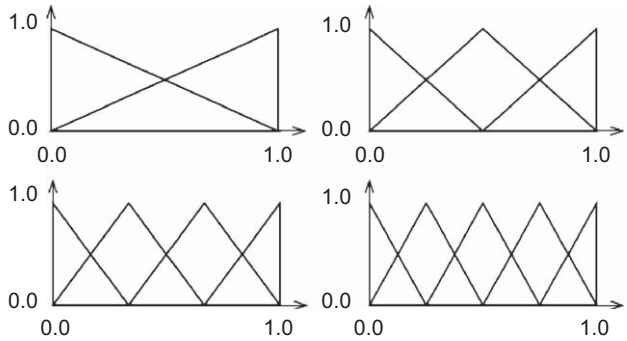

Fig. 1. Four fuzzy partitions for each attribute membership function.

As fuzzy reasoning method we use the winner rule method (classical approach) for classifying new patterns with the rule set. Every new pattern is classified as the consequent class of a single winner rule which is determined as

$$Y_k = \max\{b_j^k, \quad j = 1, \ldots, L \text{ and } k = C_j\}. \tag{5}$$

4. *Classification*. We apply a decision function F over the soundness degree of the system for the pattern classification for all classes. This function will determine the class label l corresponding to the maximum value:

$$F(Y_1, \ldots, Y_M) = l \quad \text{such that } Y_l = \{\max(Y_k), k = 1, \ldots, M\}. \tag{6}$$

2.2. Learning approach: fuzzy hybrid genetic based machine learning method

The basis of this algorithm described here, FH-GBML, consists of a hybrid Pittsburgh and Michigan genetic learning approach [18]:

- The Pittsburgh approach in which each rule set is handled as an individual.
- The Michigan approach (where an individual represents an unique rule), which is used as a kind of heuristic mutation for partially modifying each rule set, because of its high search ability to efficiently find good fuzzy rules.

This method simultaneously uses four fuzzy set partitions for each attribute, as shown in Fig. 1. As a result, each antecedent attribute is initially associated with 14 fuzzy sets generated by these four partitions as well as a special "do not care" set (i.e., 15 in total).

The main steps of this algorithm are described below:

Step 1: Generate N_{pop} rule sets with N_{rule} fuzzy rules.

Step 2: Calculate the fitness value of each rule set in the current population.

Step 3: Generate $(N_{pop} - 1)$ rule sets by the selection, crossover and mutation in the same manner as the Pittsburgh-style algorithm. Apply a single iteration of the Michigan-style algorithm (i.e., the rule generation and the replacement) to each of the generated rule sets with a pre-specified probability.

Step 4: Add the best rule set in the current population to the newly generated $(N_{pop} - 1)$ rule sets to form the next population.

Step 5: Return to Step 2 if the pre-specified stopping condition is not satisfied.

Next, we will describe every step of the algorithm:

- Initialization: N_{rule} training patterns are randomly selected. Then, a fuzzy rule from each of the selected training patterns is generated by choosing probabilistically (as shown in (7)) an antecedent fuzzy set from the 14 candidates $B_k(k = 1, 2, \ldots, 14)$ (see Fig. 1) for each attribute. Then each antecedent fuzzy set of the generated fuzzy rule is replaced by the *don't care* condition using a pre-specified probability $P_{don't\ care}$:

$$P(B_k) = \frac{\mu_{B_k}(x_{pi})}{\sum_{j=1}^{14} \mu_{B_j}(x_{pi})}. \tag{7}$$

- Fitness computation: The fitness value of each rule set S_i in the current population is calculated as the number of correctly classified training patterns by S_i. For the Michigan approach the computation follows the same scheme.
- Selection: It is based on binary tournament.
- Crossover: The substring-wise and bit-wise uniform crossover is applied in the Pittsburgh part. In the case of the Michigan part only the bit-wise uniform crossover is considered.
- Mutation: Each fuzzy partition of the individuals is randomly replaced by a different fuzzy partition using a pre-specified mutation probability for both approaches.

In our study, we have used the following parameters' values for the Ishibuchi and Yamamoto's FH-GBML method:

- Number of fuzzy rules: $5 \times p$ rules (where p is the number of examples in the data set).
- Number of rule sets (N_{pop}): 200 rule sets.
- Crossover probability: 0.9.
- Mutation probability: $1/p$ (where p is the number of examples in the data set).
- Number of replaced rules: All rules except the best-one (Pittsburgh-part, elitist approach), number of rules/5 (Michigan-part).
- Total number of generations: 1000 generations.
- Don't care probability: 0.5.
- Probability of the application of the Michigan iteration: 0.5.

For more details about this proposal, please refer to [18].

2.3. Genetic fuzzy systems

A GFS is basically a fuzzy system augmented by a learning process based on evolutionary computation, which includes genetic algorithms, genetic programming, and evolutionary strategies, among other evolutionary algorithms (EAs) [11].

The automatic definition of a fuzzy rule based system (FRBS) can be seen as an optimization or search problem. EAs are a well known and widely used global search technique with the ability to explore a large search space for suitable solutions only requiring a performance measure. In addition to their ability to find near optimal solutions in complex search spaces, the generic code structure and independent performance features of EAs make them suitable candidates to incorporate *a priori* knowledge. In the case of FRBSs, this *a priori* knowledge may be in the form of linguistic variables, fuzzy membership function parameters, fuzzy rules, number of rules, etc. These capabilities extended the use of EAs in the development of a wide range of approaches for designing FRBSs over the last few years, as has been pointed out in the last international journal special issues on GFSs [4,5,9,6].

Finally, an extensive review of the most recent developments of GFS and FRBS can be found in [12]. The web site http://sci2s.ugr.es/gfs/ provides complete information and material on the topic.

3. Data complexity measures

In the following subsections, we first present a short review on recent studies on data complexity metrics (Section 3.1), and then we describe the measures of overlapping (Section 3.2), measures of separability of classes (Section 3.3) and measures of geometry (Section 3.4) used in our study.

3.1. Recent studies on data complexity

As we have mentioned, data complexity measures are a series of metrics that quantify data set characteristics which imply some difficulty to the classification task. In the following we gather several recent publications related to these complexity measures and their applications. They can show a picture of the most recent developments in the topic:

- In [14], Ho and Basu propose some complexity measures for binary classification problems, gathering metrics of three types: overlaps in feature values from different classes; separability of classes; and measures of geometry, topology, and density of manifolds.
- In [23], Singh offers a review of data complexity measures and proposes two new ones.

Table 1
Complexity metrics used in this study.

Measure	Description
F2	Volume of overlap region
F3	Maximum (individual) feature efficiency
L1	Minimized sum of error distance by linear programming
L2	Error rate of linear classifier by linear programming
N2	Ratio of average intra/inter class NN distance
N3	Error rate of 1-NN classifier
N4	Nonlinearity of 1-NN classifier
T2	Average number of points per dimension

- In [3], Bernadó-Mansilla and Ho investigate the domain of competence of XCS by means of a methodology that characterizes the complexity of a classification problem by a set of geometrical descriptors.
- In [20], Li et al. analyse some omnivariate decision trees using the measure of complexity based in data density proposed by Ho and Basu.
- Baumgartner and Somorjai define specific measures for regularized linear classifiers in [2], using Ho and Basu's measures as reference.
- Sánchez et al. analyse the effect of the data complexity in the nearest neighbours (NNs) classifier in [22].
- Dong and Kothari propose in [10] a feature selection algorithm based on a complexity measure defined by Ho and Basu.
- Mollineda et al. in [21] extend some of Ho and Basu's measure definitions for problems with more than two classes. They analyse these generalized measures in two classic prototype selection algorithms and remark that Fisher's discriminant ratio is the most effective for prototype selection.

In our study we will study eight of the measures proposed in [14] which offer information for the FH-GBML method. They are summarized in Table 1.

In the following subsections we describe the measures we have used, classified by their family.

3.2. Measures of overlaps in feature values from different classes

These measures focus on the effectiveness of a single feature dimension in separating the classes, or the composite effects of a number of dimensions. They examine the range and spread of values in the data set within each class, and check for overlaps among different classes.

F2: Volume of overlap region. Let the maximum and minimum values of each feature f_i in class C_j be $\max(f_i, C_j)$ and $\min(f_i, C_j)$, then the overlap measure F2 is defined as

$$F2 = \prod_i \frac{MINMAX_i - MAXMIN_i}{MAXMAX_i - MINMIN_i},$$

where $i = 1, \ldots, d$ for a d-dimensional problem, and

$$MINMAX_i = MIN(\max(f_i, C_1), \max(f_i, C_2)),$$
$$MAXMIN_i = MAX(\min(f_i, C_1), \min(f_i, C_2)),$$
$$MAXMAX_i = MAX(\max(f_i, C_1), \max(f_i, C_2)),$$
$$MINMIN_i = MIN(\min(f_i, C_1), \min(f_i, C_2)).$$

F2 measures the amount of overlap of the bounding boxes of two classes. It is the product of per-feature overlap ratios, each of which is the width of the overlap interval normalized by the width of the entire interval encompassing the two classes. The volume is zero as long as there is at least one dimension in which the value ranges of the two classes are disjoint.

F3: Maximum (individual) feature efficiency. In a procedure that progressively removes unambiguous points falling outside the overlapping region in each chosen dimension [13], the efficiency of each feature is defined as *the fraction of all remaining points separable by that feature*. To represent the contribution of the most useful feature in this sense, we use the maximum feature efficiency (largest fraction of points distinguishable with only one feature) as a measure (F3). This measure considers only separating hyperplanes perpendicular to the feature axes. Therefore, even for a linearly separable problem, F3 may be less than 1 if the optimal separating hyperplane is oblique.

3.3. Measures of separability of classes

These measures give indirect characterizations of class separability. They assume that a class is made up of a single or multiple manifolds that form the support of the probability distribution of the given class. The shape, position and interconnectedness of these manifolds give hints on how well two classes are separated, but they do not describe separability by design. Some examples are shown as follows:

L1: Minimized sum of error distance by linear programming (LP). Linear classifiers can be obtained by a linear programming formulation proposed by Smith [24]. The method minimizes the sum of distances of error points to the separating hyperplane (subtracting a constant margin):

$$\begin{aligned} &\text{minimize} && \mathbf{a}^{\mathbf{t}}\mathbf{t} \\ &\text{subject to} && \mathbf{Z}^{\mathbf{t}}\mathbf{w} + \mathbf{t} \geqslant \mathbf{b}, \\ & && \mathbf{t} \geqslant \mathbf{0}, \end{aligned}$$

where $\mathbf{a}$, $\mathbf{b}$ are arbitrary constant vectors (both chosen to be 1), $\mathbf{w}$ is the weight vector to be determined, $\mathbf{t}$ is an error vector, and $\mathbf{Z}$ is a matrix where each column $\mathbf{z}$ is defined on an input vector $\mathbf{x}$ (augmented by adding one dimension with a constant value 1) and its class C (with value C_1 or C_2) as follows:

$$\begin{cases} \mathbf{z} = +\mathbf{x} & \text{if } C = C_1, \\ \mathbf{z} = -\mathbf{x} & \text{if } C = C_2. \end{cases}$$

The value of the objective function in this formulation is used as a measure (L1). The measure has a zero value for a linearly separable problem. Its value can be heavily affected by outliers located in the wrong side of the optimal hyperplane. The measure is normalized by the number of points in the problem and also by the length of the diagonal of the hyperrectangular region enclosing all training points in the feature space. It is zero for a linearly separable problem. We should notice that this measure can be heavily affected by the presence of outliers in the data set.

L2: Error rate of linear classifier by linear programming. This measure is the error rate of the linear classifier defined for L1, measured with the training set. With a small training set this can be a severe underestimate of the true error rate.

N2: Ratio of average intra/inter class nearest neighbour distance. For each input instance x_p, we calculate the distance to its nearest neighbour within the class ($intraDist(x_p)$) and the distance to nearest neighbour of any other class ($interDist(x_p)$). Then, the result is the ratio of the sum of the intra-class distances to the sum of the inter-class distances for each input example, i.e.,

$$N2 = \frac{\sum_{i=0}^{m} intraDist(x_i)}{\sum_{i=0}^{m} interDist(x_i)},$$

where m is the number of examples in the data set. This metric compares the within-class spread with the distances to the nearest neighbours of other classes. Low values of this metric suggest that the examples of the same class lay closely in the feature space. Large values indicate that the examples of the same class are disperse. It is sensitive to the classes of the closest neighbours to a point, and also to the difference in magnitude of the between-class distances and that of the within-class distances.

N3: Error rate of 1-NN classifier. This is simply the error rate of a nearest neighbour classifier measured with the training set. The error rate is estimated by the leave-one-out method. The measure denotes how close the examples of different classes are. Low values of this metric indicate that there is a large gap in the class boundary.

3.4. Measures of geometry, topology, and density of manifolds

These measures evaluate to what extent two classes are separable by examining the existence and shape of the class boundary. The contributions of individual feature dimensions are combined and summarized in a single score, usually a distance metric, rather than evaluated separately. Two measures from this family are described as follows:

N4: Nonlinearity of 1-NN classifier. This is the nonlinearity measure, as defined by linear programming. Hoekstra and Duin [16] proposed a measure for the nonlinearity of a classifier with respect to a given data set. Given a training set, the method first creates a test set by linear interpolation (with random coefficients) between randomly drawn pairs of points from the same class. Then the error rate of the classifier (trained by the given training set) on this test set is measured. Here we use such a nonlinearity measure for the linear classifier defined for L1. In the case of N4, error is calculated for a nearest neighbour classifier. This measure is for the alignment of the nearest neighbour boundary with the shape of the gap or overlap between the convex hulls of the classes.

T2: Average number of points per dimension. This is a simple ratio of the number of points in the data set over the number of feature dimensions, i.e.,

$$T2 = \frac{m}{n},$$

where m is the number of examples in the data set and n is the number of attributes of the data set. This measure is included mostly for connection with prior studies on sample sizes. Because the volume of a region scales exponentially with the number of dimensions, a linear ratio between the two is not a good measure of sampling density.

4. Data sets choice for the experimental study

We evaluate FH-GBML on a set of 438 binary classification problems. These problems are generated from pairwise combinations of the classes of 21 problems from the University of California, Irvine (UCI) repository [1]. The selected ones are *iris*, *wine*, *new-thyroid*, *solar-flare*, *led7digit*, *zoo*, *yeast*, *tae*, *balanced*, *car*, *contraceptive*, *ecoli*, *hayes-roth*, *shuttle*, *australian*, *pima*, *monks*, *bupa*, *glass*, *haberman*, and *vehicle*.

In order to do that, first we take each data set and extract the examples belonging to each class. Then we construct a new data set with the combination of the examples from two different classes. This will result in a new data set with only two classes and the examples which have two such classes as output. For example, one data set obtained from *iris* with this procedure could contain only the examples of *Iris-setosa* and *Iris-virginica* and not those from *Iris-versicolor*.

We perform this process for every possible pairwise combination of classes. However, if a data set obtained with this procedure proves to be linearly separable, we discard it. If the data set proves to be linearly separable, then we could classify it with a linear classifier with no error, so such a data set would not be a representative problem. The complexity measure L1 indicates if a problem is linearly separable when its value is zero, so every data set with a L1 value of zero will be discarded.

This method for generating binary data sets is limited by the combinatorics itself, and we can only obtain over 200 new data sets with the original 20 data sets with this first approach. In order to obtain additional data sets, we take the next step to combine the classes from a data set: we group the classes two by two, that is, we create a new binary data set, and each of its two classes are the combination of two original classes each. For this second approach we have used *ecoli*, *glass*, and *flare* data sets, since they have a high number of class labels. For example, using *ecoli* we can create a new binary data set combining *cp* and *im* classes into a new one (say class "A"), and *pp* and *imU* (say new class "B"). The new data set would contain only the examples of classes *cp, im, pp*, and *imU*, but their class label now would be "A" if their original class was *cp* or *im*, and "B" if it was *pp* or *imU*. Again, those data sets with an L1 value of 0 are discarded.

Finally, these pairwise combinations resulted in 438 binary classification problems which are used as our test-bed. Although simple, these two methods for creating binary data sets produce a wide range of values for the complexity measures, even from the same original data set.

To estimate the classifiers' error, we use a 10-fold cross validation test once all the measures are computed. We take the mean accuracy of training and test of the 10 partitions as a representative measure of the method's performance.

Fig. 2 contains the results of FH-GBML showing the training and test accuracy over all the 438 data sets, plotted in ascending training accuracy value. We would like to point out how over-fitting is continuously present in Fig. 2.

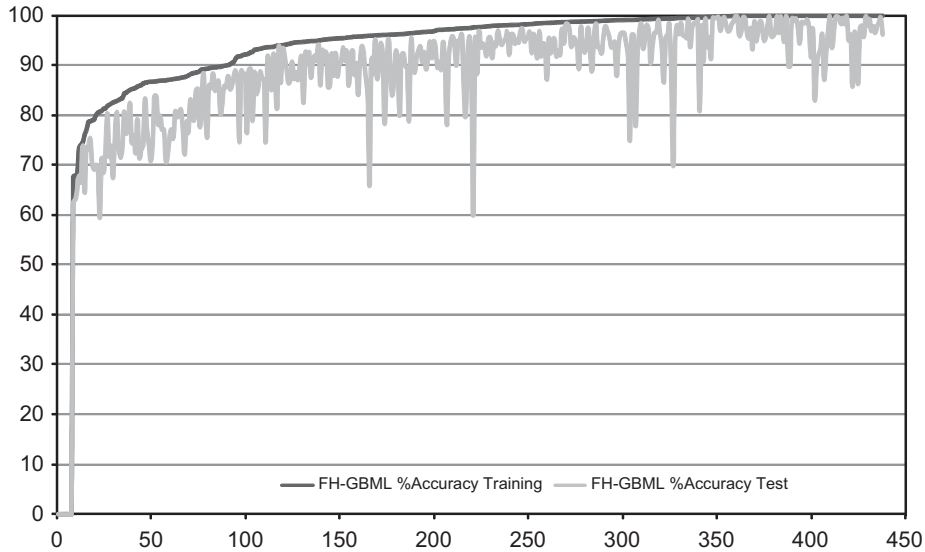

Fig. 2. Accuracy in training/test for FH-GBML sorted by training accuracy.

5. Experimental study: analysis of the FH-GBML method with data complexity measures

This study begins with the obtained results of the FH-GBML for the data sets considered. For each complexity measure, the data sets are sorted by its value, and put altogether in a figure. From these figures we obtain useful intervals which represent a good or bad behaviour of the classification method for the mentioned eight complexity measures. From these intervals we construct several rules that model the performance of the used FRBCS.

In order to do this analysis, we divide this section into the following two studies:

1. Determination of rules based on FH-GBML method's behaviour in Section 5.1.
2. Analysis of the collective evaluation of the set of rules in Section 5.2.

5.1. Determination of rules based on FH-GBML method's behaviour

First, we must point out what we understand for *good and bad behaviour* of FH-GBML:

- We understand for *good behaviour* an average high test accuracy in the interval as well as the absence of over-fitting.
- By *bad behaviour* we refer to the presence of over-fitting and/or average low test accuracy in the interval.

In the following we present the results of the execution over the 438 data sets summarized in Figs. 3–10. In each figure the results obtained by the FH-GBML method are sorted by the ascending value of the corresponding complexity measure. In the X axis we represent the data sets, not the complexity measure value, and the Y axis depicts the accuracy obtained both in training and test. The reason to do so is to give each data set the same space in the graphic representation. For those measures where we can find different *ad hoc* intervals which present good or bad behaviour of the FH-GBML, we use a vertical line to delimit the interval of the region of interest.

In Table 2 we have summarized the intervals found *ad hoc* from Figs. 3–10.

Once we have defined the *ad hoc* intervals, in Table 3 we have summarized the rules derived from them. Given a particular data set X, we get the complexity measure (CM) of X with the notation $CM[X]$. Table 3 is organized with the following columns:

- The first column corresponds to the identifier of the rule for further references.
- The "Rule" column presents the rule itself.
- The third column "Support" presents the percentage of data sets which verify the antecedent part of the rule.

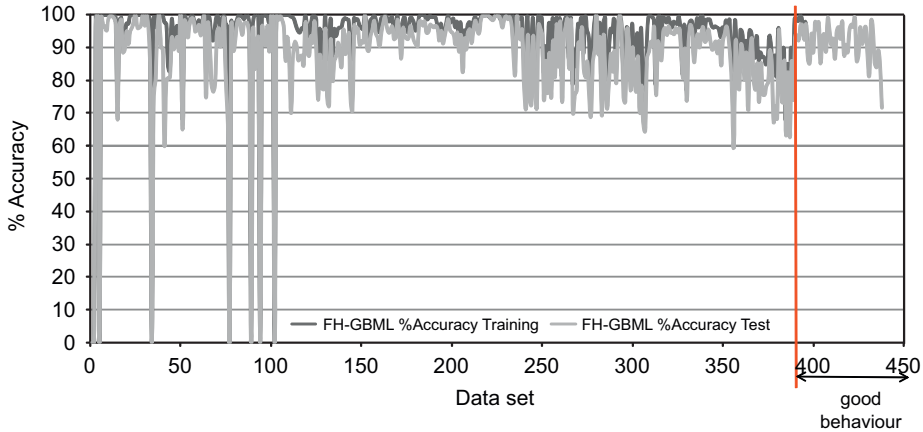

Fig. 3. Accuracy in training/test sorted by F2.

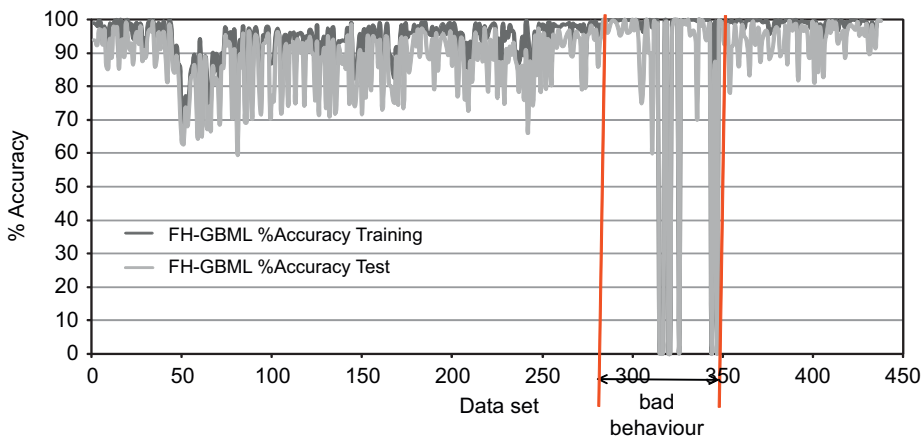

Fig. 4. Accuracy in training/test sorted by F3.

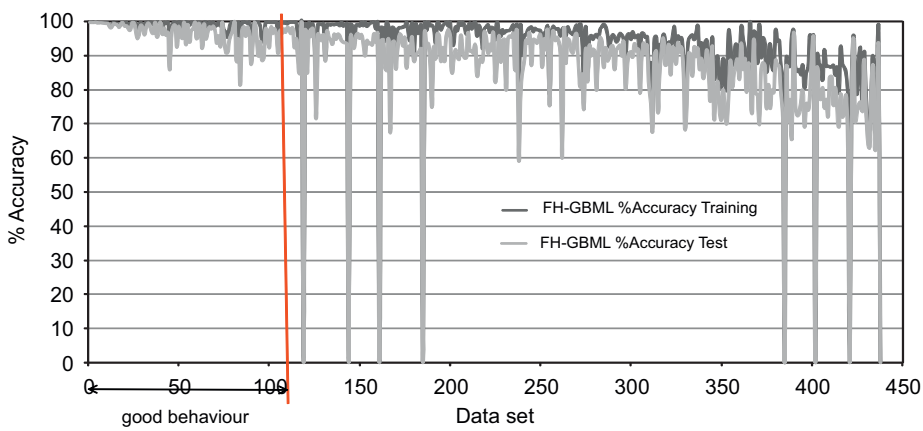

Fig. 5. Accuracy in training/test sorted by N2.

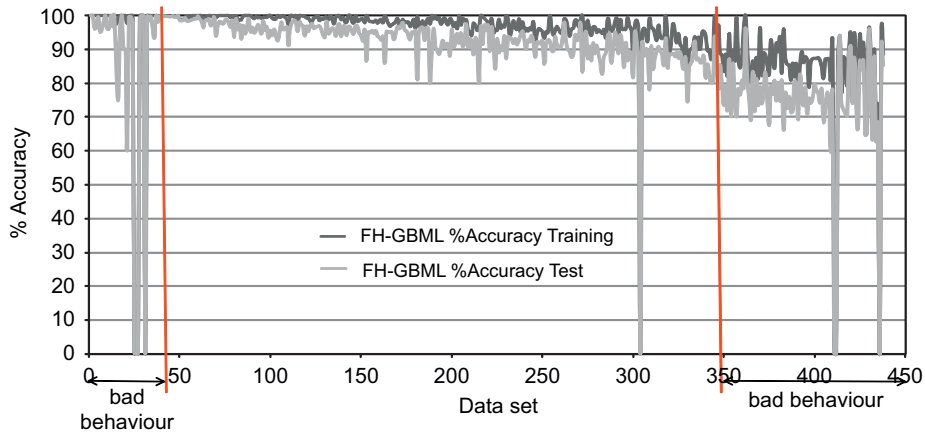

Fig. 6. Accuracy in training/test sorted by N3.

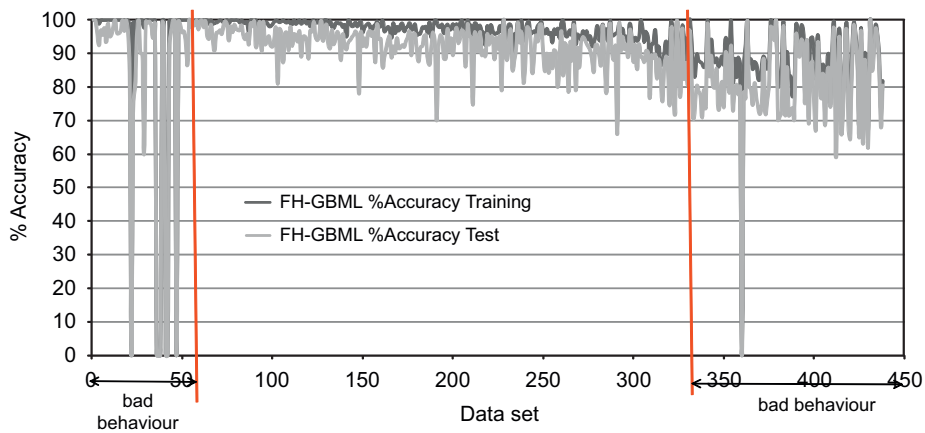

Fig. 7. Accuracy in training/test sorted by N4.

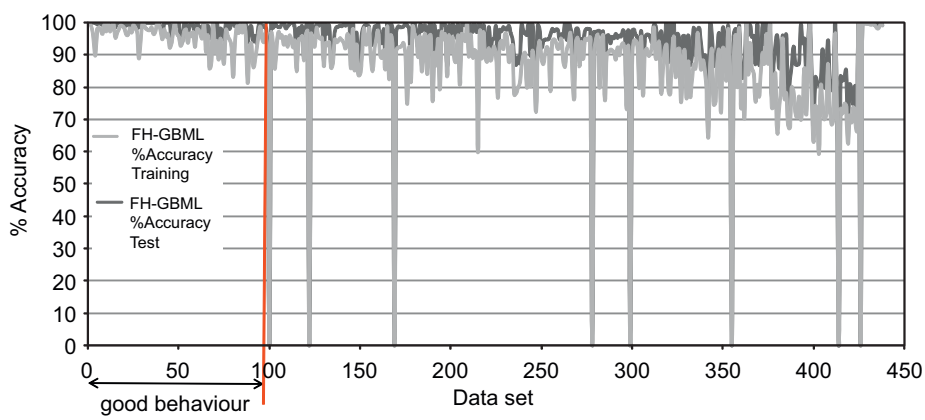

Fig. 8. Accuracy in training/test sorted by L1.

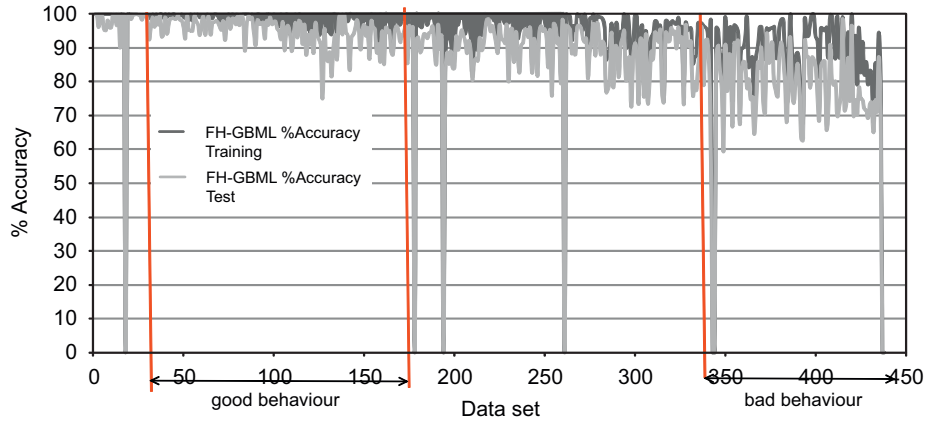

Fig. 9. Accuracy in training/test sorted by L2.

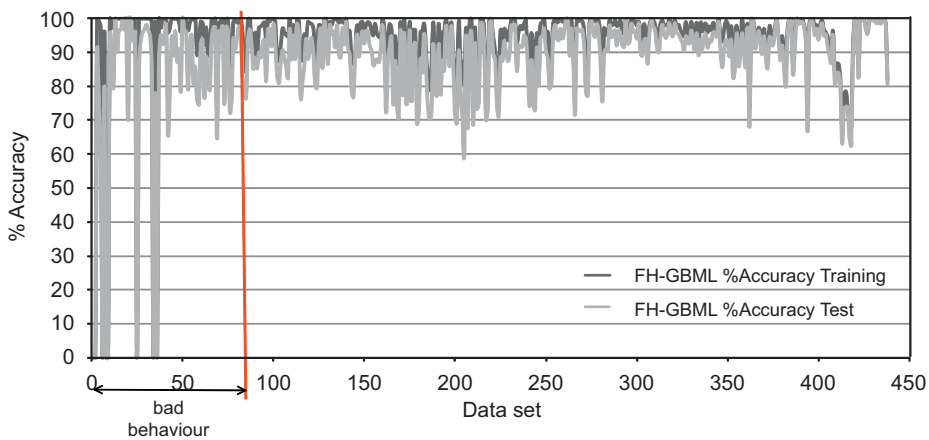

Fig. 10. Accuracy in training/test sorted by T2.

Table 2
Significant intervals.

Interval	FH-GBML behaviour
$N2 < 0.23$	*good behaviour*
$L1 < 0.2231$	*good behaviour*
$F2 = 1$	*good behaviour*
$L2 < 0.125$	*good behaviour*
$N3 = 0$	*bad behaviour*
$N3 > 0.1631$	*bad behaviour*
$N4 = 0$	*bad behaviour*
$N4 > 0.1743$	*bad behaviour*
$1 < F3 < 1.83$	*bad behaviour*
$L2 > 0.2834$	*bad behaviour*
$T2 < 12.29$	*bad behaviour*

- The column "% training, Std. Dev." shows the average accuracy in training of all the examples which are covered by the rule. The standard deviation of the average training accuracy is computed as well.
- The column "Training diff." contains the difference between the training accuracy of the rule and the training accuracy across all 438 data sets.

Table 3
One metric rules with obtained from the intervals.

Id.	Rule	Support (%)	% training, Std. Dev.	Training diff. (%)	% test, Std. Dev.	Test diff. (%)
R1+	If $N2[X] < 0.23$ then *good behaviour*	25.342	99.283, 1.340	5.713	96.854, 3.371	8.612
R2+	If $L1[X] < 0.2231$ then *good behaviour*	22.603	98.764, 1.868	5.195	95.754, 3.868	7.512
R3+	If $F2[X] = 1$ then *good behaviour*	11.187	96.060, 4.050	2.490	91.829, 5.475	3.588
R4+	If $L2[X] < 0.125$ then *good behaviour*	35.616	98.388, 2.271	4.818	95.094, 4.176	6.852
R1−	If $1 <$ F3[X] < 1.83 then *bad behaviour*	16.210	88.480, 31.537	−5.090	84.305, 30.937	−3.937
R2−	If $N3[X] = 0$ then *bad behaviour*	8.676	89.325, 30.643	−4.244	85.460, 30.272	−2.782
R3−	If $N3[X] > 0.1631$ then *bad behaviour*	21.005	83.531, 16.633	−10.038	74.521, 15.463	−13.721
R4−	If $N4[X] = 0$ then *bad behaviour*	12.557	87.083, 33.262	−6.487	82.941, 32.390	−5.301
R5−	If $N4[X] > 0.1743$ then *bad behaviour*	24.201	87.250, 11.239	−6.319	80.741, 12.707	−7.501
R6−	If $L2[X] > 0.2834$ then *bad behaviour*	22.146	85.917, 19.422	−7.652	76.114, 18.050	−12.128
R7−	If $T2[X] < 12.29$ then *bad behaviour*	17.580	87.7356, 30.143	−5.834	79.431, 28.609	−8.811

Table 4
Average FH-GBML training and test accuracy.

FH-GBML global % accuracy training	93.56955
FH-GBML global % accuracy test	88.24174

- The column "% test, Std. Dev." shows the average accuracy in test of all the examples which are covered by the rule. The standard deviation of the average test accuracy is computed as well.
- The column "Test diff." contains the difference between the test accuracy of the rule and the test accuracy across all 438 data sets.

In Table 4 we show the global training and test accuracy obtained by the FH-GBML method across all 438 data sets.

As we can see in Table 3, the positive rules (denoted with a "+" symbol in their identifier) always show a positive difference with the global average, both in training and test accuracy. The negative ones (with a "−" symbol in their identifier) verify the opposite case. The support of the rules shows us that we can characterize a wide range of data sets and obtain significant differences in accuracy, as we can see from rules R1+, R2+, R4+, R3− and R5−. Notice also that the standard deviations for the positive rules are always lower than those for the negative ones. The robustness of the method is higher in those data sets which have been covered by a positive rule. FH-GBML performs worse in the data sets under a negative rule, due usually to method's over-fitting.

From this set of rules we can state that:

- FH-GBML performs well in those data sets in which the examples of the same class lay closely in the feature space, since a low N2 offers *good* results.
- When the problem is linear or almost linearly separable, we obtain *good* results, as can be drawn from low values of L1.

Table 5
Disjunction and intersection rules from all simple rules.

Id.	Rule	Support (%)	% training, Std. Dev.	Training diff. (%)	% test, Std. Dev.	Test diff.
PRD	If R1+ or R2+ or R3+ or R4+ *good behaviour*	42.694	98.360, 3.241	4.791	95.170, 6.213	6.929
NRD	If R1− or R2− or R3− or R4− or R5− or R6− or R7− *bad behaviour*	55.479	90.325, 18.168	−3.245	83.864, 18.46784	−4.378
PRD∧ NRD	If PRD and NRD then *good behaviour*	22.602	98.319, 3.316	4.750	95.048, 5.424	6.806
PRD∧¬NRD	If PRD and not NRD then *good behaviour*	20.091	98.406, 1.837	4.837	95.308, 3.455	7.067
NRD∧¬PRD	If NRD and not PRD then *bad behaviour*	32.877	84.829, 21.801	−8.741	76.175, 20.254	−12.066
Not characterized	If not PRD and not (NRD and not PRD) then *good behaviour*	24.429	96.960, 2.097	3.391	92.372, 3.036	4.130

- A low error rate of the LP classifier in the classification, defined by L1 and measured by L2, results in an *acceptable good* behaviour of FH-GBML. On the other hand, when the error rate of the LP classifier grows sufficiently, the behaviour of FH-GBML becomes *bad* with a difference of −12.13%.
- When the volume of overlap (F2) is 1, the FH-GBML obtains *acceptable good* results with an improvement of 6.85% with respect to the global average.
- When the error obtained by the 1-NN method (N3) is 0, the FH-GBML method usually does not perform well as well as when the nonlinearity of the 1-NN method (N4) is 0. With these conditions, FH-GBML obtains a *relative bad* behaviour with differences of −2.78% and −5.30%, respectively.
- The parallel behaviour of N3 and N4 measures is present when they have high values as well. In particular when the error obtained by the 1-NN method (N3) is above 0.1631 and when the non-linearity of the 1-NN method (N4) is above 0.1743. With these conditions, FH-GBML obtains a *bad* behaviour with differences of −13.72% and −7.50%, respectively.
- An interesting fact is that a ratio between the number of examples and the number of attributes (T2) lower than 12.29 generally results in *bad* performance.

Although we have obtained some interesting rules, we can extend our study by considering the combination of these complexity metrics in order to obtain more precise and descriptive rules.

5.2. Collective evaluation of the set of rules

The objective of this section is to jointly analyse the good rules, as well as the bad rules. Thus we can arrive at a more general description, with a wider support, of the behaviour of the FH-GBML method with these joint rules. We perform the disjunctive combination of all the positive rules to obtain a single rule. The same is done with all the negative ones, so we obtain another rule. The new disjunctive rule will be activated if any of the component rules' antecedents are verified.

Since the support of the joint rules will be high, we also compute the intersection of these disjunctive rules (the data sets which activate both disjunctive rules). With the intersection of the disjunction, we try to see the global relations and competence between positive and negative intervals.

Thus we obtain three different kinds of intersections:

- Intersection of positive disjunction and *not* the negative disjunction.
- Intersection of positive disjunction and the negative disjunction.
- Intersection of negative disjunction and *not* the positive disjunction.

In Table 5 we summarize both disjunctions and the three intersections.

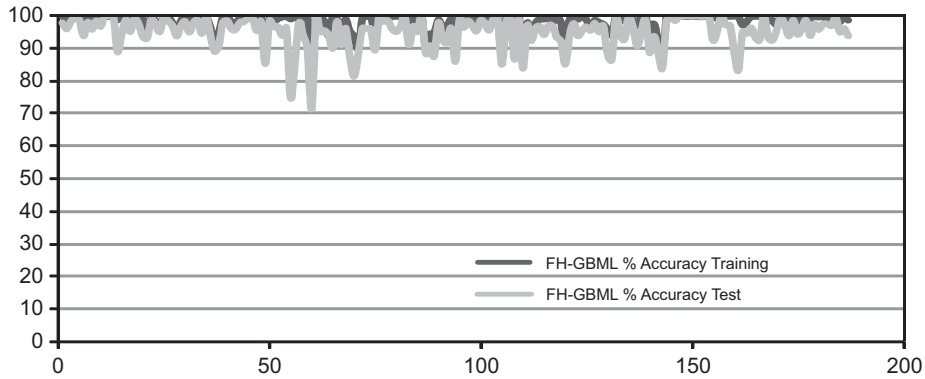

Fig. 11. Accuracy results for data sets covered by PRD.

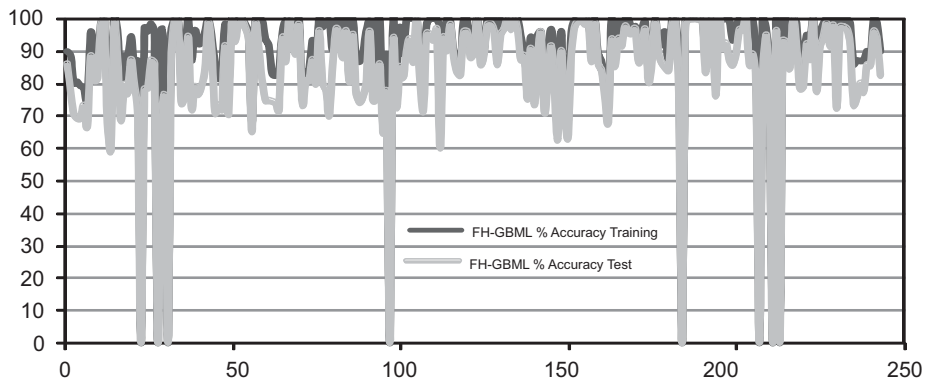

Fig. 12. Accuracy results for data sets covered by NRD.

From the new obtained rules, we can point out that:

- The positive rule disjunction (PRD) offers a high support (almost the half of the considered data sets), and it gives a good test accuracy (over the 95%). In spite of its wide support, it presents a low standard deviation of 6.21. Fig. 11 shows FH-GBML's results for data sets covered by this rule.
- The negative rule disjunction (NRD) obtains a wide support as well (over the 50%). However, it is not very accurate in indicating the data sets with low FH-GBML methods's performance as we can see from its high standard deviation and low difference. Fig. 12 shows FH-GBML's results for the data sets included in the support of this rule.
- The positive and negative rule disjunction (PRD$\wedge$NRD) is more specific than PRD in isolation. However, it presents a similar standard deviation. It is also similar to PRD in the training and test accuracy difference. Fig. 13 shows the FH-GBML accuracy results of the data sets covered by this latter rule. The Positive and Not Negative Rule Disjunction (PRD$\wedge\neg$NRD) has a lower support than PRD$\wedge$NRD and a lower standard deviation, but its difference is somewhat higher, since the data sets with low accuracy for FH-GBML have been removed by PRD$\wedge$NRD. Fig. 14 shows the accuracy results of FH-GBML for the data sets covered by this rule.
- The negative and not positive rule disjunction (NRD$\wedge\neg$PRD) is a good rule to describe the bad behaviour of FH-GBML. It has a decent support and both a high difference in training and test sets. Fig. 15 shows the FH-GBML accuracy results for the data sets in the support of this rule.

From all these new rules, we can present PRD as a representative description of good data sets, and NRD$\wedge\neg$PRD as a representative description for bad data sets, when using FH-GBML. We can consider three blocks of data sets with

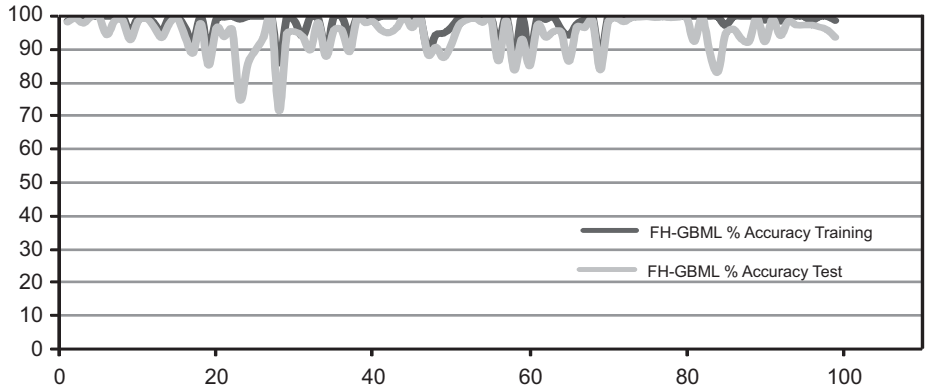

Fig. 13. Accuracy results for data sets covered by PRD∧NRD.

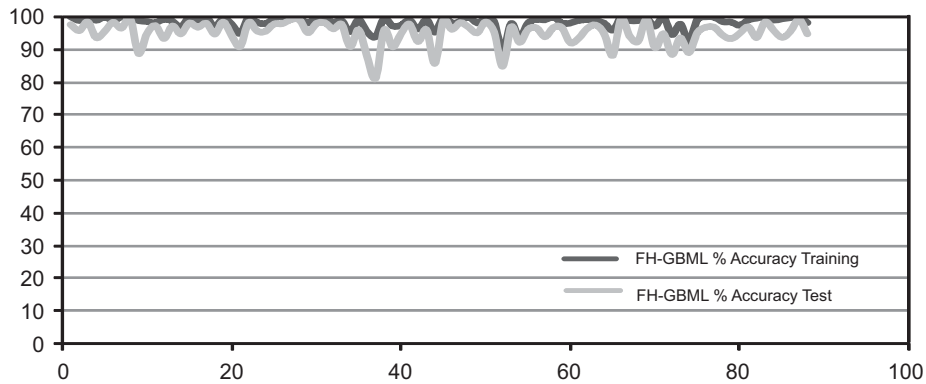

Fig. 14. Accuracy results for data sets covered by PRD∧¬NRD.

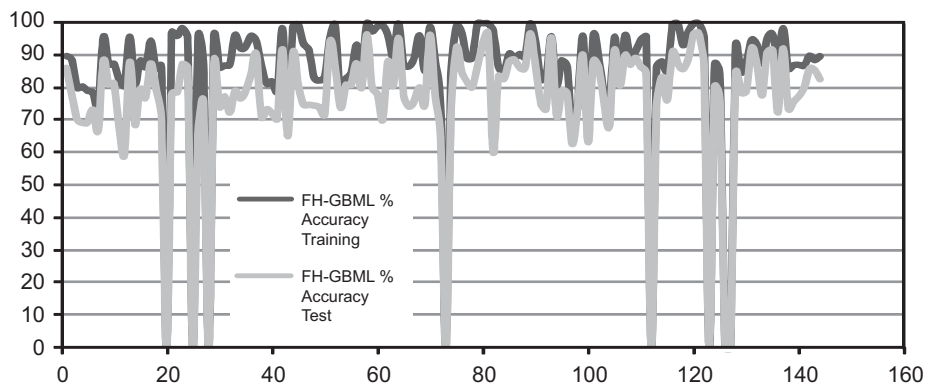

Fig. 15. Accuracy results for data sets covered by NRD∧¬PRD.

their respective support, as depicted in Fig. 17 (with no particular data set order within each block):

- The first block (the left-side one) represents those data sets covered by the PRD rule. They are the data sets recognized as being those in which FH-GBML has good accuracy.
- The second block (the middle one) plots the data sets for the rule NRD∧¬PRD, which are bad data sets for FH-GBML.

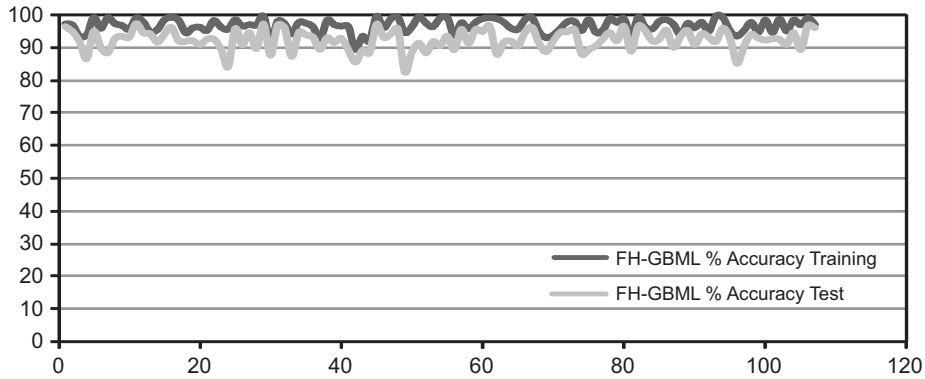

Fig. 16. Accuracy results for data sets not covered either by PRD and NRD∧¬PRD.

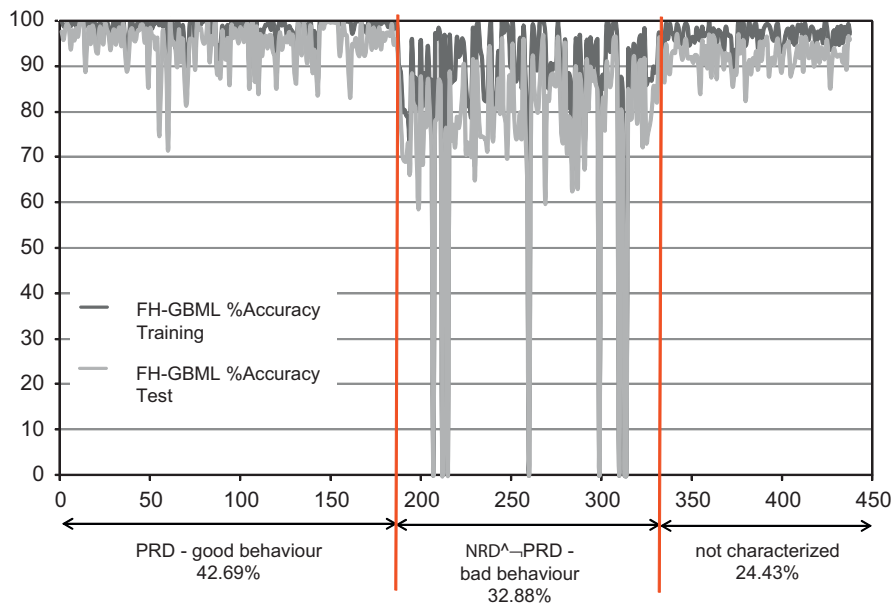

Fig. 17. Three blocks representation for PRD, NRD∧¬PRD and not covered data sets.

- The third and last block (the right-side one) contains the unclassified data sets by the previous two rules. This uncovered bunch of data sets is represented in the last row of Table 5. Fig. 16 also depicts these data sets.

We can see that the 75% of the analysed data sets are covered by these two rules, and hence the *good behaviour* and *bad behaviour* consequents properly represent the accuracy obtained by the FH-GBML method.

6. Conclusions

We have performed a study over a set of binary data sets with the FH-GBML method. We have computed some data complexity measures for the data sets in order to obtain intervals of such metrics in which the method's performance is significantly good or bad. We have constructed descriptive rules as well as studied the interaction between the intervals and the proper rules.

We have obtained two rules which are simple, interpretable and precise to describe both good and bad performance of the FH-GBML method. Furthermore, we present the possibility of determining which data sets would be prove to FH-GBML to perform well or badly prior to their execution, using the Data Complexity measures.

We must point out that this is a particular study for one specific method, the FH-GBML. On the other hand, this work presents a new challenge that could be extended to other FRBCSs, to analyse their domains of competence, and to develop new measures which could provide us with more information on the behaviour of FRBCSs for pattern recognition.

References

[1] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2007 ⟨http://www.ics.uci.edu/∼mlearn/MLRepository.html⟩.

[2] R. Baumgartner, R.L. Somorjai, Data complexity assessment in undersampled classification, Pattern Recognition Letters 27 (2006) 1383–1389.

[3] E. Bernadó-Mansilla, T.K. Ho, Domain of competence of XCS classifier system in complexity measurement space, IEEE Transactions on Evolutionary Computation 9 (1) (2005) 82–104.

[4] B. Carse, A.G. Pipe, Introduction: genetic fuzzy systems, International Journal of Intelligent Systems 22 (9) (2007) 905–907.

[5] J. Casillas, F. Herrera, R. Pérez, M.J. del Jesus, P. Villar, Special issue on genetic fuzzy systems and the interpretability—accuracy trade-off—editorial, International Journal of Approximate Reasoning 44 (1) (2007) 1–3.

[6] J. Casillas, B. Carse, Preface: genetic fuzzy systems: recent developments and future directions, Special issue on genetic fuzzy systems: recent developments and future directions, Soft Computing 13 (2009) 417–418.

[7] 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.

[8] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (1) (2004) 5–31.

[9] O. Cordón, R. Alcalá, J. Alcalá-Fdez, I. Rojas, Genetic fuzzy systems, Special issue on genetic fuzzy systems: What's next?—editorial, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 533–535.

[10] M. Dong, R. Kothari, Feature subset selection using a new definition of classificability, Pattern Recognition Letters 24 (2003) 1215–1225.

[11] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computation, Springer, Berlin, 2003.

[12] F. Herrera, Genetic fuzzy systems: taxonomy, current research trends and prospects, Evolutionary Intelligence 1 (2008) 27–46.

[13] T.K. Ho, H.S. Baird, Pattern classification with compact distribution maps, Computer Vision and Image Understanding 70 (1) (1998) 101–110.

[14] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 289–300.

[15] M. Basu, T.K. Ho, Data Complexity in Pattern Recognition, Springer, Berlin, 2006.

[16] A. Hoekstra, R.P.W, Duin, On the nonlinearity of pattern classifiers, in: Proc. 13th Internat. Conf. on Pattern Recognition, Vienna, 1996, pp. 271–275.

[17] H. Ishibuchi, T. Nakashima, M. Nii, Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining, Springer, Berlin, 2004.

[18] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification problems, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 35 (2) (2005) 359–365.

[19] L. Kuncheva L, Fuzzy Classifier Design, Springer, Berlin, 2000.

[20] Y. Li, M. Dong, R. Kothari, Classificability-based omnivariate decision trees, IEEE Transactions on Neural Networks 16 (6) (2005) 1547–1560.

[21] R.A. Mollineda, J.S. Sánchez, J.M. Sotoca, Data characterization for effective prototype selection, in: First Edition of the Iberian Conf. on Pattern Recognition and Image Analysis (IbPRIA 2005), Lecture Notes in Computer Science, vol. 3523, Springer, Berlin, 2005, pp. 27–34.

[22] J.S. Sánchez, R.A. Mollineda, J.M. Sotoca, An analysis of how training data complexity affects the nearest neighbor classifiers, Pattern Analysis and Applications 10 (3) (2007) 189–201.

[23] S. Singh, Multiresolution estimates of classification complexity, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1534–1539.

[24] F.W. Smith, Pattern classifier design by linear programming, IEEE Transactions on Computers 17 (4) (1968) 367–372.