Expert Systems with Applications 40 (2013) 4820–4831

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Analysis of data complexity measures for classification ☆

José-Ramón Cano *

Department of Computer Science, University of Jaén, Jaén, Spain

ARTICLE INFO

Keywords:
Data complexity
Class overlapping
Class separability
Classification

ABSTRACT

The study of data complexity metrics is an emergent area in the field of data mining and is focused on the analysis of several data set characteristics to extract knowledge from them. This information can be used to support the election of the proper classification algorithm.

This paper addresses the analysis of the relationship between data complexity measures and classifiers behavior. Each one of the metrics is evaluated covering its range of values and studying the classifiers accuracy on these values.

The results offer information about the usefullness of these measures, and which of them allow us to analyze the nature of the input data set and help us to decide which classification method could be the most promising one.

1. Introduction

There exist multiple classifiers systems which offer different nature classification models (Cherkassky & Mulier, 1998; Kininenko & Kukar, 2007). The behavior of these classification models is significantly data dependent, so the analysis of this dependency allow us to reach further advances in the construction and exploitation of these models (Ho, 2008; Kim, 2010; Lebourgeois & Emptoz, 1996; Yildiz, 2011).

To study the relation between classifier performances and the data sets used as input there are several studies in the literature offering data complexity measures to characterize the data sets (Basu & Ho, 1999, 2006; Ho & Basu, 2002; Singh, 2003). Some of the factors which are been able to affect to the classifiers are, for example, the shapes of the classes, the shape of the decision boundary, the amount of overlap among classes, the proximity among classes, etc.

These measures, in addition to the data characterization, have been used in the past as diagnose tool to take decisions about algorithms application (Bernadó-Mansilla & Ho, 2005; Garcıa, Cano, Bernadó-Mansilla, & Herrera, 2009; Kim & Oommen, 2008; Mollineda, Sánchez, & Sotoca, 2005). In the literature, the authors consider real life data sets obtained from the UCI repository (Asuncion & Newman, 2007), to be analyzed by the metrics.

In this paper, we study the complexity measures considering a different perspective. The objective consists in evaluate the usefulness of the metrics and its traditional use. To address this, we analyze each one of the complexity measures proposed in Ho and Basu (2002) covering its range of values, and study the relation between the measures and classifiers performances.

Each complexity measure is dependent of data set characteristics, like for example, number of instances, features, classes or instances distribution. This situation produces that two data sets with very different characteristics (different number of instances, features, classes or even instances distribution) can present the same metric value, and used as input in the same classifier, produce accuracy rates which are not related at all. This study uses a data generation algorithm, whose objective consist in generating data sets changing the instance distribution to reach concrete metric values. Using the data sets generated, we analyze the effect of the measure evolution on the accuracy of models learned by different classifiers. The graphical representation of these results allows us to analyze the relationships between the measures and classifiers for a fixed environment, which can help us in any kind of diagnose or suggestion about which classifiers is more promising for a concrete data set.

In order to do this, the paper is set out as follows. Section 2 presents and describes the data complexity measures. In Section 3 the dependence of data complexity metrics with respect to data set characteristics is studied. The algorithm used in the generation of the artificial data sets is given in Section 4. Details of empirical experiments, results obtained and their analysis are reported in Section 5. Finally, Section 6 contains a brief summary of the work and conclusions reached.

2. Preliminaries: data complexity measures

The prediction capabilities of classifiers are strongly dependent on data complexity. This is the reason why various recent papers

☆ This work was supported by the Spanish Ministry of Education, Social Policy and Sports under projects TIN-2008-06681-C06-02, and by the Andalusian Research Plan under project P08-TIC-03928 (FEDER funds).

* Tel.: +34 953648585.
E-mail address: jrcano@ujaen.es

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.02.025

have introduced the use of measures to characterize the data and to relate those characteristics to classifier performance (Sánchez, Mollineda, & Sotoca, 2007).

In Ho and Basu (2002), define some complexity measures for binary classes data sets. Singh (2003) offers a review of data complexity measures and proposes two new ones. Dong and Kothari (2003) propose a feature selection algorithm based in one of the complexity measures defined by Ho and Basu. Bernadó-Mansilla and Ho (2005) investigate the domain of competence of XCS by means of a methodology that characterizes the complexity of a classification problem by a set of geometrical descriptors. In Li, Dong, and Kothari (2005), Li et al. analyze some omnivariate decision trees using the measure of complexity based on data density proposed by Ho and Basu. Baumgartner and Somorjai (2006) define specific measures for regularized linear classifiers, using the Ho and Basu measures as reference. Mollineda et al. (2005) extend some Ho and Basu's measure definitions for problems with two or more classes. They analyze these generalized measures in two classic prototype selection algorithms and remark that the Fisher's discriminant ratio is the most effective for prototype selection. Sánchez et al. (2007) analyze the effect of the data complexity in the nearest neighbors classifiers. García et al. (2009) diagnose the effectiveness of evolutionary prototype selection (Cano, 2012) based on an overlapping measure. Kim and Oommen (2009) present a proposal to enhance the computation of volume-based inverclass overlap measures by means of prototype reduction schemes. Luengo, Fernández, García, and Herrera (2011) analyze the usefulness of the data complexity measures in order to evaluate the behavior of undersampling and oversampling methods. In Elizondo, Birkenhead, Gámez, García, and Alfaro (2012) evaluate the relationship between linear separability and the level of complexity of classification data sets.

Considering the literature, we have focused the study in the supervised learning scope. The metrics chosen are independent of the classifier, so we can study the measures impact in the model extracted by different nature classifiers.

The metrics describe the regularities and irregularities contained in the data set in terms of the chosen geometrical primitives. These descriptors are noted as measures of the geometrical complexity of a data set. This information is interesting in pattern recognition domain where most classifiers can also be characterized by geometrical descriptions of their decision regions.

The measures considered can be divided into three categories, which will be described in the following subsections.

2.1. Measures of overlaps in feature values from different classes

These measures are focused on the effectiveness of a single feature dimension in separating the classes, or the composite effects of a number of dimensions. The measures which belong to this group are:

2.1.1. Maximum Fisher's discriminant ratio (noted as F1)

The plain version of Fisher's discriminant ratio offered by Ho and Basu (2002) computes how separated are two classes according to a specific feature. It compares the difference between class means with the sum of class variances. Fisher's discriminant ratio for a feature i is defined as follows:

$$f_i = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{1}$$

where $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$ are the means and the variances of the two classes respectively.

The value for one dimension is f_i. Considering multidimensional problems, the maximum value of f_i over all feature indicate the feature which contributes to the maximal discrimination and is referred to as F1.

$$F1 = MAX(f_i) \tag{2}$$

being f_i each feature, with $i = 1,\ldots,d$ for a d-dimensional problem.

The range of this measure is $[0,+\infty[$. Small values of F1 represent strong overlapping.

2.1.2. Volume of overlap region (noted as F2)

This measure is defined as the overlap of the tails of the two class-conditional distributions. It can be computed by finding, for each feature, the maximum and minimum values of each class and computing the length of the overlap region, as the following expression indicates:

$$F2 = \prod_i \frac{MIN(max(f_i,c_1),max(f_i,c_2)) - MAX(min(f_i,c_1),min(f_i,c_2))}{MAX(max(f_i,c_1),max(f_i,c_2)) - MIN(min(f_i,c_1),min(f_i,c_2))} \tag{3}$$

where $max(f_i,c_j)$ and $min(f_i,c_j)$ are the maximum and minimum values of each feature f_i and class c_j, being $i = 1,\ldots,d$ for a d-dimensional problem, and $j = 1,2$ for binary problem.

Small values of F2 indicate small overlap volume. The range of the F2 metric is $[0,1]$.

2.1.3. Maximal (individual) feature efficiency (noted as F3)

This is a measure of efficiency of individual features that describe how much each feature contributes to the separation of the two classes.

The process followed assumes, for each feature, all points of the same class have values falling between the maximum and minimum of this class. If there is a overlap, we consider the classes ambiguous in that overlap in that dimension. So, a problem is easy if there exists one feature dimension where both classes do not overlap. The efficiency of each feature is defined as the fraction of all remaining points separable by that feature. In this sense, F3 is considered as the maximum feature efficiency.

The range in this case is $[0,1]$. Small values of F3 indicate high overlap.

2.2. Measures of separability of classes

These measures illustrate 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 value, usually a distance metric, instead of than evaluated separately. The measures present in this group are:

2.2.1. Minimized sum of error distance by linear programming (noted as L1)

Linear programming methods far outperform the adaptive methods in terms of definiteness and correctness of decisions and time efficiency (Basu & Ho, 1999). To handle both separable and nonseparable cases, we use a formulation proposed by Smith (1968) that minimizes an error function:

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

where a, b are arbitrary constant vectors (both chosen to be 1), w is the weight vector, t is an error vector and Z is a matrix where each column z is defined on an input vector x (augmented by adding one dimension with a constant value 1) and its class c (with value c_1 or c_2):

$$\begin{cases} z = +x & \text{if } c = c_1 \\ z = -x & \text{if } c = c_2 \end{cases}$$

This measure is zero for a linearly separable problem. This measure can be affected by outliers that happen to be on the wrong side of the optimal hyperplane.

Small values of L1 indicates that the data set is linearly separable. The measure L1 has its domain in the range [0, 1].

2.2.2. Error rate of linear classifier by linear programming (noted as L2)

Considering the description of the previous measure, L2 computes the error rate of such linear classifier on the original training set.

The domain of L2 metric is in the range [0, 1]. Large values of this measure indicates that the data set is not linearly separable.

2.2.3. Fraction of points on class boundary (noted as N1)

N1 is calculated by means of constructing a class-blind minimum spanning tree over the entire data set, counting the number of points incident to an edge which goes across the two classes. The minimum spanning tree connects all the points to their nearest neighbors (regardless of class) and the number of points connected to the opposite class by and edge are counted (Smith, 1998). These are considered to be points lying next to the class boundary. The fraction of such points over all points in the data set is used as measure. This metric provides an estimate of the length of the class boundary.

The N1 metric domain is [0, 1]. Large values of the measure indicate that the majority of points lay closely to the class boundary which could affect to the classifier to define the class boundary accurately.

2.2.4. Ratio of average intra/inter class nearest neighbor distance (noted as N2)

In the case of heavily interleaved or randomly labeled data, most of the points will appear next to the class boundary. The same can be true for a linearly separable problem with margins narrower that the distance between points of the same class.

To check this situation, the N2 measure is proposed, which is closely related to the previous measure N1. In this case, we first compute the Euclidean distance from each point to its nearest neighbor within or outside the class. We then take the average of all the distances to intra-class nearest neighbors, and the average of all the distance to inter-class nearest neighbors. The ration of the two averages is used as measure N2, as expression 4 shows:

$$N2 = \frac{\sum_{i=0}^{N} intraDist(ex_i)}{\sum_{i=0}^{N} interDist(ex_i)} \quad (4)$$

where, for each input instance ex_i, we calculate the distance to its nearest neighbor within the class (intraDist (ex_i)) and the distance to nearest neighbor of any other class (interDist (ex_i)). The result is the ratio of the sum of the intra-class distances to the sum of the inter-class distances for each input example, being N the number of examples in the data set.

This measure compares the within-class spread to the size of the gap between classes.

The domain of N2 metric is in the range $[0, +\infty[$. Low values suggest that the examples of the same class lay closely in the feature space. Large ones indicate that examples of the same class are disperse.

2.2.5. Error rate of 1 nearest neighbor classifier (noted as N3)

This is the error rate of a nearest neighbor classifier considering the training set, estimated by the leave-one-out method. This measure shows how close the examples of different classes are.

The N3 metric has its range in the interval [0, 1]. Low values of this metric indicate that there is a large gap in the class boundary.

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

These measures offer indirect characterizations of class separability. The shape, position, and interconnectedness of these manifolds give hints on how well two classes are separated, but they do not describe separability by design. In this case, the measures which belong to this group are:

2.3.1. Nonlineality of linear classifier by linear programming (noted as L3)

This metric defines a measure on nonlinearity proposed by Hoekstra and Duin (1996). L3 is calculated by means of the creation of a test set by linear interpolation between randomly pairs of points from the same class, given a training set. The error rate of the classifier on this test set is measured. For this metric, we consider the nonlinearity of the linear classifier described in Section 2.2.1.

The L3 metric has its values in the range [0, 1]. Small values of this measure indicates that the data set is not linearly separable.

2.3.2. Nonlineality of 1 nearest neighbor classifier (noted as N4)

It is calculated in a similar way than L3, but considers as classifier the 1 nearest neighbor.

The domain of N4 metric is in the range [0, 1]. Low values of N4 indicates that the data set is linearly separable.

2.3.3. Fraction of points with associated adherence subsets retained (noted as T1)

Lebourgeois and Emptoz proposed this metric in Lebourgeois and Emptoz (1996), which describes the shapes of class manifolds with the notion of adherence subset. An adherence subset can be considered as a sphere centered on an example of the data set which is grown till touch any example of another class. Each adherence subset contains a set of examples of the same class and cannot grow including examples of other classes. The measure removes all adherence subsets included in others, considering only the biggest ones.

T1 counts the number of biggest adherence subsets needed to cover each class, where each ball is centered at a training point and grown to the maximal size before it touches another class. It is normalized by the total number of points.

This metric is a interior description rather than a boundary description as given by the measures based on minimal spanning tree. The number and order of the retained adherence subsets 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 class, each adherence subset is retained and is of a low order.

T1 metric has its domain in the range [0, 1]. Small values of this measure indicates that the instances which compose the data set are highly grouped and the boundaries are clearly defined.

3. Dependence of data complexity metrics with respect to data set characteristics

In the literature, there are many contributions in the data complexity domain where the complexity metrics are evaluated using real life data sets (Bernadó-Mansilla & Ho, 2005; Mollineda et al., 2005; Singh, 2003). In them, one of the usual objectives consists on the use of these complexity metrics to diagnose the effectiveness of classifiers. In some cases, the diagnose is developed considering the relationships among metrics values on several different real life data sets (Kim & Oommen, 2008; Mollineda et al., 2005).

The drawback of this consideration is that two data sets with very different characteristics (different number of instances, fea-

Table 1
Different data sets with the same F1 value = 0.5, but different number of instances and instances distribution.

(a) Data set 1. (b) Data set 2. (c) Data set 3.

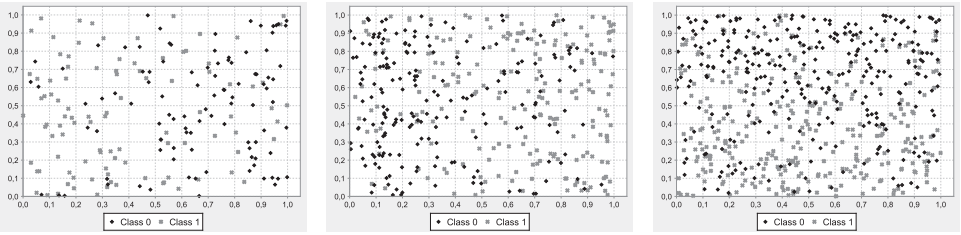


Table 2
Accuracy rates of different classifiers using the data set 1, 2 and 3.

	Data set 1 (%)	Data set 2 (%)	Data set 3 (%)
3-Nearest neighbor	86.66	80.52	68.88
C4.5	91.35	81.25	68.33
Support Vector Machine	90.64	82.04	68.56

tures, classes or instances distribution) can present the same measure value, and used as input in a classifier, their accuracy rates are not related at all.

As example, we consider three data sets with different characteristics but the same metric value (F1 = 0.5 in Table 1). Keeping the number of features in two and binary classes to represent graphically the data sets, in this example the number of instances (200, 400 and 600) and the instance distribution are changed.

As classification algorithms we consider three of the algorithms suitable for tackling classification tasks with real data among the identified in Wu and Kumar (2007) as the top 10 data mining algorithms: C4.5 (Quinlan, 1993), *k*-Nearest Neighbor (Papadopoulos & Manolopoulos, 2004) and Support Vector Machines (SVM, Steinwart & Christmann (2008)).

Table 1 shows the instance distribution of each one of the three data sets plotted. Considering these data sets as input, the accuracy rates (following a ten fold cross validation) offered by 3 nearest neighbor classifier, C4.5 and Support Vector Machine respectively appear in Table 2.

As we can see in Table 2, the classifiers offer different accuracy rates for each data set with the same F1 value.

These results indicate that the meaning of the measures cannot be directly used to establish relationship among different data sets, because the value obtained for each measure depends of the data set characteristics (number of classes, features, instances and instances distribution). If we compute one metric using a data set as input, the value obtained present meaning by itself (being high or low), and it should not be used to offer a diagnose based on similar metric values in other different data sets.

4. Data generation to cover data complexity measures range

We are interested in the analysis of each one of these metrics, covering their complete domain of values to test the usefulness of the measure. Due to the dependence between data set characteristics and complexity metrics, the generation of artificial data sets is decided, because there is not any set of data sets which cover the whole range of measure values.

Artificial data sets have been generated considering a fixed environment (a data set fixing the number of instances, features and classes), just modifying the instances distribution. The data sets generated correspond to values of the measure inside the range of the complexity metric.

We have considered the definition of measures offered in the literature, referred to binary classes problems (Bernadó-Mansilla & Ho, 2005; Ho & Basu, 2002), so the data sets generated by the algorithm satisfy that constraint. The algorithm adjusts the data set according to the measure value introduced as parameter.

Our objective is to use the data sets generated to analyze the behavior of the classifiers which extract different models over them.

In this section we present the algorithm applied to the process of generation of data sets. The algorithm follows a Montecarlo schema. These are the steps which compose the algorithm (Fig. 1):

As input (see Fig. 1) in step 1, the algorithm receives the number of instances (*N_Inst*), features (*N_Feat*) and classes (*N_Class*) of the data set to be generated, and the value of the measure we are looking for (*Measure_Value_Searched*). In the steps 2 and 3, the data set is randomly generated by the function *Random_Data_Set_Generation* and its associated value of complexity measure is computed (using the function *Evaluate_Complexity_Measure*). Between the steps 4 and 14 the iterative search is developed. To address this, in step 5, one instance is selected randomly from the data set, being mutated in step 6 (by means of the function *Random_Mutation_of_Features* which receives one instance and returned a new one with the features changed). In steps 7 and 8, the instance mutated is replaced (the function *Replace_Instance* interchange the original instance by the mutated one) in the data set and its complexity metric is recalculated. The step 9 is dedicated to check if the new metric value is closer to the initial metric value passed as argument. If it is the closest one, we keep the new instance. In other case, we replace it by the original not mutated one (step 12). This process is repeated till the data set has associated a value of the metric equal to the one introduced as parameter. When this situation appears, the loop ends and the data set generated is returned.

5. Experimental study

In this section we present the study developed considering the metrics previously described. We analyze the relation of these measures on different kind of classifiers.

To address this, first of all we present in Section 5.1 the experimental methodology. From Sections 5.1–5.12 the results and analysis for each one of the metrics are reported. The final Section 5.13 is a brief resume of the conclusions reached.

5.1. Experimental methodology

In this section we present the data sets generated, the classification algorithms considered and the methodology followed in the experiments.

5.1.1. Data sets

For each one of the complexity measures, their range has been split in one hundred intervals. Due to the dependence between the data set characteristics (in this case the instance distribution which

```
1. algorithm DS_Generator(N_Inst, N_Feat, N_Class, Measure_Value_Searched)
2. dataset= Random_Data_Set_Generation(N_Inst, N_Feat, N_Classes)
3. best_complex_measure=Evaluate_Complexity_Measure(dataset)
4. Repeat
   5. instance = Select_Randomly_one_instance(dataset)
   6. instance_mutated = Random_Mutation_of_Features(instance)
   7. dataset=Replace_Instance(dataset, instance, instance_mutated)
   8. complex_measure = Evaluate_Complexity_Measure(dataset)
   9. If complex_measure Nearer to Measure_Value_Searched Than
      best_complex_measure Then
      10. best_complex_measure=complex_measure
   11. elseif
      12. dataset=Replace_Instance(dataset, instance_mutated, instance)
   13. Endif
14. until best_complex_measure=Measure_Value_Searched
15. return dataset
```

Fig. 1. Pseudocode of data generation algorithm.

is modified) and the classifiers performances, we generate one hundred data sets per interval. We are interested in the study of the robust or weak dependence between the measure value and the accuracy rate with respect to the only characteristic that change among data sets: the instances distribution. In this way, we have one hundred instance distributions for each interval which have associated the same complexity measure value.

Most of the metrics have their domain in the range [0,1]. In the case when the maximal value of the metric is $+\infty$, then this maximal value is approximated by means of the execution of the generation algorithm having as objective the maximization of the fitness function (maximizing the value of the measure searched).

To generate the data sets using the algorithm described in Section 4 we have fixed the number of instances in 200, the number of features in 2 with feature range in [0,1] and the number of classes in 2 (because we are analyzing measures defined for binary classes).

5.1.2. Classification algorithms

The election of the classification algorithms is based in the top 10 data mining algorithms (Wu & Kumar, 2007). Three of them are suitable for tackling classification tasks with real data:

- C4.5 (Quinlan, 1993): Algorithm proposed by Quinlan in which generates decision trees. It uses a divide-and-conquer strategy to grow an initial tree from a data set *D*.
- *k*-Nearest Neighbor (Papadopoulos & Manolopoulos, 2004): This algorithm finds a group of *k* objects in the training set that are closest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighborhood.
- Support Vector Machines (SVM) (Cortes & Vapnik, 1995; Meyer, Leisch, & Hornik, 2003): This machine learning method is based on the concept of decision planes that define decision boundaries. A decision plane is one that separate between a set of objects having different class memberships.

5.1.3. Presentation of the results

The result are organized using Classification accuracy figures. This kind of figure shows the evolution of the accuracy rate in the classifiers analyzed. It considers as input the one hundreds data sets generated for the one hundred intervals with the metrics between minimal and maximal value. In this figures, for each snap (one interval), we present the maximal, average and minimal accuracy rates. In addition, we consider three regions in the graphic, representing minimal, medium and large values of the measure and we compute the area which appears between maximal and minimal value of the region. Large areas indicate that the relation between the measure value and the classification behavior is weak because values maximal and minimal depends too much of the instances distribution. Small areas represents that the relation between the metric value and the accuracy rates is closer and robust, with non dependence of the instance distribution.

5.2. Results and analysis for maximum Fisher's discriminant ratio (F1)

The results in this case correspond to Fig. 2. The analysis of the results is the following:

As it was noted in Section 2.1.1, small values of the measure indicate strong overlapping. As the metric increases its value, the overlapping is reduced and both classes look clearly differentiable.

Paying attention to the accuracy rates offered by the classifiers (Fig. 2(a)–(c)) we can see that all of them present similar behavior. In the first third (zone dedicated to Area 1 in the figures), when the overlapping is high (snaps 0–33), all of them suffer low accuracy rates and large value for the areas, which means that the relation between the measure value and the accuracy rate of the classifiers is not clear. In the first 10 snaps the accuracy is affected negatively. Areas 2 and 3 are smaller, where the difference between maximal and minimal accuracy is not very significant. In these cases, the accuracy rates increase their values and the reduction in the size of the areas let us detect the effect of the measure value in the accuracy rate of the classifiers.

As larger the F1 value is, higher the accuracy rate of the classifier is, and more robust (non dependent of the instance distribution).

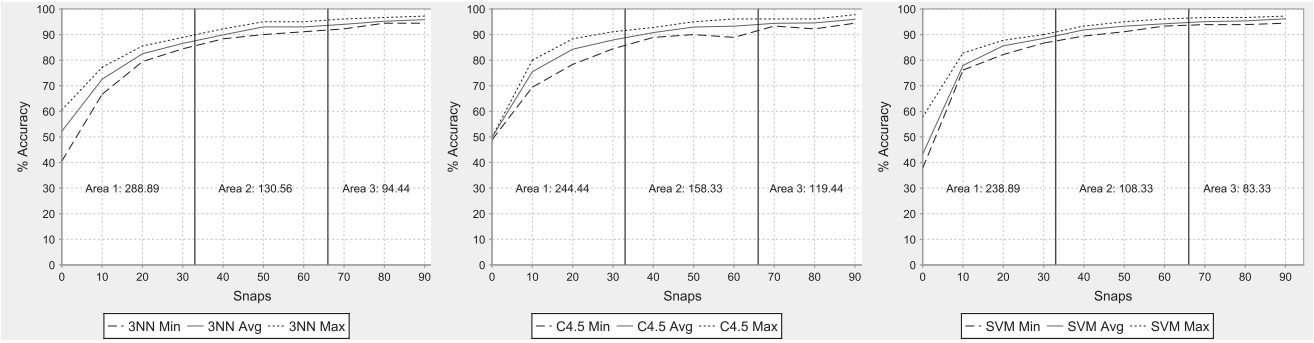
So, analyzing this metric in a data set and notifying that it is small, let us to know that the classification accuracy will suffer, as with low values as in consistency due to the overlapping drawback which is present in the data set.

5.3. Results and analysis for volume of overlap region (F2)

The results for F2 measure correspond to Fig. 3. The analysis of this figure is the following:

As it was noted in Section 2.1.2, small values of the measure indicates small overlap volume. As the metric increases its value, the overlapping between classes appears.

The analysis of the measure F2 considering the accuracy rates shows that the volume overlapping affects notably the classification accuracy. When the overlapping is small the accuracy rate is shortly affected in all the classifiers. Paying attention to the size of the areas, all the algorithms are sensible to this measure. The

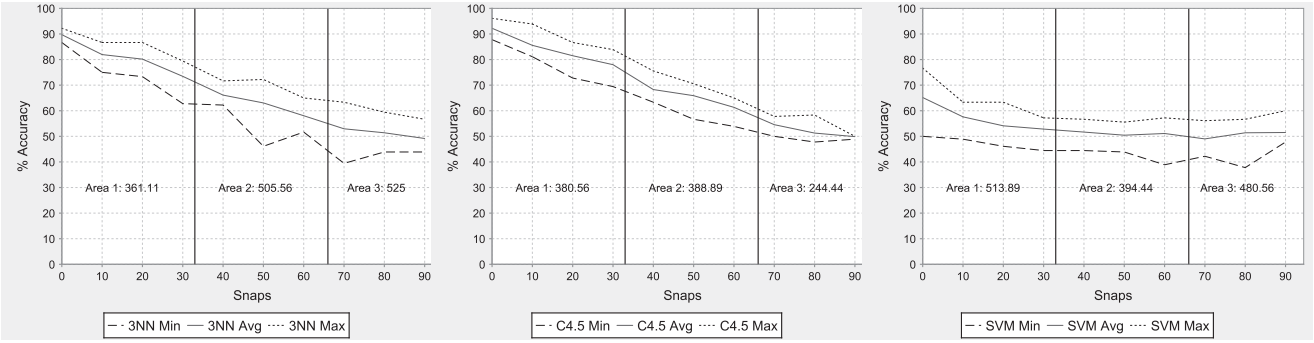


(a) Accuracy results associated to the classifier 3NN in all snaps for F1.

(b) Accuracy results associated to the classifier C4.5 in all snaps for F1.

(c) Accuracy results associated to the classifier SVM in all snaps for F1.

Fig. 2. Behavior of different classifiers for the domain of values of F1.



(a) Accuracy results associated to the classifier 3NN in all snaps for F2.

(b) Accuracy results associated to the classifier C4.5 in all snaps for F2.

(c) Accuracy results associated to the classifier SVM in all snaps for F2.

Fig. 3. Behavior of different classifiers for the domain of values of F2.

most robust one is C4.5 for medium or large values of the metric (Fig. 3(b)), while 3NN is the best when the value is small (Fig. 3(a)). This measure does not offer information regarding to SVM (see Fig. 3(c)), where SVM presents the smallest and less robust accuracy rates.

So, analyzing this metric in a data set and notifying that its is higher, let us to know that the classification accuracy will suffer due to the overlapping drawback. In the case of small values of the metric, the 3NN is the most promising one, while for medium or large values, C4.5 is the remarkable. This measure does not produce sensible information to be used in the case of SVM classifier.

5.4. Results and analysis for maximal (individual) feature efficiency (F3)

The results in the case of F3 measure correspond to Fig. 4. The analysis of this figure is the following:

As it was noted in Section 2.1.3, small values of the feature efficiency represents high overlapping because there is not any feature which contributes enough to the separation of classes. As the metric increases its value, the overlapping between classes disappears and the separability increases.

Like in the case of the metric F1 and F2, the analysis of the measure F3 shows that the feature efficiency affects notably the classification accuracy. When the feature efficiency is small the accuracy rate is negatively affected in all the classifiers. When the measure present small or medium values, the robustness of all the classifiers is committed which can be seen in the large size of these areas (see Areas 1 and 2 in Fig. 4(a)–(c)). In the case of large values of the metric, 3NN and C4.5 offer the highest accuracy rates. SVM presents a similar accuracy rate, but more robust.

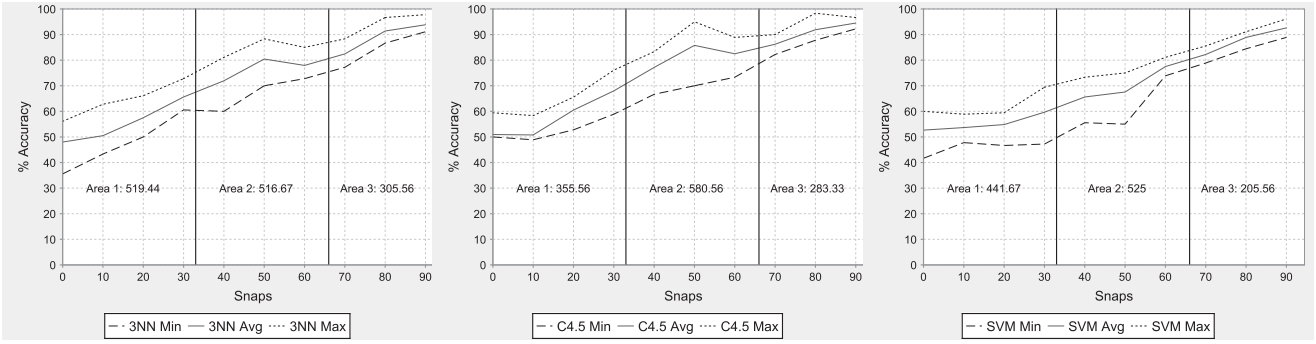
So, analyzing the feature efficiency in a data set and notifying that its is small, let us to know that the classification accuracy will be affected and due to the lack of robustness is not clear the algorithm election. All of them are sensible in accuracy and consistency. When the metric reaches a medium value, 3NN and C4.5 improves their accuracy rates, but not their robustness. In the case of large values of the measure, all the algorithms offer better behavior in accuracy rate and robustness, reducing the variability of their results.

5.5. Results and analysis for minimized sum of error distance by linear programming (L1)

The results for L1 measure correspond to Fig. 5. The analysis of this figure is the following:

Small values of this metric indicate high linear separability of the problem. As the metric increases its value, the linear separability is reduced.

C4.5 (see Fig. 5(b)) has more difficulties to approximate models with oblique bounds, and it becomes more difficult as the linear separability disappears. The case of SVM (see Fig. 5(c)) is opposite to C4.5. For the polynomial SVM that we have analyzed is very easy to learn models with oblique bounds, covering perfectly the frontier between classes. Its difficulties arrives when the linear separa-

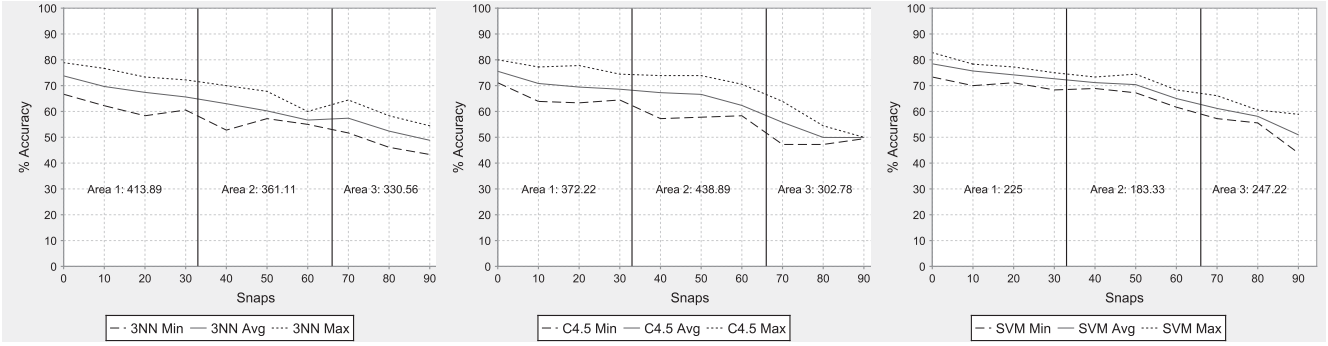


(a) Accuracy results associated to the classifier 3NN in all snaps for F3.

(b) Accuracy results associated to the classifier C4.5 in all snaps for F3.

(c) Accuracy results associated to the classifier SVM in all snaps for F3.

Fig. 4. Behavior of different classifiers for the domain of values of F3.



(a) Accuracy results associated to the classifier 3NN in all snaps for L1.

(b) Accuracy results associated to the classifier C4.5 in all snaps for L1.

(c) Accuracy results associated to the classifier SVM in all snaps for L1.

Fig. 5. Behavior of different classifiers for the domain of values of L1.

bility disappears. 3NN, like in the rest of previous examples, adjusts significantly the model to the input data (see Fig. 5(a)).

The linear separability, as the previous metrics analyzed, affects to the accuracy capabilities of the classification algorithms. When the data set presents linear separability, for the classifiers is easier to find quality predictive models. This quality is being reduced as the linear separability is removed. In this case, SVM is the algorithm which offers the best accuracy rates and robustness in all the range of the metric (Fig. 5(c)). The rest of the classifiers do not present consistency in their results, with larger areas in the whole range (Fig. 5(a) and (b)).

So, analyzing the linear separability in a data set and being smaller the value associated to this metric, let us to know that the classifiers are able to extract high quality models, in concrete, the SVM classifier.

5.6. Results and analysis for error rate of linear classifier by linear programming (L2)

In the case of L2 measure, their results correspond to Fig. 6. The analysis of this figure is the following:

This metric is correlated with L1. Small values of this metric (meaning small error rate by linear programming) represents high linear separability of the problem. As the metric increases its value (increasing the error), the linear separability is reduced.

In this case, the effect of the L2 variation in the accuracy rates of the classifiers is not significative as we can see in Fig. 6(a)–(c). The reduction in accuracy rate is just remarkable when the metric reach its maximal values. In this case, there is not an important accuracy rate reduction like in the case of other measures. The robustness of the algorithms is higher when the metric presents small values, deteriorating as the value becomes higher. Considering accuracy rates, the C4.5 is the one which produces the highest accuracy rates.

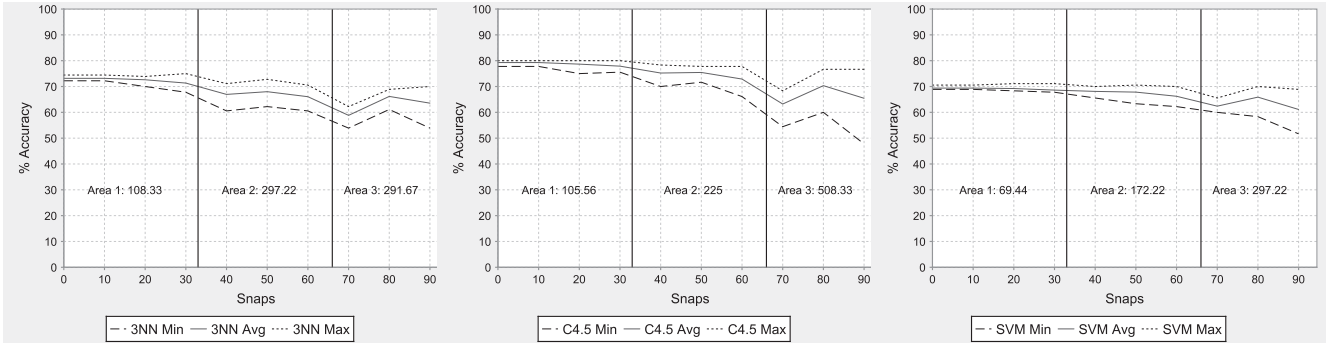
This metric does not offer significant information in concern the correlation of the metric values and the classification capabilities of the algorithms. The information that it produce is that if the measure presents a small value, the accuracy rate generated by each classifier is high and robust, while in the case of high values of the measure, the conclusion is that the accuracy rate may not be stable among instance distributions and classifiers.

5.7. Results and analysis for fraction of points on class boundary (N1)

The results for N1 measure correspond to Fig. 7. The analysis of this figure is the following:

The minimal value for this metric is reached when most of the instances of the spanning tree are connected to instances to the same class, which increases the classification capabilities of the 3NN classifier.

The nature of the measure (its nearest neighbor base definition) makes that C4.5 and SVM are not able to approximate their models to the original instance distribution.

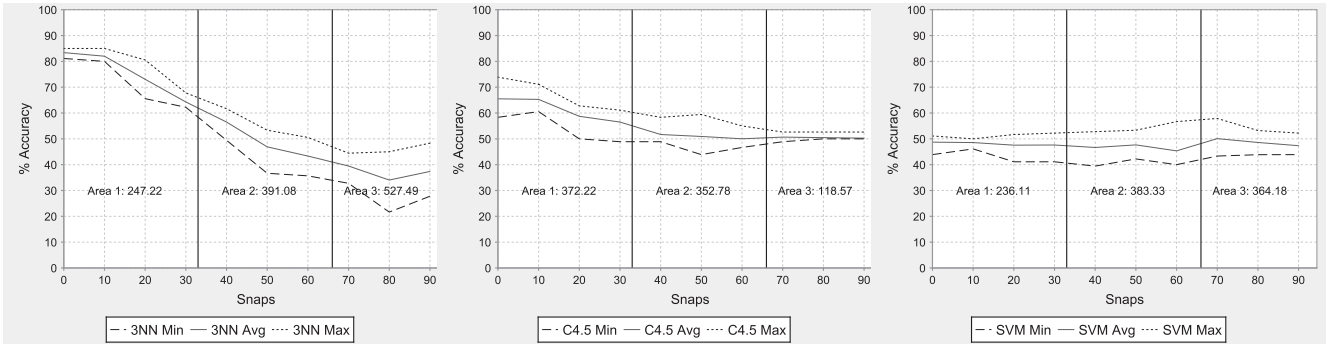


(a) Accuracy results associated to the classifier 3NN in all snaps for L2.

(b) Accuracy results associated to the classifier C4.5 in all snaps for L2.

(c) Accuracy results associated to the classifier SVM in all snaps for L2.

Fig. 6. Behavior of different classifiers for the domain of values of L2.



(a) Accuracy results associated to the classifier 3NN in all snaps for N1.

(b) Accuracy results associated to the classifier C4.5 in all snaps for N1.

(c) Accuracy results associated to the classifier SVM in all snaps for N1.

Fig. 7. Behavior of different classifiers for the domain of values of N1.

The effect of the measure is remarkable in the case of 3NN. As we can see in the Fig. 7(a), when the measure reaches the minimal value the accuracy rate of 3NN improves considerably. C4.5 and SVM present discrete accuracy rates due, as we have mentioned before, to the nature of this metric (Fig. 7(b) and (c)). The robustness in the classification behavior is poor in this case, much more when the metric reaches large values.

So, this metric looks interesting to be applied in a data set to consider the use or not as input for a 3NN classifier. If the metric offers a small value, 3NN is the most promising classification algorithm to be applied. For C4.5 and SVM classifiers this measure does not look interesting.

5.8. Results and analysis for ratio of average intra/inter class nearest neighbor distance (N2)

The results for N2 measure correspond to Fig. 8. The analysis of this figure is the following:

When the minimal value for this metric is reached, it suggests that the examples of the same class lie closely in the feature space. As the measure value increases, the examples of the same class are more disperse.

The 3NN classifier is affected notably (see Fig. 8(a)). For the minimal values of the measure, 3NN classifier improves its behavior, which is decreasing till the highest value of the metric is reached. C4.5 and SVM present discrete accuracy rates due, as we have mentioned before, to the nature of this metric.

So, this metric looks interesting to be applied in a data set to consider the use or not as input for a 3NN classifier. If the metric offers a small value, 3NN is the most promising classification algorithm to be applied. For C4.5 and SVM classifiers this measure does not offer remarkable information.

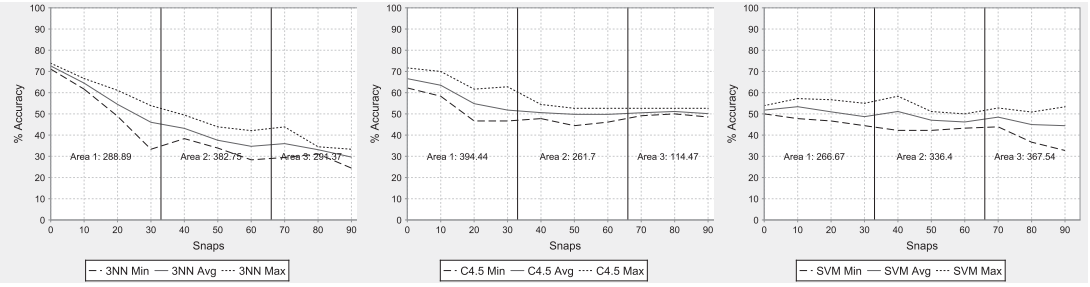
5.9. Results and analysis for error rate of 1 nearest neighbor classifier (N3)

In the case of N3 measure, their results correspond to Figure from Fig. 9. The analysis of this figure is the following:

This metric represents the error rate of 1 nearest neighbor classifier, indicating how close the examples of different class are. When the maximal value for this metric is reached, it suggests that the there is a large gap in the class boundary.

This measure has a clear correspondence with 3NN classifier (see Fig. 9(a)). For the minimal values of the measure, 3NN classifier improves its behavior but the big size of the Area 1 indicates that the behavior is not consistent. When the results become consistent (small Areas), the accuracy rate decreases till the highest value of the metric is reached. C4.5 and SVM present discrete accuracy rates due, as we have mentioned before, to the nature of this metric.

So, the metric does not offer to much information, because it only has sense in the case of 3NN and it presents too close correspondence with it.

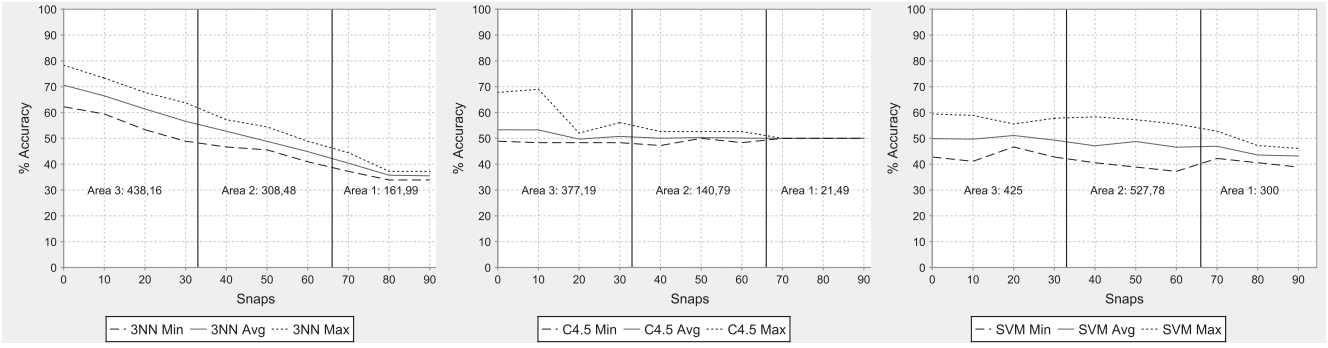


(a) Accuracy results associated to the classifier 3NN in all snaps for N2.

(b) Accuracy results associated to the classifier C4.5 in all snaps for N2.

(c) Accuracy results associated to the classifier SVM in all snaps for N2.

Fig. 8. Behavior of different classifiers for the domain of values of N2.



(a) Accuracy results associated to the classifier 3NN in all snaps for N3.

(b) Accuracy results associated to the classifier C4.5 in all snaps for N3.

(c) Accuracy results associated to the classifier SVM in all snaps for N3.

Fig. 9. Behavior of different classifiers for the domain of values of N3.

5.10. Results and analysis for nonlineality of linear classifier by linear programming (L3)

In the case of L3 measure, their results correspond to Figure from Fig. 10. The analysis of this figure is the following:

This metric is a measure of nonlinearity, as it was defined in Section 2.3.1. The evolution of the instance distributions show linearly separable data sets in the first snaps till non separable ones when the maximum value is reached.

Due to the data generation process and the data distribution reached, C4.5 and SVM are able to reach easy and interpretable models with generalization capabilities.

As in the rest of the measures, the nonlinearity of the instance distribution affects to the accuracy rates of the classifiers analyzed. In this case, the accuracy rate is significantly reduced as de measure increase its value, but the robustness of this accuracy rate is reduced significatively (see Areas in Fig. 10(a)–(c)).

The interesting information that this measure is able to present appears when the metric reach very small values (first 10 snaps), which means that the data set has associated high linear separability, but in most of the metric range it is not possible to establish any relation with the classifiers.

5.11. Results and analysis for nonlineality of 1 nearest neighbor classifier (N4)

This metric, as the previous L3 one, is a measure of nonlinearity but in this case it is based in the nearest neighbor classifier. The evolution of the instance distributions produce linearly separable data sets in the smallest values of the measure till non separable ones when the maximum value is reached (see Fig. 11).

Due to the data generation process in this case, based on the nearest neighbor rule, C4.5 present difficulties to reach easy and interpretable models with generalization capabilities.

In this case, the effect of this measure is not clear in each one of the classifiers considered due to their accuracy behavior is not consistent. This metric does not seem to help to take any kind of decision of prediction about behavior of the classifiers.

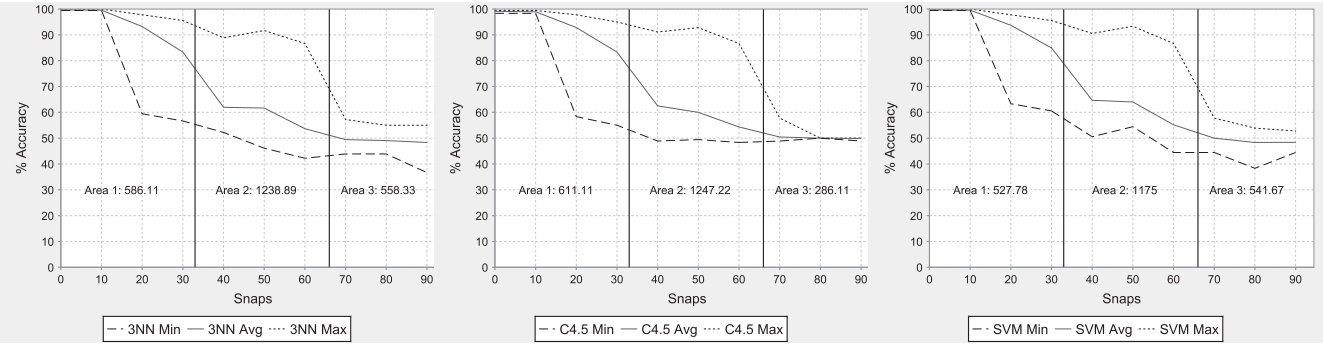
5.12. Results and analysis for fraction of points with associated adherence subsets retained (T1)

This measure represents how much points are clustered in hyperspheres or distributed in thinner structures, so when the measure reaches its minimal values (with minimal number of hyperspheres and the biggest ones) the instances are grouped and next to each other. While the measure increases its value, the instances are dispersed and the size of the hyperspheres needs to be reduce and their number increased.

This metric offers very discrete accuracy rates in all classifiers independently of the value of the measure (see Fig. 12(a)–(c)). This fact indicates that the T1 metric does not offer useful information about classifiers behavior.

5.13. Resume analysis of complexity metrics and their classifiers relation

In this section we offer a resume of the previous results and analysis.

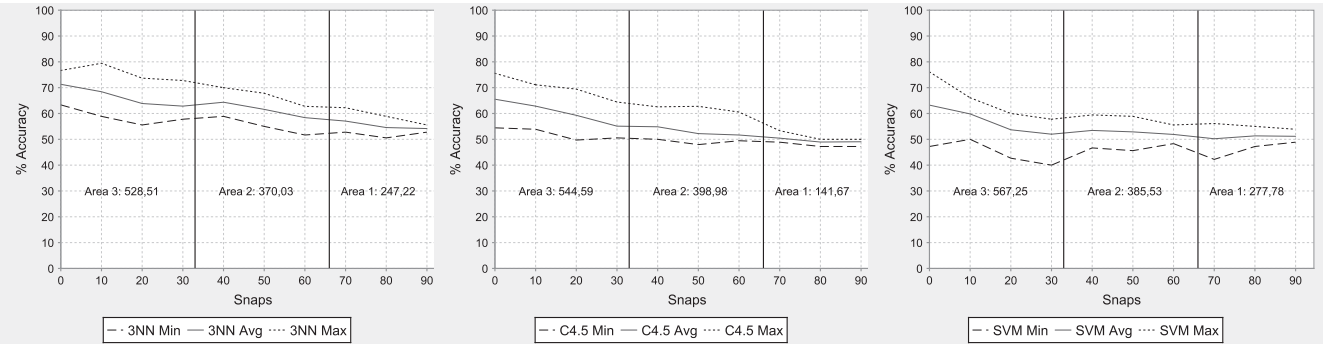


(a) Accuracy results associated to the classifier 3NN in all snaps for L3.

(b) Accuracy results associated to the classifier C4.5 in all snaps for L3.

(c) Accuracy results associated to the classifier SVM in all snaps for L3.

Fig. 10. Behavior of different classifiers for the domain of values of L3.



(a) Accuracy results associated to the classifier 3NN in all snaps for N4.

(b) Accuracy results associated to the classifier C4.5 in all snaps for N4.

(c) Accuracy results associated to the classifier SVM in all snaps for N4.

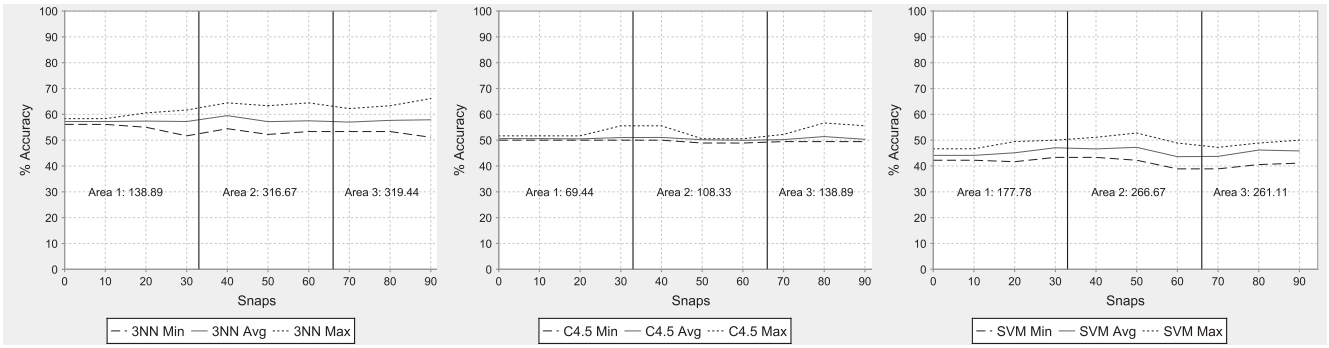
Fig. 11. Behavior of different classifiers for the domain of values of N4.

Considering the complexity metric perspective we can point out:

- Metrics of Overlaps in feature values: The overlapping between classes has demonstrated to affect significantly to the classification task. The variation of all the measures situated in this group (F1, F2 and F3) has shown this effect. F1 is the one in this group which presents the most defined effect on all the classifiers accuracy, being the most robust effect at the same time (which increase its reliability). F2 shows an interesting behavior for 3NN and C4.5, but it is more robust in the C4.5 case. In the case of F3, its variation affects to all classifiers, but their accuracy rates are robust just when the metric reaches its maximal values.
- Separability measures: The behavior of these metrics is different among them. Some of them show highlighted effects on classifiers, while other has not got any king of effect on them. L1 is the one whose variation affects significantly to all the classifiers, being the SVM the most robust one. In the case of L2, there is not clear effect on the classifiers behavior. N1, which is based in nearest neighbor rule, shows a remarkable effect on 3NN classifier, but this effect is not detected on C4.5 and SVM. N2 presents a similar behavior than N1 (useful for the 3NN classifier) with respect to the classifiers, but it is not as robust as the N1 case. Finally, N3 as the previous N1 and N2 measures, reflects its variations in the 3NN classifier, but the accuracy rates suffer in robustness.
- Measures of geometry, topology and density of manifolds: The effect of this group of metrics on the classifiers benefits is not representative. In some cases their effect in the accuracy rate is minimal and when it is significative, it does not appear in a robust manner. The L3 metric is the one which affects to the accuracy rate of all classifiers, but the variability of the accuracy rates produces that this information cannot be used to establish relations with the classifiers. The effect of N4 and T1 is more robust, but the accuracy rates of the classifiers does not offer significant changes.

Regarding to the classifiers considered in this study and the metrics which offer information about their application we can point out the following:

- 3NN: The measures F1,F2 and F3 are interesting tools to predict the effect of the overlapping drawback which can be present in a input data set. In the case of separability metrics, the most valuable ones are those based in nearest neighbor rule (N1, N2 and N3), but all of them suffer of the robustness which produce a reduction in predictive precision. Among the third set of measures, none of them offer to much information, just N4, but it is too close to the classifier so it is most effective and reliable to use the classifier instead the metric.
- C4.5: For this classifier, all the overlapping metrics report interesting information to be considered for the use of it (F1, F2 and F3). The separability metrics, which are based in nearest



(a) Accuracy results associated to the classifier 3NN in all snaps for T1.

(b) Accuracy results associated to the classifier C4.5 in all snaps for T1.

(c) Accuracy results associated to the classifier SVM in all snaps for T1.

Fig. 12. Behavior of different classifiers for the domain of values of T1.

neighbor rule or linear programming, do not present useful information for this classifier. The models that C4.5 generates do not suit with the heuristic considered in this group to calculate the metrics. The situation is similar with the measures of geometry, topology and density of manifolds. None of them offer remarkable relationship between them and the C4.5.

- SVM: As for the previous classifiers, the overlapping metrics are interesting to be analyzed before the use of the SVM. All of them, F1, F2 and F3 report interesting information to predict the performances of the SVM over an input data set. Considering the separability measures, those based on nearest neighbor rule can be discarded, but the ones which use linear programming to be calculated can be remarked. In concrete, L1, whose relation with the SVM appears in accuracy with enough robustness. In the last group of metrics there is not anyone to be highlighted regard to this classifier.

6. Conclusion

This paper addresses the analysis of data complexity metrics by means of artificial data sets to analyze their relation with different nature classifiers.

An experimental study has been carried out generating artificial data sets covering the range of each metric. These data sets and their measures associated have been analyzed evaluating the relation among them and classifiers benefits. The data sets are used as input for C4.5, 3NN and SVM, representing graphically the effect of the measures range over the accuracy rate of the classifiers. The main conclusions reached are the following:

- As Section 3 presents, the meaning of the measures cannot be directly used to establish relationship among data sets with different characteristics (different number of instances, features, classes or instances distribution).
- There exist a clear relation between classifier performances and complexity metrics, which appear in some of the measures analyzed:
  - In the case of overlapping measures, as larger the overlapping is, smaller are the benefits on prediction of all the classifiers analyzed. If we evaluate one data set by means of these measures and they reach the overlapping significant values (small F1, large F2 or small F3), we could diagnose that the classifiers will present a reduction in their prediction capabilities due to the presence of overlapping. F1 is the highlighted metric in this group due to it robust behavior.
  - Paying attention to the separability metrics, we must point out that in this metric exist two groups. One particularly suited to analyze data sets for linear classifiers, considering just the metrics L1 and L2, and the second one, to analyze data sets for nearest neighbor classifiers, based on N1 and N2. In both groups, the evaluation of these metrics and the presence of class separability indicates that the classifiers offer promising prediction capabilities. In this group, the interesting metrics are L1, related to SVM classifier, and N1, in relation to the 3NN. L2 does not offer interesting relations with classifiers. N2 and N3 show a similar behavior than N1, but they are not as robust as N1.
  - The measures related to geometry, topology and density of manifolds offer different information for each classifier. The L3 measure shows an interesting behavior in the accuracy of the classifiers, but its results are not consistent. The effect that the metrics N4 and T1 produce are robust, but they does not affect to the classifier behavior. In these group od metrics, none of them let us establish any significant relation with the classifiers studied.

As final conclusion, we can resume that the use of some of these measures can let us to analyze the nature of the input data set and gives us guidelines to approximate predictions about the behavior of certain classifiers.

References

Asuncion, A. & Newman, D.J. (2007). UCI Machine Learning Repository. <http://www.ics.uci.edu/mlearn/MLRepository.html>, Irvine, CA: University of California, Department of Information and Computer Science.

Basu, M., & Ho, T.K. (1999). The learning behavior of single neuron classifiers on linearly separable or nonseparable input. In *Proceedings of the 1999 International Joint Conference on Neural Networks*, Washinton DC, July.

Basu, M., & Ho, T. K. (2006). *Data complexity in pattern recognition*. Springer.

Baumgartner, R., & Somorjai, R. L. (2006). Data complexity assessment in undersampled classification of high-dimensional biomedical data. *Pattern Recognition Letters, 27*(12), 1383–1389.

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

Cano, J.-R. (2012). Predictive-collaborative model as recovery and validation tool. Case of study: Psychiatric emergency department decision support. *Expert Systems with Applications, 39*, 4044–4048.

Cherkassky, V., & Mulier, F. (1998). *Learning from data: Concepts, theory, and methods*. Wiley-Interscience.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

Dong, M., & Kothari, R. (2003). Feature subset selection using a new definition of classifiability. *Pattern Recognition Letters, 24*(9-10), 1215–1225.

Elizondo, D. A., Birkenhead, R., Gámez, M., García, N., & Alfaro, E. (2012). Linear separability and classification complexity. *Expert Systems with Applications, 39*, 7796–7807.

García, S., Cano, J.-R., Bernadó-Mansilla, E., & Herrera, F. (2009). Diagnose of effective evolutionary prototype selection using an overlapping measure. *International Journal of Pattern Recognition and Artificial Intelligence, 23*(8), 1527–1548.

Ho, T. K. (2008). Data complexity analysis: Linkage between context and solution in classification. *Lecture Notes in Computer Science, 5342*, 986–995.

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

Hoekstra, A. & Duin, R. (1996). On the nonlinearity of pattern classifiers. In *Proceddings of the International Conference on Pattern Recognition* (pp. 271–275).

Kim, Y. (2010). Performance evaluation for classification methods: A comparative simulation study. *Expert Systems with Applications, 37*(3), 2292–2306.

Kim, S. W., & Oommen, B. J. (2008). A fast computation of inter-class overlap measures using prototype reduction schemes. *Lecture Notes in Computer Science, 5032*, 173–184.

Kim, S. W., & Oommen, B. J. (2009). On using prototype reduction schemes to enhance the computation of volume-based inter-class overlap measures. *Pattern Recognition, 42*(2), 2695–2704.

Kininenko, I., & Kukar, M. (2007). *Machine learning and data mining: Introduction to principles and algorithms*. Horwood Publishing Limited.

Lebourgeois, F. & Emptoz, H. (1996). Pretopological approach for supervised learning.In *Proceedings of the 13th International Conference on Pattern Recognition* (pp. 256–260).

Li, Y. H., Dong, M., & Kothari, R. (2005). Classifiability-based omnivariate decision trees. *IEEE Transactions On Neural Networks, 16*(6), 1547–1560.

Luengo, J., Fernández, A., García, S., & Herrera, F. (2011). Addressing data complexity for imbalanced data sets: Analysis of SMOTE-based oversampling and evolutionary undersampling. *Soft Computing, 15*, 1909–1936.

Meyer, D., Leisch, F., & Hornik, K. (2003). The support vector machine under test. *Neurocomputing, 55*, 169–186.

Mollineda, R. A., Sánchez, J. S., & Sotoca, J. M. (2005). Data characterization for effective prototype selection. *Lecture Notes in Computer Science, 3523*, 27–34.

Papadopoulos, A. N., & Manolopoulos, Y. (2004). *Nearest neighbor search: A database perspective*. Springer Verlag.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufman Publishers.

Sánchez, J. S., Mollineda, R. A., & Sotoca, J. M. (2007). An analysis of how training data complexity affects the nearest neighbours classifiers. *Pattern Analysis & Applications, 10*, 189–201.

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

Smith, F. W. (1968). Pattern classifier design by linear programming. *IEEE Transactions on Computers, C-17*(4), 367–372.

Smith, S. P. (1998). A test to determine the multivariate normality of a data set. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 10*(5), 757–761.

Steinwart, I., & Christmann, A. (2008). *Support vector machines information science and statistics*. Springer.

Wu, X., & Kumar, V. (2007). *Top 10 algorithms in data mining*. Chapman & Hall/CRC.

Yildiz, O. T. (2011). Mapping classifiers and datasets. *Expert Systems with Applications, 38*(4), 3697–3702.