# Equalizing imbalanced imprecise datasets for genetic fuzzy classifiers

**Ana M. Palacios** [1], **Luciano Sánchez** [1], **Inés Couso** [2]

[1] *Departamento de Informática, Universidad de Oviedo*
*E-33203 Gijón, Asturias, Spain*
*E-mail: [palaciosana,luciano]@uniovi.es*

[2] *Departamento de Estadística e I.O. y D.M, Universidad de Oviedo*
*E-33203 Gijón, Asturias, Spain*
*E-mail: couso@uniovi.es*

**Abstract**

Determining whether an imprecise dataset is imbalanced is not immediate. The vagueness in the data causes that the prior probabilities of the classes are not precisely known, and therefore the degree of imbalance can also be uncertain. In this paper we propose suitable extensions of different resampling algorithms that can be applied to interval valued, multi-labelled data. By means of these extended preprocessing algorithms, certain classification systems designed for minimizing the fraction of misclassifications are able to produce knowledge bases that are also adequate under common metrics for imbalanced classification.

*Keywords:* Genetic Fuzzy Systems, Interval Valued Data, Imbalanced Classification, Low Quality Data

## 1. Introduction

The prevalent criterion for measuring the quality of a classifier is the expected misclassification rate. As it is widely known, the optimal solution of this problem is the so called minimum error Bayes rule[2], which is defined in terms of the conditional probabilities of the classes. Nonetheless, a system obtaining the minimum error rate is not well suited to certain tasks[21,34]. Common examples include medical diagnosis or fraud detection, where the false negatives do not incur the same cost as false positives.

Imbalanced classification is arguably a particular case of this cost sensitive classification[14,13]. These problems are characterized because the prior probabilities of the classes are much different among them. Given that the minimum error Bayes rule does not necessarily equalize the misclassification rate for the different classes, it might happen that the false negative rate of the best classification system (in terms of the global error) is so high that it cannot be assumed. The methods of choice for these last problems consist in finding the classifiers that optimize a cost function different than the expected error rate, as done for instance with the the minimum risk Bayes rule[2] or resampling the available data for equalizing the prior probabilities of the classes[39].

Genetic Fuzzy Systems (GFSs) are not an exception to this. For using a GFS with an imbalanced dataset, the standard fitness function has to be altered, or else changes must be effected on the dataset for raising the importance of misclassifying objects of the minority class. Both techniques have been well studied in the context of GFSs: there are works that deal with the use of costs in fuzzy classifiers for imbalanced datasets[11,38,41,42,46], and other studies suggest employing a preprocessing step in order to balance the training data before the training[1,16,17,18,19]. In this last respect, we can highlight the re-sampling procedure named "Synthetic Minority Oversampling Technique" or SMOTE[4]; it has been shown that SMOTE is one the most efficient preprocessing algorithms for imbalanced data in relation with Fuzzy Rule Based Systems (FRBS)[16].

Notwithstanding, the learning of Fuzzy Rule-based Systems (FRBSs) from datasets that are both imprecisely perceived and imbalanced has not yet been addressed from the perspective of the preprocessing of the training data. In this paper we are chiefly interested in mechanisms for preprocessing these low quality imbalanced dataset and in studying the properties of GFSs applied to imprecise data that has been rebalanced. In this regard, observe that it is easy to estimate the prior probabilities of the classes from the training data in crisp datasets, however in interval or fuzzy datasets there is imprecision in the perception of the classes, thus these prior probabilities cannot be precisely determined. In certain cases it cannot be decided whether a dataset is imbalanced or not; a low specificity in the output variable easily leads to a possibly imbalanced dataset, as we will show in the sections that follow.

We have studied and extended different preprocessing stages, organized in three categories: under-sampling, over-sampling and hybrid[1,16]. For extending these preprocessing mechanisms to fuzzy data we have taken into account different fuzzy arithmetic operators[6,10], and different rankings of fuzzy numbers. Ranking methods play a crucial role in this work; the concept was first introduced for ordering fuzzy numbers[22,23], however ranking or comparing fuzzy numbers has been given many different interpretations; in this paper we will focus on the centroid index ranking method[8,9,12,28,37,43] which is a common technique for ranking numbers[36]. Moreover, the extension of these preprocessing mechanisms to imprecise data poses additional problems because, as we have just mentioned, an inaccurate perception of the classes means that the percentage of the instances that belong to each class is also imprecise. Therefore, we can obtain, for example, that the frequency of the class "A" is between $[0.3, 0.7]$ and for "B" is also between $[0.3, 0.7]$ so, if $f_A = 0.3$ and $f_B = 0.7$ the minority class will be represented by the class "A" but also can happen that $f_A = 0.7$ and $f_B = 0.3$ where the minority class now is "B". That is to say, when the data is imprecise it might happen that either alternative can be regarded as the minority class.

The structure of this paper is as follows: in Section 2 we introduce the problem of imbalanced datasets and discuss from a theoretical point of view the impact of preprocessing data for solving imbalanced problems. In Section 3 we generalize the imbalanced classification problem to imprecise data, review some preprocesing techniques, focusing on SMOTE[4], ENN[45], NCL[27] and CNN[20] algorithms, and propose new algorithms for re-balancing low quality imbalanced datasets, taking into account the possibly imprecise outputs. In Section 4 we introduce a metric of evaluation for these imprecisely perceived datasets. After describing these new algorithms for balancing datasets, we will analyze the behaviour of GFS for imprecise data[31] when the data is preprocessed before the learning phase, and compare the results obtained in several real-world problems about the diagnosis of dyslexic children[32] and the future performance of athletes in a competition[31]. In Section 5 we show these results. The paper finishes with the conclusions and future works, in Section 6.

## 2. A statistical characterization of the imbalanced classification problem

The problem of imbalanced datasets in classification occurs when the number of instances of one class is much lower than that of the other classes[5]. Some authors have named this problem "datasets with rare classes[44]". The minority class often represents the concept of interest[24,29,33].

According to certain authors[44,35] optimizing the average error leads to erroneous conclusions in imbalanced problems, since the minority or positive class has very little impact on the accuracy as compared to the majority or negative class. The answer to that concern, as we have mentioned before, consists in determining a suitable set of misclassification costs and in solving a cost sensitive classification problem[14,13], or alternatively, in preprocessing the training data, resampling it for equalizing the prior probabilities of the classes[1,16]. In this section we propose a theoretical justification of these preprocessing algorithms, and study the relation between them and certain cost sensitive classifications.

### *2.1. Notation for two-classes problems*

Many concepts in imbalanced classification are originated in two-classes problems. First and foremost, the confusion matrix, an example of which is shown in Table 2, divides the results of classifying a set of instances into four different categories: True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). Data in the first class are labelled “positives” or “minority”, and the second class is labelled “negatives” or “majority”.

The error in a test set is defined by total number of misclassified examples divided by the available examples,

$$\mathrm{Err} = \frac{\mathrm{FP}+\mathrm{FN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{1}$$

and the accuracy is $1-\mathrm{Err}$. We will also use the terms

$$\mathrm{TP}_{\mathrm{rate}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \quad \mathrm{FN}_{\mathrm{rate}} = \frac{\mathrm{FN}}{\mathrm{TP}+\mathrm{FN}} \tag{2}$$

$$\mathrm{TN}_{\mathrm{rate}} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \quad \mathrm{FP}_{\mathrm{rate}} = \frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}. \tag{3}$$

### *2.2. Metrics for two classes problems*

As we will explain in the next section, minimizing the global error in imbalanced problems might be biased towards the majority class. In other words, the instances that belong to the minority class might be misclassified more often than the other classes.

For preventing this, the cost of a mistake should depend on the class of the object. There is a wide agreement about the fact that the benefits of classifiers in similar domains must be assessed by more appropriate criteria than the average classification error[26]. Commonly used metrics for two classes problems[16,17,18] include the arithmetic and geometric means of the sensitivity $acc^{+} = \mathrm{TP}_{\mathrm{rate}}$ and the specificity $acc = \mathrm{TN}_{\mathrm{rate}}$. In particular, the geometric mean of both values is an interesting indicator of the quality of a classifier for imbalanced data, because it is high when both $acc^{+}$ and $acc$ are high or when the difference between $acc+$ and $acc$ is small[25]. Optimizing the geometric mean is a compromise intended for maximizing the accuracy on both classes while keeping these accuracies balanced[26]. These criteria will be studied in depth and extended to multiclass problems in the next subsection.

### *2.3. Statistical characterization: costs and resampling in imbalanced problems*

In this subsection we illustrate the mechanisms through which preprocessing the data, or using costs, influences the quality of a classification system for imbalanced data.

Let $(\mathbf{x},\mathbf{c})$ be a random variable pair taking values in $\mathbb{R}^d \times \mathscr{C}$, where the continuous random vector $\mathbf{x}$ is the feature or input vector, and the discrete variable $\mathbf{c} \in \mathscr{C} = \{c_1, c_2, \ldots, c_C\}$ is the class. Let $f(x)$ be the density function of the random vector $\mathbf{x}$, and $f(x|c)$ the density function of this vector, conditioned on the class is $\mathbf{c} = c$. $P(c_i)$ is the a priori probability of class $c_i$, $i = 1, \ldots, C$. $P(c_i|x)$ is the a posteriori probability of $c_i$, given that $\mathbf{x} = x$.

A classifier $\Phi$ is a mapping $\Phi : \mathbb{R}^d \to \mathscr{C}$, where $\Phi(x) \in \mathscr{C}$ denotes the class that an object is assigned when it is perceived through the feature vector $x$. A classifier defines so many decision regions $\mathscr{D}_i$ as classes,

$$\mathscr{D}_i = \{x \in \mathbb{R}^d \mid \Phi(x) = c_i\}, \quad i = 1, 2, \ldots C. \tag{4}$$

The performance of a classifier can be measured by the expected fraction of correct classifications

$$T(\Phi) = \sum_{i=1}^{C} \int_{\mathscr{D}_i} P(c_i|x) f(x) \mathrm{d}x. \tag{5}$$

It is widely known that the decision rule maximizing this rate is

$$\Phi_B(x) = \arg\max_{c \in \mathscr{C}} P(c|x), \tag{6}$$

| | Positive class ($C_0$) | Negative class ($C_1$) |
|---|---|---|
| Positive prediction ($C_0$) | True Positive (TP) | False Positive (FP) |
| Negative prediction ($C_1$) | False Negative (FN) | True Negative (TN) |

Table 1: Confusion matrix for two classes problems

the so called "minimum error Bayes rule[2]". The same value in Eq. (5) can also be expressed as

$$\begin{aligned} T(\Phi) = & \sum_{i=1}^{C} P(c_i) \int_{\mathscr{D}_i} f(x|c_i)\mathrm{d}x \\ = & \sum_{i=1}^{C} P(c_i) T_i(\Phi) \end{aligned} \qquad (7)$$

where the term

$$T_i(\Phi) = \int_{\mathscr{D}_i} f(x|c_i)\mathrm{d}x \qquad (8)$$

is the expected fraction of elements of the $i$-th class that are correctly classified by $\Phi$. That is to say, the expected fraction of correct classifications is the weighted average of successes restricted to each class, $T_i(\Phi)$, $i = 1,\dots,C$. Observe that, for two classes problems, $T_1$ is the expected value of $\mathrm{TP}_{\mathrm{rate}}$ and $T_2$ is the expected value of $\mathrm{TN}_{\mathrm{rate}}$.

This design is optimal in the sense that no other classification system can improve its expected success rate, but it is not without problems. In particular, in this paper we are interested in the case where the proportion of individuals of the interest class is very small. If a value of a $P(c_i)$ is near zero, its associated rate $T_i(\Phi_B)$ can also be low while at the same time $T(\Phi_B)$ can still be high. In words, it is possible that the percentage of failures in the minority classes, or "false negatives" of the minimum error Bayes rule is not admissible for certain applications.

This question is solved by minimizing a different criteria than the misclassification rate. For instance, one can search for the classifier $\Phi^{\mathrm{avg}}$ maximizing the unweighted mean

$$T^{\mathrm{avg}}(\Phi) = \sum_{i=1}^{C} \frac{1}{C} T_i(\Phi) \qquad (9)$$

which is a generalization to multiclass problems of the arithmetic mean of sensitivity and accuracy that we have mentioned in the preceding section. We can also regard it as a multiclass generalization of the Bradley's approximation to the area under the ROC curve[3]

$$\begin{aligned} & E(1 - F^{\mathrm{AUC}}(\Phi)) = E(\frac{\mathrm{TP}_{\mathrm{rate}} - \mathrm{FP}_{\mathrm{rate}}}{2}) \\ & = \frac{1}{2} + \frac{1}{2}(T_1(\Phi) + T_2(\Phi)). \end{aligned} \qquad (10)$$

The classifier $\Phi^{\mathrm{GM}}$ generalizing the geometric mean also seen before is assessed by the value

$$T^{\mathrm{GM}}(\Phi) = \prod_{i=1}^{C} T_i(\Phi)^{\frac{1}{C}}. \qquad (11)$$

In this paper we enclose these two metrics and many others in a common framework, and propose that the imbalanced classification is regarded as a multicriteria optimization problem, where the objective vector of a classifier $\Phi$ comprises its success rates for all classes,

$$(T_1(\Phi), \dots, T_C(\Phi)) \qquad (12)$$

and the dominance between two classifiers, whose success rates are $(t_1 \dots t_c)$ and $(u_1 \dots u_c)$, is given by

$$\begin{aligned} & (t_1 \dots t_c) \succeq (u_1 \dots u_c) \iff \\ & t_i \geqslant u_i \text{ for all } i \text{ and } t_i > u_i \text{ for some } i. \end{aligned} \qquad (13)$$

Let $\phi$ be the Pareto front of this problem,

$$\phi(T_1(\Phi), \dots, T_C(\Phi)) = 0. \qquad (14)$$

Observe that the different solutions to the imbalanced problems (including the minimum error classifier, as a particular case) are points in this Pareto front.

For proving this assert, suppose that there exists a scalar metric $M$ whose maximum is used for finding the best classifier (for instance, $M$ can be the AUC criterion, or the geometric mean mentioned before). Imagine also that there exists a classifier $\Phi'$ such that the maximum value of this metric is $M(\Phi')$ and $\phi(T_1(\Phi'),\ldots,T_C(\Phi')) \neq 0$. That classifier $\Phi'$ would be dominated by at least one element $\Phi''$ in the Pareto front (because if not, $\phi(T_1(\Phi'),\ldots,T_C(\Phi')) = 0$, by definition). Therefore, there would exist a classifier $\Phi''$ for which $M(\Phi'') < M(\Phi')$ and at the same time the success rate of $\Phi''$ in both the minority and the majority classes would be better or equal than that of $\Phi'$, which makes not sense.

Therefore, the best classifier $\Phi_0$ with respect to any coherent scalar metric $M$ must be a point of the Pareto front. Now we want to determine whether there exists a convex combination of the functions $T_i(\Phi)$ whose minimum is also $\Phi_0$, when constrained to $\phi(T_1(\Phi),\ldots,T_C(\Phi)) = 0$. In particular, let us consider the scalar metric that follows:

$$M^w(\Phi) = \sum_{i=1}^{C} w_i T_i(\Phi). \tag{15}$$

Introducing a Lagrange multiplier, the best classifier for $M^w$ is the maximum of

$$M^w(\Phi) + \lambda\phi(T_1(\Phi),\ldots,T_C(\Phi)) \tag{16}$$

and the first order Karush-Kuhn-Tucker conditions in this maximum are

$$\frac{\partial M^w}{\partial T_i}(\Phi_0) + \lambda\frac{\partial \phi}{\partial T_i}(\Phi_0) = 0. \tag{17}$$

We want to know whether there is a set of weights $w_i$ such that the maximum of Eq. (16) is the same classifier that maximizes the arbitrary scalar metric $M$ mentioned at the beginning of this paragraph. The answer is positive, as it is well known that the convex part of a Pareto front can be reached by minimizing weighted combinations of the objectives. For instance, the assignment

$$w_i = \frac{\frac{\partial\phi}{\partial T_i}(\Phi_0)}{\sum_{j=1}^{C}\frac{\partial\phi}{\partial T_j}(\Phi_0)} \tag{18}$$

makes that the equations

$$\frac{\frac{\partial\phi}{\partial T_i}(\Phi_0)}{\sum_{j=1}^{C}\frac{\partial\phi}{\partial T_j}(\Phi_0)} + \lambda\frac{\partial \phi}{\partial T_i}(\Phi_0) = 0 \tag{19}$$

are fulfilled in $\Phi_0$, and the value of the Lagrange multiplier is

$$\lambda = -(\sum_{j=1}^{C}\frac{\partial\phi}{\partial T_j}(\Phi_0))^{-1}. \tag{20}$$

Now we will show that $M^w$ can be regarded as the dual of the risk of a classifier for a given cost matrix. Let $B = [b_{ij}] \in \mathbb{R}^{C\times C}$, where $b_{ij} = \mathrm{cost}(c_i, c_j)$ is the cost of deciding that an object is of class $c_i$ when its actual class is $c_j$. Cost sensitive classifiers minimize the risk function

$$R(\Phi) = \sum_{i=1}^{C}\int_{\mathscr{D}_i}\sum_{j=1}^{C} b_{ij}P(c_j|x)f(x)\mathrm{d}x. \tag{21}$$

Let $w_i$ the value defined in Eq. (18), and let also be

$$K = \max_{k=1\ldots C}\left\{\frac{w_k}{P(c_k)}\right\}, \tag{22}$$

and let the cost matrix be

$$b_{ij} = \begin{cases} K - \dfrac{w_i}{P(c_i)} & \text{for } i = j \\ K & \text{else.} \end{cases} \tag{23}$$

The risk $R^w(\Phi)$ associated to this particular cost matrix is computed as follows:

$$R^w(\Phi) = R^{w1} - R^{w2} \tag{24}$$

where

$$\begin{aligned} R^{w1}(\Phi) &= \sum_{i=1}^{C}\int_{\mathscr{D}_i}\sum_{j=1}^{C} KP(c_j|x)f(x)\mathrm{d}x \\ &= K\int_{\mathbb{R}^d}\sum_{j=1}^{C} P(c_j|x)f(x)\mathrm{d}x \end{aligned} \tag{25}$$

does not depend on the classifier, and

$$\begin{aligned} R^{w2}(\Phi) &= \sum_{i=1}^{C}\int_{\mathscr{D}_i}\frac{w_i}{P(c_i)}P(c_i|x)f(x)\mathrm{d}x \\ &= \sum_{i=1}^{C}\frac{w_i}{P(c_i)}\int_{\mathscr{D}_i}P(c_i|x)f(x)\mathrm{d}x \\ &= \sum_{i=1}^{C} w_i T_i(\Phi) \\ &= M^w(\Phi) \end{aligned} \tag{26}$$

thus finding the classifier $\Phi$ that minimizes $R^{w1} - R^{w2}$ is equivalent to finding the classifier maximizing $M^w$, as desired.

As a consequence of this, it is also possible to find solutions to imbalanced classification problems by resampling the training set. Such resampling should be designed for obtaining a new dataset, ideally a random sample from an hypothetical population with the same densities $f(x|c_i)$ and $f(x)$, but whose prior probabilities of the classes are $P_{\text{resample}}(c_i) = w_i$. It is immediate that the best classifier for this hypothetical population, according to the minimum error based criterion, is

$$\begin{aligned}\Phi_{\text{resample}}(x) &= \arg\max_{i=1\ldots C} \frac{\frac{w_i}{P(c_i)}P(c_i|x)}{\sum_{k=1}^{C}\frac{w_k}{P(c_k)}P(c_k|x)} \\ &= \arg\max_{i=1\ldots C} \frac{w_i}{P(c_i)}P(c_i|x)\end{aligned} \tag{27}$$

and this classifier also minimizes the risk associated to the cost matrix in Eq. (23) for the original problem.

The consequences of these results can be summarized in the following two points:

1. Any criteria for defining the quality of imbalanced classifiers fulfilling Eq. (13) can be replaced by a cost-sensitive classification system with the cost matrix defined by Eq. (23). Observe that this cost matrix is not unique.

2. The minimum risk solution to an imbalanced classifier can also be obtained by applying the minimum error Bayes rule to a resampled dataset whose prior probabilities are $w_i$.

Observe also that our arguments in this section support the fact that for each scalar quality measure (i.e, average of sensitivity and specificity, geometric mean, AUC, etc.) there is a cost matrix for which the minimum risk classifier is the same as the imbalanced classifier, but we have not provided a method for obtaining the values of the elements of this cost matrix in the general case.

## 3. Preprocessing possibly imbalanced datasets

According to the results in the preceding section, an imbalanced dataset problem can be dealt with by altering the objective function of the classifier, for making it to depend on a cost matrix that better takes into account the minority class, or else we can leave the classification system as is, but equalize the data in advance, in order to minimize the effect caused by their class imbalance. In this last respect, there are three categories of preprocessing methods in the literature[1,16]:

- Under-sampling methods: Obtain a subset of the original dataset by eliminating some of the examples of the majority class. This category comprises the Condensed Nearest Neighbour rule (CNN)[20], Tomek links[40], One-sided selection (OSS)[26], Neighbourhood cleaning rule (NCL)[27] based on the Wilson's Edited Nearest Neighbour (ENN)[45] and the random under-sampling.
- Over-sampling methods: Obtain a superset of the original dataset by replicating some of the examples of the minority class or creating new ones from the original minority class instances. These methods are Synthetic Minority Over-sampling Technique (SMOTE)[4] and random over-sampling.
- Hybrid methods: These combine over-sampling and under-sampling, and obtain a set by combining the two previous methods. For instance, SMOTE+Tomek Link and SMOTE+ENN.

### *3.1. Possibly imbalanced datasets*

When a dataset contains imprecision in the output variables, the degree of imbalance is also uncertain. For instance, if an instance is labeled as "class $\{$A, B$\}$" or, in words, if we do not know whether the true class of certain instance is A or B, then the total number of instances of types "A" and "B" in the training set is also an imprecise value and the same can be said about the imbalance ratio of that dataset. To this we can add that, if the specificity of these imprecise labels is low, most datasets will be possibly imbalanced. For instance, imagine a problem with three classes where, after computing the ranges of the relative frequencies of the classes, we obtain that $f_A \in [0.05, 0.25]$, $f_B \in [0.05, 0.35]$ and $f_C \in [0.4, 0.9]$. This means that the actual frequencies might be 0.25, 0.35 and 0.4, which is not, strictly speaking, an unbalanced problem, but it is also possible that they are 0.05, 0.05 and 0.9. In this last case, it is widely admitted that a classification system might not perform well on classes A and B unless we equalize the training set. Since we cannot precise whether the actual imbalance ratio is as low as 0.4/0.25 or as high as 0.9/0.05, it seems reasonable to us to adapt those techniques used for preprocessing imbalanced crisp datasets to low quality data and use them also in problems that, at a first sight, would have not been regarded as umbalanced. Because of this fact, new algorithms based on SMOTE, NCL, ENN and CNN will be explained in this section.

### 3.2. Catalog of methods to be extended: ENN, NCL, CNN and SMOTE

As we have mentioned before, there are different categories of preprocessing mechanisms. All of them are susceptible of being extended to interval and fuzzy data. Our selection of methods is chosen as follows:

**Under-sampling:** In this category we have considered NCL, ENN and CNN. This is because both CNN and OSS have worse performance (in the framework of FRBCS) than the straight use of the unprocessed data, but CNN is known to improve OSS. In turn, applying NCL is better than not applying a preprocessing mechanism but it is not expected to improve SMOTE, being comparable to the "Tomek links" algorithm[16]. We have also extended the ENN algorithm because this algorithm not only removes elements from the majority class, but also discard instances from the remaining classes. Given that the concept of "minority class" is not precise under our assumptions, this behavior is more convenient for us than that of these algorithms discarding elements of a unique, pre-selected class.

**Over-sampling:** We apply SMOTE because this algorithm consistently obtains better results than the random-over-sampling in the context of FRBCS[16].

**Hybrid method:** We have selected SMOTE+ENN because both this and SMOTE+Tomek links obtain similar results, near to those obtained with SMOTE. In addition to this, ENN tends to remove more examples than the Tomek links does, so it is expected to produce a deeper data cleaning[16].

### 3.3. Outline and generalization of the preprocessing methods

Since there are many parts in common among the methods that will be generalized, we describe all of them first, and then detail the common parts in a separate subsection later.

**SMOTE:** In the SMOTE algorithm, the minority class is over-sampled. Each minority class example is selected, and new synthetic examples are introduced along the line segments joining any or all of its nearest neighbors of the minority class. Depending on the amount of over-sampling required, some of these nearest neighbors are randomly chosen[4]. For example, if the implementation uses four nearest neighbors and the amount of over-sampling needed is 200%, only two neighbors from the four nearest neighbors are chosen and one sample is generated in the direction of each. In Figure 1 an example is shown where $x_i$ is the selected point, $x_{i1}$ to $x_{i4}$ are some of the selected nearest neighbors and $r_1$ to $r_2$ are the synthetic data points created by the randomized interpolation.

For generating synthetic samples, the difference between the feature vector of the sample under consideration and its nearest neighbor is taken. This difference is multiplied by a random number between 0 and 1, and added to the initial feature vector. This causes that a random point along the line segment between two specific features is selected. This approach effectively forces the decision region of the minority class to become more general[4]. A numerical example of this procedure is detailed in Table 2.

**ENN:** This preprocessing mechanism is a data cleaning method that removes any example whose class label differs from the class of at least two of its

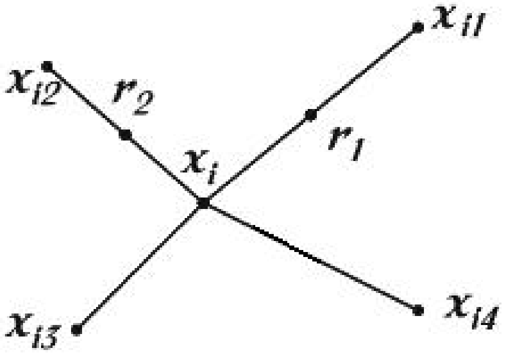

Figure 1: Creation of synthetic data points in the SMOTE algorithm.

Table 2: Example of the SMOTE method.

| |
|---|
| Consider an instance (6,4) and let (4,3) be its nearest neighbor. |
| (6,4) is the instance for which the nearest neighbors are being identified. |
| (4,3) is one of its k-nearest neighbors. |
| Let: |
| f1_1 = 6 f2_1 = 4 f2_1 - f1_1 = -2 |
| f1_2 = 4 f2_2 = 3 f2_2 - f1_2 = -1 |
| The new samples will be generated as |
| (f1',f2') = (6,4) + rand(0-1) * (-2,-1) |
| rand(0-1) generates a random number between 0 and 1. |

three nearest neighbors[45]. The steps of this algorithm are:

1. For each instance of the training set, their nearest neighbors are found.
2. The class represented or associated with the largest number of these nearest neighbours is selected.
3. If the class of the instance is different than the class found in the preceding step, the instance is removed.

**NCL:** This preprocessing mechanism is a undersampling method based on ENN that removes examples from the majority class:

1. For each input of the training set, their nearest neighbors are found.
2. The class represented or associated with the largest number of nearest neighbours is selected.
3. If the class of the instance belongs to the majority class and is different than the class fond in the preceding step, the instance is removed. In addition to this, if the class of the instance belongs to the minority class and the class fond in the preceding step is not the same as the class of the instance then all the instances among the nearest neighbours that belong to the majority class are also removed.

**CNN:** This method is used to find a consistent subset of examples. The concept of a consistent subset of a training set was proposed in reference[20], in combination with the algorithm "Condensed NN rule". To determine this consistent subset from the training set, this algorithm uses two additional sets $S$ and $T$ in the following way:

- $S$ is initialized with examples of the training set. There are different procedures for this inicialization. For instance, the first examples of the training set can be chosen[20].
- $T$ is initialized with those instances of the training set that are not in $S$.
- A nearest-neighbor classifier is defined using the contents of $S$. Each instance in $T$ is classified with this nearest-neighbor system. If the actual class of an object in $T$ does not match the output of the $S$-based classifier, this object is added to $S$.

To this we can add that some authors defend that objects of the minority class should not be removed[15], thus in this case the initialization of $S$ consists in inserting one example of the majority class of the training set and all the elements of the minority class.

## 3.4. Generalization of SMOTE to Low Quality Data: SMOTE-LQD

There are three aspects in our generalization of SMOTE to low quality data that deserve a detailed study:

1. Selection of the minority class and the amount of synthetic examples. Given that the imbalance ratio is not precisely known, it might happen that more than one class can be regarded as the minority class.

2. Computation of the nearest neighbors of any example. The implementation applied in this work uses the euclidean distance to select the nearest neighbors and it also uses fuzzy arithmetic operators and a fuzzy ranking, as we will explain later.

3. Generation of synthetic examples from the minority class. We will use fuzzy arithmetic operators, and control the values that may be out of range for the different attributes.

### *3.4.1. Selection of the minority class*

The inputs to the SMOTE algorithm[4] are the minority classes and the amount of synthetic examples that will be generated for each class. In our extension, the minority class is the set of all classes but the most frequent. Instances with more than one label are not considered in this step.

Suppose that the features and the classes of the objects in the dataset cannot be accurately perceived, but we are given intervals that contain them:

$$\overline{\mathscr{D}} = \{(\mathscr{X}_k, \mathscr{Y}_k)\}_{k=1}^{N} \tag{28}$$

where $\mathscr{X}_k \subset \mathbb{R}^d$ and $\mathscr{Y}_k \subset \{1,\dots,C\}$. Let us define the vector $(m_1,\dots,m_C)$ of absolute frequencies of the classes in the dataset, whose components are

$$m_i = \#\{k \mid \mathscr{Y}_k = \{i\}\}. \tag{29}$$

Let $c^*$ be the majority class, $m_{c^*} = \max_{i=1,\dots,C} m_i$. We will consider that all classes but $c^*$ are minority classes. Each instance will be used to generate a number of synthetic resamples that depends on its class, and this number is

$$\left(\frac{m_{c^*}}{m_1},\dots,\frac{m_{c^*}}{m_C}\right). \tag{30}$$

### *3.4.2. Computation of the nearest neighbors*

In a first step we collect all the examples that possibly belong to the minority class. This set includes those whose class we know and those whose class we cannot affirm is different than the minority:

$$\{(\mathscr{X}_k, \mathscr{Y}_k) \mid c^* \notin \mathscr{Y}_k\}. \tag{31}$$

The second step consists in obtaining the $k$ nearest neighbors of the example, where the meaning of "nearest" is given by a generalized euclidean distance and a certain method for ranking these distances. That is to say, the euclidean distance between two vectors of fuzzy numbers $(\widetilde{A}_{i1},\dots,\widetilde{A}_{in})$ and $(\widetilde{B}_{j1},\dots,\widetilde{B}_{jn})$ is generalized as follows:

$$\widetilde{D}_{ij} = \left[\bigoplus_{m=1}^{n} (\widetilde{A}_{im} \ominus \widetilde{B}_{jm})^2\right]^{\frac{1}{2}} \tag{32}$$

where all fuzzy numbers are trapezoidal, $\widetilde{A} = (a,b,c,d)$ and all the arithmetic operators are also fuzzy[6,10]. We will consider that $\widetilde{D}_{ij}$ is a generalized trapezoidal fuzzy number.

We have used the operation "ranking" for determing the $k$ nearest neighbours of a given example. It is well known[36] that no single ranking method is superior to all other methods; each ranking appears to have some advantages as well as disadvantages. In this proposal we use the method defined in[43,12].

### 3.4.3. Generation of the synthetic examples

The generation of the synthetic examples consists in taking the difference between the feature vector under consideration and its nearest neighbor[4]. This difference is multiplied by a random number between 0 and 1, and added to the feature of the synthetic example. It is remarked that these operations involve fuzzy arithmetic and we have to control the values that are out of range in the different attributes due to these operations.

## 3.5. ENN_LQD and NCL_LQD

For both ENN_LQD and NCL_LQD, we have to consider the following factors:

1. *The computation of the three nearest neighbors:* As in SMOTE_LQD we need to introduce fuzzy arithmetic operators and a fuzzy ranking for determining the nearest neighbors of a given example.

2. *Removal of the instances that differ in two of its three neighbors:* Having into account that the instances may have more than one label, the options are:

   - If the instance has a precise output and at least one of its neighbors has an imprecise output, then all possible alternatives of the set of elements are studied (see Figure 2). If one of these alternatives prevents the removal, the instance is not deleted from the training set.
   - If the instance is multilabelled and the most frequent class among their nearest neighbors is contained in the set of labels (see Figure 3 for an example) then the instance is not deleted. However, in this case the uncertainty is removed and the element is assigned the most frequent class. In the example in Figure 3, the instance labelled with $\{1,0\}$ is relabelled as $\{0\}$. In Figure 4 the instance keeps its imprecise output.

In addition to this, the NCL_LQD algoritm defines the following directives:

1. The selection of the number of instances that belong to each class, used for determining the minority classes, always takes into account the instances with imprecise outputs, applying Eq. (29).

2. If the instance $(\mathscr{X}_k, \mathscr{C}_k)$ is part of the minority classes and its class differs at least in two of its three nearest neighbors, it can happen that (see Figure 5):

   - None of the neighbours belongs to the majority class: none is eliminated.
   - The class of a multi-labelled neighbor contains the majority class: this neighbour is also kept.

## 3.6. CNN_LQD

The generalization of this algorithm is as follows:

1. *Computation of the nearest neighbours:* As done in SMOTE_LQD we need to introduce fuzzy arithmetic operators and a fuzzy ranking for determining the nearest neighbors of a given example.

2. *Initialization of the set S:* This set is composed by $C$ instances of the training set (where $C$ is the number of classes). Elements with multiple labels are not allowed in this set.

3. *Addition of imprecise instances to the set T:* If the label of any of the nearest neighbors is included in the set of labels of the element being classified, we will consider that this element has been correctly classified and then we do not include it in the set $S$ (see Figure 6).

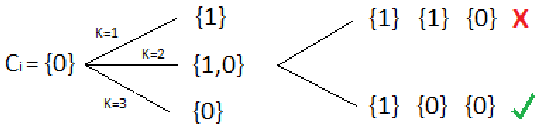

Figure 2: Instance which is not removed because some of its neigbors are imprecise

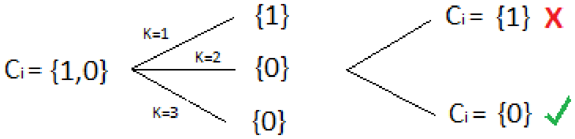

Figure 3: Instance with imprecise output that is relabelled according to their neighbors

## 4. Metric of evaluation with low quality data and imbalanced datasets

The multicriteria evaluation of the performance of a classifier $\Phi$ is a numerical estimation of the values $T_i(\Phi)$ seen before, or the scalar value of a suitable metric $M(\Phi)$. Most of times, these metrics depend of the terms of the confusion matrix. We have mentioned before that $\mathrm{TP}_{\mathrm{rate}}$ is an estimation of $T_1$ and $\mathrm{TN}_{\mathrm{rate}}$ is an estimation of $T_2$.

Generally speaking, let $\mathscr{D} = \{X_k, y_k\}_{k=1,\dots,N}$ be a crisp dataset and let $S(\Phi, \mathscr{D}) = [s_{ij}]$ be the confusion matrix of a classifier $\Phi$ on the dataset $\mathscr{D}$. This matrix comprises the elements $s_{ij}$, that are the number of elements in the sample for which the output $\Phi(x_k)$ of the classifier is $c_i$ and $j = y_k$ (i.e. the class of the $k$-th element is $c_{y_k}$). Let us express this as follows:

$$s_{ij} = \sum_{k=1}^{N} \delta_{c_i,\Phi(x_k)} \delta_{j,y_k}. \tag{33}$$

In case that the features and the classes of the objects in the dataset cannot be accurately perceived, but we are given intervals that contain them, the dataset is

$$\overline{\mathscr{D}} = \{(\mathscr{X}_k, \mathscr{Y}_k)\}_{k=1}^{N} \tag{34}$$

where $\mathscr{X}_k \subset \mathbb{R}^d$ and $\mathscr{Y}_k \subset \{1,\dots,C\}$. The most precise output of the classifier $\Phi$ for a set-valued input $\mathscr{X}$ is

$$\Phi(\mathscr{X}) = \{\Phi(x) \mid x \in \mathscr{X}\}. \tag{35}$$

In this case, the elements of the confusion matrix $\overline{S}$ are also sets. Let us define, for simplicity in the notation, the set-valued function $\overline{\delta} : \mathscr{C} \times \mathscr{P}(\mathscr{C}) \to \mathscr{P}(\{0,1\})$

$$\overline{\delta}_{a,\mathscr{A}} = \begin{cases} 1 & \{a\} = \mathscr{A} \\ 0 & a \notin \mathscr{A} \\ \{0,1\} & \text{else.} \end{cases} \tag{36}$$

With the help of this function, the confusion matrix in the preceding subsection is generalized to an interval-valued matrix $\overline{S} = [\overline{s}_{ij}]$, as follows:

$$\overline{s}_{ij} = \sum_{k=1}^{N} \overline{\delta}_{c_i,\Phi(\mathscr{X}_k)} \overline{\delta}_{j,\mathscr{Y}_k}. \tag{37}$$

and the estimation of the values $T_i$ with imprecise data are

$$\widehat{T_i}(\Phi, \overline{\mathscr{D}}) = \frac{\overline{s}_{ij}}{\overline{n}_j} \tag{38}$$

where $\overline{n}_j$ is an interval estimation of the number of elements of the $j$-th class,

$$\overline{n}_j = [\#\{k \mid \{j\} = \mathscr{Y}_k\}, \#\{k \mid j \in \mathscr{Y}_k\}]. \tag{39}$$

Observe that these last expressions make use of set-valued addition and multiplication,

$$\mathscr{A} + \mathscr{B} = \{a + b \mid a \in \mathscr{A}, b \in \mathscr{B}\} \tag{40}$$

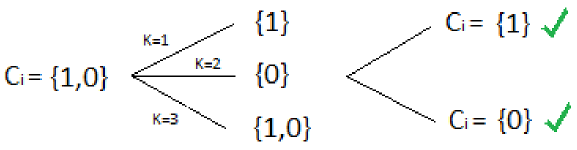

Figure 4: Instance with imprecise output that is not relabelled.

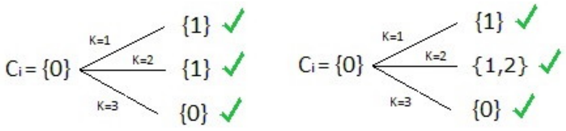

Figure 5: Instance and neighbours kept in the training set. The classes are $\{0,1,2\}$, where the class "2" is the majority.

$$\mathscr{A} \cdot \mathscr{B} = \{ab \mid a \in \mathscr{A}, b \in \mathscr{B}\}. \tag{41}$$

## 5. Numerical Results

In this section we will study some real-world problems; some of them are related to medical diagnosis (the diagnostic of dyslexia in children[32]) while others study the future performance of athletes in certain tests[31]. These datasets are summarized as follows:

- Athletism datasets: This experimentation is composed by 8 datasets that are used to predict whether an athlete will improve certain threshold in the long jump, 100 meters and 200 meters, given several relevant indicators of each event. All the features are interval-valued.
- Dyslexia datasets: This subset is composed by 3 datasets that are used for diagnosing whether one child has dyslexia or not. These datasets contain a mix of interval and crisp data.

It is remarked that all these cases have a certain degree of imbalance and vagueness in the perception of the features and the class. With this experimentation we will compare the performance of GFSs designed for being used with low quality data when applied to both unprocessed and preprocessed datasets.

### *5.1. Settings*

The datasets comprising this experimentation have been taken from previous works[31,32] for an easier reproducibility, and all of them have imprecise inputs and outputs. A brief descripcion of them is provided in Table 3, showing for each dataset the name, the number of examples (Ex), number of attributes (Atts), the classes and the percentage of patterns of each class.

All the experiments have been run with a population size of 100, probabilities of crossover and mutation of 0.9 and 0.1, respectively, and limited to 100 generations. The fuzzy partitions of the labels are uniform and their size is 5 in athletism datasets and 4 in datasets of dyslexia. All the imprecise experiments were repeated 100 times with bootstrapped resamples of the training set. The preprocessing methods applied in this work (SMOTE_LQD, SMOTE+ENN_LQD, ENN_LQD and NCL_LQD) use the three nearest neighbors, except CNN_LQD that only uses one, and balance all the classes taking into account the imprecise outputs, where the number of duplicates of the minority instances is estimated by the algorithms, when necessary. All the methods have been used for preprocessing 100 bootstrapped resamples of the training set, where the "out of the bag" instances are the test sets.

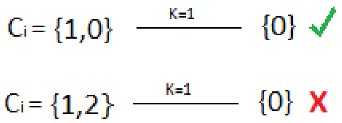

Figure 6: Classification of the instances with imprecise outputs.

| Dataset | Ex. | Atts. | Classes | %Classes |
|---|---|---|---|---|
| Long-4 | 25 | 4 | (0,1) | ([0.36,0.64],[0.36,0.64]) |
| BLong-4 | 25 | 4 | (0,1) | ([0.36,0.64],[0.36,0.64]) |
| 100ml-4-I | 52 | 4 | (0,1) | ([0.44,0.63],[0.36,0.55]) |
| 100ml-4-P | 52 | 4 | (0,1) | ([0.44,0.63],[0.36,0.55]) |
| B100ml-I | 52 | 4 | (0,1) | ([0.44,0.63],[0.36,0.55]) |
| B100ml-P | 52 | 4 | (0,1) | ([0.44,0.63],[0.36,0.55]) |
| B200ml-I | 19 | 4 | (0,1) | ([0.47,0.73],[0.26,0.52]) |
| B200ml-P | 19 | 5 | (0,1) | ([0.47,0.73],[0.26,0.52]) |
| Dyslexic-12 | 65 | 12 | (0,1,2,4) | ([0.32,0.43],[0.07,0.16], [0.24,0.35],[0.12,0.35]) |
| Dyslexic-12-01 | 65 | 12 | (0,1,2) | ([0.44,0.53],[0.24,0.35], [0.12,0.30]) |
| Dyslexic-12-12 | 65 | 12 | (0,1,2) | ([0.32,0.43],[0.32,0.52] [0.12,0.30]) |

Table 3: Summarized descriptions of the datasets.

Lastly, it is remarked that we have used two different metrics: the arithmetic and the geometric means of the values $T_i$ mentioned in the preceding sections. These metrics have been labelled $\mathrm{Acc}_{\mathrm{Tst}}$ and $\mathrm{GM}_{\mathrm{Tst}}$. Observe that we have only included the upper bound of GM, which is the most pessimistic estimation, as the lower bound is less informative, and tends to be an overly optimistic lower bound of the results.

### *5.2. Athletic's Datasets*

The behaviour of the GFS able to use low quality data when applied to both unprocessed and preprocessed Athletic's datasets is shown in Table 4. This Table is composed by several columns. The first column, "Dataset", contains the name of the datasets. The second one, "GFS", shows the accuracy obtained by the GFS with the original datasets (unprocessed). The rest of the columns show the accuracy of the GFS when the datasets are preprocessed with different preprocessing mechanisms.

The datasets are divided in two groups. On the one hand, those based on the long run and 200 meters. On the other hand, 100 meters run. This division responds to the fact that the events in the first group have a higher imprecision in the number of instances in each class, because the number of multilabelled instances is also higher. We have used this property for comparing the different behavior of the algorithms in datasets with low and high imprecision in the output variable.

We have observed that SMOTE_LQD is the alternative of choice in both cases, and applying this preprocessing algorithm is, generally speaking, better than leaving the data unprocessed even for those datasets where their imbalance is not self-evident. On the contrary, we have found cases where the use of NCL_LQD is not recommended: this last algorithm should not be applied to those datasets where the imprecision is high and the number of instances low, as happens in 100 meters. The removal of instances loses relevant information and this difficults the convergence of the GFS.

Summarizing the results in Table 4, we have detected a different behavior of the preprocessing algorithms as a function of the imprecision in the output variable. These differences are as follows:

- Using SMOTE_LQD improves the convergence of GFS for imprecise data in all the problems.
- The use of SMOTE+ENN_LQD is our second best choice. There are differences between SMOTE_LQD and SMOTE+ENN_LQD that make us to prefer the first alternative, but either algorithm is better than leaving the data unprocessed.
- The use of NCL_LQD has a quality coherent with the findings of other authors[16] but too many instances may be removed (this effect is clearly perceived in the 100 meters problem). When the size of the dataset is large, NCL_LQD improves the results of ENN_LQD and CNN_LQD.
- Using CNN_LQD is not recommended as leaving the data unprocessed can outperform this preprocessing, again confirming the results in the literature[16].
- ENN_LQD cleans both majority and minority instances, improving CNN_LQD, as expected. The results of this algorithm are intermediate between NCL_LQD and CNN_LQD, however the need for applying this algorithm is not justified, as their results do not significantly improve those obtained when the GFS uses the raw data.

We can conclude that our results corroborate the expected properties of the crisp versions of the algorithms[16]. In Figure 7 we have pictured the differences among these algorithms according to the degree of imprecision in the data. Lastly, in Figure 8 we have displayed the same results with respect to the Bradley's approximation to the AUC metric mentioned before. As we have shown, these are similar to the average of the success rates. The highest improvements are obtained through the use of the algorithm SMOTE_LQD: in our experimentation, the mean improvement in the most pessimistic estimation of GM was near 7%, and these improvements were significant in about 65% of our cases. The highest improvements were measured in the datasets whose imprecision in the output variable is higher.

It is also interesting that in the dataset "B200ml-P" the preprocessing of the data seems to be irrelevant if the averaged accuracy is the metric of choice. Notwithstanding, there is a measurable gain if the GM metric is used: 61.91% for SMOTE_LQD, with respect to 40.70% without preprocessing. In Table 5 we have shown that the number of successful classifications is indeed similar, however the preprocessing has equalized the number of correct classifications for all the classes, from 3973 in the majority class and 571 in the minority class to near 2000 in either class (2518 and 2084 successes). Also in this Table 5 we can check how FN and FP decrease in some datasets, with benefits in long run and 200 meters.

### 5.3. *Dyslexic's Datasets*

The behaviour of the GFS able to use low quality data in the Dyslexia datasets is shown in Table 6, where the accuracy of the GFS is shown when the datasets are unprocessed or preprocessed with different mechanisms. In the first column of the table we have shown the names of the datasets and the values of the majority class and the number of extra instances needed for balancing the datasets either with SMOTE_LQD or SMOTE+ENN_LQD. These two parameters have been obtained through the study of the confusion matrix obtained with the original datasets. In the same column of the same table we display the behavior of the GFS without preprocessing the data, and the remaing part of that table contains the performances of the GFS with and without preprocessing the imprecise data with SMOTE_LQD, SMOTE+ENN_LQD, ENN_LQD, NCL_LQD and CNN_LQD.

As it happened with athletism datasets, we detect that using SMOTE_LQD uniformly improves the results of all the algorithms in our selection. The quality of the results is mostly coherent with the results in the preceding subsection and also in the crisp ver-

Table 4: Means of 100 repetitions of the GFS from the low quality Athletic's datasets with 5 labels/variable with the original dataset and preprocessed with several preprocessing mechanism.

| | **GFS** | | **GFS+SMOTE_LQD** | | **GFS+SMOTE+ENN_LQD** | |
|---|---|---|---|---|---|---|
| **Dataset** | $Acc_{Tst}$ | $GM_{Tst}$ | $Acc_{Tst}$ | $GM_{Tst}$ | $Acc_{Tst}$ | $GM_{Tst}$ |
| Long-4 | [0.408,0.677] | 52.52 | [0.486,0.755] | 59.16 | [0.417,0.686] | 49.69 |
| BLong-4 | [0.375,0.674] | 50.45 | [0.446,0.746] | 57.55 | [0.443,0.742] | 57.06 |
| B200ml-I | [0.524,0.768] | 45.21 | [0.630,0.875] | 61.82 | [0.590,0.834] | 57.62 |
| B200ml-P | [0.520,0.738] | 40.70 | [0.521,0.739] | 61.91 | [0.516,0.734] | 54.51 |
| **Mean** | [0.457,0.715] | 47.22 | **[0.521,0.779]** | **60.11** | [0.492,0.749] | 54.72 |
| 100ml-4-I | [0.622,0.824] | 67.50 | [0.625,0.826] | 68.94 | [0.623,0.824] | 66.78 |
| 100ml-4-P | [0.645,0.824] | 69.03 | [0.653,0.832] | 70.15 | [0.650,0.829] | 69.60 |
| B100ml-I | [0.631,0.828] | 67.80 | [0.633,0.831] | 69.42 | [0.583,0.781] | 62.20 |
| B100ml-P | [0.651,0.84] | 69.78 | [0.65,0.839] | 70.41 | [0.645,0.834] | 68.72 |
| **Mean** | [0.638,0.829] | 68.53 | **[0.641,0.832]** | **69.73** | [0.626,0.817] | 66.83 |
| **Total Mean** | [0.547,0.772] | 57.87 | **[0.581,0.806]** | **64.92** | [0.559,0.783] | 60.77 |

| | **GFS+ENN_LQD** | | **GFS+NCL_LQD** | | **GFS+CNN_LQD** | |
|---|---|---|---|---|---|---|
| **Dataset** | $Acc_{Tst}$ | $GM_{Tst}$ | $Acc_{Tst}$ | $GM_{Tst}$ | $Acc_{Tst}$ | $GM_{Tst}$ |
| Long-4 | [0.395,0.664] | 52.21 | [0.422,0.69] | 54.17 | [0.377,0.645] | 47.27 |
| BLong-4 | [0.373,0.673] | 50.13 | [0.374,0.674] | 50.14 | [0.373,0.673] | 42.65 |
| B200ml-I | [0.513,0.757] | 45.44 | [0.526,0.771] | 49.19 | [0.476,0.720] | 43.45 |
| B200ml-P | [0.505,0.723] | 39.64 | [0.502,0.720] | 44.78 | [0.517,0.735] | 41.36 |
| **Mean** | [0.447,0.705] | 46.85 | [0.456,0.713] | 49.57 | [0.436,0.694] | 43.68 |
| 100ml-4-I | [0.591,0.792] | 65.83 | [0.604,0.805] | 66.90 | [0.564,0.765] | 62.98 |
| 100ml-4-P | [0.652,0.830] | 68.63 | [0.644,0.822] | 68.44 | [0.497,0.676] | 54.33 |
| B100ml-I | [0.603,0.8] | 66.18 | [0.609,0.806] | 66.92 | [0.561,0.759] | 62.32 |
| B100ml-P | [0.635,0.824] | 68.26 | [0.641,0.830] | 69.28 | [0.534,0.723] | 57.51 |
| **Mean** | [0.621,0.812] | 67.22 | [0.624,0.815] | 67.88 | [0.539,0.731] | 59.28 |
| **Total Mean** | [0.534,0.758] | 57.04 | [0.54,0.764] | 58.72 | [0.488,0.712] | 51.48 |

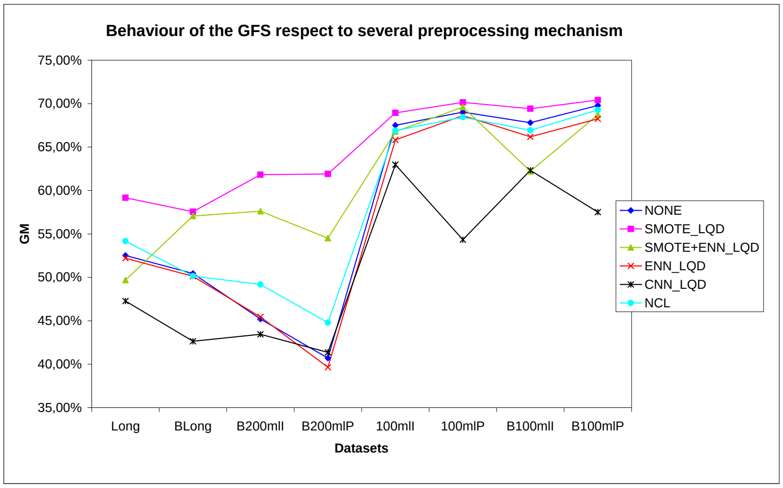

Figure 7: Behaviour of low quality data in the GFS respect to several preprocessing mechanisms (upper bound of GM metric)

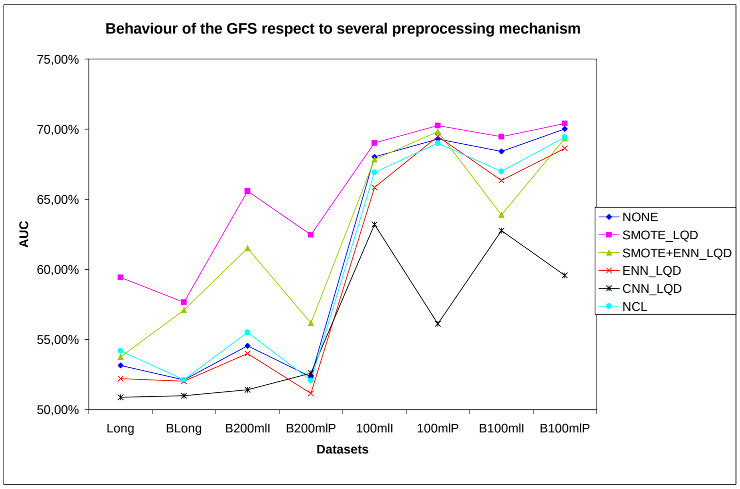

Figure 8: Behaviour of low quality data in the GFS respect to several preprocessing mechanisms. (upper bound of AUC metric)

| | GFS | | GFS+SMOTE_LQD | |
|---|---|---|---|---|
| **Long-4** | | | | |
| | Class 0 | Class 1 | Class 0 | Class 1 |
| Class 0 | 2591 | **3168** | 3098 | **2661** |
| Class 1 | **2186** | 3463 | **1974** | 3675 |
| **BLong-4** | | | | |
| | Class 0 | Class 1 | Class 0 | Class 1 |
| Class 0 | 2379 | 3720 | 3307 | 2792 |
| Class 1 | 2005 | 3764 | 2245 | 3524 |
| **100ml-4-I** | | | | |
| | Class 0 | Class 1 | Class 0 | Class 1 |
| Class 0 | 9352 | 2867 | 8044 | 4175 |
| Class 1 | 4346 | 6393 | 2986 | 7753 |
| **100ml-4-P** | | | | |
| | Class 0 | Class 1 | Class 0 | Class 1 |
| Class 0 | 9135 | 2974 | 9005 | 3104 |
| Class 1 | 3863 | 6626 | 3549 | 6940 |
| **B100ml-4-I** | | | | |
| | Class 0 | Class 1 | Class 0 | Class 1 |
| Class 0 | 9286 | 2693 | 8009 | 3970 |
| Class 1 | 4298 | 6261 | 2949 | 7610 |
| **B100ml-4-P** | | | | |
| | Class 0 | Class 1 | Class 0 | Class 1 |
| Class 0 | 9164 | 2945 | 8618 | 3491 |
| Class 1 | 3754 | 6775 | 3195 | 7334 |
| **B200ml-I** | | | | |
| | Class 0 | Class 1 | Class 0 | Class 1 |
| Class 0 | 3983 | 696 | 4093 | 586 |
| Class 1 | 2355 | 744 | 1745 | 1354 |
| **B200ml-P** | | | | |
| | Class 0 | Class 1 | Class 0 | Class 1 |
| Class 0 | 3973 | 686 | 2518 | 2141 |
| Class 1 | 2368 | 571 | 855 | 2084 |

Table 5: Confusion matrix obtained in the low quality dasates of Athletics with unprocessed and preprocessed data (SMOTE_LQD).

sion of the algorithms[16]. SMOTE_LQD improves SMOTE+ENN_LQD, but the difference is not too relevant. On the contrary, the use of NCL_LQD is not justified for this data, and it was expected to be. This can be explained by the fact that the datasets studied here are multiclass problems, but the conclusions in the literature were intended for binary problems. It also worths mentioning that the results of ENN_LQD are near to those of NCL_LQD (see the maximum of the optimistic estimations and the minimum of the pessimistic estimations in Table 6). ENN_LQD is worse than NCL_LQD in Dyslexic-12 and Dyslexic-12-01 For the last part, remark that CNN_LQD obtains, as it did in athletism datasets, the worst results.

## 6. Conclusions and Future Works

In this work we have considered the use of low quality imbalanced datasets in combination with certain GFSs that are able to use low quality data. The results have shown us that the behavior of a GFS for imprecise data can be improved with suitable generalizations of preprocessing algorithms for imbalanced data. This is because the uncertainty in the output label causes that many datasets become pos-

| | GFS | GFS+<br>SMOTE_LQD | GFS+SMOTE+<br>ENN_LQD | GFS+<br>ENN_LQD | GFS+<br>NCL_LQD | GFS+<br>CNN_LQD |
|---|---|---|---|---|---|---|
| **Dataset** | $Acc_{Tst}$ | $Acc_{Tst}$ | $Acc_{Tst}$ | $Acc_{Tst}$ | $Acc_{Tst}$ | $Acc_{Tst}$ |
| **Dyslexic-12** | | | | | | |
| M=[0,1,2,4]<br>N=[1,2,2,1] | [0.410,0.557] | [**0.453,0.578**] | [0.434,0.568] | [0.411,0.540] | [0.425,0.551] | [0.393,0.508] |
| **Dyslexic-12-01** | | | | | | |
| M=[0,1,2]<br>N=[1,2,1] | [0.524,0.656] | [0.550,0.663] | **[0.551,0.670]** | [0.493,0.595] | [0.521,0.62] | [0.472,0.556] |
| **Dyslexic-12-12** | | | | | | |
| M=[0,1,2]<br>N=[2,1,2] | [0.443,0.614] | [**0.484,0.645**] | [0.477,0.642] | [0.418,0.674] | [0.445,0.577] | [0.388,0.516] |
| **Mean** | [0.459,0.609] | **[0.496,0.629]** | [0.488,0.627] | [0.441,0.603] | [0.463,0.582] | [0.418,0.527] |

Table 6: Means of 100 repetitions of the GFS from low quality datasets of Dyslexic with 4 labels/variable with the original dataset and preprocessed with several preprocessing mechanism.

sibly imbalanced datasets, as there exist imbalanced selections of the imprecise data that are compatible with our incomplete knowledge of the problem.

Our experimentation concludes that the generalizations called SMOTE_LQD and SMOTE+ENN_LQD are a balanced choice for a vast majority of problems, as they tend to improve the results of the classification and rarely degrade the results obtained if the preprocessing stage is obviated. Furthermore, we have shown that the ranking of the different generalized alternatives is coherent with the ranking obtained by other authors for crisp data and GFS for binary problems. Also in multiclass problems this ranking holds to a certain degree, being remarkable the good behavior of SMOTE_LQD and SMOTE+ENN_LQD.

In future works, we intend to incorporate information about the confusion matrix of the minimum error-based GFS into the preprocessing algorithm. This information can be used to fine tune the synthesis of instances in combination with a particular GFS. We have also observed that multiclass datasets might be better suited for an internal approach that takes into account the cost of misclassification for each pair of classes (i.e. a minimum risk-based approach). In the last place, we think possible that, in those cases where the output variable is vague with high probability, and therefore we are not sure that the dataset is imbalanced, some techniques used in semi-supervised learning can be introduced in the preprocessing stage.

## Acknowledgments

This work was supported by the Spanish Ministry of Education and Science, under grants TIN2008-06681-C06-04, TIN2007-67418-C03-03, and by Principado de Asturias, PCTI 2006-2009.

1. Batista G., Prati R., Monard M., A study of the behaviour of several methods for balancing machine learning training data. SIGKDD Explorations 6 (1), 20-29 (2004).
2. Berger, J. Statistical decision theory and Bayesian Analysis. Springer-Verlag.
3. Bradley, A.P., The use of the area under the ROC curve in the evaluation ofmachine learning algorithms, Pattern Recognition 30(7) 1145–1159 (1997)
4. Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P., SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligent Research 16, 321-357 (2002).
5. Chawla N.V., Japkowicz N., Kolcz A., Editorial: Special issue on learning from imbalanced data sets. SIGKDD Explorations 6 (1), 1-6 (2004).
6. Chen S. H, Operations on fuzzy numbers with function principal. Tamkang Journal of Management Sciences, 6(1), 13-25.(1985)
7. Cheng C.H., A new approach for ranking fuzzy numbers by distance method. Fuzzy Sets and Systems 95 (1998) 307-317.
8. Chen S. J., Chen S. M., A new method for handling multicriteria fuzzy decision making problems using

FN-IOWA operators. Cybernatics and Systems, 34, 109-137. (2003)

9. Chen S. J., Chen S. M., Fuzzy risk analysis based on the ranking of generalized trapezoidal fuzzy numbers. Applied Intelligence, 26(1), 1-11. (2007)
10. Chen S. H., Ranking generalized fuzzy number with graded mean integration. In Proceedings of the eighth international fuzzy systems association world congress, Vol. 2. (pp. 899-902) (1999).
11. Crockett K., Bandar Z., O'Shea J., On producing balanced fuzzy decision tree classifiers. IEEE Internat. Conf. on Fuzzy Systems 1756-1762, 2006.
12. Chu T. C., Tsao C. T., Ranking fuzzy numbers with an area between the centroid point and original point. Computers and Mathematics with Applications, 43, 111-117 (2002)
13. Dmochowski, J. P., Sajda, P., Parra, L. C. Maximum Likelihood in Cost-Sensitive Learning: Model Specification, Approximators, and Upper Bounds. Journal of Machine Learning Research. (2010) In press.
14. Elkan, C., The Foundations of Cost-Sensitive Learning. Proceedings of the IJCAI-01. 973-?978. 2001.
15. Fawcett T., Provost F.J., Adaptive fraud detection. Data Mining Knowledge Discovery: 1 (3) 291-316 (1997).
16. Fernández A., Garcia S., del Jesús M.J., Herrera F., A study behaviour of linguistic fuzzy rule based classification system in the framework of imbalanced data-sets. Fuzzy Sets and Systems 159, 2378-2398 (2008).
17. Fernández A., del Jesús M.J., Herrera F., On the influence of an adaptive inference system in fuzzy rule based classification systems for imbalanced data-sets. Expert Systems with Applications 36, 9805-9812 (2009).
18. Fernández A., del Jesús M.J., Herrera F., Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. International Journal of Approximate Reasoning 50, 561-577 (2009).
19. Fernández A., del Jesús M.J., Herrera F., On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets. Information Sciences. DOI: 10.1016/j.ins.2009.12.014 (2010).
20. Hart P., The condensed nearest neighbor rule. IEEE Trans. Inform. Theory 14, 515-516 (1968).
21. Japkowicz N., Stephen S., The class imbalance problem: a systematic study. Intelligent Data Anal. 6 (5), 429-450, 2002.
22. Jain R., Decision-making in the presence of fuzzy variables, IEEE Trans. Systems Man and Cybernet. SMC- 6, 698-703, (1976).
23. Jain R., A procedure for multi-aspect decision making using fuzzy sets, Internat. J. Systems Sci. 8, 1-7, (1978).
24. Kilic K., Uncu O., Türksen I.B., Comparison of different strategies of utilizing fuzzy clustering in structure identification. Information Sciences 177 (23), 5153-5162 (2007).
25. Kubat M., Holte R., Matwin S., Learning when Negative Examples Abound. Proccedings of the 9th European Conference on Machine Learning. ECML (1997).
26. Kubat M., Matwin S., Addressing the curse of imbalanced training sets: one-sided selection. Internat. Conf. Machine Learning, 170-186 (1997).
27. Laurikkala J., Improving identification of difficult small classes by balancing class distribution. T.R. A-2001-2, University of Tampere (2001).
28. Liang C., Wu J., Zhang J., Ranking indices and rules for fuzzy numbers based on gravity center point. Paper presented at the 6th World Congress on Intelligent Control and Automation, Dalian, China.(2006)
29. Mazurowski M., Habas P., Zurada J., Lo J., Baker J., Tourassi G., Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Networks 21 (2-3), 427-436 (2008).
30. Murakami S., Maeda S., Imamura S., Fuzzy decision analysis on the development of centralized regional energy control system, IFAC Syrup. on Fuzzy Inform. Knowledge Representation and Decision Anal., 363-368 (1983).
31. Palacios, A., Couso, I., Sánchez, L. Future performance modeling in athletism with low quality data-based GFSs. (2010). In press.
32. Palacios, A., Sánchez, L., Couso, I. Diagnosis of dyslexia from vague data with Genetic Fuzzy System., 51, 993-1009. (2010)
33. Peng X., King I., Robust BMPM training based on second-order cone programming and its application in medical diagnosis, Neural Networks 21 (2-3), 450-457 (2008).
34. Phua C., Alahakoon D., Lee V., Minority report in fraud detection: classification of skewed data. SIGKDD Explorations Newsletter 6 (1), 50-59, (2004)
35. Provost F., Fawcett T., Robust classification systems for imprecise enviroments. In: Proc. AAAI pp 706-713 (1998).
36. Ramli N., Mohamad D., A comparative analysis of centroid methods in ranking fuzzy numbers. European Journal of Scientific Research, 28 (3): 492-501 (2009)
37. Shieh B.S., An approach to centroids of fuzzy numbers. International Journal of Fuzzy Systems, 9 (1), 51-54.(2007)
38. Soler V., Cerquides J., Sabria J., Roig J., Prim M., Imbalanced datasets classification by fuzzy rule extraction and genetic algorithms. IEEE Internat. Conf. Data Mining –Workshops, 330-336, 2006.

39. Sun, Y., Wong, A. K. C., Kemel, M. Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence 23, 4, 687–719. (2009)
40. Tomek I., Two modifications of cnn. IEEE Trans. Systems Man Comm. 6, 769-772 (1976)
41. Visa S., Ralescu A., Learning imbalanced and overlapping classes using fuzzy sets. Internat. Conf. Machine Learning –Workshop on Learning from Imbalanced Datasets II, 2003.
42. Visa S., Ralescu A., The effect of imbalanced data class distribution on fuzzy classifiers–experimental study. IEEE Internat. Conf. on Fuzzy Systems, 749-754, 2005.
43. Wang Y. J., Lee H. S., The revised method of ranking fuzzy numbers with an area between the centroid and original points. Computers and Mathematics with Applications, 55, 2033-2042.(2008)
44. Weiss G., Mining with rarity: a unifying framework. SIGKDD Explorations 6 (1), 7-19 (2004).
45. Wilson D.R., Asymptotic properties of nearest neighbour rules using edited data. IEEE Trans. Systems Man Comm. 2(3), 408-421 (1972).
46. Xu L., Chow M., Taylor L., Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-algorithm. IEEE Trans. Power Systems 22(1), 164-171, 2007.