

# A First Study on the Noise Impact in Classes for Fuzzy Rule Based Classification Systems

José A. Sáez, Julián Luengo and Francisco Herrera  
Dept. of Computer Science and Artificial Intelligence  
CITIC - University of Granada, 18071 Granada, Spain  
{smja, julianlm, herrera}@decsai.ugr.es

**Abstract**—The presence of noise is common in any real data set and may adversely affect the accuracy, construction time and complexity of the classifiers. Models built by Fuzzy Rule Based Classification Systems are recognised for their interpretability, but traditionally these methods have not considered the presence of noise in the data, so it would be interesting to quantify its effect on them.

The aim of this contribution is to study the behavior and robustness of Fuzzy Rule Based Classification Systems in presence of noise. In order to do this, 69 synthetic data sets have been created from 23 data sets from the UCI repository. Different levels of noise have been introduced artificially in the class in order to analyze the FRBCSs when noise is present. The methods of Chi et al. and PDFC have been considered as a case study, analyzing the accuracy of the models created. From the results obtained, it is possible to deduce that Fuzzy Rule Based Classification Systems have a good tolerance to class noise.

## I. INTRODUCTION

The classification task [1] consists of making generalizations from a set of training examples. The knowledge learned from them can be applied on a set of unobserved examples to predict their classes. The primary goal of a classification algorithm is to perform at the same level as human experts. Classification algorithms can provide many benefits to an organization such as reducing decision making time, improving the consistency of decisions, and reducing dependence on scarce human experts. Successful applications of classification in business decision making have been reported for fault diagnosis in semiconductor manufacturing, loan approval, bank failure prediction, and industry and occupation code prediction as shown in [2].

Among the most important factors to determine the accuracy of a classifier are the quality of the training data and the inductive ability of the learning algorithm. Thus, given a particular learning algorithm, its classification accuracy depends crucially on the quality of training data, which is determined by a large number of components [3], [4]. Some are the source of that data comes from and the input of data, which are inherently subject to error. Although efforts have been carried out in order to solve this problem, errors in large data sets are common and can be serious. Unless exhaustive measures are taken to avoid them, error rates can reach 5% or more [5]–[7]. Thus, real data sets rarely lack of noise and they usually have corruptions that can affect the interpretations, decisions taken and the models created from the data, as well as the system performance.

While a description of the complete taxonomy of data noise is an open research issue, there are generally two types of noise in a given data set [8]:

- Class noise or labeling errors occur when an instance belongs to the incorrect class. Class noise can be attributed to several causes, including subjectivity during the labeling process, data entry errors, or the absence of some representative attributes.
- Attribute noise, in contrast, reflects erroneous values for one or more attributes of the data set.

We focus only on class noise because this type of noise is most frequent in real data sets and has been most studied in the literature [9].

Fuzzy Rule Based Classification Systems (FRBCSs) [10], [11] stand out for being capable of building a linguistic model interpretable by humans. Although they have been widely studied in the literature [11], the effects of noise in the results have not been addressed yet.

The aim of this study is to analyze the behavior of FRBCSs with noisy training data, based on the accuracy of the classifier built. In order to achieve our objective, we will analyze the influence of class noise in the performance and robustness of FRBCSs. In order to do this, we will consider 23 data sets from the UCI repository [12]. Three levels of noise will be introduced in class, 5%, 10% and 20%, to create 69 new synthetic data sets. We will consider Chi et al. [13] and PDFC [14] FRBCSs with different configurations. For the Chi et al. algorithm, the number of labels will be varied. For PDFC, different type of labels will be considered, paying attention to how they influence the tolerance to noise. Finally, we will analyze the impact of noise on the performance of each system.

The rest of this contribution is organized as follows. In Section II, we describe the used FRBCSs: the Chi et al. algorithm and the PDFC method. Then, in Section III, we develop the description of the noise considered in this study. In Section IV we describe the experimental framework, where we indicate the used data sets, the class noise introduction schema and the parameters configuration for the employed algorithms in the experimentation. We present the analysis of the results obtained in Section V. In Section VI, we point out our conclusions on studied FRBCSs when dealing with noisy data. Finally in Appendix A the result tables of test accuracy for Chi et al. and PDFC are depicted.

## II. FUZZY RULE BASED CLASSIFICATION SYSTEMS

In this section we introduce the FRBCSs used in this work: the rule generation method of Chi et al. in Subsection II-A and the PDFC algorithm in Subsection II-B.

FRBCSs are widely employed due to their capability to build a linguistic model interpretable to the users with the possibility of mixing different information such as that coming from expert knowledge and information coming from mathematical models or empiric measures. New FRBCS models have been proposed on standard classification [15], [16] and data streams [17] among others. They have also been also applied widely including, but not limited to, the detection of intrusions, medical applications and in the imbalanced data framework [18].

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

### A. Rule Generation Method of Chi et al.

This method uses fuzzy rules as follows:

$$\text{Rule } R_j : \text{ If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_n \text{ is } A_{jn} \quad (1) \\ \text{then Class} = C_j \text{ with } RW_j$$

where  $R_j$  is the label of the  $j$ -th rule,  $x = (x_1, \dots, x_n)$  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 [19].

Fuzzy learning methods are the basis to build a FRBCS. The first algorithm discussed is the method proposed by Chi et al. [13], [20]. To generate the fuzzy rule base, this method determines the relationship between the variables of the problem and establishes an association between the space of the features and the space of the classes by means of the following steps:

- 1) *Establishment of the linguistic partitions.* Once the domain of variation of each feature  $A_i$  is determined, the fuzzy partitions are computed.
- 2) *Generation of a fuzzy rule for each example*  $x_p = (x_{p1}, \dots, x_{pn}, C_p)$ . To do this is necessary:
  - 2.1 To compute the matching degree  $\mu(x_p)$  of the example to the different fuzzy regions using a conjunction operator.
  - 2.2 To assign the example  $x_p$  to the fuzzy region with the greatest membership degree.
  - 2.3 To generate a rule for the example, with antecedent determined by the selected fuzzy region and consequent equal to the class of the example.
  - 2.4 To compute the rule weight.

Rules with the same antecedent can be generated during the learning. If they have the same class in the consequent we remove one of them, but if it is different, only the rule with the highest weight is kept in the rule base.

### B. PDFC Method (Positive Definite Fuzzy Classifier)

The PDFC algorithm [14] considers a fuzzy model with  $m$  fuzzy rules of the form:

$$\text{Rule } j : \text{ If } A_j^1 \text{ AND } A_j^2 \text{ AND } \dots \text{ AND } A_j^n \quad (2) \\ \text{THEN } b_j$$

where  $A_j^k$  is a fuzzy set with membership function  $a_j^k : \mathbb{R} \rightarrow [0, 1]$ ,  $j = 1, \dots, m$ ,  $k = 1, \dots, n$ ,  $b_j \in \mathbb{R}$ . The input output mapping,  $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}$ , of the model is defined as

$$\mathcal{F}(x_p) = \frac{b_0 + \sum_{j=1}^m b_j \prod_{k=1}^n a_j^k(x_k)}{1 + \sum_{j=1}^m \prod_{k=1}^n a_j^k(x_k)}. \quad (3)$$

where  $b_0 \in \mathbb{R}$ , the membership functions  $a_0^k(x_k) \equiv 1$  for  $k = 1, \dots, n$  and any  $x_p \in \mathbb{R}^n$ . Then, the system induces a binary fuzzy classifier,  $f$ , with decision rule

$$f(x_p) = \text{sign}(\mathcal{F}(x_p) + t) \quad (4)$$

where  $t \in \mathbb{R}$  is a threshold. We can assume  $t = 0$  without a loss of generality.

The membership functions for a binary fuzzy classifier defined above are generated from a reference function (the type of the labels)  $a^k$  through location transformation [21], and the classifiers defined in them.

The decision rule of the binary fuzzy classifier, then, can be written as:

$$f(x_p) = \text{sign} \left( \sum_{j=1}^m b_j K(x_p, z_j) + b_0 \right) \quad (5)$$

where  $z_j = [z_j^1, z_j^2, \dots, z_j^n]^T \in \mathbb{R}$  contains the location parameters of  $a_j^k$ .  $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0, 1]$  is a translation invariant kernel (Mercer Kernel [22]) defined as

$$K(x_p, z_j) = \prod_{k=1}^n a^k(x_p^k - z_j^k) \quad (6)$$

Finally, the decision rule of a binary fuzzy classifier is

$$f(x_p) = \text{sign} \left( b_0 + \sum_{j=1}^m b_j \prod_{k=1}^n a_j^k(x_p^k) \right) \quad (7)$$

In order to find the fuzzy rules from the training set, it is necessary to construct a Mercer kernel from the positive definite reference functions, as given in the Equation 6. Theorem 3.12 in [14] states that the decision rule of a PDFC can be viewed as a hyperplane in  $\mathbb{F}$ . A SVM algorithm [23] is used to find an optimal hyperplane  $\mathbb{H}$  and, once it is obtained, fuzzy rules can easily be extracted from the SVM decision rule as follows:

- $b_0 \leftarrow b$ , where  $b$  is the constant of  $\mathbb{H}$ .
- For each support vector  $x_i$ , create a fuzzy rule  $z_j$  centering the reference functions in support vector  $z_j \leftarrow x_i$ , and assign the consequent of the rule  $b_j \leftarrow y_i \alpha_i$ , where  $\alpha_i$  is the  $i$ -th Lagrange multiplier obtained by solving the quadratic programming problem of SVM, and  $y_i$  is the class of  $i$ -th support vector.

### III. PRESENCE OF CLASS NOISE IN DATA

Noise is a problem that can affect any real data set [9] and may influence the accuracy, construction time, interpretability and size of the model built. Although there are methods for filtering noise [8] they usually cannot produce data with similar characteristics to the original [24]. For this reason it is interesting to know the robustness offered by the classification methods without external treatment for incorrect data.

In data sets for classification there are two different types of noise: class noise or labeling error (when the noise affects the dependent variable) and attribute noise (when the noise affects one or more independent variables).

Class noise occurs when the class label of an instance is not appropriate, which may occur in mislabeled or contradictory examples (instances that appear more than once with different classes). There are several approaches to mitigate its effects [8], mainly based on the elimination of noise instances (which tend to improve the accuracy of the classifier, as shown in [9]) or the correction of the wrong labeled examples.

In [9], a complete study of the class noise impact was done. They concluded the existence of class noise will decrease classification accuracy, where the accuracies decline almost linearly with the increase of the noise level. Also they pointed out that removing those noisy instances will improve the classification accuracy. Even though the use of pruning techniques of the decision trees employed in this study that partially addresses the impact of class noise, class noise can still drastically affect the classification accuracy, as long as the noise exists in the training set.

In addition to the impact of class noise in classification accuracy, the research from [8] and [25] also suggests that class noise handling could shrink the size of the decision trees and save the time in training a classifier comprehensively.

In order to introduce class noise in the data sets, we adopt the scheme shown in [25] which is described next. Given the pair of classes  $(X, Y)$ , being  $X$  the majority class and  $Y$  the second majority class, and a noise level  $x\%$ , an instance with label  $X$  has a probability of  $x\%$  to be incorrectly labeled as  $Y$ . As indicated in [9], this scheme is appropriate because it is more likely that only certain types of classes are mislabeled.

### IV. EXPERIMENTAL FRAMEWORK

In this section, first we show in Subsection IV-A the data sets selected for experimentation. In Subsection IV-B, we give the details for the insertion of class noise. In Subsection IV-C, we indicate the algorithms used for the study along with the parameters used in their execution.

#### A. Data sets

Table I summarizes the properties of selected data sets for experimentation. For each data set the number of examples (#Ex.), the number of numeric attributes (#Attr.) and the number of classes (#Cl.) are presented.

Data set	#Ex.	#Attr.	#Cl.
banana	5,300	2	2
contraceptive	1,473	9	3
ecoli	336	7	8
glass	214	9	7
heart	270	13	2
ionosphere	351	33	2
iris	150	4	3
magic	19,020	10	2
monk-2	432	6	2
page-blocks	5,472	10	5
penbased	10,992	16	10
phoneme	5,404	5	2
pima	768	8	2
ring	7,400	20	2
satimage	6,435	36	7
segment	2,310	19	7
sonar	208	60	2
spambase	4,597	57	2
thyroid	7,200	21	3
twonorm	7,400	20	2
wdbc	569	30	2
wine	178	13	3
yeast	1,484	8	10

TABLE I  
SUMMARY DESCRIPTION FOR CLASSIFICATION DATA SETS.

The accuracy estimation of each classifier is obtained by means of 5-fold cross validation. We divide the data set into five partition sets with equal number of examples and maintaining the proportion between classes. Each partition set is used as test for the model learned from the four remaining partitions. We use five partitions because most of the used data sets in our experiments are small, so they would have a low number of examples in the test partition if a higher fold number is considered. Moreover, since each partition has a larger number of examples the noise effects will be more notable facilitating their analysis.

#### B. Class Noise Introduction Schema

Following the class noise introduction schema explained in Section III, from 23 data sets from UCI repository we have introduced class noise with levels  $x = 5\%$ ,  $x = 10\%$  and  $x = 20\%$  obtaining 69 new data sets with class noise.

In all created data sets noise is introduced only in the training partitions, while the test sets remain unchanged. Thus it is possible to observe how noise affects the accuracy of the classifiers when training with noisy data.

To estimate the loss of accuracy ( $L_x$ ) produced in a classifier in the presence of determined noise percentage, we use the formula presented in Equation 8:

$$L_x = \frac{Acc_{0\%} - Acc_{x\%}}{Acc_{0\%}} \quad (8)$$

where  $Acc_{x\%}$  is the mean accuracy of the classifier with a noise level of  $x\%$  in all data sets.

### C. Parameters Configuration

The algorithms have been executed with the KEEL tool <sup>1</sup> [26] with the recommended values from the authors shown in Table II.

Algorithm	Parameters
Chi et al.	Number of labels = 3, 5 and 7 T-norm for the computation of the compatibility = Product Rule weight = Penalized certainty factor Fuzzy reasoning method = Winning rule
PDFC	C = 100 d = 0.25 Tolerance = 0.001 epsilon = $10^{-12}$ Labels type = Gaussian and symmetric triangle Preprocess = Normalize in [0,1]

TABLE II  
PARAMETER SPECIFICATION FOR THE METHODS EMPLOYED IN THE EXPERIMENTATION.

For the Chi et al. algorithm different numbers of labels have been used in order to analyze its influence on tolerance to noise. For PDFC different types of labels have been employed in the experimentation to analyze how noise affects the created model, since this FRBCS adjusts the number of labels automatically.

### V. ANALYSIS OF THE INFLUENCE OF CLASS NOISE IN FRBCSS

In this section we analyze the noise robustness of the two FRBCSSs, considering two measures: the classical accuracy and the loss of accuracy described in Equation 8.

In order to complete the analysis given in this section and get more details about the concrete obtained results, we present the tables of the Appendix A. For each studied algorithm and for each configuration, we show the average of test accuracy for the 5 executions done in each data set. So the Table V, Table VI and Table VII depict the results for the Chi et al. algorithm with 3, 5, and 7 labels respectively. Table VIII shows the results for PDFC with gaussian labels and Table IX for PDFC with triangular labels. In these tables, we can observe how the average of test accuracy in each data set is affected when introducing different noise levels in classes. The name of the data set, and the average of test accuracy in each level of noise is present (a 0% noise level indicates that no-noise is introduced).

Summary description of the Appendix A is present in Table III that shows the mean accuracy in test with different levels of noise for the 92 data sets with class noise. The column (Method) shows the analyzed method, with the configuration parameter considered in brackets: in the Chi et al. algorithm correspond to the number of linguistic labels. For the PDFC algorithm, it indicates the type of labels, with (T) for the symmetric triangle function and (G) for gaussian function. The column (Mean) shows the mean result at all noise levels, and

the rest of columns, the mean accuracy with a concrete level of noise. Table IV shows the loss of accuracy for the different noise levels considered.

Method	Accuracy				
	0%	5%	10%	20%	Mean
Chi (3)	69.82	69.90	70.18	68.88	69.69
Chi (5)	66.88	66.61	66.27	64.85	66.15
Chi (7)	58.40	57.97	57.49	55.90	57.44
PDFC (G)	86.40	85.84	85.23	<b>83.40</b>	85.22
PDFC (T)	<b>88.42</b>	<b>87.31</b>	<b>86.56</b>	83.39	<b>86.42</b>

TABLE III  
TEST ACCURACY WITH CLASS NOISE AT 5%, 10% AND 20%.

Method	Loss			
	5%	10%	20%	Mean
Chi (3)	<b>0.00</b>	<b>-0.01</b>	<b>0.01</b>	<b>0.00</b>
Chi (5)	<b>0.00</b>	0.01	0.03	0.01
Chi (7)	0.01	0.02	0.04	0.02
PDFC (G)	0.01	0.01	0.03	0.02
PDFC (T)	0.01	0.02	0.06	0.03

TABLE IV  
TEST LOSS WITH CLASS NOISE AT 5%, 10% AND 20%.

Based on these experimental results with class noise, we observe that the PDFC algorithm in its two configurations works better than the Chi et al. algorithm since it achieves better accuracy results in all cases (with and without noise). The overall best method is PDFC using the triangular type of labels.

Considering the Chi et al. algorithm, a smaller number of linguistic labels provides a better noise tolerance, as shown in Table IV, where the loss is compounded by the introduction of higher noise levels. Similarly, the type of label for the PDFC algorithm also affects the method noise sensitivity. In some cases, as shown by the negative result for the algorithm Chi (3) with 10% of noise in Table IV (-0.01), some algorithms can benefit by introducing very low levels of noise, since it may happen that data considered as outliers previously now fall within the boundaries of their class when varying their characteristics.

As mentioned, the PDFC method obtains higher accuracy in average than Chi et al., but suffers of a higher absolute loss of accuracy when levels of class noise are increased. Due to PDFC obtains a model that fits the training data better, the increase in noise is more damaging to the adjustment when compares to the Chi et al. method. Despite having more loss, PDFC stills obtains higher accuracy than the Chi et al algorithm.

In both cases, the percentages of loss of the algorithms studied are very small indicating the high robustness against class noise of the FRBCSSs.

<sup>1</sup>www.keel.es

## VI. CONCLUSIONS

In this contribution we have analyzed the influence of class noise in FRBCSs. In order to do so, we have carried out an analysis using the Fuzzy Rule Based Classification Systems formally known as Chi et al. and PDFC. We have analyzed the class noise disrupting the original data sets with noise levels of  $x = 5\%$ ,  $x = 10\%$  and  $x = 20\%$ .

From the results and their analysis, it follows that a better fitted model obtained with a higher accuracy in test when training with noise-free data usually obtains a higher loss of accuracy when introducing class noise. The percentages of loss of accuracy indicate that FRBCSs are robust against class noise, obtaining a loss of accuracy lower than 6% with high noise levels.

In the case of Chi et al. the number of linguistic labels has shown its influence in the class noise sensitivity: a lower number of labels implies a higher noise tolerance. PDFC (T) with triangular labels has shown a high robustness against class noise when compared with gaussians ones.

This is a preliminary study for two FRBCSs. As future work, it would be possible to incorporate new FRBCSs to make a better generalization of their behaviors against noise, and the application of filtering techniques or the inclusion of mechanisms for dealing with noise in FRBCS itself.

### APPENDIX A

#### TEST ACCURACY TABLES FOR ALL DATA SETS

In this appendix the test accuracy tables are presented. In each row the results for the data set are depicted, and in each column the considered noise level are shown.

Data set	0%	5%	10%	20%
banana	60.32	64.64	73.77	45.47
contraceptive	40.26	39.04	40.19	38.29
ecoli	72.64	73.53	74.13	76.50
glass	57.95	53.72	48.59	48.12
heart	51.11	50.37	51.48	47.78
ionosphere	65.27	64.41	63.84	60.71
iris	92.67	92.67	92.67	92.67
magic	76.80	77.40	78.10	79.21
monk-2	41.45	39.60	38.22	36.60
page-blocks	91.92	91.79	91.79	91.85
penbased	97.71	97.70	97.69	97.59
phoneme	71.93	72.30	72.37	73.32
pima	72.53	74.09	74.48	72.00
ring	55.27	56.22	58.12	64.80
satimage	48.30	47.52	46.39	44.52
segment	85.71	85.67	85.54	83.59
sonar	59.18	57.74	55.33	57.76
spambase	71.22	72.79	74.59	81.10
thyroid	92.01	92.03	91.93	91.39
twonorm	90.54	90.74	90.36	84.70
wdbc	92.62	92.27	91.39	92.10
wine	92.67	91.54	90.41	88.73
yeast	25.74	29.92	32.75	35.44
Mean	69.82	69.90	<b>70.18</b>	68.88

TABLE V

TEST ACCURACY RESULTS FOR ALL DATA SETS EMPLOYED IN THE EXPERIMENTATION FOR CHI (3)

Data set	0%	5%	10%	20%
banana	80.15	80.30	79.08	65.92
contraceptive	30.01	29.47	29.74	29.13
ecoli	81.26	80.66	80.96	80.07
glass	58.89	57.51	54.67	52.82
heart	20.74	20.74	20.74	17.78
ionosphere	54.43	53.29	52.15	51.01
iris	96.00	96.00	96.00	96.00
magic	80.94	80.91	81.03	80.38
monk-2	0.00	0.00	0.00	0.00
page-blocks	94.12	93.91	94.21	94.06
penbased	95.48	95.46	95.38	95.27
phoneme	78.55	78.55	79.52	78.83
pima	70.17	70.96	69.39	67.58
ring	64.28	65.11	66.26	68.46
satimage	52.03	51.90	52.17	52.93
segment	89.70	89.70	90.00	89.74
sonar	14.48	14.01	11.58	13.03
spambase	81.05	81.34	82.42	82.16
thyroid	92.01	92.00	91.83	91.32
twonorm	90.31	89.39	88.35	85.07
wdbc	84.18	83.48	82.25	79.44
wine	73.02	72.44	71.90	70.78
yeast	56.33	54.92	54.51	49.80
Mean	<b>66.88</b>	66.61	66.27	64.85

TABLE VI

TEST ACCURACY RESULTS FOR ALL DATA SETS EMPLOYED IN THE EXPERIMENTATION FOR CHI (5)

Data set	0%	5%	10%	20%
banana	87.45	86.87	86.11	80.43
contraceptive	28.78	27.97	27.97	28.04
ecoli	68.47	67.58	68.17	66.38
glass	59.83	59.35	57.00	52.80
heart	7.04	7.04	7.04	4.44
ionosphere	42.73	40.74	38.46	39.32
iris	88.67	88.67	88.67	88.67
magic	81.47	81.38	81.27	80.93
monk-2	0.00	0.00	0.00	0.00
page-blocks	94.12	93.97	94.12	93.99
penbased	82.81	82.73	82.46	82.07
phoneme	82.07	81.90	81.85	80.85
pima	62.76	62.11	60.15	58.20
ring	52.53	52.34	52.22	51.22
satimage	78.59	77.81	77.26	75.80
segment	91.30	91.17	91.34	91.13
sonar	2.93	2.93	2.93	2.93
spambase	77.70	77.64	77.31	75.61
thyroid	91.39	91.31	91.11	90.13
twonorm	23.57	23.03	22.53	21.22
wdbc	63.98	62.75	61.16	52.91
wine	18.44	18.44	17.87	17.87
yeast	56.67	55.53	55.26	50.88
Mean	<b>58.40</b>	57.97	57.49	55.90

TABLE VII

TEST ACCURACY RESULTS FOR ALL DATA SETS EMPLOYED IN THE EXPERIMENTATION FOR CHI (7)

Data set	0%	5%	10%	20%
banana	68.68	66.17	64.30	58.13
contraceptive	54.24	54.17	53.63	50.17
ecoli	83.05	83.64	84.24	82.45
glass	68.26	66.87	63.61	55.65
heart	80.74	79.63	78.15	72.59
ionosphere	93.45	90.33	92.88	90.59
iris	95.33	95.33	95.33	94.67
magic	86.15	86.09	86.00	85.62
monk-2	97.22	96.30	96.99	94.66
page-blocks	95.87	95.36	95.36	95.34
penbased	99.50	99.47	99.46	99.40
phoneme	80.13	79.74	79.37	78.92
pima	76.56	76.95	77.21	73.57
ring	97.99	97.84	97.81	97.36
satimage	87.40	87.21	87.30	87.24
segment	95.84	95.80	95.80	95.76
sonar	89.38	87.97	82.17	83.19
spambase	93.93	93.30	92.84	92.06
thyroid	94.86	94.49	94.29	93.89
twonorm	97.65	97.35	97.61	97.27
wdbc	97.36	97.54	97.19	97.19
wine	97.17	96.62	95.46	92.67
yeast	56.53	56.06	53.30	49.73
Mean	<b>86.40</b>	85.84	85.23	83.40

TABLE VIII  
TEST ACCURACY RESULTS FOR ALL DATA SETS EMPLOYED IN THE  
EXPERIMENTATION FOR PDFC (G)

Data set	0%	5%	10%	20%
banana	89.72	89.40	89.06	88.34
contraceptive	51.46	49.42	49.89	48.47
ecoli	81.85	80.95	77.99	73.52
glass	77.57	76.19	73.39	73.38
heart	82.22	79.26	82.22	72.22
ionosphere	95.45	92.89	93.45	88.62
iris	94.00	92.67	92.67	88.67
magic	87.66	87.27	87.13	84.94
monk-2	100.00	97.22	94.45	90.97
page-blocks	97.09	97.00	96.82	96.29
penbased	99.57	99.34	99.03	97.90
phoneme	87.27	87.51	86.97	84.84
pima	70.57	69.40	67.71	66.27
ring	97.43	95.81	94.05	89.95
satimage	91.08	90.55	89.54	86.76
segment	98.57	98.27	97.58	96.19
sonar	87.47	86.99	86.02	81.27
spambase	94.95	92.78	91.56	87.71
thyroid	99.26	98.67	98.17	95.96
twonorm	97.46	96.55	95.57	91.18
wdbc	98.24	96.49	95.26	88.05
wine	98.30	97.19	96.06	92.11
yeast	56.40	56.33	56.40	54.38
Mean	<b>88.42</b>	87.31	86.56	83.39

TABLE IX  
TEST ACCURACY RESULTS FOR ALL DATA SETS EMPLOYED IN THE  
EXPERIMENTATION FOR PDFC (T)

#### ACKNOWLEDGMENTS

This work is supported by the Spanish Ministry of Science and Technology under Projects TIN2007-65981 and TIN2008-06681-C06-01. J. Luengo is funded by a FPU scholarship from the Ministry of Science and Innovation.

#### REFERENCES

- [1] C. M. Bishop, *Pattern Recognition And Machine Learning (Information Science And Statistics)*. Springer, 2007.
- [2] M. Mannino, Y. Yang, and Y. Ryu, "Classification algorithm sensitivity to training data with non representative attribute noise," *Decision Support Systems*, vol. 46, p. 743–751, 2009.
- [3] R. Wang, V. Storey, and C. Firth, "A framework for analysis of data quality research," *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 4, p. 623–639, 1995.
- [4] R. Wang, D. Strong, and L. Guarascio, "Beyond accuracy: What data quality means to data consumers," *Journal of Management Information Systems*, vol. 12, no. 4, pp. 5–34, 1996.
- [5] X. Wu, *Knowledge Acquisition from Databases*. Ablex Publishing Corp., 1995.
- [6] K. Orr, "Data quality and systems theory," *Communications of the ACM*, vol. 41, no. 2, p. 66–71, 1998.
- [7] J. Maletic and A. Marcus, "Data cleansing: Beyond integrity analysis," *Proceedings of the Conference on Information Quality*, 2000.
- [8] C. Brodley and M. Friedl, "Identifying mislabeled training data," *Journal of Artificial Intelligence Research*, vol. 11, pp. 131–167, 1999.
- [9] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial Intelligence Review*, vol. 22, no. 3, pp. 177–210, 2004.
- [10] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining*. Springer-Verlag, 2004.
- [11] L. Kuncheva, *Fuzzy classifier design*. Springer-Verlag, 2000.
- [12] A. Asuncion and D. Newman, *UCI Machine Learning Repository - http://mlr.cs.umass.edu/ml/index.html*, 2007.
- [13] Z. Chi, H. Yan, and T. Pham, *Fuzzy algorithms with applications to image processing and pattern recognition*. World Scientific, 1996.
- [14] Y. Chen and J. Z. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 716–728, 2003.
- [15] J. Hühn and E. Hüllermeier, "Fr3: A fuzzy rule learner for inducing reliable classifiers," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, p. 138–149, 2009.
- [16] E. G. Mansoori, M. J. Zolghadri, and S. D. Katebi, "Sgerd: A steady-state genetic algorithm for extracting fuzzy classification rules from data," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, p. 1061–1071, 2008.
- [17] P. P. Angelov and X. Zhou, "Evolving fuzzy-rule-based classifiers from data streams," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, p. 1462–1475, 2008.
- [18] A. Fernández, M. del Jesus, and F. Herrera, "On the influence of an adaptive inference system in fuzzy rule based classification systems for imbalanced data-sets," *Expert Systems with Applications*, vol. 36, p. 9805–9812, 2009.
- [19] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 506–515, 2001.
- [20] L. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *In Proceedings of the 1991 IEEE International Symposium on Intelligent Control, Arlington, Virginia, U.S.A.*, pp. 263–268, 1991.
- [21] D. Dubois and H. Prade, "Operations on fuzzy numbers," *International Journal of Systems Science*, vol. 9, no. 6, pp. 613–626, 1978.
- [22] N. Cristianini and J. Shawe-Taylor, *An introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [23] V. Vapnik, *Statistical Learning Theory*. New York, U.S.A.: Wiley, 1998.
- [24] X. Wu and X. Zhu, "Mining with noise knowledge: Error-aware data mining," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 38, no. 4, pp. 917–932, 2008.
- [25] X. Zhu, X. Wu, and S. Chen, "Eliminating class noise in large datasets," *Proceedings of the 20th ICML International Conference on Machine Learning, Washington D.C.*, p. 920–927, 2003.
- [26] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "Keel: a software tool to assess evolutionary algorithms form data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2008.