Pattern Recognition 46 (2013) 355–364

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification

José A. Sáez [a,*], Julián Luengo [b], Francisco Herrera [a]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, CITIC-UGR, Granada 18071, Spain
[b] Department of Civil Engineering, LSI, University of Burgos, Burgos 09006, Spain

ARTICLE INFO

Article history:
Received 8 March 2012
Received in revised form
6 July 2012
Accepted 14 July 2012
Available online 23 July 2012

Keywords:
Classification
Noisy data
Noise filtering
Data complexity measures
Nearest neighbor

ABSTRACT

Classifier performance, particularly of instance-based learners such as *k*-nearest neighbors, is affected by the presence of noisy data. Noise filters are traditionally employed to remove these corrupted data and improve the classification performance. However, their efficacy depends on the properties of the data, which can be analyzed by what are known as data complexity measures. This paper studies the relation between the complexity metrics of a dataset and the efficacy of several noise filters to improve the performance of the nearest neighbor classifier. A methodology is proposed to extract a rule set based on data complexity measures that enables one to predict in advance whether the use of noise filters will be statistically profitable. The results obtained show that noise filtering efficacy is to a great extent dependent on the characteristics of the data analyzed by the measures. The validation process carried out shows that the final rule set provided is fairly accurate in predicting the efficacy of noise filters before their application and it produces an improvement with respect to the indiscriminate usage of noise filters.

1. Introduction

Real-world data is commonly affected by noise [1,2]. The building time, complexity and, particularly, the performance of the model, are usually deteriorated by noise in classification problems [3–5]. Several learners, e.g., C4.5 [6], are designed taking these problems into account and incorporate mechanisms to reduce the negative effects of noise. However, many other methods ignore these issues. Among them, instance-based learners, such as *k*-nearest neighbors (*k*-NN) [7–9], are known to be very sensitive to noisy data [10,11].

In order to improve the classification performance of noise-sensitive methods when dealing with noisy data, noise filters [12–14] are commonly applied. Their aim is to remove potentially noisy examples before building the classifier. However, both correct examples and examples containing valuable information can also be removed. This fact implies that these techniques do not always provide an improvement in performance. As indicated by Wu and Zhu [1], the success of these methods depends on several circumstances, such as the kind and nature of the data errors, the quantity of noise removed or the capabilities of the classifier to deal with the loss of useful information related to the filtering. Therefore, the efficacy of noise filters, i.e., whether their usage causes an improvement in classifier performance, depends on the noise-robustness and the generalization capabilities of the classifier used, but it also strongly depends on the characteristics of the data.

Data complexity measures [15] are a recent proposal to represent characteristics of the data which are considered difficult in classification tasks, e.g., the overlapping among classes, their separability or the linearity of the decision boundaries.

This paper proposes the computation of these data complexity measures to predict in advance when the usage of a noise filter will statistically improve the results of a noise-sensitive learner: the nearest neighbor classifier (1-NN). This prediction can help, for example, to determine an appropriate noise filter for a concrete noisy dataset – that filter providing a significant advantage in terms of the results – or to design new noise filters which select more or less aggressive filtering strategies considering the characteristics of the data. Choosing a noise-sensitive learner facilitates the checking of when a filter removes the appropriate noisy examples in contrast to a robust learner—the performance of classifiers built by the former is more sensitive to noisy examples retained in the dataset after the filtering process. In addition, this paper has the following objectives:

1. To analyze the relation between the characteristics of the data and the efficacy of several noise filters.
2. To find a reduced set of the most appropriate data complexity measures for predicting the noise filtering efficacy.

* Corresponding author. Tel.: +34 958 240598; fax: +34 958 243317.
E-mail addresses: smja@decsai.ugr.es, tschigorine@gmail.com (J.A. Sáez), jluengo@ubu.es (J. Luengo), herrera@decsai.ugr.es (F. Herrera).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.07.009

3. Even though each noise filter may depend on concrete characteristics of the data to work correctly, it would be interesting to identify common characteristics of the data under which most of the noise filters work properly.
4. To provide a set of interpretable rules which a practitioner can use to determine whether to use a noise filter with a classification dataset.

A web page with the complementary material of this paper is available at http://sci2s.ugr.es/filtering-efficacy. It includes the details of the experimentation, the datasets used, the performance results of the noise filters and the distribution of the data complexity metrics of the datasets.

The rest of this paper is organized as follows. Section 2 presents data complexity measures. Section 3 introduces the noise filters and enumerates those considered in this paper. Section 4 describes the method employed to extract the rules predicting the noise filtering efficacy. Section 5 shows the experimental study performed and the analysis of results. Finally, Section 6 enumerates some concluding remarks.

2. Data complexity measures

In this section, first a brief review of recent studies on data complexity metrics is presented (Section 2.1). Then, the measures of overlapping (Section 2.2), the measures of separability of classes (Section 2.3) and the measures of geometry (Section 2.4) used in this paper are described.

2.1. Recent studies on data complexity

There are some methods used in classification, either learner or preprocessing techniques, which work well with concrete datasets, while other techniques work better with different ones. This is due to the fact that each classification dataset has particular characteristics that define it. Issues such as the generality of the data, the inter-relationships among the variables and other factors are key for the results of such methods. An emergent field proposes the usage of a set of data complexity measures to quantify these particular sources of the problem on which the behavior of classification methods usually depends [15].

A seminal work on data complexity is [16], in which some complexity measures for binary classification problems are proposed, gathering metrics of three types: overlaps in feature values from different classes; separability of classes; and measures of geometry, topology and density of manifolds. Extensions can also be found in the literature, such as in the work of Singh [17], which offers a review of data complexity measures and proposes two new ones.

From these works, different authors attempt to address different data mining problems using these measures. For example, Baumgartner and Somorjai [18] define specialized measures for regularized linear classifiers. Other authors try to explain the behavior of learning algorithms using these measures, optimizing the decision tree creation in the binarization of datasets [19] or to analyze fuzzy-UCS and the model obtained when applied to data streams [20]. The data complexity measures have been referred to other related fields, such as gene expression analysis in Bioinformatics [21,22].

The research efforts in data complexity are currently focused on two fronts. The first aims to establish suitable problems for a given classification algorithm, using only the data characteristics, and thus determining their domains of competence. In this line of research recent publications, e.g., the works of Luengo and Herrera [23] and Bernadó-Mansilla and Ho [24], provide a first insight into the determination of an individual classifier's domains of competence. Parallel to this, Sánchez et al. [25] study the effect of data complexity on the nearest neighbor classifier. The relationships between the domains of competence of similar classifiers were analyzed by Luengo and Herrera [26], indicating that related classifiers benefit from common sources of complexity of the data.

Data complexity measures are increasingly used in order to characterize when a preprocessing stage will be beneficial to a subsequent classification algorithm in many challenging domains. García et al. [27] firstly analyzed the behavior of the evolutionary prototype selection strategy using one complexity measure based on overlapping. Further developments resulted in a characterization of when the preprocessing in imbalanced datasets is beneficial [28]. The data complexity measures can also be used online in the data preparation step. An example of this is the work of Dong [29], in which a feature selection algorithm based on complexity measures is proposed.

This paper follows the second research line. It aims to characterize when a filtering process is beneficial using the information provided by the data complexity measures. Noise will affect the geometry of the dataset, and thus the values of the data complexity metrics. It can be expected that such metrics will enable one to know in advance whether noise filters will be useful for the given dataset.

In this study, 11 of the metrics proposed by Ho and Basu [16] will be analyzed. In the following subsections, these measures, classified by their family, are briefly presented. For a deeper description of their characteristics, the reader may consult [16].

2.2. Measures of class overlapping

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

- F1—*maximum Fisher's discriminant ratio*: This is the value of Fisher's discriminant ratio of the attribute that enables one to better discriminate between the two classes, computed as

$$\mathrm{F1} = \max_{i=1,\ldots,d} \frac{(\mu_{i,1}-\mu_{i,2})^2}{\sigma_{i,1}^2+\sigma_{i,2}^2} \tag{1}$$

  where d is the number of attributes, and $\mu_{i,j}$ and $\sigma_{i,j}^2$ are the mean and variance of the attribute i in the class j, respectively.
- F2—*volume of the overlapping region*: This measures the amount of overlapping of the bounding boxes of the two classes. Let $\max(f_i,C_j)$ and $\min(f_i,C_j)$ be the maximum and minimum values of the feature f_i in the set of examples of class C_j, let minmax_i be the minimum of $\max(f_i,C_j),(j=1,2)$ and maxmin_i be the maximum of $\min(f_i,C_j),(j=1,2)$ of the feature f_i. Then, the measure is defined as

$$\mathrm{F2} = \prod_{i=1\ldots d} \frac{\mathrm{minmax}_i-\mathrm{maxmin}_i}{\max(f_i,C_1\cup C_2)-\min(f_i,C_1\cup C_2)} \tag{2}$$

- F3—*maximum feature efficiency*: This is the maximum fraction of points distinguishable with only one feature after removing unambiguous points falling outside of the overlapping region in this feature [30].

2.3. Measures of separability of classes

These give indirect characterizations of class separability. They assume that a class is made up of single or multiple manifolds

that form the support of the probability distribution of the given class. The shape, position and interconnectedness of these manifolds give hints of how well the two classes are separated, but they do not describe separability by design.

- L1—*minimized sum of error distance by linear programming*: This is the value of the objective function that tries to minimize a linear classifier obtained by the linear programming formulation proposed by Smith [31]. The method minimizes the sum of distances of error points to the separating hyperplane. The measure is normalized by the number of points in the problem and also by the length of the diagonal of the hyper-rectangular region enclosing all training points in the feature space.
- L2—*error rate of linear classifier by linear programming*: This measure is the error rate of the linear classifier defined for L1, measured with the training set.
- N1—*rate of points connected to the opposite class by a minimum spanning tree*: N1 is computed using a minimum spanning tree [32], which connects all the points to their nearest neighbors. Then the number of points connected to the opposite class by an edge of this tree are counted. These are considered to be the points lying next to the class boundary. N1 is the fraction of such points over all points in the dataset.
- N2—*ratio of average intra/inter class nearest neighbor distance*: This is computed as

$$N2 = \frac{\sum_{i=0}^{m} intra(x_i)}{\sum_{i=0}^{m} inter(x_i)} \quad (3)$$

  where m is the number of examples in the dataset, $intra(x_i)$ the distance to its nearest neighbor within the class, and $inter(x_i)$ the distance to the nearest neighbor of any other class. This metric compares the within-class spread with the distances to the nearest neighbors of other classes. Low values of this metric suggest that the examples of the same class lie close in the feature space, whereas large values indicate that the examples of the same class are dispersed.
- N3—*error rate of the 1-NN classifier*: This is the error rate of a nearest neighbor classifier estimated by the leave-one-out method. This measure denotes how close the examples of different classes are. Low values of this metric indicate that there is a large gap in the class boundary.

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

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

- L3—*nonlinearity of a linear classifier by linear programming*: Hoekstra and Duin [33] propose a measure for the nonlinearity of a classifier with respect to a given dataset. Given a training set, the method first creates a test set by linear interpolation (with random coefficients) between randomly drawn pairs of points from the same class. Then, the error rate of the classifier (trained by the given training set) on this test set is measured.
- N4—*nonlinearity of the 1-NN classifier*: The error is calculated for a nearest neighbor classifier. This measure is for the alignment of the nearest neighbor boundary with the shape of the gap or overlap between the convex hulls of the classes.
- T1—*ratio of the number of hyperspheres, given by* ϵ-neighborhoods, *by the total number of points*: The local clustering properties of a point set can be described by an ϵ-neighborhood pretopology [34]. Instance space can be covered by ϵ-neighborhoods by means of hyperspheres (the procedure to compute them can be found in [16]). A list of such hyperspheres needed to cover the two classes is a composite description of the shape of the classes. The number and size of the hyperspheres indicate how much the points tend to be clustered in hyperspheres or distributed in thinner structures. In a problem where each point is closer to points of the other class than points of its own, each hypersphere is retained and is of a low size. T1 is the normalized count of the retained hyperspheres by the total number of points.

3. Corrupted data treatment by noise filters

Noise filters are preprocessing mechanisms designed to detect and eliminate noisy examples in the training set. The result of noise elimination in preprocessing is a reduced and improved training set which is then used as an input to a machine learning algorithm.

There are several of these filters based on using the distance between examples to determine their similarity and create neighborhoods. These neighborhoods are used to detect suspicious examples which can then be eliminated. The Edited Nearest Neighbor [12] or the Prototype Selection based on Relative Neighborhood Graphs [35] are some examples of methods that can be found within this group of noise filters.

Another group of noise filters creates classifiers over several subsets of the training data in order to detect noisy examples. Brodley and Friedl [13] trained multiple classifiers built by different learning algorithms, such as k-NN [7], C4.5 [6] and a Linear Discriminant Analysis [36], from a corrupted dataset and then used them to identify mislabeled data, which are characterized as the examples that are incorrectly classified by the multiple classifiers. Similar techniques have been widely developed considering the building of several classifiers with the same learning algorithm [37,38]. Instead of using multiple classifiers learned from the same training set, Gamberger et al. [37] suggest a Classification Filter (CF) approach, in which the training set is partitioned into n subsets, then a set of classifiers is trained from the union of any $n-1$ subsets; those classifiers are used to classify the examples in the excluded subset, eliminating the examples that are incorrectly classified.

The noise filters analyzed in this paper are shown in Table 1. They have been chosen due to their good behavior with many real-world problems.

4. Obtaining rules to predict the noise filtering efficacy

In order to provide a rule set based on the characteristics of the data which enables one to predict whether the usage of noise filters will be statistically beneficial, the methodology shown in

Table 1
Noise filters employed in the experimentation.

Filter	Reference	Abbreviation
Classification filter	[37]	CF
Cross-validated committees filter	[38]	CVCF
Ensemble filter	[13]	EF
Edited nearest neighbor with estimation of probabilities threshold	[39]	ENNTh
Edited nearest neighbor	[12]	ENN
Iterative-partitioning filter	[40]	IPF
Nearest centroid neighborhood edition	[41]	NCNEdit
Prototype selection based on relative neighborhood graphs	[35]	RNG

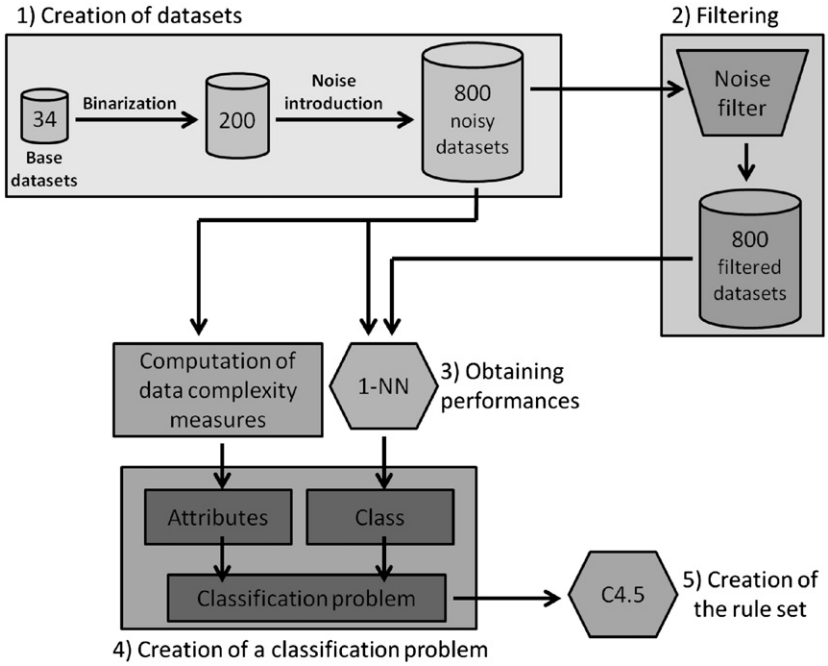

Fig. 1. Methodology to obtain the rule set predicting the noise filtering efficacy.

Table 2
Base datasets and their number of instances (#INS), attributes (#ATT) and classes (#CLA). (R/I/N) refers to the number of real, integer and nominal attributes.

Dataset	#INS	#ATT (R/I/N)	#CLA	Dataset	#INS	#ATT (R/I/N)	#CLA
australian	690	14 (3/5/6)	2	led7digit	500	7 (7/0/0)	10
balance	625	4 (4/0/0)	3	mammographic	830	5 (0/5/0)	2
banana	5300	2 (2/0/0)	2	monk-2	432	6 (0/6/0)	2
bands	365	19 (13/6/0)	2	mushroom	5644	22 (0/0/22)	2
bupa	345	6 (1/5/0)	2	pima	768	8 (8/0/0)	2
car	1728	6 (0/0/6)	4	ring	7400	20 (20/0/0)	2
chess	3196	36 (0/0/36)	2	saheart	462	9 (5/3/1)	2
contraceptive	1473	9 (0/9/0)	3	sonar	208	60 (60/0/0)	2
crx	653	15 (3/3/9)	2	spambase	4597	57 (57/0/0)	2
ecoli	336	7 (7/0/0)	8	tae	151	5 (0/5/0)	3
flare	1066	11 (0/0/11)	6	tic-tac-toe	958	9 (0/0/9)	2
glass	214	9 (9/0/0)	7	titanic	2201	3 (3/0/0)	2
hayes-roth	160	4 (0/4/0)	3	twonorm	7400	20 (20/0/0)	2
heart	270	13 (1/12/0)	2	wdbc	569	30 (30/0/0)	2
housevotes	232	16 (0/0/16)	2	wine	178	13 (13/0/0)	3
ionosphere	351	33 (32/1/0)	2	wisconsin	683	9 (0/9/0)	2
iris	150	4 (4/0/0)	3	yeast	1484	8 (8/0/0)	10

Fig. 1 has been designed. The complete process[1] is described as follows.

1. 800 different classification datasets are built as follows (these are common to all noise filters):
   - The 34 datasets shown in Table 2 have been selected from the KEEL-dataset repository[2] [42].
   - 200 binary datasets – with more than 100 examples in each one – are built from these 34 datasets. Multi-class datasets are used to create other binary datasets by means of the selection and/or combination of their classes. Only problems with two classes are considered as the data complexity measures are only well defined to work on binary problems. The amount of examples of the two classes has been taken into account in order to create the datasets; they are intended to be as similar as possible. Let *IR* be the fraction between the number of examples of the majority and the minority class—formally known as *imbalanced ratio* [43]. In order to control the size of both classes, only datasets with a low imbalanced ratio were created, specifically with $1 \leq IR \leq 2.25$. Therefore, the size of both classes is sufficiently similar. This prevents filtering methods from deleting all the examples from the minority class, which can occur if a high imbalanced ratio is present in the data since the filtering methods used do not take into account the class imbalance and may consider these examples to be noise.
   - Finally, in order to study the behavior of the noise filters in several circumstances, several noise levels x (0%, 5%, 10% and 15%) are introduced into these 200 datasets, resulting in 800 datasets. Noise is introduced in the same way as in [3], a reference paper in the framework of noisy data in classification. Each attribute A_i is corrupted separately: x% of the examples are chosen and the A_i value of each of these

[1] The datasets used in this procedure and the performance results of 1-NN – with and without the usage of noise filters – can be found on the web page of this paper.

[2] http://www.keel.es/datasets.php.

examples is assigned a random value of the domain of that attribute following a uniform distribution. One must take into account that these 200 datasets may contain noise, so the real noise level after the noise introduction process may be higher.

2. These 800 datasets are filtered with a noise filter, leading to 800 new filtered datasets.
3. The test performance of 1-NN [7,44,45] on each of the 800 datasets, both with and without the application of the noise filter, is computed. The estimation of the classifier performance is obtained by means of three runs of a 10-fold cross-validation and their results are averaged. The AUC metric [46] is used due it being commonly employed when working with binary datasets and the fact that it is less sensitive to class imbalance. The performance estimation is used to check which datasets are improved in their performance by 1-NN when using the noise filter.
4. A classification problem is created with each example being one of the datasets built and in which:
   - The attributes are the 11 data complexity metrics for each dataset. The distribution of the values of each data complexity measure can be found on the web page with complementary material for this paper.
   - The class label represents whether the usage of the noise filter implies a statistically significant improvement of the test performance. Wilcoxon's statistical test [47] – with a significance level of $\alpha = 0.1$ – is applied to compare the performance results of the 3×10 test folds with and without the usage of the noise filter. Depending on whether the usage of the noise filter is statistically better than the lack of filtering, each example is labeled as *positive* or *negative*, respectively.
5. Finally, similar to the method of Orriols-Puig and Casillas [20], the C4.5 algorithm [6] is used to build a decision tree on the aforementioned classification problem, which can be transformed into a rule set. The performance estimation of this rule set is obtained using a 10-fold cross-validation. By means of the analysis of the decision trees built by C4.5, it is possible to check which are the most important data complexity metrics to predict the noise filtering efficacy, i.e., those in the top levels of the tree and appearing more times, and their performance examining the test results.

5. Experimental study

The experimentation is organized in five different parts, each one in a different subsection and with a different objective:

1. *To check to what extent the noise filtering efficacy can be predicted using data complexity measures* (Section 5.1). In order to do this, the procedure described in Section 4 is followed with each noise filter. Thus, a rule set based on all the data complexity measures is learned to predict the efficacy of each noise filter. Its performance, which is estimated using a 10-fold cross-validation, gives a measure of the relation existing between the data complexity metrics and the noise filtering efficacy—a higher performance will imply a stronger relation.
2. *To provide a reduced set of data complexity metrics that best determine whether to use a noise filter and do not cause the prediction capability to deteriorate* (Section 5.2). The decision trees built in the above step by C4.5 are analyzed, studying two elements:
   - The *order*, from 1 to 11, in which the first node corresponding with each data complexity metric appears in the decision tree, starting from the root. This order is averaged over the 10 folds.
   - The *percentage* of nodes of each data complexity metric in the decision tree, averaged over the 10 folds.

   This analysis will provide the better discriminating metrics and those appearing more times in the decision trees—they are not necessarily placed in the top positions of the tree but are still important to discriminate between the two classes. In this way, the rule sets obtained in the above step are simplified and thus become more interpretable.
3. *To find common characteristics of the data on which the efficacy of all noise filters depends* (Section 5.3). Each noise filter may depend on concrete values of the data complexity metrics, i.e., on concrete characteristics of the data, to work properly. However, it is interesting to investigate whether there are common characteristics of the data under which all noise filters work properly. To do this, the rule set learned with each noise filter will be applied to predict the efficacy of the rest of the noise filters. The rule set achieving the highest performance predicting the efficacy of the different noise filters will have rules more similar to the rest of noise filters, i.e., the rules will cover similar areas of the domain.
4. *To provide the rule set which works best predicting the noise filtering efficacy of all the noise filters* (Section 5.4). The study of the above point will provide the rule set which best represents the characteristics under which the majority of the noise filters work well. The behavior of these rules with each noise filter will be analyzed in this section, paying attention to the *coverage* of each rule—the percentage of examples covered, and its *accuracy*—the percentage of correct classifications among the examples covered.
5. *To perform an additional validation of the chosen rule set* (Section 5.5). Even though the behavior of each rule set is validated using a 10-fold cross-validation in each of the above steps, a new validation phase with new datasets is performed in this section. These datasets are used to check if the chosen rule set is really more advantageous than the indiscriminate application of the noise filters to all the datasets.

5.1. Data complexity measures and noise filtering efficacy

The procedure described in Section 4 has been followed with each one of the noise filters. Table 3 shows the performance results of the rule sets obtained with C4.5 on the training and test sets for each noise filter when predicting the noise filtering efficacy, i.e., when discriminating between the aforementioned *positive* and *negative* classes.

The training performance is very high for all the noise filters – it is close to the maximum achievable performance – and there are no differences between the eight noise filters. The test performance results, although not at the same level as

Table 3
Performance results of C4.5 predicting the noise filtering efficacy (11 data complexity measures used).

Noise filter	Training	Test
CF	0.9979	0.8446
CVCF	0.9966	0.8353
EF	0.9948	0.8176
ENNTh	0.9958	0.8307
ENN	0.9963	0.8300
IPF	0.9973	0.8670
NCNEdit	0.9945	0.8063
RNG	0.9969	0.8369
Mean	**0.9963**	**0.8335**

Table 4
Averaged order of the data complexity measures in the decision trees.

Metric	CF	CVCF	EF	ENNTh	ENN	IPF	NCNEdit	RNG	Mean
F1	3.70	4.80	5.90	8.60	6.40	4.50	6.00	8.20	**6.01**
F2	1.40	1.00	1.00	1.00	1.00	1.00	1.50	1.00	**1.11**
F3	2.50	3.40	10.10	5.80	4.10	3.30	7.20	4.50	**5.11**
N1	10.50	9.90	9.10	10.30	8.40	7.10	8.10	8.50	8.99
N2	6.20	2.00	3.30	8.00	2.30	3.00	4.60	2.70	**4.01**
N3	8.80	8.50	7.80	11.00	7.00	9.50	7.90	8.70	8.65
N4	7.40	9.70	9.90	7.20	11.00	10.50	8.20	5.60	8.69
L1	9.20	10.00	7.90	9.70	11.00	6.00	9.40	9.60	9.10
L2	8.10	6.80	9.30	8.40	10.30	10.00	11.00	10.50	9.30
L3	7.80	8.70	4.60	8.60	11.00	5.90	7.80	8.40	7.85
T1	6.70	6.80	5.20	3.50	4.80	11.00	6.30	4.50	**6.10**

the training results, are also noteworthy. All of them have more than 0.8 success, with the averaged test performance of all the noise filters higher than 0.83. These results show that noise filtering efficacy can be predicted with a good performance by means of data complexity measures. Therefore, a clear relation can be seen between both concepts, i.e., data complexity metrics and filtering efficacy.

5.2. Metrics that best predict the noise filtering efficacy

In order to find the subset of data complexity measures that enables the best decision to be made of whether a noise filter should be used, the decision trees built by C4.5 in the previous section are analyzed. Table 4 shows the averaged *order* of each data complexity measure in which it appears in the decision trees built for each noise filter.

These results show that the three best measures are generally F2, N2 and F3:

- F2 is the first measure for all noise filters.
- N2 is placed in six of the eight noise filters as the second metric.
- F3 is placed between the second and third positions in another six of the eight noise filters.

The following two measures in importance are T1 and F1:

- T1 appears in seven of the eight noise filters between the second and fifth positions.
- F1 appears in six of the eight noise filters between the third and fifth positions.

The rest of the measures have a lower discriminative power, due their positions being worse. Averaged results for all noise filters also support these conclusions. Therefore, the aforementioned measures (F2, N2, F3, T1 and F1) are the most important for all the noise filters, even though the concrete order can vary slightly from some filters to others.

From these results, the measures of overlapping among the classes (F1, F2 and F3) are the group of metrics that most influence predictions of the filtering efficacy. The filtering efficacy is particularly dependent on the volume of the overlapping region (F2) and, to a lesser degree, on the rest of the overlapping metrics (F3 and F1) which, using different methods, compute the discriminative power of the attributes. The dispersion of the examples within each class (N2) and the shape of the classes and the complexity of the decision boundaries (T1) must also be taken into account to predict the filtering efficacy. In short, all these metrics provide information about the shape of the classes and the overlapping among them, which may be key factors in the success of any noise filtering technique.

Since the efficacy of the noise filters has been studied over the results of the 1-NN classifier, one could expect a greater influence of measures based on 1-NN, such as N3 and N4. These measures are based on the error rate of the 1-NN classifier –the former is computed on the training set whereas the latter is computed on an artificial test set. It is important to point out that 1-NN is very sensitive to the closeness of only one example to others belonging to a different class [16,25] and a similar error rate may be due to multiple situations where the filtering may be beneficial or not, for example:

1. Existence of isolated noisy examples.
2. A large overlapping between the classes.
3. Closeness between the classes (although overlapping does not exist).

A noise filtering method is likely to be beneficial in the first scenario because isolated noisy examples are likely to be identified and removed, improving the final performance of the classifier. However, the situation is not so clear in the other two scenarios: the filtering may delete important parts of the domain and disturb the boundaries of the classes or, on the contrary, it may clean up the overlapping region and create more regular class boundaries [1,48]. Therefore, the multiple causes on which the error rate of 1-NN depends imply that measures based on it, such as N3 and N4, are not always good indicators of the noise filtering efficacy.

Table 5 shows the *percentage* of nodes referring to each data complexity measure in the decision trees for each of the noise filters. These results provide similar conclusions to those of the *order* results, with the most representative measures again being F1, F2, F3, N2 and T1, while the rest of the measures have lower percentages.

The *order* and *percentage* results show that the measures F1, F2, F3, N2 and T1 are the most discriminative and have a higher number of nodes in the decision trees. It is aimed to attain a reduced set, from among these five metrics, that enables filtering efficiency to be predicted without a loss in accuracy with respect to all the measures. In order to avoid the study of all the existing combinations of the five metrics, the following experimentation is mainly focused on the measures F2, N2 and F3, the most discriminative ones—since the *order* results can be considered more important than the *percentage* results. The incorporation into this set of T1, F1 or both is also studied. The prediction capability of the measure F2 alone, since is the most discriminative one, is also shown. All these results are presented in Table 6.

The training results of these combinations do not change with respect to the usage of all the metrics. However, the test performance results improve in many cases the results of using all the metrics, particularly in the cases of F2–N2–F3–T1–F1 and

Table 5
Percentage of the number of nodes of each data complexity measure in the decision trees.

Metric	CF	CVCF	EF	ENNTh	ENN	IPF	NCNEdit	RNG	Mean
F1	22.45	21.24	14.94	8.47	17.07	20.66	18.67	5.71	**16.15**
F2	15.31	11.50	14.94	18.64	12.20	9.09	14.67	9.52	**13.23**
F3	16.33	23.01	2.30	13.56	20.73	23.14	12.00	18.10	**16.15**
N1	2.04	2.65	3.45	1.69	8.54	8.26	5.33	3.81	4.47
N2	8.16	11.50	17.24	6.78	19.51	9.92	16.00	19.05	**13.52**
N3	5.10	5.31	5.75	0.00	6.10	3.31	6.67	4.76	4.62
N4	8.16	3.54	2.30	13.56	0.00	0.83	6.67	12.38	5.93
L1	3.06	2.65	8.05	3.39	0.00	7.44	2.67	5.71	4.12
L2	6.12	7.08	5.75	8.47	1.22	2.48	0.00	0.95	4.01
L3	5.10	3.54	16.09	6.78	0.00	14.88	9.33	4.76	7.56
T1	8.16	7.96	9.20	18.64	14.63	0.00	8.00	15.24	**10.23**

Table 6
Performance results of C4.5 predicting the noise filtering efficacy (measures used: F2, N2, F3, T1 and F1).

Noise filter	F2		F2–N2–F3–T1–F1		F2–N2–F3–F1		F2–N2–F3–T1		F2–N2–F3	
	Training	Test	Training	Test	Training	Test	Training	Test	Training	Test
CF	0.9991	0.7766	0.9975	0.8848	0.9986	0.8623	0.9983	0.8949	0.9972	0.8713
CVCF	1.0000	0.5198	0.9997	0.8102	0.9983	0.7943	0.9994	0.8165	0.9977	0.8152
EF	1.0000	0.7579	0.9993	0.8102	0.9991	0.8101	0.9997	0.8297	0.9997	0.8421
ENNTh	1.0000	0.8419	0.9996	0.8309	0.9996	0.8281	0.9907	0.8052	0.9992	0.8302
ENN	1.0000	0.7361	0.9928	0.8942	0.9935	0.8662	0.9966	0.8948	0.9967	0.7946
IPF	1.0000	0.7393	0.9975	0.8378	0.9989	0.8119	0.9986	0.8019	0.9985	0.7725
NCNEdit	0.9981	0.8024	0.9977	0.8164	0.9982	0.8231	0.9983	0.8436	0.9912	0.8136
RNG	0.9993	0.7311	0.9967	0.8456	0.9983	0.8086	0.9989	0.8358	0.9980	0.7754
Mean	0.9996	0.7381	0.9976	0.8413	0.9981	0.8256	0.9976	0.8403	0.9973	0.8144

Table 7
Ranks computed by Wilcoxon's test $R+/R-$, representing the ranks obtained by the combination of the row and the column, respectively. *All* refers to the usage of all the complexity metrics.

Metrics	F2–N2–F3	F2–N2–F3–T1	F2–N2–F3–F1	F2–N2–F3–F1–T1	All
F2–N2–F3	–	6/30	12/24	8/28	11/25
F2–N2–F3–T1	30/6	–	30/6	19/17	20/16
F2–N2–F3–F1	24/12	6/30	–	3/33	13/23
F2–N2–F3–F1–T1	28/8	17/19	33/3	–	23/13
All	25/11	16/20	23/13	13/23	–

F2–N2–F3–T1. This is because the unnecessary measures to predict the filtering efficacy which can introduce a bias into the datasets have been removed. However, the usage of the measure F2 alone to predict the noise filtering efficacy with a good performance can be discarded, since its results are not good enough compared with the cases where more than one measure is considered. This fact reflects that the usage of single measures does not provide enough information to achieve a good filtering efficacy prediction result. Therefore, it is necessary to combine several measures which examine different aspects of the data.

In order to determine which combination of measures is chosen as the most suitable one, Wilcoxon's statistical test is performed, comparing the test results of Tables 3 and 6 of each noise filter. Table 7 shows the ranks obtained by each combination of metrics.

From these results, the combinations of metrics F2–N2–F3–T1 and F2–N2–F3–T1–F1 are noteworthy. Removing some data complexity metrics improves the performance with respect to all the metrics. However, it is necessary to retain a minimum number of metrics representing as much information as possible. Note that these two sets contain measures of three different types: overlapping, separability of classes and geometry of the dataset. Therefore, even though the differences are not significant in all cases, the combination with more ranks and a lower number of measures, i.e., F2–N2–F3–T1, can be considered the most appropriate and will be chosen for a deeper study.

5.3. Common characteristics of the data on which the efficacy of the noise filters depends

From the results shown in Table 6, the rules learned with any noise filter can be used to accurately predict filtering efficacy because they obtain good test performance results. However, these rules should be used to predict the behavior of the filter from which they have been learned.

It would be interesting to provide a single rule set, better adapting the behavior of all the noise filters. In order to do this, the rules learned to predict the behavior of one filter will be tested to predict the behavior of the rest of the noise filters (see Table 8). From these results, the prediction performance of the rules learned for the RNG filter is clearly the more general, since they are applicable to the rest of the noise filters obtaining the best prediction results—see the last column with an average of 0.8786. Therefore, this rule set has rules that are more similar to the rest of the noise filters and thus, it represents better the common characteristics on which the efficacy of all noise filters depends.

5.4. Analysis of the chosen rule set

The rule set chosen to predict the filtering efficacy of all the noise filters is shown in Table 9. The analysis of such rules is shown in Table 10, where the *coverage* (Cov) and the *accuracy* (Acc) of each rule is shown.

These results show that the rules with the highest coverage in predicting the behavior of all noise filters are R6, R5 and R10. Moreover, the rules predicting the *positive* examples have a very high accuracy rate, close to 100%. The rule R5 has the highest coverage among the rules predicting the *negative* class, although its accuracy is a bit lower than that of the rules R6 and R10. This could be due to the fact that the datasets in which the application of a noise filter implies a disadvantage are more widely dispersed in the search space and, that being so, creating general rules is more complex. The rest of the rules have a lower coverage, although their accuracy is generally high, so they are more specific rules.

The rules R6 and R10 are characterized by having a value of F2 higher than 0.43. Moreover, the rule R6 requires a value of T1 lower than 0.9854, i.e., a large part of the domain of the metric T1. However, as reflected in the experimentation in [16] and also on the web page with complementary material for this paper, a large number of datasets have a T1 value of around 1. The incorporation, therefore, of the measure T1 into the rules and the multiple values between 0.9 and 1 of this metric in the antecedents should not be surprising.

By contrast, the rule R5 has a value of F2 lower than 0.43. Other metrics are also included in this rule, such as N2 with a value higher than 0.41 and F3 with a value higher than 0.1.

Table 8
Performance results of the rules learned with the method in the column predicting the efficacy of the noise filter in the row.

Noise filter	CF	CVCF	EF	ENN	ENNTh	IPF	NCNEdit	RNG
CF	–	0.8848	0.8631	0.9049	0.8114	**0.9230**	0.8590	0.9172
CVCF	0.8030	–	0.7656	0.8884	0.7373	0.9024	0.7747	**0.9115**
EF	0.8756	0.8044	–	0.8597	0.8540	0.8425	0.8824	**0.8901**
ENN	0.7795	0.8588	0.7804	–	0.7512	0.8161	0.7804	**0.8865**
ENNTh	0.7900	0.7681	0.8176	0.8083	–	0.8114	**0.8362**	0.8267
IPF	0.8455	**0.9092**	0.7922	0.8680	0.7164	–	0.7915	0.8694
NCNEdit	0.8313	0.7644	0.8462	0.7897	0.8120	0.8333	–	**0.8487**
RNG	0.7959	0.7988	0.8069	**0.8251**	0.7538	0.8128	0.8130	–
Mean	0.8173	0.8269	0.8103	0.8491	0.7766	0.8488	0.8196	**0.8786**

Table 9
Rule set chosen to predict the noise filtering efficacy.

Rule	F2	N2	T1	F3	Filter
R1	≤ 0.439587	≤ 0.264200	≤ 0.995100		Positive
R2	≤ 0.439587	≤ 0.264200	> 0.995100		Negative
R3	≤ 0.439587	(0.2642, 0.419400]			Negative
R4	≤ 0.439587	> 0.419400		≤ 0.101900	Positive
R5	≤ 0.439587	> 0.419400		> 0.101900	Negative
R6	> 0.439587		≤ 0.985400		Positive
R7	> 0.439587	≤ 0.298600	(0.985400, 0.994900]		Positive
R8	> 0.439587	(0.298600, 0.344700]	(0.985400, 0.994900]		Negative
R9	> 0.439587	≤ 0.344700	> 0.994900		Negative
R10	> 0.439587	(0.344700, 0.836984]	(0.985400, 0.996005]		Positive
R11	> 0.439587	(0.344700, 0.515300]	> 0.996005	≤ 0.294916	Negative
R12	> 0.439587	(0.515300, 0.836984]	> 0.996005	≤ 0.294916	Positive
R13	> 0.439587	(0.344700, 0.836984]	> 0.996005	> 0.294916	Negative
R14	> 0.439587	> 0.836984	> 0.985400	≤ 0.011076	Negative
R15	> 0.439587	> 0.836984	> 0.985400	> 0.011076	Positive

Table 10
Analysis of the behavior of the chosen rule set, which comes from the RNG filter, with all the noise filters.

Rule	CF		CVCF		EF		ENN		ENNTh		IPF		NCNEdit	
	Cov	Acc	Cov	Acc	Cov	Acc	Cov	Acc	Cov	Acc	Cov	Acc	Cov	Acc
R1	4.05	100.00	6.47	100.00	3.24	100.00	4.46	100.00	3.72	66.67	5.58	100.00	3.50	85.71
R2	1.21	33.33	1.29	33.33	0.46	0.00	1.34	33.33	1.24	100.00	1.20	33.33	1.00	50.00
R3	1.21	33.33	1.72	25.00	0.46	100.00	1.79	75.00	2.48	100.00	1.59	25.00	2.50	100.00
R4	3.64	100.00	3.02	100.00	2.31	100.00	1.34	100.00	1.65	25.00	3.19	100.00	2.00	75.00
R5	12.96	75.00	8.19	57.89	11.11	62.50	18.30	75.61	23.14	85.71	11.95	60.00	21.00	80.95
R6	38.06	98.94	42.67	100.00	42.13	98.90	38.84	97.70	35.95	94.25	41.04	98.06	34.00	97.06
R7	3.24	100.00	3.02	100.00	2.78	100.00	2.68	100.00	2.48	100.00	1.99	100.00	2.00	100.00
R8	0.40	0.00	0.86	0.00	0.46	0.00	0.89	0.00	0.83	0.00	1.20	0.00	1.00	0.00
R9	4.05	10.00	3.45	25.00	3.70	0.00	5.80	69.23	4.55	18.18	3.19	25.00	4.00	12.50
R10	14.98	100.00	14.66	100.00	15.28	100.00	12.95	96.55	12.40	93.33	14.74	100.00	11.00	90.91
R11	0.81	50.00	0.86	50.00	0.93	0.00	1.34	100.00	2.07	40.00	1.20	33.33	0.50	0.00
R12	6.48	100.00	7.76	94.44	7.87	94.12	4.46	90.00	4.13	90.00	6.37	93.75	8.00	93.75
R13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
R14	3.64	55.56	2.16	20.00	3.70	50.00	2.68	50.00	2.89	42.86	2.79	42.86	4.50	66.67
R15	4.45	100.00	3.88	100.00	5.56	100.00	2.68	83.33	1.65	75.00	2.79	100.00	4.00	87.50

From the analysis of these three rules, which are the most representative, it can be concluded that a high value of F2 generally leads to a statistical improvement in the results of the nearest neighbor classifier if a noise filter is used. If the classification problem is rather simple, with a lower value of F2, the application of a noise filter is generally not necessary. The high values of the measure N2 in the rule R5 reflects the fact that the examples of the same class are dispersed. Thus, when dealing with complex problems with high degrees of overlapping, filtering can improve the classification performance. However, if the problem is rather simple, with low degrees of overlapping, and moreover the examples of the same class are dispersed, e.g., if there are many clusters with low overlapping among them, noise

Table 11
Base datasets used for the validation phase.

Dataset	#INS	#ATT (R/I/N)	#CLA
abalone	4174	8 (7/0/1)	28
breast	277	9 (0/0/9)	2
dermatology	358	34 (0/34/0)	6
german	1000	20 (0/7/13)	2
page-blocks	5472	10 (4/6/0)	5
phoneme	5404	5 (5/0/0)	2
satimage	6435	36 (0/36/0)	7
segment	2310	19 (19/0/0)	7
vehicle	846	18 (0/18/0)	4
vowel	990	13 (10/3/0)	11

Table 12
Ranks obtained applying the final rule set ($R+$) and the indiscriminate usage of the filter ($R-$).

Dataset	CF	CVCF	EF	ENN	ENNTh	IPF	NCNEdit	RNG
$R+$	32 132.5	26 865.5	28 103.5	33 297.5	37 238.0	30 871.5	31 497.5	30 718.5
$R-$	13 017.5	17 984.5	16 746.5	11 552.5	7612.0	14 278.5	13 352.5	14 431.5
p-**Value**	0.000001	0.002265	0.000089	0.000001	0.000001	0.000001	0.000001	0.000001

filtering is not usually necessary—since the filtering may remove any of those clusters and be detrimental to the test performance.

5.5. Validation of the chosen rule set

In order to validate the usefulness of the rule set provided in the previous section to discern when to apply a noise filter to a concrete dataset, an additional experimentation has been prepared considering the 10 datasets shown in Table 11. From these datasets, another 300 binary ones have been created in the same way as explained in Section 4, but increasing the noise levels up to 25%.

For each noise filter, the test performance of 1-NN is computed for these datasets in two different cases:

1. Indiscriminately applying the noise filter to each training dataset.
2. Applying the noise filter to a training dataset only if the rule set of Section 5.4 so indicates. Concretely, the rule set indicates that noise filters must be applied in a 56% of the cases.

Then, the test results of both cases are compared using Wilcoxon's test. Table 12 shows the ranks obtained by case 1 ($R-$) and case 2 ($R+$) along with the corresponding p-values.

The results of this table show that, with some noise filters such as ENNTh and ENN, the advantage of using the rule set is more accentuated, whereas with others, such as CVCF and EF, this difference is less remarkable. However, very low p-values have been obtained in all the comparisons, which implies that the usage of the rule set to predict when to apply filtering is clearly positive with all the noise filters considered. Therefore, the conclusions obtained in the previous sections are maintained in this validation phase, even though a wider range of noise levels have been considered in the latter.

6. Concluding remarks

This paper has studied to what extent noise filtering efficacy can be predicted using data complexity measures when the nearest neighbor classifier is employed. A methodology to extract a rule set based on data complexity measures to predict in advance when a noise filter will statistically improve the results has been provided.

The results obtained have shown that there is a notable relation between the characteristics of the data and the efficacy of several noise filters, as the rule sets have good prediction performances. The most influential metrics are F2, N2, F3 and T1. Moreover, a single rule set has been proposed and tested to predict the noise filtering efficacy of all the noise filters, providing a good prediction performance. This shows that the conditions under which a noise filter works well are similar for other noise filters.

The analysis of the rule set provided shows that, generally, noise filtering statistically improves the classifier performance of the nearest neighbor classifier when dealing with problems with a high value of overlapping among the classes. However, if the problem has several clusters with a low overlapping among them, noise filtering is generally unnecessary and can indeed cause the classification performance to deteriorate.

This paper has focused on the prediction of noise filtering efficacy with the nearest neighbor classifier due it being perhaps the most noise-sensitive learner and then, the true filtering efficacy was checked. In future works, how noise filtering efficacy can be predicted for other classification algorithms with different noise-tolerance will be studied.

Acknowledgment

Supported by the Spanish Ministry of Science and Technology under Projects TIN2011-28488 and TIN2010-15055, and also by Regional Project P10-TIC-6858. J.A. Sáez holds an FPU Scholarship from the Spanish Ministry of Education and Science.

References

[1] X. Wu, X. Zhu, Mining with noise knowledge: error-aware data mining, IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 38 (4) (2008) 917–932.
[2] D. Liu, Y. Yamashita, H. Ogawa, Pattern recognition in the presence of noise, Pattern Recognition 28 (7) (1995) 989–995.
[3] X. Zhu, X. Wu, Class noise vs. attribute noise: a quantitative study, Artificial Intelligence Review 22 (2004) 177–210.
[4] Y. Li, L.F.A. Wessels, D. de Ridder, M.J.T. Reinders, Classification in the presence of class noise using a probabilistic kernel Fisher method, Pattern Recognition 40 (12) (2007) 3349–3357.
[5] R. Kumar, V.K. Jayaraman, B.D. Kulkarni, An SVM classifier incorporating simultaneous noise reduction and feature selection: illustrative case examples, Pattern Recognition 38 (1) (2005) 41–49.
[6] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993.
[7] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1967) 21–27.
[8] J. Toyama, M. Kudo, H. Imai, Probably correct k-nearest neighbor search in high dimensions, Pattern Recognition 43 (4) (2010) 1361–1372.
[9] Y. Liaw, M. Leou, C. Wu, Fast exact k nearest neighbors search using an orthogonal search tree, Pattern Recognition 43 (6) (2010) 2351–2358.
[10] I. Kononenko, M. Kukar, Machine Learning and Data Mining: Introduction to Principles and Algorithms, Horwood Publishing Limited, 2007.
[11] Y. Wu, K. Ianakiev, V. Govindaraju, Improved k-nearest neighbor classification, Pattern Recognition 35 (10) (2002) 2311–2318.
[12] D. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man and Cybernetics 2 (3) (1972) 408–421.
[13] C. Brodley, M. Friedl, Identifying mislabeled training data, Journal of Artificial Intelligence Research 11 (1999) 131–167.
[14] X. Zhu, X. Wu, Q. Chen, Eliminating class noise in large datasets, in: Proceeding of the 20th International Conference on Machine Learning, 2003, pp. 920–927.
[15] M. Basu, T. Ho, Data Complexity in Pattern Recognition, Springer, Berlin, 2006.
[16] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 289–300.
[17] S. Singh, Multiresolution estimates of classification complexity, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1534–1539.
[18] R. Baumgartner, R.L. Somorjai, Data complexity assessment in undersampled classification, Pattern Recognition Letters 27 (2006) 1383–1389.
[19] A.C. Lorena, A.C.P.L.F. de Carvalho, Building binary-tree-based multiclass classifiers using separability measures, Neurocomputing 73 (2010) 2837–2845.
[20] A. Orriols-Puig, J. Casillas, Fuzzy knowledge representation study for incremental learning in data streams and classification problems, Soft Computing 15 (12) (2011) 2389–2414.

[21] A.C. Lorena, I.G. Costa, N. Spolar, M.C.P. de Souto, Analysis of complexity indices for classification problems: cancer gene expression data, Neurocomputing 75 (1) (2012) 33–42.
[22] O. Okun, H. Priisalu, Dataset complexity in gene expression based cancer classification using ensembles of k-nearest neighbors, Artificial Intelligence in Medicine 45 (2009) 151–162.
[23] J. Luengo, F. Herrera, Domains of competence of fuzzy rule based classification systems with data complexity measures: a case of study using a fuzzy hybrid genetic based machine learning method, Fuzzy Sets and Systems 161 (2010) 3–19.
[24] E. Bernadó-Mansilla, T.K. Ho, Domain of competence of XCS classifier system in complexity measurement space, IEEE Transactions on Evolutionary Computation 9 (1) (2005) 82–104.
[25] J.S. Sánchez, R.A. Mollineda, J.M. Sotoca, An analysis of how training data complexity affects the nearest neighbor classifiers, Pattern Analysis and Applications 10 (3) (2007) 189–201.
[26] J. Luengo, F. Herrera, Shared domains of competence of approximate learning models using measures of separability of classes, Information Sciences 185 (2012) 43–65.
[27] S. García, J.R. Cano, E. Bernadó-Mansilla, F. Herrera, Diagnose Effective Evolutionary Prototype Selection Using an Overlapping Measure, 2009.
[28] J. Luengo, A. Fernández, S. García, F. Herrera, Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling, Soft Computing—A Fusion of Foundations, Methodologies and Applications 15 (2011) 1909–1936.
[29] R.K.M. Dong, Feature subset selection using a new definition of classificability, Pattern Recognition Letters 24 (2003) 1215–1225.
[30] T.K. Ho, H.S. Baird, Pattern classification with compact distribution maps, Computer Vision and Image Understanding 70 (1) (1998) 101–110.
[31] F.W. Smith, Pattern classifier design by linear programming, IEEE Transactions on Computers 17 (4) (1968) 367–372.
[32] S.P. Smith, A.K. Jain, A test to determine the multivariate normality of a data set, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (5) (1988) 757–761.
[33] A. Hoekstra, R.P.W. Duin, On the nonlinearity of pattern classifiers, in: 13th International Conference on Pattern Recognition, 1996, pp. 271–275.
[34] F. Lebourgeois, H. Emptoz, Pretopological approach for supervised learning, in: 13th International Conference on Patter Recognition, 1996, pp. 256–260.
[35] J. Sánchez, F. Pla, F. Ferri, Prototype selection for the nearest neighbor rule through proximity graphs, Pattern Recognition Letters 18 (1997) 507–513.
[36] G. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley and Sons, 2004.
[37] D. Gamberger, N. Lavrac, C. Groselj, Experiments with noise filtering in a medical domain, in: 16th International Conference on Machine Learning (ICML99), 1999, pp. 143–151.
[38] S. Verbaeten, A. Assche, Ensemble methods for noise elimination in classification problems, in: 4th International Workshop on Multiple Classifier Systems (MCS 2003), Lecture Notes on Computer Science, vol. 2709, Springer, 2003, pp. 317–325.
[39] F. Vazquez, J. Sánchez, F. Pla, A stochastic approach to Wilsonś editing algorithm, in: 2nd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA05), Lecture Notes on Computer Science, vol. 3523, Springer, 2005, pp. 35–42.
[40] T. Khoshgoftaar, P. Rebours, Improving software quality prediction by noise filtering techniques, Journal of Computer Science and Technology 22 (2007) 387–396.
[41] J. Sánchez, R. Barandela, A. Márques, R. Alejo, J. Badenas, Analysis of new techniques to obtain quality training sets, Pattern Recognition Letters 24 (2003) 1015–1022.
[42] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2–3) (2011) 255–287.
[43] A. Orriols-Puig, E. Bernadó-Mansilla, Evolutionary rule-based systems for imbalanced data sets, Soft Computing 13 (3) (2009) 213–225.
[44] N.A. Samsudin, A.P. Bradley, Nearest neighbour group-based classification, Pattern Recognition 43 (10) (2010) 3458–3467.
[45] I. Triguero, S. García, F. Herrera, Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification, Pattern Recognition 44 (4) (2011) 901–916.
[46] J. Huang, C.X. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Transactions on Knowledge and Data Engineering 17 (3) (2005) 299–310.
[47] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[48] J.S. Sánchez, R. Barandela, A.I. Marqués, R. Alejo, J. Badenas, Analysis of new techniques to obtain quality training sets, Pattern Recognition Letters 24 (7) (2003) 1015–1022.

José A. Sáez received his M.Sc. in Computer Science from the University of Granada, Granada, Spain, in 2009. He is currently a Ph.D. student in the Department of Computer Science and Artificial Intelligence in the University of Granada. His main research interests include noisy data in classification, discretization methods and imbalanced learning.

Julián Luengo received the M.S. degree in Computer Science and the Ph.D. degree from the University of Granada, Granada, Spain, in 2006 and 2011, respectively. His research interests include machine learning and data mining, data preparation in knowledge discovery and data mining, missing values, data complexity and fuzzy systems.

Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain.

He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has had more than 200 papers published in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001).

He currently acts as Editor in Chief of the international journal "Progress in Artificial Intelligence" (Springer) and serves as Area Editor of the Journal Soft Computing (area of evolutionary and bioinspired algorithms) and International Journal of Computational Intelligence Systems (area of information systems). He acts as Associated Editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Knowledge and Information Systems, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, Swarm and Evolutionary Computation.

He received the following honors and awards: ECCAI Fellow 2009, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", and International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition, 2010).

His current research interests include computing with words and decision making, data mining, bibliometrics, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.