How Complex Is Your Classification Problem?: A Survey on Measuring Classification Complexity

Article *in* ACM Computing Surveys · September 2019

DOI: 10.1145/3347711

Ana Carolina Lorena
Instituto Tecnologico de Aeronautica
120 PUBLICATIONS **1,473** CITATIONS

Luís Paulo Faina Garcia
University of Brasília
17 PUBLICATIONS **207** CITATIONS

Jens Lehmann
University of Bonn
292 PUBLICATIONS **12,439** CITATIONS

Marcilio Carlos Pereira de Souto
Université d'Orléans
94 PUBLICATIONS **1,183** CITATIONS

How Complex Is Your Classification Problem?: A Survey on Measuring Classification Complexity

ANA C. LORENA, Instituto Tecnológico de Aeronáutica
LUÍS P. F. GARCIA, University of Brasilia
JENS LEHMANN, University of Bonn
MARCILIO C. P. SOUTO, University of Orléans
TIN KAM HO, IBM Watson

Characteristics extracted from the training datasets of classification problems have proven to be effective predictors in a number of meta-analyses. Among them, measures of classification complexity can be used to estimate the difficulty in separating the data points into their expected classes. Descriptors of the spatial distribution of the data and estimates of the shape and size of the decision boundary are among the known measures for this characterization. This information can support the formulation of new data-driven pre-processing and pattern recognition techniques, which can in turn be focused on challenges highlighted by such characteristics of the problems. This article surveys and analyzes measures that can be extracted from the training datasets to characterize the complexity of the respective classification problems. Their use in recent literature is also reviewed and discussed, allowing to prospect opportunities for future work in the area. Finally, descriptions are given on an R package named Extended Complexity Library (ECoL) that implements a set of complexity measures and is made publicly available.

CCS Concepts: • **Computing methodologies** → **Supervised learning by classification**;

Additional Key Words and Phrases: Supervised machine learning, classification, complexity measures

ACM Reference format:
Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin Kam Ho. 2019. How Complex Is Your Classification Problem?: A Survey on Measuring Classification Complexity. *ACM Comput. Surv.* 52, 5, Article 107 (September 2019), 34 pages.
https://doi.org/10.1145/3347711

1 INTRODUCTION

The work from Ho and Basu (2002) was seminal in analyzing the difficulty of a classification problem by using descriptors extracted from a learning dataset. Given that no Machine Learning (ML)

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors would also like to thank the financial support of the foundations FAPESP (grant 2012/22608-8), CNPq (grants 308858/2014-0 and 305291/2017-3), CAPES (grant 88887.162551/2018-00) and CAPES-COFECUB.

Authors' addresses: A. C. Lorena, Instituto Tecnológico de Aeronáutica, Praça Marechal Eduardo Gomes, 50, São José dos Campos, São Paulo, 12228-900, Brazil; email: aclorena@ita.br; L. P. F. Garcia, University of Brasilia, Asa Norte, Brasília, Distrito Federal, 70910-900, Brazil; email: luis.garcia@unb.br; J. Lehmann, University of Bonn, Römerstraße, 164, Bonn, North Rhine-Westphalia, 53117, Germany; email: jens.lehmann@cs.uni-bonn.de; M. C. P. Souto, University of Orléans, Léonard de Vinci, 6759, Orleans, 6759, France; email: marcilio.desouto@univ-orleans.fr; T. K. Ho, IBM Watson, 1101 Kitchawan Road, Yorktown Heights, New York, USA; email: tho@us.ibm.com.

0360-0300/2019/09-ART107 $15.00
https://doi.org/10.1145/3347711

technique can consistently obtain the best performance for every classification problem (Wolpert 1996), this type of analysis allows to understand the scenarios in which a given ML technique succeeds and fails (Ali and Smith 2006; Flores et al. 2014; Luengo and Herrera 2015; Muñoz et al. 2018). Furthermore, it guides the development of new data-driven pre-processing and pattern recognition techniques, as done in Dong and Kothari (2003), Garcia et al. (2015), Hu et al. (2010), Mollineda et al. (2005), and Smith et al. (2014a). This data-driven approach enables a better understanding of the peculiarities of a given application domain that can be explored to get better prediction results.

According to Ho and Basu (2002), the complexity of a classification problem can be attributed to a combination of three main factors: (i) the ambiguity of the classes; (ii) the sparsity and dimensionality of the data; and (iii) the complexity of the boundary separating the classes. The ambiguity of the classes is present in scenarios in which the classes can not be distinguished using the data provided, regardless of the classification algorithm employed. This is the case for poorly defined concepts and the use of non-discriminative data features. These problems are known to have non-zero Bayes error. An incomplete or sparse dataset also hinders a proper data analysis. This shortage leads to some input space regions to be underconstrained. After training, subsequent data residing in those regions are classified arbitrarily. Finally, Ho and Basu (2002) focus on the complexity of the classification boundary, and present a number of measures that characterize the boundary in different ways. The complexity of classification boundary is related to the size of the smallest description needed to represent the classes and is native of the problem itself (Macià 2011). Using the Kolmogorov complexity concept (Ming and Vitanyi 1993), the complexity of a classification problem can be measured by the size of the smallest algorithm that is able to describe the relationships between the data (Ling and Abu-Mostafa 2006). In the worst case, it would be necessary to list all the objects along with their labels. However, if there is some regularity in the data, a compact algorithm can be obtained. In practice, the Kolmogorov complexity is incomputable and approximations are made, as those based on the computation of indicators and geometric descriptors drawn from the learning datasets available for training a classifer (Ho and Basu 2002; Singh 2003a). We refer to those indicators and geometric descriptors as data complexity measures or simply *complexity measures* from here on.

This article surveys the main complexity measures that can be obtained directly from the data available for learning. It extends the work from Ho and Basu (2002) by including more measures from literature that may complement the concepts already covered by the measures proposed in their work. The formulations of some of the measures are also adapted so they can give standardized results. The usage of the complexity measures through recent literature is reviewed, too, highlighting various domains where an advantageous use of the measures can be achieved. Besides, the main strengths and weakness of each measure are reported. As a side result, this analysis provides insights into adaptations needed with some of the measures and into new unexplored areas where the complexity measures can succeed.

All measures detailed in this survey were assembled into an R package named ECoL (*Extended Complexity Library*). It contains all the measures from the DCoL (*Data Complexity*) library (Orriols-Puig et al. 2010), which were standardized and reimplemented in R, and a set of novel measures from the related literature. The added measures were chosen to complement the concepts assessed by the original complexity measures. Some corrections into the existent measures are also discussed and detailed in the article. The ECoL package is publicly available at CRAN[1] and GitHub.[2]

This article is structured as follows: Section 2 presents the most relevant complexity measures. Section 3 presents and analyzes the complexity measures included in the ECol package. Section 4

[1] https://cran.r-project.org/package=ECoL.

[2] https://github.com/lpfgarcia/ECoL.

presents some applications of the complexity measures in the ML literature. Section 5 concludes this work.

2 COMPLEXITY MEASURES

Geometric and statistical data descriptors are among the most used in the characterization of the complexity of classification problems. Among them are the measures proposed in Ho and Basu (2002) to describe the complexity of the boundary needed to separate binary classification problems, later extended to multiclass classification problems in works such as Ho et al. (2006), Mollineda et al. (2005, 2006), and Orriols-Puig et al. (2010). Ho and Basu (2002) divide their measures into three main groups: (i) measures of overlap of individual feature values; (ii) measures of the separability of classes; and (iii) geometry, topology, and density of manifolds measures. Similarly, Sotoca et al. (2005) divide the complexity measures into the categories: (i) measures of overlap; (ii) measures of class separability; and (iii) measures of geometry and density. In this article, we group the complexity measures into more categories, as follows:

1. **Feature-based measures**, which characterize how informative the available features are to separate the classes;
2. **Linearity measures**, which try to quantify whether the classes can be linearly separated;
3. **Neighborhood measures**, which characterize the presence and density of same or different classes in local neighborhoods;
4. **Network measures**, which extract structural information from the dataset by modeling it as a graph;
5. **Dimensionality measures**, which evaluate data sparsity based on the number of samples relative to the data dimensionality;
6. **Class imbalance measures**, which consider the ratio of the numbers of examples between classes.

To define the measures, we consider that they are estimated from a learning dataset T (or part of it) containing n pairs of examples $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$ and $y_i \in \{1, \ldots, n_c\}$. That is, each example $\mathbf{x}_i$ is described by m predictive features and has a label y_i out of n_c classes. Most of the measures are defined for features with numerical values only. In this case, symbolic values must be properly converted into numerical values prior to their use. We also use an assumption that linearly separable problems can be considered simpler than classification problems requiring non-linear decision boundaries. Finally, some measures are defined for binary classification problems only. In that case, a multiclass problem must first be decomposed into multiple binary sub-problems. Here we adopt a pairwise analysis of the classes; that is, a one-versus-one (OVO) decomposition of the multiclass problem (Lorena et al. 2008). The measure for the multiclass problem is then defined as the average of the values across the different sub-problems. To standardize the interpretation of the measure values, we introduce some modifications into the original definitions of some of the measures. The objective was to make each measure assume values in bounded intervals and also to make higher values of the measures indicative of a higher complexity, while lower values indicate a lower complexity.

2.1 Feature-based Measures

These measures evaluate the discriminative power of the features. In many of them, each feature is evaluated individually. If there is at least one very discriminative feature in the dataset, the problem can be considered simpler than if there is no such attribute. All measures from this category require the features to have numerical values. Most of the measures are also defined for binary classification problems only.

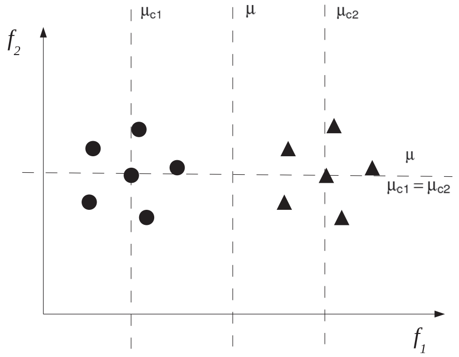


Fig. 1. Example of F1 computation for a two-class dataset.

2.1.1 Maximum Fisher's Discriminant Ratio (F1). The first measure presented in this category is the maximum Fisher's discriminant ratio, denoted by F1. It measures the overlap between the values of the features in different classes and is given by:

$$F1 = \frac{1}{1 + \max_{i=1}^{m} r_{f_i}}, \tag{1}$$

where r_{f_i} is a discriminant ratio for each feature f_i. Originally, F1 takes the value of the largest discriminant ratio among all the available features. This is consistent with the definition that if at least one feature discriminates the classes, the dataset can be considered simpler than if no such attribute exists. In this article, we take the inverse of the original F1 formulation into account, as presented in Equation (1). Herewith, the F1 values become bounded in the $(0, 1]$ interval and higher values indicate more complex problems, where no individual feature is able to discriminate the classes.

Orriols-Puig et al. (2010) present different equations for calculating r_{f_i}, depending on the number of classes or whether the features are continuous or ordinal (Cummins 2013). One straightforward formulation is:

$$r_{f_i} = \frac{\sum_{j=1}^{n_c} \sum_{k=1, k \neq j}^{n_c} p_{c_j} p_{c_k} (\mu_{c_j}^{f_i} - \mu_{c_k}^{f_i})^2}{\sum_{j=1}^{n_c} p_{c_j} (\sigma_{c_j}^{f_i})^2}, \tag{2}$$

where p_{c_j} is the proportion of examples in class c_j, $\mu_{c_j}^{f_i}$ is the mean of feature f_i across examples of class c_j, and $\sigma_{c_j}^{f_i}$ is the standard deviation of such values. An alternative for r_{f_i} computation that can be employed for both binary and multiclass classification problems is given in Mollineda et al. (2005). Here, we adopt this formulation, which is similar to the clustering validation index from Caliński and Harabasz (1974):

$$r_{f_i} = \frac{\sum_{j=1}^{n_c} n_{c_j} (\mu_{c_j}^{f_i} - \mu^{f_i})^2}{\sum_{j=1}^{n_c} \sum_{l=1}^{n_{c_j}} (x_{li}^{j} - \mu_{c_j}^{f_i})^2}, \tag{3}$$

where n_{c_j} is the number of examples in class c_j, $\mu_{c_j}^{f_i}$ is the same as defined for Equation (2), μ^{f_i} is the mean of the f_i values across all the classes, and x_{li}^{j} denotes the individual value of the feature f_i for an example from class c_j. Taking, for instance, the dataset shown in Figure 1, the most discriminative feature would be f_1. F1 correctly indicates that the classes can be easily separable using this feature. Feature f_2, however, is non-discriminative, since its values for the two classes overlap with the same mean and variance.

The denominator in Equation (3) must go through all examples in the dataset. The numerator goes through the classes. Since the discriminant ratio must be computed for all features, the total asymptotic cost for the F1 computation is $O(m \cdot (n + n_c))$. As $n \geq n_c$ (there is at least one example per class), $O(m \cdot (n + n_c))$ can be reduced to $O(m \cdot n)$.

Roughly, the F1 measure computes the ratio of inter-class to the intra-class scatter for each feature. Using the formulation in Equation (1), low values of the F1 measure indicate that there is at least one feature whose values show little overlap among the different classes; that is, it indicates the existence of a feature for which a hyperplane perpendicular to its axis can separate the classes fairly. Nonetheless, if the required hyperplane is oblique to the feature axes, F1 may not be able to reflect the underlying simplicity. To deal with this issue, Orriols-Puig et al. (2010) propose to use a F1 variant based on a *Directional Vector*, to be discussed next. Finally, Hu et al. (2010) note that the F1 measure is most effective if the probability distributions of the classes are approximately normal, which is not always true. On the contrary, there can be highly separable classes, such as those distributed on the surfaces of two concentric hyperspheres, that would yield a very high value for F1.

2.1.2 The Directional-vector Maximum Fisher's Discriminant Ratio (F1v). This measure is used in Orriols-Puig et al. (2010) as a complement to F1. It searches for a vector that can separate the two classes after the examples have been projected into it and considers a directional Fisher criterion defined in Malina (2001) as:

$$dF = \frac{\mathbf{d}^t\mathbf{B}\mathbf{d}}{\mathbf{d}^t\mathbf{W}\mathbf{d}}, \tag{4}$$

where $\mathbf{d}$ is the directional vector onto which data are projected to maximize class separation, $\mathbf{B}$ is the between-class scatter matrix, and $\mathbf{W}$ is the within-class scatter matrix. $\mathbf{d}$, $\mathbf{B}$, and $\mathbf{W}$ are defined according to Equations (5), (6), and (7), respectively.

$$\mathbf{d} = \mathbf{W}^{-1}(\mu_{c_1} - \mu_{c_2}), \tag{5}$$

where μ_{c_i} is the centroid (mean vector) of class c_i and $\mathbf{W}^{-1}$ is the pseudo-inverse of $\mathbf{W}$.

$$\mathbf{B} = (\mu_{c_1} - \mu_{c_2})(\mu_{c_1} - \mu_{c_2})^t, \tag{6}$$

$$\mathbf{W} = p_{c_1}\Sigma_{c_1} + p_{c_2}\Sigma_{c_2}, \tag{7}$$

where p_{c_i} is the proportion of examples in class c_i and Σ_{c_1} is the scatter matrix of class c_i.

Taking the definition of dF, the F1v measure is given by:

$$F1v = \frac{1}{1 + dF}. \tag{8}$$

According to Orriols-Puig et al. (2010), the asymptotic cost of the F1v algorithm for a binary classification problem is $O(m \cdot n + m^3)$. Multiclass problems are first decomposed according to the OVO strategy, producing $\frac{n_c(n_c-1)}{2}$ subproblems. In the case that each one of them has the same number of examples—that is, $\frac{n}{n_c}$—the total cost of the F1v measure computation is $O(m \cdot n \cdot n_c + m^3 \cdot n_c^2)$.

Lower values in F1v defined by Equation (8), which are bounded in the $(0, 1]$ interval, indicate simpler classification problems. In this case, a linear hyperplane will be able to separate most if not all of the data in a suitable orientation with regard to the features axes. This measure can be quite costly to compute due to the need for the pseudo-inverse of the scatter matrix. Like F1, it is based on the assumption of normality of the classes distributions.

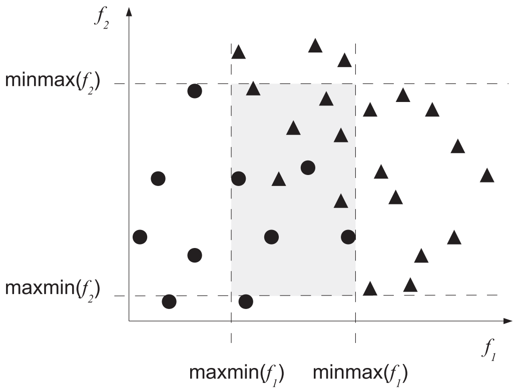


Fig. 2. Example of overlapping region.

2.1.3 Volume of Overlapping Region (F2). The F2 measure calculates the overlap of the distributions of the feature values within the classes. It can be determined by finding, for each feature f_i, its minimum and maximum values in the classes. The range of the overlapping interval is then calculated, normalized by the range of the values in both classes. Finally, the obtained values are multiplied, as shown in Equation (9):

$$F2 = \prod_{i}^{m} \frac{overlap(f_i)}{range(f_i)} = \prod_{i}^{m} \frac{\max\{0, \min\max(f_i) - \max\min(f_i)\}}{\max\max(f_i) - \min\min(f_i)}, \tag{9}$$

where:

$$\begin{aligned} \min\max(f_i) &= \min(\max(f_i^{c_1}), \max(f_i^{c_2})), \\ \max\min(f_i) &= \max(\min(f_i^{c_1}), \min(f_i^{c_2})), \\ \max\max(f_i) &= \max(\max(f_i^{c_1}), \max(f_i^{c_2})), \\ \min\min(f_i) &= \min(\min(f_i^{c_1}), \min(f_i^{c_2})). \end{aligned}$$

The values $\max(f_i^{c_j})$ and $\min(f_i^{c_j})$ are the maximum and minimum values of each feature in a class $c_j \in \{1, 2\}$, respectively. The numerator becomes zero when the per-class value ranges are disjoint for at least one feature. This equation uses a correction that was made in Souto et al. (2010) and Cummins (2013) to the original definition of F2, which may yield negative values for non-overlapping feature ranges. The asymptotic cost of this measure is $O(m \cdot n \cdot n_c)$, considering a OVO decomposition in the case of multiclass problems. The higher the F2 value, the greater the amount of overlap between the problem classes. Therefore, the problem's complexity is also higher. Moreover, if there is at least one non-overlapping feature, the F2 value should be zero. Figure 2 illustrates the region that F2 tries to capture (as the shaded area), for a dataset with two features and two classes.

Cummins (2013) points to an issue with F2 for the cases illustrated in Figure 3. In Figure 3(a), the attribute is discriminative but the minimum and maximum values overlap in the different classes; and in Figure 3(b), there is one noisy example that disrupts the measure values. Cummins (2013) proposes to deal with these situations by counting the number of feature values in which there is an overlap, which is only suitable for discrete-valued features. Using this solution, continuous features must be discretized *a priori*, which imposes the difficulty of choosing a proper discretization technique and associated parameters, an open issue in data mining (Kotsiantis and Kanellopoulos 2006).

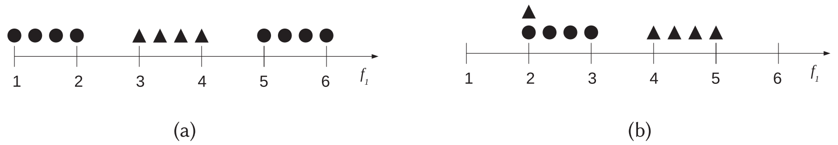


Fig. 3. Problematic situations for F2.

It should be noted that the situations shown in Figure 3 can also be harmful for the F1 measure. As noted by Hu et al. (2010), F2 does not capture the simplicity of a linear oblique border either, since it assumes again that the linear boundary is perpendicular to one of the features axes. Finally, the F2 value can become very small depending on the number of operands in Equation (9); that is, it is highly dependent on the number of features a dataset has. This worsens for problems with many features, so their F2 values may not be comparable to those of other problems with fewer features. Souto et al. (2010), Lorena et al. (2012), and, more recently, Seijo-Pardo et al. (2019) use a sum instead of the product in Equation (9), which partially solves the problems identified. Nonetheless, the result is not an overlapping volume and corresponds to the amount or size of the overlapping region. In addition, the measure remains influenced by the number of features the dataset has.

2.1.4 Maximum Individual Feature Efficiency (F3). This measure estimates the individual efficiency of each feature in separating the classes and considers the maximum value found among the m features. Here, we take the complement of this measure so higher values are obtained for more complex problems. For each feature, it checks whether there is overlap of values between examples of different classes. If there is overlap, the classes are considered to be ambiguous in this region. The problem can be considered simpler if there is at least one feature that shows low ambiguity between the classes, so F3 can be expressed as:

$$F3 = \min_{i=1}^{m} \frac{n_o(f_i)}{n}, \tag{10}$$

where $n_o(f_i)$ gives the number of examples that are in the overlapping region for feature f_i and can be expressed by Equation (11). Low values of F3, computed by Equation (10), indicate simpler problems, where few examples overlap in at least one dimension. As with F2, the asymptotic cost of the F3 measure is $O(m \cdot n \cdot n_c)$.

$$n_o(f_i) = \sum_{j=1}^{n} I(x_{ji} > \max\min(f_i) \wedge x_{ji} < \min\max(f_i)) \tag{11}$$

In Equation (11), I is the indicator function, which returns 1 if its argument is true and 0 otherwise, while $\max\min(f_i)$ and $\min\max(f_i)$ are the same as defined for F2.

Figure 4 depicts the computation of F3 for the same dataset from Figure 2. While for feature f_1 the proportion of examples that are in the overlapping region is $\frac{14}{30}$ (Figure 4(a)), for f_2 this proportion is $\frac{25}{30}$ (Figure 4(b)), resulting in a F3 value of $\frac{14}{30}$.

Since $n_o(f_i)$ is calculated taking into account the minimum and maximum values of the feature f_i in different classes, it entails the same problems identified for F2 with respect to: classes in which the feature has more than one valid interval (Figure 3(a)), susceptibility to noise (Figure 3(b)) and the fact that it is assumed that in linearly separable problems, the boundary is perpendicular to an input axis.

2.1.5 Collective Feature Efficiency (F4). The F4 measure was proposed in Orriols-Puig et al. (2010) to get an overview of how the features work together. It successively applies a procedure similar to that adopted for F3. First the most discriminative feature according to F3 is selected; that

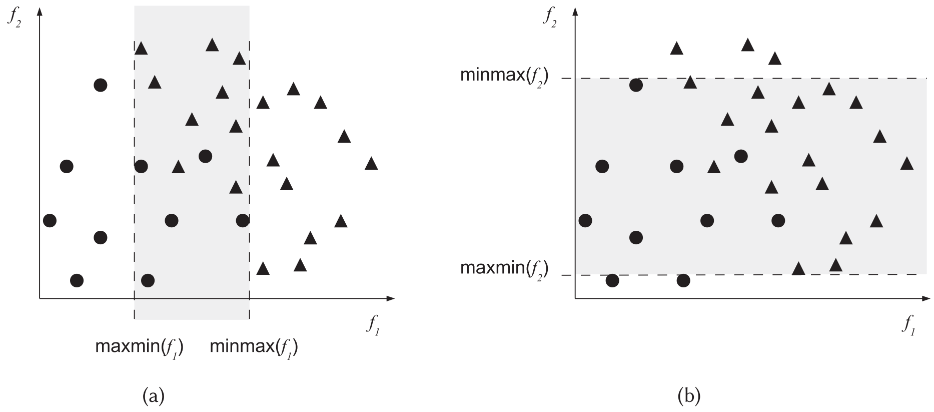


Fig. 4. Calculating F3 for the dataset from Figure 2.

is, the feature that shows less overlap between different classes. All examples that can be separated by this feature are removed from the dataset and the previous procedure is repeated: the next most discriminative feature is selected, excluding the examples already discriminated. This procedure is applied until all the features have been considered and can also be stopped when no example remains. F4 considers the ratio of examples that have not been discriminated, as presented in Equation (12). F4 is computed after l rounds are performed through the dataset, where l is in the range $[1, m]$. If one of the input features is already able to discriminate all the examples in T, l is 1, while it can get up to m in the case all features have to be considered. Its equation can be denoted by:

$$F4 = \frac{n_o(f_{min}(T_l))}{n}, \tag{12}$$

where $n_o(f_{min}(T_l))$ measures the number of points in the overlapping region of feature f_{min} for the dataset from the lth round (T_l). This is the current most discriminative feature in T_l. Taking the ith iteration of F4, the most discriminative feature in dataset T_i can be found using Equation (13), adapted from F3:

$$f_{min}(T_i) = \left\{ f_j | \min_{j=1}^{m} (n_o(f_j)) \right\}_{T_i} \tag{13}$$

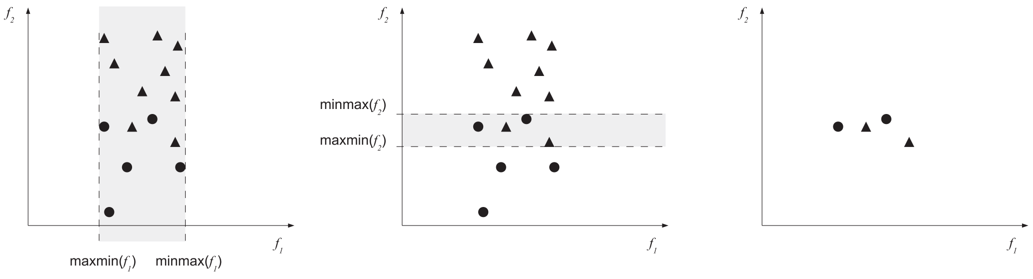
where $n_o(f_j)$ is computed according to Equation (11). While the dataset at each round can be defined as:

$$T_1 = T, \tag{14}$$

$$T_i = T_{i-1} - \{\mathbf{x}_j | x_{ji} < \max\min(f_{min}(T_{i-1})) \vee x_{ji} > \min\max(f_{min}(T_{i-1})). \tag{15}$$

That is, the dataset at the ith round is reduced by removing all examples that are already discriminated by the previous considered feature $f_{min}(T_{i-1})$. Therefore, the computation of F4 is similar to that of F3, except that it can be applied to reduced datasets. Lower values of F4 computed by Equation (12) indicate that it is possible to discriminate more examples and, therefore, the problem is simpler. The idea is to get the number of examples that can be correctly classified if hyperplanes perpendicular to the axes of the features are used in their separation. Since the overlapping measure applied is similar to that used for F3, they share the same problems in some estimates (as discussed for Figures 3(a) and 3(b)). F4 applies the F3 measure multiple times and at most it will iterate for all input features, resulting in a worst-case asymptotic cost of $O(m^2 \cdot n \cdot n_c)$.

Figure 5 shows the F4 operation for the dataset from Figure 2. Feature f_1 is the most discriminative in the first round (Figure 4(a)). Figure 5(a) shows the resulting dataset after all examples



(a) (b) (c)

Fig. 5. Calculating F4 for the dataset from Figure 2.

correctly discriminated by f_1 are disregarded. Figure 5(c) shows the final dataset after feature f_2 has been analyzed in Figure 5(b). The F4 value for this dataset is $\frac{4}{30}$.

2.2 Measures of Linearity

These measures try to quantify to what extent the classes are linearly separable; that is, if it is possible to separate the classes by a hyperplane. They are motivated by the assumption that a linearly separable problem can be considered simpler than a problem requiring a non-linear decision boundary. To obtain the linear classifier, Ho and Basu (2002) suggest to solve an optimization problem proposed by Smith (1968), while in Orriols-Puig et al. (2010) a linear Support Vector Machine (SVM) (Cristianini and Shawe-Taylor 2000) is used instead. Here, we adopt the SVM solution.

The hyperplane sought in the SVM formulation is the one that separates the examples from different classes with a maximum margin while minimizing training errors. This hyperplane is obtained by solving the following optimization problem:

$$\underset{\mathbf{w},b,\epsilon}{\text{Minimize}} \quad \frac{1}{2}\,||\mathbf{w}||^2 + C\left(\sum_{i=1}^{n} \varepsilon_i\right) \tag{16}$$

$$\text{Subject to:}\begin{cases} y_i\,(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \varepsilon_i, \\ \varepsilon_i \geq 0, i = 1, \ldots, n, \end{cases} \tag{17}$$

where C is the trade-off between the margin maximization, achieved by minimizing the norm of $\mathbf{w}$, and the minimization of the training errors, modeled by ε. The hyperplane is given by $\mathbf{w}\cdot\mathbf{x} + b = 0$, where $\mathbf{w}$ is a weight vector and b is an offset value. SVMs are originally proposed to solve binary classification problems with numerical features. Therefore, symbolic features must be converted into numerical values and multiclass problems must first be decomposed.

2.2.1 Sum of the Error Distance by Linear Programming (L1). This measure assesses if the data are linearly separable by computing, for a dataset, the sum of the distances of incorrectly classified examples to a linear boundary used in their classification. If the value of L1 is zero, then the problem is linearly separable and can be considered simpler than a problem for which a non-linear boundary is required.

Given the SVM hyperplane, the error distance of the erroneous instances can be computed by summing up the ε_i values. For examples correctly classified, ε_i will be zero while it indicates the distance of the example to the linear boundary otherwise. This is expressed in Equation (18). The ε_i values are determined in the SVM optimization process.

$$SumErrorDist = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i. \tag{18}$$

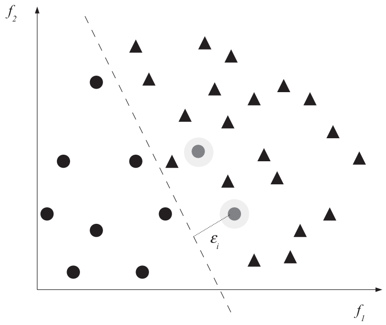


Fig. 6. Example of L1 and L2 computation. The examples misclassified by the linear SVM are highlighted in gray.

The L1 value can then be computed as:

$$L1 = 1 - \frac{1}{1 + SumErrorDist} = \frac{SumErrorDist}{1 + SumErrorDist}. \tag{19}$$

Low values for L1 (bounded in $[0, 1)$) indicate that the problem is close to being linearly separable—that is, simpler. Figure 6 illustrates an example of L1 application. After a linear boundary is obtained, the ϵ_i values of the misclassified examples (gray circles) are summed up and subject to Equation (19).

L1 does not allow to check if a linearly separable problem is simpler than another that is also linearly separable. Therefore, a dataset for which data are distributed narrowly along the linear boundary will have a null L1 value, and so will a dataset in which the classes are far apart with a large margin of separation. The asymptotic computing cost of the measure is dependent on that of the linear SVM, and can take $O(n^2)$ operations in the worst case (Bottou and Lin 2007). In multiclass classification problems decomposed according to OVO, this cost would be $O(n_c^2 \cdot (\frac{n}{n_c})^2)$, which resumes to $O(n^2)$, too.

2.2.2 Error Rate of Linear Classifier (L2). The L2 measure computes the error rate of the linear SVM classifier. Let $h(\mathbf{x})$ denote the linear classifier obtained. L2 is then given by:

$$L2 = \frac{\sum_{i=1}^{n} I(h(\mathbf{x}_i) \neq y_i)}{n}. \tag{20}$$

Higher L2 values denote more errors and therefore a greater complexity regarding the aspect that the data cannot be separated linearly. For the dataset in Figure 6, the L2 value is $\frac{2}{30}$. L2 has similar issues with L1 in that it does not differentiate between problems that are barely linearly separable (i.e., with a narrow margin) from those with classes that are very far apart. The asymptotic cost of L2 is the same of L1, that is, $O(n^2)$.

2.2.3 Non-Linearity of a Linear Classifier (L3). This measure uses a methodology proposed by Hoekstra and Duin (1996). It first creates a new dataset by interpolating pairs of training examples of the same class. Herewith, two examples from the same class are chosen randomly and they are linearly interpolated (with random coefficients), producing a new example. Figure 7 illustrates the generation of six new examples (in gray) from a base training dataset. Then a linear classifier is trained on the original data and has its error rate measured in the new data points. This index is sensitive to how the data from a class are distributed in the border regions and also on how much the convex hulls which delimit the classes overlap. In particular, it detects the presence of concavities in the class boundaries (Armano and Tamponi 2016). Higher values indicate a greater

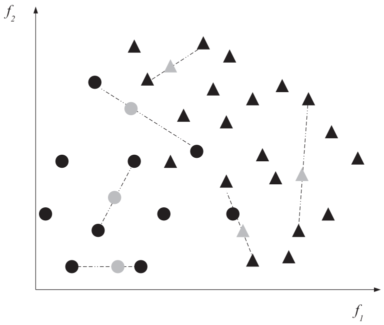


Fig. 7. Example of how new points are generated in measures L3 and N4.

complexity. Letting $h_T(\mathbf{x})$ denote the linear classifier induced from the original training data T, the L3 measure can be expressed by:

$$L3 = \frac{1}{l}\sum_{i=1}^{l} I(h_T(\mathbf{x}'_i) \neq y'_i), \tag{21}$$

where l is the number of interpolated examples $\mathbf{x}'_i$ and their corresponding labels are denoted by y'_i. In ECoL, we generate the interpolated examples maintaining the proportion of examples per class from the original dataset and use $l = n$. The asymptotic cost of this measure is dependent on both the induction of a linear SVM and the time taken to obtain the predictions for the l test examples, resulting in $O(n^2 + m \cdot l \cdot n_c)$.

2.3 Neighborhood Measures

These measures try to capture the shape of the decision boundary and characterize the class overlap by analyzing local neighborhoods of the data points. Some of them also capture the internal structure of the classes. All of them work over a distance matrix storing the distances between all pairs of points in the dataset. To deal with both symbolic and numerical features, we adopt a heterogeneous distance measure named Gower (1971). For symbolic features, the Gower metric computes if the compared values are equal, while for numerical features, a normalized difference of values is taken.

2.3.1 Fraction of Borderline Points (N1). In this measure, first a Minimum Spanning Tree (MST) is built from data, as illustrated in Figure 8. Herewith, each vertex corresponds to an example and the edges are weighted according to the distance between them. N1 is obtained by computing the percentage of vertices incident to edges connecting examples of opposite classes in the generated MST. These examples are either on the border or in overlapping areas between the classes. They can also be noisy examples surrounded by examples from another class. Therefore, N1 estimates the size and complexity of the required decision boundary through the identification of the critical points in the dataset: those very close to each other but belonging to different classes. Higher N1 values indicate the need for more complex boundaries to separate the classes and/or that there is a large amount of overlapping between the classes. N1 can be expressed as:

$$N1 = \frac{1}{n}\sum_{i=1}^{n} I((\mathbf{x}_i, \mathbf{x}_j) \in MST \;\wedge\; y_i \neq y_j). \tag{22}$$

To build the graph from the data, it is necessary to first compute the distance matrix between all pairs of elements, which requires $O(m \cdot n^2)$ operations. Next, using *Prim's algorithm* for obtaining

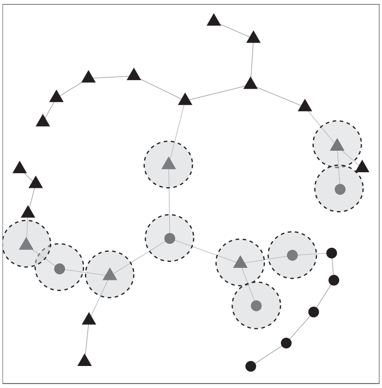


Fig. 8. Example of MST generated for the dataset from Figure 2 and the detected points in the decision border.

the MST requires $O(n^2)$ operations in the worst case. Therefore, the total asymptotic complexity of N1 is $O(m \cdot n^2)$.

N1 is sensitive to the type of noise where the closest neighbors of noisy examples have a different class from their own, as typical in the scenario where erroneous class labels are introduced during data preparation. Datasets with this type of noise are considered more complex than their clean counterparts, according to the N1 measure, as observed in Lorena et al. (2012) and Garcia et al. (2015).

Another issue is that there can be multiple MSTs valid for the same set of points. Cummins (2013) propose to generate 10 MSTs by presenting the data points in different orderings and reporting an average N1 value. Basu and Ho (2006) also report that the N1 value can be large even for a linearly separable problem. This happens when the distances between borderline examples are smaller than the distances between examples from the same class. However, Ho (2002) suggests that a problem with a complicated nonlinear class boundary can still have relatively few edges among examples from different classes as long as the data points are compact within each class.

2.3.2 Ratio of Intra/Extra Class Nearest Neighbor Distance (N2). This measure computes the ratio of two sums: (i) the sum of the distances between each example and its closest neighbor from the same class (intra-class); and (ii) the sum of the distances between each example and its closest neighbor from another class (extra-class). This is shown in Equation (24).

$$intra\_extra = \frac{\sum_{i=1}^{n} d(\mathbf{x_i}, NN(\mathbf{x}_i) \in y_i)}{\sum_{i=1}^{n} d(\mathbf{x_i}, NN(\mathbf{x}_i) \in y_j \neq y_i)}, \tag{23}$$

where $d(\mathbf{x_i}, NN(\mathbf{x}_i) \in y_i)$ corresponds to the distance of example $\mathbf{x}_i$ to its nearest neighbor (NN) from its own class y_i and $d(\mathbf{x_i}, NN(\mathbf{x}_i) \in y_j \neq y_i)$ represents the distance of $\mathbf{x}_i$ to the closest neighbor from another class $y_j \neq y_i$ ($\mathbf{x}_i$'s nearest enemy). Based on the intra/extra class calculation, N2 can be obtained as:

$$N2 = 1 - \frac{1}{1 + intra\_extra} = \frac{intra\_extra}{1 + intra\_extra}. \tag{24}$$

The computation of N2 requires obtaining the distance matrix between all pairs of elements in the dataset, which requires $O(m \cdot n^2)$ operations. Figure 9 illustrates the intra- and extra-class distances for a particular example in a dataset.

Low N2 values are indicative of simpler problems, in which the overall distance between examples of different classes exceeds the overall distance between examples from the same class. N2 is sensitive to how data are distributed within classes and not only to how the boundary between

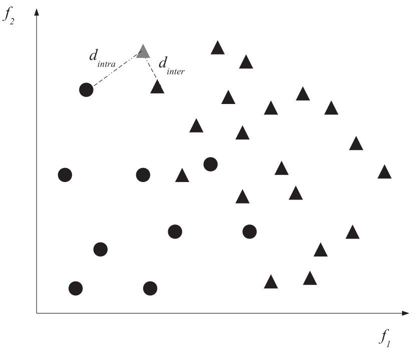


Fig. 9. Example of intra and inter class distances for a particular example.

the classes is like. It can also be sensitive to labeling noise in the data, just like N1. According to Ho (2002), a high N2 value can also be obtained for a linearly separable problem where the classes are distributed in a long, thin, and sparse structure along the boundary. It must also be observed that N2 is related to F1 and F1v, since they all assess intra and inter class variabilities. However, N2 uses a distance that summarizes the joint relationship between the values of all the features for the concerned examples.

2.3.3 Error Rate of the Nearest Neighbor Classifier (N3). The N3 measure refers to the error rate of a 1NN classifier that is estimated using a leave-one-out procedure. The following equation denotes this measure:

$$N3 = \frac{\sum_{i=1}^{n} I(NN(\mathbf{x}_i) \neq y_i)}{n}, \tag{25}$$

where $NN(\mathbf{x}_i)$ represents the nearest neighbor classifier's prediction for example $\mathbf{x}_i$ using all the others as training points. High N3 values indicate that many examples are close to examples of other classes, making the problem more complex. N3 requires $O(m \cdot n^2)$ operations.

2.3.4 Non-Linearity of the Nearest Neighbor Classifier (N4). This measure is similar to L3, but uses the NN classifier instead of the linear predictor. It can be expressed as:

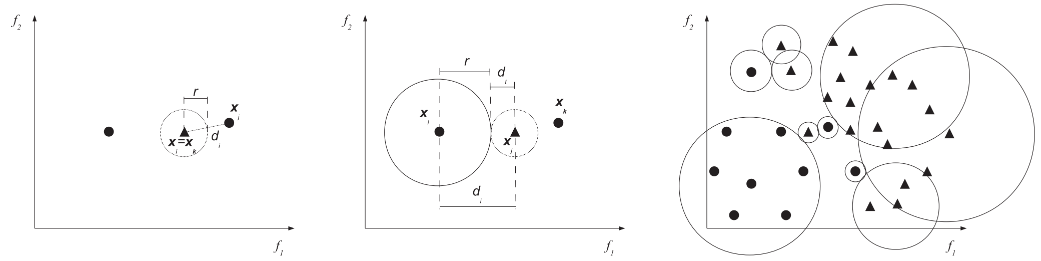
$$N4 = \frac{1}{l} \sum_{i=1}^{l} I(NN_T(\mathbf{x}'_i) \neq y'_i), \tag{26}$$

where l is the number of interpolated points, generated as illustrated in Figure 7. Higher N4 values are indicative of problems of greater complexity. In contrast to L3, N4 can be applied directly to multiclass classification problems without the need to decompose them into binary subproblems first. The asymptotic cost of computing N4 is $O(m \cdot n \cdot l)$ operations, as it is necessary to compute the distances between all possible testing and training examples.

2.3.5 Fraction of Hyperspheres Covering Data (T1). This is regarded as a topological measure in Ho and Basu (2002). It uses a process that builds hyperspheres centered at each one of the examples. The radius of each hypersphere is progressively increased until the hypersphere reaches an example of another class. Smaller hyperspheres contained in larger hyperspheres are eliminated. T1 is defined as the ratio between the number of the remaining hyperspheres and the total number of examples in the dataset:

$$T1 = \frac{\sharp Hyperspheres(T)}{n} \tag{27}$$

where $\sharp Hyperspheres(T)$ gives the number of hyperspheres that are needed to cover the dataset.



(a) Hyperspheric radiuses of two examples that are mutual nearest enemies

(b) Radius r obtained recursively

(c) Final hyperspheres for a dataset

Fig. 10. Calculating T1 for a dataset.

The hyperspheres represent a form of adherence subsets as discussed in Lebourgeois and Emptoz (1996). The idea is to obtain an adherence subset of maximum order for each example such that it includes only examples from the same class. Subsets that are completely included in other subsets are discarded. In principle, the adherence subsets can be of any form (e.g., hyperectangular), and hyperspheres are chosen in the definition of this measure, because it can be defined with relatively few parameters (i.e., only a center and a radius). Fewer hyperspheres are obtained for simpler datasets. This happens when data from the same class are densely distributed and close together. Herewith, this measure also captures the distribution of data within the classes and not only their distribution near the class boundary.

In this article, we propose an alternative implementation of T1. It involves a modification of the definition to stop the growth of the hypersphere when the hyperspheres centered at two points of opposite classes just start to touch. With this modification, the radius of each hypersphere around an example can be directly determined based on distance matrix between all examples. The radius computation for an example $\mathbf{x}_i$ is shown in Algorithm 1, in which the nearest enemy (*ne*) of an example corresponds to the nearest data point from an opposite class ($ne(\mathbf{x}_i) = NN(\mathbf{x}_i) \in y_j \neq y_i$). If two points are mutually nearest enemies of each other (line 3 in Algorithm 1), the radiuses of their hyperspheres correspond to half of the distance between them (lines 4 and 5; see also Figure 10(a)). The radiuses of the hyperspheres around other examples can be determined recursively (lines 7 and 8), as illustrated in Figure 10(b).

Once the radiuses of all hyperspheres are found, a post-processing step can be applied to verify which hyperspheres are absorbed: those lying inside larger hyperspheres. The hyperspheres obtained for our example dataset is shown in Figure 10(c). The most demanding operation in T1 is to compute the distance matrix between all the examples in the dataset, which requires $O(m \cdot n^2)$ operations.

2.3.6 Local Set Average Cardinality (LSC). According to Leyva et al. (2014), the Local-Set (LS) of an example $\mathbf{x}_i$ in a dataset (T) is defined as the set of points from T whose distance to $\mathbf{x}_i$ is smaller than the distance from $\mathbf{x}_i$ to $\mathbf{x}_i$'s nearest enemy (Equation (28)).

$$LS(\mathbf{x}_i) = \{\mathbf{x}_j | d(\mathbf{x}_i, \mathbf{x}_j) < d(\mathbf{x}_i, ne(\mathbf{x}_i))\}, \tag{28}$$

where $ne(\mathbf{x}_i)$ is the nearest enemy from example $\mathbf{x}_i$. Figure 11 illustrates the local set of a particular example ($\mathbf{x}$, in gray) in a dataset.

The cardinality of the LS of an example indicates its proximity to the decision boundary and also the narrowness of the gap between the classes. Therefore, the LS cardinality will be lower for examples separated from the other class with a narrow margin. According to Leyva et al. (2014), a high number of low-cardinality local sets in a dataset suggests that the space between classes

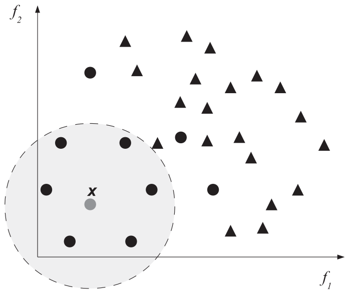


Fig. 11. Local set of an example $\mathbf{x}$ in a dataset.

ALGORITHM 1: Computing the radius of the hypersphere of an example $\mathbf{x}_i$.

```
Require: A distance matrix $D_{nxn}$, a label vector $\mathbf{y}$, a data index $i$;
$\mathbf{x}_j = ne(\mathbf{x}_i)$;
$d_i$ = distance of $\mathbf{x}_i$ to $\mathbf{x}_j$;
$\mathbf{x}_k = ne(\mathbf{x}_j)$;
if ($\mathbf{x}_i = \mathbf{x}_k$) then
    return $\frac{d_i}{2}$;
else
    $d_t$ = radius($D$, $\mathbf{y}$, $j$);
    return $d_i - d_t$;
end if
```

is narrow and irregular; that is, the boundary is more complex. The local set average cardinality measure (LSC) is calculated here as:

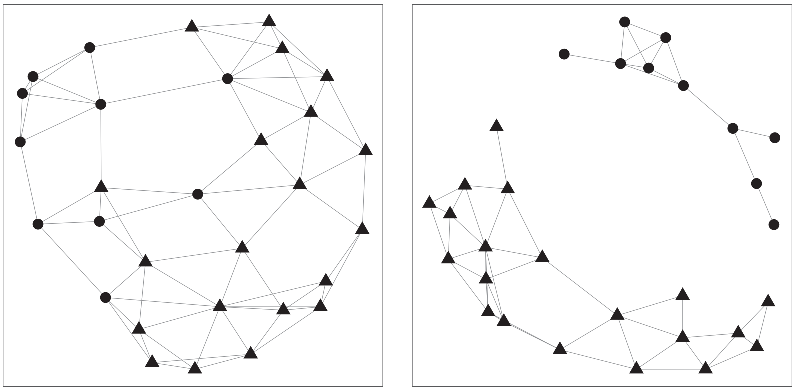
$$LSC = 1 - \frac{1}{n^2} \sum_{i=1}^{n} |LS(\mathbf{x}_i)|, \tag{29}$$

where $|LS(\mathbf{x}_i)|$ is the cardinality of the local set for example $\mathbf{x}_i$. This measure can complement N1 and L1 by also revealing the narrowness of the between-class margin. Higher values are expected for more complex datasets, in which each example is nearest to an enemy than to other examples from the same class. In that case, each example will have a local set of cardinality 1, resulting in a LSC of $1 - \frac{1}{n}$. The asymptotic cost of LSC is dominated by the computation of pairwise distances between all examples, resulting in $O(m \cdot n^2)$ operations.

2.4 Network Measures

Morais and Prati (2013) and Garcia et al. (2015) model the dataset as a graph and extract measures for the statistical characterization of complex networks (Kolaczyk 2009) from this representation. In Garcia et al. (2015) low correlation values were observed between the basic complexity measures of Ho and Basu (2002) and the graph-based measures, which supports the relevance of exploring this alternative representation of the data structure. In this article, we highlight the best measures for the data complexity induced by label noise imputation (Garcia et al. (2015)), with an emphasis on those with low correlation between each other.

To use these measures, it is necessary to represent the classification dataset as a graph. The obtained graph must preserve the similarities or distances between examples for modeling the data relationships. Each example from the dataset corresponds to a node or vertex of the graph, while undirected edges connect pairs of examples and are weighted by the distances between the



(a) Building the graph (unsupervised) (b) Pruning process (supervised)

Fig. 12. Building a graph using ϵ-NN.

examples. As in the neighborhood measures, the Gower distance is employed. Two nodes i and j are connected only if $dist(i,j) < \epsilon$. This corresponds to the ϵ-NN method for building a graph from a dataset in the attribute-value format (Zhu et al. 2005). As in Morais and Prati (2013) and Garcia et al. (2015), in ECoL the ϵ value is set to 0.15 (note that the Gower distance is normalized to the range [0,1]). Next, a post-processing step is applied to the graph, pruning edges between examples of different classes. Figure 12 illustrates the graph building process for the dataset from Figure 2. Figure 12(a) shows the first step, when the pairs of vertices with $dist(\mathbf{x}_i, \mathbf{x}_j) < \epsilon$ are connected. This first step is unsupervised, since it disregards the labels of connected points. Figure 12(b) shows the graph obtained after the pruning process is applied to disconnect examples from different classes. This step can be regarded as supervised, in which the label information is taken into account to obtain the final graph.

For a given dataset, let $G = (V, E)$ denote the graph built by this process. By construction, $|V| = n$ and $0 \leq |E| \leq \frac{n(n-1)}{2}$. Let the ith vertex of the graph be denoted as v_i and an edge between two vertices v_i and v_j be denoted as e_{ij}. The extracted measures are described next. All the measures from this category require building a graph based on the distance matrix between all pairs of elements, which requires $O(m \cdot n^2)$ operations. The asymptotic cost of all the presented measures is dominated by the computation of this matrix.

2.4.1 *Average Density of the Network (Density).* This measure considers the number of edges that are retained in the graph built from the dataset normalized by the maximum number of edges between n pairs of data points.

$$Density = 1 - \frac{2|E|}{n(n-1)} \tag{30}$$

Lower values for this measure are obtained for dense graphs, in which many examples get connected. This will be the case for datasets with dense regions from a same class in the dataset. This type of dataset can be regarded as having lower complexity. However, a low number of edges will be observed for datasets of low density (examples are far apart in the input space) and/or for which examples of opposite classes are near each other, implying a higher classification complexity.

2.4.2 *Clustering Coefficient (ClsCoef).* The clustering coefficient of a vertex v_i is given by the ratio of the number of edges between its neighbors and the maximum number of edges that could possibly exist between them. We take as a complexity measure the value:

$$ClsCoef = 1 - \frac{1}{n}\sum_{i=1}^{n} \frac{2|e_{jk} : v_j, v_k \in N_i|}{k_i(k_i - 1)}, \tag{31}$$

where $N_i = \{v_j : e_{ij} \in E\}$ denotes the neighborhood set of a vertex v_i (those nodes directly connected to v_i) and k_i is the size of N_i. The sum calculates, for each vertex v_i, the ratio of existent edges between its neighbors by the total number of edges that could possibly be formed.

The Clustering coefficient measure assesses the grouping tendency of the graph vertexes, by monitoring how close to form cliques neighborhood vertexes are. As computed by Equation (31), it will be smaller for simpler datasets, which will tend to have dense connections among examples from the same class.

2.4.3 Hub Score (Hubs). The hub score scores each node by the number of connections it has to other nodes, weighted by the number of connections these neighbors have. Herewith, highly connected vertexes that are also connected to highly connected vertices will have a larger hub score. This is a measure of the influence of each node of the graph. Here, we take the formulation:

$$Hubs = 1 - \frac{1}{n}\sum_{i=1}^{n} hub(v_i). \tag{32}$$

The values of $hub(v_i)$ are given by the principal eigenvector of A^tA, where A is the adjacency matrix of the graph. Here, we take an average value for all vertices.

In complex datasets, in which a high overlapping of the classes is observed, strong vertexes will tend to be less connected to strong neighbors. However, for simple datasets there will be dense regions within the classes and higher hub scores. Therefore, according to Equation (32), smaller Hubs values are expected for simpler datasets.

2.5 Dimensionality Measures

The measures from this category give an indicative of data sparsity. They are based on the dimensionality of the datasets, either original or reduced. The idea is that it can be more difficult to extract good models from sparse datasets, due to the probable presence of regions of low density that will be arbitrarily classified.

2.5.1 Average Number of Features Per Dimension (T2). Originally, T2 divides the number of examples in the dataset by their dimensionality (Basu and Ho 2006). In this article, we take the inverse of this formulation to obtain higher values for more complex datasets, so:

$$T2 = \frac{m}{n}. \tag{33}$$

T2 can be computed at $O(m + n)$. In some work, the logarithmic function is applied to the measure (e.g., Lorena et al. (2012)), because T2 can take arbitrarily large or small values. Though, this can take the measure into negative values when the number of examples is larger than the number of features.

T2 reflects the data sparsity. If there are many predictive attributes and few data points, they will probably be sparsely distributed in the input space. The presence of low density regions will hinder the induction of an adequate classification model. Therefore, lower T2 values indicate less sparsity and therefore simpler problems.

2.5.2 Average Number of PCA Dimensions Per Points (T3). The measure T3 (Lorena et al. 2012) is defined with a Principal Component Analysis (PCA) of the dataset. Instead of the raw dimensionality of the feature vector (as in T2), T3 uses the number of PCA components needed to represent 95% of data variability (m') as the base of data sparsity assessment. The measure is calculated as:

$$T3 = \frac{m'}{n}. \tag{34}$$

The value m' can be regarded as an estimate of the intrinsic dataset dimensionality after the correlation among features is minimized. As in the case of T2, smaller values will be obtained for simpler datasets, which will be less sparse. Since this measure requires performing a PCA analysis of the dataset, its worst cost is $O(m^2 \cdot n + m^3)$.

2.5.3 Ratio of the PCA Dimension to the Original Dimension (T4). This measure gives a rough measure of the proportion of relevant dimensions for the dataset (Lorena et al. 2012). This relevance is measured according to the PCA criterion, which seeks a transformation of the features to uncorrelated linear functions of them that are able to describe most of the data variability. T4 can be expressed by:

$$T4 = \frac{m'}{m}. \tag{35}$$

The larger the T4 value, the more of the original features are needed to describe data variability. This indicates a more complex relationship of the input variables. The asymptotic cost of the measure is $O(m^2 \cdot n + m^3)$.

2.6 Class Imbalance Measures

These measures try to capture one aspect that may largely influence the predictive performance of ML techniques when solving data classification problems: class imbalance; that is, a large difference in the number of examples per class in the training dataset. Indeed, when the differences are severe, most of the ML classification techniques tend to favor the majority class and present generalization problems.

In this section, we present some measures for capturing class imbalance. If the problem has a high imbalance in the proportion of examples per class, it can be considered more complex than a problem for which the proportions are similar.

2.6.1 Entropy of Class Proportions (C1). The C1 measure was used in Lorena et al. (2012) to capture the imbalance in a dataset. It can be expressed as:

$$C1 = -\frac{1}{\log(n_{c_i})} \sum_{i=1}^{n_c} p_{c_i} \log(p_{c_i}), \tag{36}$$

where p_{c_i} is the proportion of examples in each of the classes. This measure will achieve maximum value for balanced problems; that is, problems in which all proportions are equal. These can be considered simpler problems according to the class balance aspect. The asymptotic cost for computing this measure is $O(n)$ for obtaining the proportions of examples per class.

2.6.2 Imbalance Ratio (C2). The C2 measure is a well-known index computed for measuring class balance. Here, we adopt a version of the measure that is also suited for multiclass classification problems (Tanwani and Farooq 2010):

$$C2 = 1 - \frac{1}{IR}, \tag{37}$$

where:

$$IR = \frac{n_c - 1}{n_c} \sum_{i=1}^{n_c} \frac{n_{c_i}}{n - n_{c_i}}, \tag{38}$$

where n_{c_i} is the number of instances from the ith class. These numbers can be computed at $O(n)$ operations. Larger values of C2 are obtained for imbalanced problems. The minimum value of C2 is achieved for balanced problems, in which $n_i = n_j$ for all $i, j = 1, \ldots, n_c$.

2.7 Other Measures

This section gives an overview of some other measures that can be used to characterize the complexity of classification problems found in the related literature. Part of these measures was not described previously, because they capture similar aspects already measured by the described measures. Other measures were excluded, because they have a high computational cost.

Walt and Barnard (2007) present some variations of the T1 measure. One of them is quite similar to the LSC measure, with a difference on the normalization used by LSC. Another variation first generates an MST connecting the hypersphere centers given by T1 and then counts the number of vertexes that connect examples from different classes. There is also a measure that computes the density of the hyperspheres. We believe that the LSC measure complements T1 at a lower computational cost.

Mollineda et al. (2006) present some density measures. The first one, named D1, gives the *average number of examples per unit of volume* in the dataset. The *volume of local neighborhood* (D2) measure gives the average volume occupied by the k nearest neighbors of each example. Finally, the *class density in overlap region* (D3) determines the density of each class in the overlap regions. It counts, for each class, the number of points lying in the same region of a different class. Although these measures give an overview of data density, we believe that they do not allow to extract complementary views of the problem complexity already captured by the original neighborhood-based measures. Furthermore, they may have a higher computational cost and present an additional parameter (e.g., the k in k nearest neighbors) to be tuned.

Some of the measures found in the literature propose to analyze the dataset using a divisive approach or in multiple resolutions. Usually they show a high computational cost that can be prohibitive for datasets with a moderate number of features. Singh (2003a) reports some of such measures. Their partitioning algorithm generates hypercuboids in the space at different resolutions (with increasing numbers of intervals per feature from 0 to 31). At each resolution, the data points are assigned into cells. *Purity* measures whether the cells contain examples from a same class or from mixed classes. The *nearest neighbor separability* measure counts, for each example of a cell, the proportion of its nearest neighbors that share its class. The cell measurements are linearly weighted to obtain a single estimate and the overall measurement across all cells at a given resolution is exponentially weighted. Afterwards, the area under the curve defined by one separability measure versus the resolution defines the overall data separability. In Singh (2003b), two more measures based on the space partitioning algorithm are defined: *collective entropy*, which is the level of uncertainty accumulated at different resolutions; and *data compactness*, related to the proportion of non-empty cells at different resolutions.

In Armano and Tamponi (2016), a method named *Multi-resolution Complexity Analysis* (MRCA) is used to partition a dataset. Like in T1, hyperspheres of different amplitudes are drawn around the examples and the imbalance regarding how many examples of different classes they contain is measured. A new dataset of profile patterns is obtained, which is clustered. Afterwards, each cluster is evaluated and ranked according to a complexity metric called *Multiresolution Index* (MRI).

Armano (2015) presents how to obtain a class signature that can be used to identify, for instance, the discriminative capability of the input features. This could be regarded as a feature-based complexity measure, although more developments are necessary, since the initial studies considered binary-valued features only.

Mthembu and Marwala (2008) present a *Separability Index* SI, which takes into account the average number of examples in a dataset that have a nearest neighbor with the same label. This is quite similar to what is captured by N3, except for using more neighbors in NN classification. Another measure named *Hypothesis margin* (HM) takes the distance between the nearest neighbor of an object of the same class and a nearest enemy of another class. This largely resembles the N2 computation.

Similarly to D3, Mollineda et al. (2006) and Anwar et al. (2014) introduce a complexity measure that also focuses on local information for each example by employing the nearest neighbor algorithm. If the majority of the k nearest neighbors of an example share its label, this point can be regarded as easy to classify. Otherwise, it is a difficult point. An overall complexity measure is given by the proportion of data points classified as difficult.

Leyva et al. (2014) define some measures based on the concept of Local Sets previously described, which employ neighborhood information. Besides LSC, Leyva et al. (2014) also propose to cluster the data in the local sets and then count the number of obtained clusters. This measure is related to T1. The third measure is named *number of invasive points* (Ipoints), which uses the local sets to identify borderline instances and is related to N1, N2, and N3.

Smith et al. (2014a) propose a set of measures devoted to understand why some data points are harder to classify than others. They are called *instance hardness* measures. One advantage of such an approach is to reveal the difficulty of a problem at the instance level, rather than at the aggregate level with the entire dataset. Nonetheless, the measures can be averaged to give an estimate at the dataset level. As shown in some recent work on dynamic classifier selection (Cruz et al. 2017b, 2018), the concept of instance hardness is very useful for pointing out classifiers able to perform well in confusing or overlapping areas of the dataset, giving indicatives of a local level of competence. Most of the complexity measures previously presented, although formulated for obtaining a complexity estimate per dataset, can be adapted to assess the contribution of each example to the overall problem difficulty. Nonetheless, this is beyond the scope of this article.

One very effective instance hardness measure from Smith et al. (2014a) is the *k-Disagreeing Neighbors* (kDN), which gives the percentage of the k nearest neighbors that do not share the label of an example. This same concept was already explored in the works of Anwar et al. (2014), Mthembu and Marwala (2008), and Sotoca et al. (2005). The *Disjunct Size* (DS) corresponds to the size of a *disjunct* that covers an example divided by the largest disjunct produced, in which disjuncts are obtained using the C4.5 learning algorithm. A related measure is the *Disjunct Class Percentage* (DCP), which is the number of data points in a disjunct that belong to a same class divided by the total number of examples in the disjunct. The *Tree Depth* (TD) returns the depth of the leaf node that classifies an instance in a decision tree. The previous measures give estimates from the perspective of a decision tree classifier. In addition, the *Minority Value* (MV) index is the ratio of examples sharing the same label of an example to the number of examples in the majority class. The *Class Balance* (CB) index presents an alternative to measuring the class skew. The C1 and C2 measures previously described are simple alternatives already able to capture the class imbalance aspect.

Elizondo et al. (2012) focus their study on the relationship between linear separability and the level of complexity of classification datasets. Their method uses *Recursive Deterministic Perceptron* (RDP) models and counts the number of hyperplanes needed to transform the original problem, which may not be linearly separable, into a linearly separable problem.

In Skrypnyk (2011), various class separability measures are presented, focusing on feature selection. Some parametric measures are the *Mahalanobis* and the *Bhattacharyya* distances between the classes and the *Normal Information Radius.* These measures are computationally intensive due to the need to compute covariance matrices and their inverse. An information theoretic measure is the *Kullback-Leibler* distance. It quantifies the discrepancy between two probability distributions. Based on discriminant analysis, a number of class separability measures can also be defined. This family of techniques is closely related to measures F1v and N2, discussed in this survey.

Cummins (2013) also defines some alternative complexity measures. The first, named N5, consists of multiplying N1 by N2. According to Fornells et al. (2007), the multiplication of N1 and N2 emphasizes extreme behavior concerning class separability. Another measure (named *Case Base Complexity Profile*) retrieves the k nearest neighbors of an example $\mathbf{x}$ for increasing values of k, from 1 up to a limit K. At each round, the proportion of neighbors that have the same label as $\mathbf{x}$ is counted. The obtained values are then averaged. Although interesting, this measure can be considered quite costly to compute.

More recently, Zubek and Plewczynski (2016) presented a complexity curve based on the *Hellinger* distance of probability distributions, assuming that the input features are independent. It takes subsets of different sizes from a dataset and verifies if their information content is similar to that of the original dataset. The computed values are plotted, and the area under the obtained curve is used as an estimate of data complexity. The proposed measure is also applied in data pruning. The measure values computed turned out to be quite correlated to T2.

In the recent literature, there are also studies on generalizations of the complexity measures for other types of problems. In Lorena et al. (2018), these measures are adapted to quantify the difficulty of regression problems. Charte et al. (2016) present a complexity score for multi-label classification problems. Smith-Miles (2009) surveys some strategies for measuring the difficulty of optimization problems.

3 THE ECOL PACKAGE

Based on the review performed, we assembled a set of 22 complexity measures into an R package named ECoL (*Extended Complexity Library*), available at CRAN[3] and GitHub.[4] Table 1 summarizes the characteristics of the complexity measures included in the package. It presents the category, name, acronym, and the limit values (minimum and maximum) assumed by these measures. Taking the measure F1 as an example, according to Table 1 its higher limit is 1 (when the average values of the attributes are the same for all classes), and its lower value is approximately null. In our implementations, all measures assume values that are in bounded intervals. Moreover, for all measures, the higher the value, the greater the complexity measured. We also present the worst-case asymptotic time complexity cost for computing the measures, where n stands for the number of points in a dataset, m corresponds to its number of features, n_c is the number of classes, and l is the number of novel points generated in the case of the measures L3 and N4. All distance- and network-based measures are based on information from a distance matrix between all pairs of examples of the dataset, which can be computed only once and reused for obtaining the values of all those measures. The same reasoning applies to the linearity measures, since all of them involve training a linear SVM, from which the required information for computing the individual measures can be obtained.

Another relevant observation is that although each measure gives an indication into the complexity of the problem according to some characteristics of its learning dataset, a unified

[3] https://cran.r-project.org/package=ECoL.
[4] https://github.com/lpfgarcia/ECoL.

Table 1. Characteristics of the Complexity Measures

Category	Name	Acronym	Min	Max	Asymptotic cost
Feature-based	Maximum Fisher's discriminant ratio	F1	≈ 0	1	$O(m \cdot n)$
	Directional vector maximum Fisher's discriminant ratio	F1v	≈ 0	1	$O(m \cdot n \cdot n_c + m^3 \cdot n_c^2)$
	Volume of overlapping region	F2	0	1	$O(m \cdot n \cdot n_c)$
	Maximum individual feature efficiency	F3	0	1	$O(m \cdot n \cdot n_c)$
	Collective feature efficiency	F4	0	1	$O(m^2 \cdot n \cdot n_c)$
Linearity	Sum of the error distance by linear programming	L1	0	≈ 1	$O(n^2)$
	Error rate of linear classifier	L2	0	1	$O(n^2)$
	Non linearity of linear classifier	L3	0	1	$O(n^2 + m \cdot l \cdot n_c)$
Neighborhood	Fraction of borderline points	N1	0	1	$O(m \cdot n^2)$
	Ratio of intra/extra class NN distance	N2	0	≈ 1	$O(m \cdot n^2)$
	Error rate of NN classifier	N3	0	1	$O(m \cdot n^2)$
	Non linearity of NN classifier	N4	0	1	$O(m \cdot n^2 + m \cdot l \cdot n)$
	Fraction of hyperspheres covering data	T1	0	1	$O(m \cdot n^2)$
	Local set average cardinality	LSC	0	$1 - \frac{1}{n}$	$O(m \cdot n^2)$
Network	Density	Density	0	1	$O(m \cdot n^2)$
	Clustering Coefficient	ClsCoef	0	1	$O(m \cdot n^2)$
	Hubs	Hubs	0	1	$O(m \cdot n^2)$
Dimensionality	Average number of features per dimension	T2	≈ 0	m	$O(m + n)$
	Average number of PCA dimensions per points	T3	≈ 0	m	$O(m^2 \cdot n + m^3)$
	Ratio of the PCA dimension to the original dimension	T4	0	1	$O(m^2 \cdot n + m^3)$
Class imbalance	Entropy of classes proportions	C1	0	1	$O(n)$
	Imbalance ratio	C2	0	1	$O(n)$

interpretation of their values is not easy. Each measurement has an associated limitation (for example, the feature separability measures cannot cope with situations where an attribute has different ranges of values for the same class; see Figure 3(a)) and must then be considered only as an estimate of the problem complexity, which may have associated errors. Since the measures are made on a dataset T, they also give only an apparent measurement of the problem complexity (Ho and Basu 2002). This reinforces the need to analyze the measures together to provide more robustness to the reached conclusions. There are also cases where some caution must be taken, such as in the case of F2, whose final values depend on the number of predictive attributes in the dataset. This particular issue is pointed out by Singh (2003b), which states that the complexity measures ideally should be conceptually uncorrelated to the number of features, classes, or number of data points a dataset has, making the complexity measure values for different datasets more comparable. This requirement is clearly not fulfilled by F2. Nonetheless, for the dimensionality- and balance-based measures, Singh (2003a)'s assertion does not apply, since they are indeed concerned with the relationship of the numbers of dimensions and data points a dataset has.

For instance, a linearly separable problem with an oblique hyperplane will have high F1, indicating that it is complex, and also a low L1, denoting that it is simple. LSC, however, will assume a low value for a very imbalanced two-class dataset in which one of the classes contains one unique example and the other class is far and densely distributed. This would be an indicative of a simple classification problem according to LSC interpretation, but data imbalance should be considered, too. In the particular case of class imbalance measures, Batista et al. (2004) show that the harmful effects due to class imbalance are more pronounced when there is also a large overlap between the classes. Therefore, these measures should be analyzed together with measures able to capture class overlap (ex. C2 with N1). Regarding network-based measures, the ϵ parameter in the $\epsilon - NN$ algorithm in ECol is fixed at 0.15, although we can expect that different values may be more appropriate for distinct datasets. With the free distribution of the ECol package, interested users are able to modify this value and also other parameters (such as the distance metric employed in various measures) and test their influence in the reported results.

Finally, while some measures are based on classification models derived from the data, others use only statistics directly derived from the data. Those that use classification models, i.e., a linear classifier or an NN classifier, are: F1v, L1, L2, L3, N3, N4. This makes these measures dependent on the classifier decisions they are based on, which in turn depends on some choices in building the classifiers, such as the algorithm to derive a linear classifier, or the distance used in nearest-neighbor classification (Bernadó-Mansilla and Ho 2005). Other measures are based on characteristics extracted from the data only, although the N1 and the network indexes involve pre-computing a distance-based graph from the dataset. Moreover, it should be noticed that all measures requiring the computation of covariances or (pseudo-)inverses are time-consuming, such as F1v, T3, and T4.

Smith et al. (2014a) highlight another notice-worthy issue that some measures are unable to provide an instance-level hardness estimate. Understanding which instances are hard to classify may be valuable information, since more efforts can be devoted to them. However, many of the complexity measures originally proposed for a dataset-level analysis can be easily adapted to give instance-level hardness estimates. This is the case of N2, which averages the intra- and inter-class distances from each example to their nearest neighbors.

4 APPLICATION AREAS

The data complexity measures have been applied to support various supervised ML tasks. This section discusses some of the main applications of the complexity measures found in the related literature. They can be roughly divided into the following categories:

(1) data analysis, where the measures are used to understand the peculiarities of a particular dataset or domain;
(2) data pre-processing, where the measures are employed to guide data-preprocessing tasks;
(3) learning algorithms, where the measures are employed for understanding or in the design of ML algorithms;
(4) meta-learning, where the measures are used in the meta-analysis of classification problems, such as in choosing a particular classifier.

4.1 Data Analysis

Following the data analysis framework, some works employ the measures to better understand how the main characteristics of datasets available for learning in an application domain affect the achievable classification performance. For instance, in Lorena et al. (2012) the complexity measures are employed to analyze the characteristics of cancer gene expression data that have most impact on the predictive performance in their classification. The measures that turned out to be the most effective in such characterization were: T2 and T3 (data sparsity), C1 (class imbalance), F1 (feature-based), and N1, N2, and N3 (neighborhood-based). The complexity measure values were also monitored after a simple feature selection strategy, which revealed the importance of such pre-processing in reducing the complexity of those high-dimensional classification problems. More recently, Morán-Fernández et al. (2017a) conducted a similar study with more classification and feature selection techniques and reached similar conclusions. They also tried to answer whether classification performance could be predicted by the complexity measure values in the case of the microarray datasets. In that study, the complexity measures with highlighted results were: F1 and F3 (feature-based), N1 and N2 (neighborhood-based) when the k-nearest neighbor (kNN) classifier is used, and L1 (linearity-based) in the case of linear classifiers.

Another interesting use of the data complexity measures has been in generating artificial datasets with controlled characteristics. This resulted in some data repositories with systematic coverage for evaluating classifiers under different challenging conditions (de Melo and Lorena 2018; Macià and Bernadó-Mansilla 2014; Macià et al. 2010; Smith et al. 2014b). In Macià et al. (2010), a multi-objective Genetic Algorithm (GA) is employed to select subsets of instances of a dataset targeting at specific ranges of values of one or more complexity measures. In their experiments, one representative of each of the categories of Ho and Basu (2002)'s complexity measures was chosen to be optimized: F2, N4, and T1. Later, in Macià and Bernadó-Mansilla (2014), the same authors analyze the UCI repository. They experimentally observed that the majority of the UCI problems are easy to learn (only 3% were challenging for the classifiers tested). To increase the diversity of the repository, Macià and Bernadó-Mansilla (2014) suggest to include artificial datasets carefully designed to span the complexity space, which are produced by their multiobjective GA. This gave rise to the UCI+ repository. In de Melo and Lorena (2018), a hill-climbing algorithm is also employed to find synthetic datasets with targeted complexity measure values. Some measures devoted to evaluate the overlapping of the classes were chosen to be optimized: F1, N1, and N3. The algorithm starts with randomly produced datasets and the labels of the examples are iteratively switched seeking to reach a given complexity measure value.

4.2 Data Pre-proprocessing

The data complexity measures have also been used to guide data pre-processing tasks, such as Feature Selection (FS) (Liu et al. 2010), noise identification (Frenay and Verleysen 2014), and dealing with data imbalance (Fernández et al. 2018; He and Garcia 2009).

In FS, the measures have been used to both guide the search for the best featues in a dataset (Okimoto et al. 2017; Singh 2003b) or to understand feature selection effects (Baumgartner et al. 2006; Pranckeviciene et al. 2006; Skrypnyk 2011). For instance, Pranckeviciene et al. (2006) propose to quantify whether FS effectively changes the complexity of the original classification problem. They found that FS was able to increase class separability in the reduced spaces, as measured by N1, N2, and T1. Okimoto et al. (2017) assert the power of some complexity measures in ranking the features contained in synthetic datasets for which the relevant features are known *a priori*. As expected, feature-based measures (mainly F1) are very effective in revealing the relevant features in a dataset, although some neighborhood measures (N1 and N2) also present highlighted results. Another interesting recent work on feature selection uses a combination of the feature-based complexity measures F1, F2, and F3 to support the choice of thresholds in the number of features to be selected by FS algorithms (Seijo-Pardo et al. 2019).

Instance (or prototype) selection (IS) has also been the theme of various works involving the data complexity measures. In one of the first works in the area, Mollineda et al. (2005) tries to predict which instance selection algorithm should be applied to a new dataset. They report highlighted results of the F1 measure in identifying situations in which an IS technique is needed. Other works include: Leyva et al. (2014) and Cummins and Bridge (2011). Leyva et al. (2014), for instance, presents some complexity measures that are claimed to be specifically designed for characterizing IS problems. Among them is the LCS measure. Kim and Oommen (2009) perform a different analysis. They are interested in investigating whether the complexity measures can be calculated at reduced datasets while still preserving the characteristics found in the original datasets. Only separability-measures are considered, among them F1, F2, F3, and N2. The results were positive for all measures, except for F1.

Under his partitioning framework, Singh (2003b) discusses how potential outliers can be identified in a dataset. Other uses of the complexity measures in the noise identification context are: Garcia et al. (2013, 2015, 2016), Saéz et al. (2013), and Smith et al. (2014a). Garcia et al. (2015), for example, investigate how different label noise levels affect the values of the complexity measures. Neighborhood-based (N1, N2, and N3), feature-based (F1 and F3), and some network-based measures (density and hubs) were found to be effective in capturing the presence of label noise in classification datasets. Two of the measures most sensitive to noise imputation were then combined to develop a new noise filter, named *GraphNN*.

Gong and Huang (2012) found that the data complexity of a classification problem is more determinant in model performance than class imbalance, and that class imbalance amplifies the effects of data complexity. Vorraboot et al. (2012) adapted the back-propagation (BP) algorithm to take into account the class overlap and the imbalance ratio of a dataset using the F1 feature-based measure and the imbalance-ratio for binary problems. López et al. (2012) use the F1 measure to analyze the differences between pre-processing techniques and cost-sensitive learning for addressing imbalanced data classification. Other works in the analysis of imbalanced classification problems include Xing et al. (2013), Anwar et al. (2014), Santos et al. (2018), and Zhang et al. (2019). More discussions on the effectiveness of data complexity analysis related to the data imbalance theme can be found in Fernández et al. (2018).

4.3 Learning Algorithms

Data complexity measures can also be employed for analysis at the level of algorithms. These analyses can be for devising, tuning or understanding the behavior of different learning algorithms. For instance, Zhao et al. (2018) use the complexity measures to understand the data transformations performed by *Extreme Learning Machines* at each of their layers. They have noticed some small changes in the complexity as measured by F1, F3, and N2, which were regarded as non-significant.

A very popular use of the data complexity measures is to outline the domains of competence of one or more ML algorithms (Luengo and Herrera 2015). This type of analysis allows to identify problem characteristics for which a given technique will probably succeed or fail. While improving the understanding of the capabilities and limitations of each technique, it also supports the choice of a particular technique for solving a new problem. It is possible to reformulate a learning procedure by taking into account the complexity measures, too, or to devise new ML and pre-processing techniques.

In the analysis of the domains of competence of algorithms, one can cite: Ho (2000, 2002) for *random decision forests*; Bernadó-Mansilla and Ho (2005) for the *XCS* classifier; Ho and Bernadó-Mansilla (2006) for *NN, Linear Classifier, Decision Tree, Subspace Decision Forest,* and *Subsample Decision Forest*; Flores et al. (2014) for finding datasets that fit for a *semi-naive Bayesian Network Classifier* (BNC) and to recommend the best semi-naive BNC to use for a new dataset; Trujillo et al. (2011) for a *Genetic Programming classifier*; Ciarelli et al. (2013) for *incremental learning* algorithms; Fornells et al. (2007); Garcia-Piquer et al. (2012) for CBR; and Britto Jr et al. (2014) for the *Dynamic Selection* (DS) of classifiers.

In Luengo and Herrera (2015), a general automatic method for extracting the domains of competence of any ML classifier is proposed. This is done by monitoring the values of the data complexity measures and relating them to the difference in the training and testing accuracies of the classifiers. Rules are extracted from the measures to identify when the classifiers will achieve a good or bad accuracy performance.

The knowledge advent from the problem complexity analysis can also be used for improving the design of existent ML techniques. For instance, Smith et al. (2014a) propose a modification of the *back-propagation* algorithm for training Artificial Neural Networks (ANNs) that embed their concept of instance hardness. Therein, the error function of the BP algorithm places more emphasis on the hard instances. Other works along this line include: Vorraboot et al. (2012), also on NN, using the measures F1 and imbalance ratio; Campos et al. (2012) in DT ensembles, using the N1 and F4 measures. Recently, Brun et al. (2018) proposed a framework for dynamic classifier selection in ensembles. It uses a subset of the complexity measures for both: selecting subsets of instances to train the pool of classifiers that compose the ensemble; and to determine the predictions that will be used for a given subproblem, which will favor classifiers trained on subproblems of similar complexity to the query subproblem. They have selected one measure from each of the Ho and Basu (2002)'s original categories, which showed low Pearson correlation with each other to be optimized by a GA suited for DS: F1, N2, and N4.

However, some works have devised new approaches for data classification based on the information of the complexity measures. This is the case of Lorena and de Carvalho (2010), in which the measures F1 and F2 are used as splitting criteria for decomposing multiclass problems. Quiterio and Lorena (2018) also work on the decomposition of multiclass problems, using the complexity measures to place the binary classifiers in *Directed Acyclic Graph* structures. No specific complexity measure among those tested in the paper (namely, F1, F3, N1, N2, and T1) could be regarded as best suited for optimizing the DAG structures, although all of them were suitable choices for evaluating the binary classifiers. Sun et al. (2019) perform hierarchical partitions of the classes minimizing classification complexity, which are estimated according to the measures F1, F2, F3, N2, N3 and a new measure based on centroids introduced in their work. The best experimental results were obtained for the measures F1, F3, and centroid-based.

Another task that can be supported by the estimates on problem complexity is to tune the parameters of the ML techniques for a given problem. In He et al. (2015), the data complexity measures are applied to describe the leak quantification problem. They employ one representative measure of each of the Ho and Basu (2002)'s categories: F2, N1, and T1. In addition, a parameter-tuning procedure that minimizes data complexity under some domain-specific constraints is proposed. Measures N1 and T1 achieved better results. Nojima et al. (2011) use the complexity measures to specify the parameter values of fuzzy classifiers. Some decision rules for binary classification problems based on measures F4, L1, L2, N1, and T2 are reported. N4 is also mentioned as a key measure in the case of multiclass problems.

4.4 Meta-Learning

In Meta-learning (MtL), meta-knowledge about the solutions of previous problems is used to aid the solution of a new problem (Vilalta and Drissi 2002). For this, a meta-dataset composed of datasets for which the solutions are known is usually built. They must be described by meta-features, which is how the complexity measures are mainly used in this area. Some works previously described have made use of meta-learning, so they also fall in this category (e.g., Leyva et al. (2014), Nojima et al. (2011), Smith et al. (2014b), and Zhang et al. (2019)).

The work of Mollineda et al. (2006) is one of the first to present a general meta-learning framework based on a number of data complexity measures. Walt and Barnard (2007) employ the data complexity measures to characterize classification problems in a meta-learning setup designed to predict the expected accuracy of some ML techniques. Krijthe et al. (2012) compare classifier selection using cross-validation with meta-learning. Ren and Vale (2012) use the data complexity measures F1, F2, F3, N1, N2, T1, and T2 to predict the behavior of the NN classifier. In Garcia et al. (2016), an MtL recommender system able to predict the expected performance of noise filters in noisy data identification tasks is presented. For such, a meta-learning database is created containing meta-features, characteristics extracted from several corrupted datasets, along with the performance of some noise filters when applied to these datasets. Along with some standard meta-learning meta-features, the complexity measures N1 and N3 have a higher contribution to the prediction results. More recent works on meta-learning include: Cruz et al. (2015, 2017a, 2018), das Dôres et al. (2016), Garcia et al. (2018), Parmezan et al. (2017), Roy et al. (2016), and Shah et al. (2018). In Garcia et al. (2018), for example, all of the complexity measures described in this work were employed to generate regression models able to predict the accuracies of four classifiers with very distinct biases: ANN, decision tree, kNN, and SVM. The estimated models were effective in such predictions. The top-ranked meta-features chosen by one particular regression technique (Random Forest—RF) were N3, N1, N2, density, and T1. All of them regard neighborhood-based information from the data (in the case of density, in the form of a graph built from the data).

Another interesting usage of the complexity measures in the meta-analysis of classification problems is presented in Muñoz et al. (2018). There, an instance space is built based on meta-features extracted from a large set of classification problems, along with the performance of multiple classifiers. Among the meta-features used are the complexity measures F3, F4, L2, N1, and N4. The instance space framework provides an interesting overview of which are the hardest and easiest datasets and also to identify strengths and weaknesses of individual classifiers. The paper also presents a method to generate new datasets that better span the instance space.

4.5 Summary

A summary of the main applications of the data complexity measures found in the literature is presented in Table 2. It can be observed that these measures have been mainly employed in the characterization of the domains of competence of various learning and also pre-processing techniques by revealing when they will perform well or not. These are generalized to the use of the measures as meta-features for describing datasets in meta-learning studies.

Concerning the usage of the individual measures, we can notice a variation per domain. As expected, feature-based measures are quite effective in FS. Among them, F1 is the most used and has shown highlighted results also in instance selection and in class imbalance analysis. LSC was proposed in the IS context. Neighborhood-based measures (mainly N1, N2, and N3) also show detached results in different domains, such as FS, noise identification, and meta-learning. But one should be aware that in most of the reviewed work there was no clear evaluation on the contribution of each of the complexity measure values in the results achieved. Indeed, most of the related work perform an ad hoc selection of which complexity measures are to be used (for example, one representative measure per category). Since each measure provides a distinct perspective on classification complexity, a combination of different measures is advised. Nonetheless, whether there is a subset of the complexity measures that can be considered core to stress the difficulty of problems from different application domains is still an open issue.

5 CONCLUSION

This article reviewed the main data complexity measures from the literature. These indices allow to characterize the difficulty of a classification problem from the perspectives of data geometry and distribution within or across the classes. They were first proposed and analyzed in Ho and

Table 2. Some Work Applying the Data Complexity Measures

Category	**Sub-type**	**References**
Data Analysis	Domain understanding	García-Callejas and Araújo (2016), Kamath et al. (2008), Lorena et al. (2012) Morán-Fernández et al. (2017a)
	Data generation	Macià and Bernadó-Mansilla (2014), Macia et al. (2008), Macià et al. (2013, 2010) de Melo and Lorena (2018), Muñoz et al. (2018), Smith et al. (2014b)
Data Pre-processing	Feature Selection	Baumgartner et al. (2006), Okimoto et al. (2017), Pranckeviciene et al. (2006), Singh (2003b) Seijo-Pardo et al. (2019), Skrypnyk (2011)
	Instance Selection	Cummins and Bridge (2011), Kim and Oommen (2009), Leyva et al. (2014), Mollineda et al. (2005)
	Noise identification	Garcia et al. (2013, 2015, 2016), Saéz et al. (2013), Singh (2003b), Smith et al. (2014a)
	Class imbalance	Gong and Huang (2012), López et al. (2012), Vorraboot et al. (2012), Xing et al. (2013) Anwar et al. (2014), Santos et al. (2018), Zhang et al. (2019)
Learning algorithms	Domain of competence	Bernadó-Mansilla and Ho (2005), Flores et al. (2014), Ho and Bernadó-Mansilla (2006) Ciarelli et al. (2013), Fornells et al. (2007), Garcia-Piquer et al. (2012), Trujillo et al. (2011) Britto Jr et al. (2014), Ho (2000), Lucca et al. (2017), Luengo and Herrera (2015)
	Algorithm design	Brun et al. (2018), Campos et al. (2012), Smith et al. (2014a), Vorraboot et al. (2012)
	Algorithm understanding	Zhao et al. (2018)
	Multiclass decomposition	Lorena and de Carvalho (2010), Morán-Fernández et al. (2017b), Quiterio and Lorena (2018) Sun et al. (2019)
	Parameter tuning	He et al. (2015), Nojima et al. (2011)
Meta-learning	Meta-features	Garcia et al. (2016, 2018), Leyva et al. (2014), Nojima et al. (2011), Smith et al. (2014b) Krijthe et al. (2012), Mollineda et al. (2006), Ren and Vale (2012), Walt and Barnard (2007) Cruz et al. (2015, 2017a), das Dôres et al. (2016), Parmezan et al. (2017), Roy et al. (2016) Muñoz et al. (2018), Shah et al. (2018), Zhang et al. (2019)

Basu (2002) and have since been extensively used in the analysis and development of classification and pre-processing techniques.

The original complexity measures and other measures found in related literature were briefly presented. Despite the presence of many methods for measuring the complexity of classification problems, they often share similar concepts. There has not been a study comparing them to reveal which ones can extract more distinct aspects regarding data complexity. Besides the characteristics of each individual measure highlighted alongside their definitions, we present next some general discussions about each category of complexity measures.

In the case of the feature-based complexity measures, there is an expectation that each feature has a certain contribution to the discrimination task, and that the axis representing the feature can be interpreted as it is. This is more likely to be true for problems where the features are meaningful explanatory variables each contributing somewhat independently to the classification. It is particularly less likely to be true in classification problems where sensory signals are directly taken as input, such as pixel values in images, where a natural unit of discriminatory information tends to involve a larger group of features (such as a patch of colors displayed over multiple pixels). For those cases, transformation of the raw feature values, such as by a directional vector projection, becomes essential. The second issue is that as we examine the overlap of the feature value ranges, there is an expectation that the unseen values in an interval that spans the seen values contribute to the discrimination task in a similar way as the seen values, i.e., there is continuity in the class definition w.r.t. that feature. This tends to be true for features in a continuous numerical scale, and is less likely for other cases. For categorical features, the notion of value ranges degenerates into specific values, and several measures in this family have difficulties.

The measures in the linearity family focus on the perspective of linear separability, which has a long history of being used as a characterization of classification difficulty. It was involved in the early debates of the limits of a certain classifier's capabilities (e.g., the debate on the perceptron in Minsky and Papert (1969)). One issue of concern is that linear separability is often characteristic of sparse data sets—consider the extreme case where only one training point is available from each class in an arbitrary classification problem, and in that case linear separability of the training data does not give much information about the nature of the underlying task. Sparse datasets in high-dimensional space are also likely to be linearly separable (see, for example, Costa et al. (2009)), which motivates techniques like SVMs that use a feature transformation to map the data to a high-dimensional space where simple linear classifiers suffice. The interactive effects of this type of measure with data size, data density, and dimensionality are illustrative of the challenges involved in data complexity discussions. Therefore, the complexity evaluations need to be anchored first on fixed datasets and followed by discussions of changes in responses to the other influences.

Measures in this neighborhood-based family characterize the datasets in ways different from those of the feature-based family and the linearity-based family. They use a distance function to summarize the relationship between points. This is best fitted for datasets where the features are on a comparable scale (e.g., per-pixel intensity values) such that a natural metric exists. For datasets that involve features of heterogeneous types and scales, a scale-normalization step or a suitable weighting scheme is needed for a summarizing metric to be properly defined. The usefulness of the measures depends critically on whether such a metric can be obtained. The Gower distance metric employed in ECol is a simple alternative for dealing with features of different types and scales, but more sophisticated distance functions could be used instead (Wilson and Martinez 1997). In addition, since these measures are influenced by within-class data distributions as well as by the data distributions near the class boundaries, the information they convey may include more than what is relevant to the discrimination task, which may cause drown-out of the critical signal about classification complexity.

The network-based measures regard on the structure of the data in the input space. They may complement the previous measures presented, although they also consider the neighborhood of examples for obtaining the graph representation. It should be noticed that a number of other complex network measures can be extracted from the graph built, as well as other strategies can be used to obtain the graph representation. The strategy chosen to build the graph from a learning dataset considers both the proximity of the examples (ϵ-NN) and the data label information (pruning step). Herewith, we expect to get an overview of both intra- and inter-class relationships.

All measures from the dimensionality group rely only on the numbers of examples and features in a dataset, disregarding the label information. Therefore, they do not give any indicative of boundary complexity, but rather give a very simplified and naïve overview on data sparsity. As discussed in the article introduction, data sparsity is one of the factors that may affect the complexity of a classification problem. Indeed, datasets with a high dimensionality and a low number of examples tend to be distributed sparsely. In many cases, this can make the classification problem look simpler than it really is, so simple classification models may not generalize well to new data points that occupy regions formerly underrepresented in the training dataset.

The measures of the class imbalance category regard on the number of examples per class. As in the case of the dimensionality measures, they do not allow to directly estimate the complexity of the classification boundary. Rather, they regard on another aspect that may influence the performance of many ML classification techniques, which is the underrepresentation of one or more classes in relation to others.

This work also provides an R package with an implementation of a set of 22 complexity measures from the previous categories. The package is expected to give interested researchers a quick start into the field. An immediate line of follow-up work is to evaluate these measures empirically and try to: (i) identify those measures with most distinct concepts, since many of them have similar computation; and (ii) compare their ability in revealing the complexity of a diverse set of classification problems. This type of investigation is expected to yield a reduced subset of core measures able to capture the most critical aspects of classification complexity.

Last, the main use cases where the measures have been applied were presented. The most common use of the measures is to characterize datasets in meta-learning studies or the domain of competence of learning and pre-processing techniques. Nonetheless, more contributions remain possible in employing the conclusions of these studies to adapt and propose new learning and pre-processing techniques. For instance, relatively few works have been done in devising new learning schemes and pre-processing techniques based on the complexity measures. This points to the potential of these measures that remain poorly explored. We believe that a better understanding of the characteristics of a given problem shall be the key to support the design of techniques with better predictive results.

Another direction that awaits to be better explored is how one can use the complexity measures to evaluate different formulations of a problem, in terms of how classes are defined or chosen, in domains where there is flexibility in such choices. An example is a text categorization task, where one may have some limited freedom to choose what is to be considered a category. Better choices can lead to lower error rates even if the classifier technology stays the same. Here the data complexity measures can serve as figures of merit to evaluate alternative class definitions.

REFERENCES

Shawkat Ali and Kate A. Smith. 2006. On learning algorithm selection for classification. *Appl. Soft Comput.* 6, 2 (2006), 119–138.

Nafees Anwar, Geoff Jones, and Siva Ganesh. 2014. Measurement of data complexity for classification problems with unbalanced data. *Statist. Anal. Data Mining* 7, 3 (2014), 194–211.

Giuliano Armano. 2015. A direct measure of discriminant and characteristic capability for classifier building and assessment. *Inform. Sci.* 325 (2015), 466–483.

Giuliano Armano and Emanuele Tamponi. 2016. Experimenting multiresolution analysis for identifying regions of different classification complexity. *Pattern Anal. Appl.* 19, 1 (2016), 129–137.

Mitra Basu and Tin K. Ho. 2006. *Data Complexity in Pattern Recognition.* Springer.

Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria C. Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newslett.* 6, 1 (2004), 20–29.

Richard Baumgartner, Tin K. Ho, Ray Somorjai, Uwe Himmelreich, and Tania Sorrell. 2006. Complexity of magnetic resonance spectrum classification. In *Data Complexity in Pattern Recognition.* Springer, 241–248.

Ester Bernadó-Mansilla and Tin K. Ho. 2005. Domain of competence of XCS classifier system in complexity measurement space. *IEEE Trans. Evol. Comput.* 9, 1 (2005), 82–104.

Léon Bottou and Chih-Jen Lin. 2007. Support vector machine solvers. *Large Scale Kern. Mach.* 3, 1 (2007), 301–320.

Alceu S. Britto Jr., Robert Sabourin, and Luiz E. S. Oliveira. 2014. Dynamic selection of classifiers—A comprehensive review. *Pattern Recog.* 47, 11 (2014), 3665–3680.

André L. Brun, Alceu S. Britto Jr., Luiz S. Oliveira, Fabricio Enembreck, and Robert Sabourin. 2018. A framework for dynamic classifier selection oriented by the classification problem difficulty. *Pattern Recog.* 76 (2018), 175–190.

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Commun. Stat.-theor. Meth.* 3, 1 (1974), 1–27.

Yoisel Campos, Carlos Morell, and Francesc J. Ferri. 2012. A local complexity based combination method for decision forests trained with high-dimensional data. In *Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA'12)*. 194–199.

Francisco Charte, Antonio Rivera, María J. del Jesus, and Francisco Herrera. 2016. On the impact of dataset complexity and sampling strategy in multilabel classifiers performance. In *Proceedings of the 11th International Conference on Hybrid Artificial Intelligence Systems (HAIS'16)*. 500–511.

Patrick M. Ciarelli, Elias Oliveira, and Evandro O. T. Salles. 2013. Impact of the characteristics of data sets on incremental learning. *Artific. Intell. Res.* 2, 4 (2013), 63–74.

Ivan G. Costa, Ana C. Lorena, Liciana R. M. P. y Peres, and Marcilio C. P. de Souto. 2009. Using supervised complexity measures in the analysis of cancer gene expression data sets. In *Proceedings of the Brazilian Symposium on Bioinformatics*. 48–59.

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Rafael M. O. Cruz, Robert Sabourin, and George D. C. Cavalcanti. 2017a. META-DES.Oracle: Meta-learning and feature selection for dynamic ensemble selection. *Inform. Fus.* 38 (2017), 84–103.

Rafael M. O. Cruz, Robert Sabourin, and George D. C. Cavalcanti. 2018. Dynamic classifier selection: Recent advances and perspectives. *Inform. Fus.* 41 (2018), 195–216.

Rafael M. O. Cruz, Robert Sabourin, George D. C. Cavalcanti, and Tsang Ing Ren. 2015. META-DES: A dynamic ensemble selection framework using meta-learning. *Pattern Recog.* 48, 5 (2015), 1925–1935.

Rafael M. O. Cruz, Hiba H. Zakane, Robert Sabourin, and George D. C. Cavalcanti. 2017b. Dynamic ensemble selection VS K-NN: Why and when dynamic selection obtains higher classification performance? In *Proceedings of the 17th International Conference on Image Processing Theory, Tools and Applications (IPTA'17)*. 1–6.

Lisa Cummins. 2013. *Combining and Choosing Case Base Maintenance Algorithms*. Ph.D. Dissertation. National University of Ireland, Cork.

Lisa Cummins and Derek Bridge. 2011. On dataset complexity for case base maintenance. In *Proceedings of the 19th International Conference on Case-Based Reasoning (ICCBR'11)*. 47–61.

Silvia N. das Dôres, Luciano Alves, Duncan D. Ruiz, and Rodrigo C. Barros. 2016. A meta-learning framework for algorithm recommendation in software fault prediction. In *Proceedings of the 31st ACM Symposium on Applied Computing (SAC'16)*. 1486–1491.

Vinícius V. de Melo and Ana C. Lorena. 2018. Using complexity measures to evolve synthetic classification datasets. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'18)*. 1–8.

Ming Dong and Rishabh P. Kothari. 2003. Feature subset selection using a new definition of classificability. *Pattern Recog. Lett.* 24 (2003), 1215–1225.

David A. Elizondo, Ralph Birkenhead, Matias Gamez, Noelia Garcia, and Esteban Alfaro. 2012. Linear separability and classification complexity. *Expert Syst. Appl.* 39, 9 (2012), 7796–7807.

Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. 2018. *Learning from Imbalanced Data Sets.* Springer.

María J. Flores, José A. Gámez, and Ana M. Martínez. 2014. Domains of competence of the semi-naive Bayesian network classifiers. *Inform. Sci.* 260 (2014), 120–148.

Albert Fornells, Elisabet Golobardes, Josep M. Martorell, Josep M. Garrell, Núria Macià, and Ester Bernadó. 2007. A methodology for analyzing case retrieval from a clustered case memory. In *Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR'07)*. 122–136.

Benoit Frenay and Michel Verleysen. 2014. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 5 (2014), 845–869.

Luís P. F. Garcia, André C. P. L. F. de Carvalho, and Ana C. Lorena. 2013. Noisy data set identification. In *Proceedings of the 8th International Conference on Hybrid Artificial Intelligent Systems (HAIS'13)*. 629–638.

Luís P. F. Garcia, André C. P. L. F. de Carvalho, and Ana C. Lorena. 2015. Effect of label noise in the complexity of classification problems. *Neurocomputing* 160 (2015), 108–119.

Luís P. F. Garcia, André C. P. L. F. de Carvalho, and Ana C. Lorena. 2016. Noise detection in the meta-learning level. *Neurocomputing* 176 (2016), 14–25.

Luís P. F. Garcia, Ana C. Lorena, Marcilio C. P. de Souto, and Tin Kam Ho. 2018. Classifier recommendation using data complexity measures. In *Proceedings of the 24th International Conference on Pattern Recognition (ICPR'18)*. 874–879.

David García-Callejas and Miguel B. Araújo. 2016. The effects of model and data complexity on predictions from species distributions models. *Ecol. Modell.* 326 (2016), 4–12.

Alvaro Garcia-Piquer, Albert Fornells, Albert Orriols-Puig, Guiomar Corral, and Elisabet Golobardes. 2012. Data classification through an evolutionary approach based on multiple criteria. *Knowl. Inform. Syst.* 33, 1 (2012), 35–56.

Rongsheng Gong and Samuel H. Huang. 2012. A Kolmogorov-Smirnov statistic based segmentation approach to learning from imbalanced datasets: With application in property refinance prediction. *Expert Syst. Appl.* 39, 6 (2012), 6192–6200.

John Gower. 1971. A general coefficient of similarity and some of its properties. *Biometrics* 27, 4 (1971), 857–871.

Haibo He and Edwardo A. Garcia. 2009. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* 21, 9 (2009), 1263–1284.

Zhi-Min He, Patrick P. K. Chan, Daniel S. Yeung, Witold Pedrycz, and Wing W. Y. Ng. 2015. Quantification of side-channel information leaks based on data complexity measures for web browsing. *Int. J. Machine Learn. Cyber.* 6, 4 (2015), 607–619.

Tin K. Ho. 2000. Complexity of classification problems and comparative advantages of combined classifiers. In *Proceedings of the International Workshop on Multiple Classifier Systems (MCS'00)*. 97–106.

Tin K. Ho. 2002. A data complexity analysis of comparative advantages of decision forest constructors. *Pattern Anal. Appl.* 5 (2002), 102–112.

Tin K. Ho and Mitra Basu. 2002. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Machine Intell.* 24, 3 (2002), 289–300.

Tin K. Ho, Mitra Basu, and Martin H. C. Law. 2006. Measures of geometrical complexity in classification problems. In *Data Complexity in Pattern Recognition*. Springer, 1–23.

Tin K. Ho and Ester Bernadó-Mansilla. 2006. Classifier domains of competence in data complexity space. In *Data Complexity in Pattern Recognition*. Springer, 135–152.

Aarnoud Hoekstra and Robert P. W. Duin. 1996. On the nonlinearity of pattern classifiers. In *Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96)*, Vol. 4. 271–275.

Qinghua Hu, Witold Pedrycz, Daren Yu, and Jun Lang. 2010. Selecting discrete and continuous features based on neighborhood decision error minimization. *IEEE Trans. Syst., Man Cyber., Part B (Cyber.)* 40, 1 (2010), 137–150.

Vidya Kamath, Timothy J. Yeatman, and Steven A. Eschrich. 2008. Toward a measure of classification complexity in gene expression signatures. In *Proceedings of the 30th International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'08)*. 5704–5707.

Sang-Woon Kim and John Oommen. 2009. On using prototype reduction schemes to enhance the computation of volume-based inter-class overlap measures. *Pattern Recog.* 42, 11 (2009), 2695–2704.

Eric D. Kolaczyk. 2009. *Statistical Analysis of Network Data: Methods and Models*. Springer.

Sotiris Kotsiantis and Dimitris Kanellopoulos. 2006. Discretization techniques: A recent survey. *GESTS International Trans. Comput. Sci. Eng.* 32, 1 (2006), 47–58.

Jesse H. Krijthe, Tin K. Ho, and Marco Loog. 2012. Improving cross-validation based classifier selection using meta-learning. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR'12)*. 2873–2876.

Frank Lebourgeois and Hubert Emptoz. 1996. Pretopological approach for supervised learning. In *Proceedings of the 13th International Conference on Pattern Recognition*, Vol. 4. 256–260.

Enrique Leyva, Antonio González, and Raúl Pérez. 2014. A set of complexity measures designed for applying meta-learning to instance selection. *IEEE Trans. Knowl. Data Eng.* 27, 2 (2014), 354–367.

Li Ling and Yaser S. Abu-Mostafa. 2006. *Data Complexity in Machine Learning*. Technical Report CaltechCSTR:2006.004. California Institute of Technology.

Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao. 2010. Feature selection: An ever evolving frontier in data mining. In *Proceedings of the 4th International Workshop on Feature Selection in Data Mining (FSDM'10)*, Vol. 10. 4–13.

Victoria López, Alberto Fernández, Jose G. Moreno-Torres, and Francisco Herrera. 2012. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Syst. Appl.* 39, 7 (2012), 6585–6608.

Ana C. Lorena, Ivan G. Costa, Newton Spolaôr, and Marcilio C. P. Souto. 2012. Analysis of complexity indices for classification problems: Cancer gene expression data. *Neurocomputing* 75, 1 (2012), 33–42.

Ana C. Lorena and André C. P. L. F. de Carvalho. 2010. Building binary-tree-based multiclass classifiers using separability measures. *Neurocomputing* 73, 16–18 (2010), 2837–2845.

Ana C. Lorena, André C. P. L. F. de Carvalho, and João M. P. Gama. 2008. A review on the combination of binary classifiers in multiclass problems. *Artific. Intell. Rev.* 30, 1 (2008), 19–37.

Ana C. Lorena, Aron I. Maciel, Pericles B. C. Miranda, Ivan G. Costa, and Ricardo B. C. Prudêncio. 2018. Data complexity meta-features for regression problems. *Machine Learning* 107, 1 (2018), 209–246.

Giancarlo Lucca, Jose Sanz, Graçaliz P. Dimuro, Benjamín Bedregal, and Humberto Bustince. 2017. Analyzing the behavior of aggregation and pre-aggregation functions in fuzzy rule-based classification systems with data complexity measures. In *Proceedings of the 10th Conference of the European Society for Fuzzy Logic and Technology (IWIFSGN'17)*. 443–455.

Julián Luengo and Francisco Herrera. 2015. An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. *Knowl. Inform. Syst.* 42, 1 (2015), 147–180.

Núria Macià. 2011. *Data Complexity in Supervised Learning: A Far-reaching Implication.* Ph.D. Dissertation. La Salle, Universitat Ramon Llull.

Núria Macià and Ester Bernadó-Mansilla. 2014. Towards UCI+: A mindful repository design. *Inform. Sci.* 261 (2014), 237–262.

Núria Macia, Ester Bernadó-Mansilla, and Albert Orriols-Puig. 2008. Preliminary approach on synthetic data sets generation based on class separability measure. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR'08).* 1–4.

Núria Macià, Ester Bernadó-Mansilla, Albert Orriols-Puig, and Tin Kam Ho. 2013. Learner excellence biased by data set selection: A case for data characterisation and artificial data sets. *Pattern Recog.* 46, 3 (2013), 1054–1066.

Núria Macià, Albert Orriols-Puig, and Ester Bernadó-Mansilla. 2010. In search of targeted-complexity problems. In *Proceedings of the 12th Conference on Genetic and Evolutionary Computation.* 1055–1062.

Witold Malina. 2001. Two-parameter Fisher criterion. *IEEE Trans. Syst., Man, Cyber., Part B (Cyber.)* 31, 4 (2001), 629–636.

Li Ming and Paul Vitanyi. 1993. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer.

Marvin Minsky and Seymour Papert. 1969. *Perceptrons: An Introduction to Computational Geometry.* The MIT Press, Cambridge, MA.

Ramón A. Mollineda, José S. Sánchez, and José M. Sotoca. 2005. Data characterization for effective prototype selection. In *Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'05).* 27–34.

Ramón A. Mollineda, José S. Sánchez, and José M. Sotoca. 2006. A meta-learning framework for pattern classification by means of data complexity measures. *Intel. Artific.* 10, 29 (2006), 31–38.

Gleison Morais and Ronaldo C. Prati. 2013. Complex network measures for data set characterization. In *Proceedings of the 2nd Brazilian Conference on Intelligent Systems (BRACIS'13).* 12–18.

Laura Morán-Fernández, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. 2017a. Can classification performance be predicted by complexity measures? A study using microarray data. *Knowl. Inform. Syst.* 51, 3 (2017), 1067–1090.

Laura Morán-Fernández, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. 2017b. On the use of different base classifiers in multiclass problems. *Prog. Artific. Intell.* 6, 4 (2017), 315–323.

Linda Mthembu and Tshilidzi Marwala. 2008. A note on the separability index. Retrieved from: *Arxiv Preprint Arxiv:0812.1107* (2008).

Mario A. Muñoz, Laura Villanova, Davaatseren Baatar, and Kate Smith-Miles. 2018. Instance spaces for machine learning classification. *Machine Learn.* 107, 1 (2018), 109–147.

Yusuke Nojima, Shinya Nishikawa, and Hisao Ishibuchi. 2011. A meta-fuzzy classifier for specifying appropriate fuzzy partitions by genetic fuzzy rule selection with data complexity measures. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ'11).* 264–271.

Lucas Chesini Okimoto, Ricardo Manhães Savii, and Ana Carolina Lorena. 2017. Complexity measures effectiveness in feature selection. In *Proceedings of the 6th Brazilian Conference on Intelligent Systems (BRACIS'17).* 91–96.

Albert Orriols-Puig, Núria Macià, and Tin K. Ho. 2010. *Documentation for the Data Complexity Library in C++.* Technical Report. La Salle, Universitat Ramon Llull.

Antonio R. S. Parmezan, Huei D. Lee, and Feng C. Wu. 2017. Metalearning for choosing feature selection algorithms in data mining: Proposal of a new framework. *Expert Syst. Appl.* 75 (2017), 1–24.

Erinija Pranckeviciene, Tin K. Ho, and Ray Somorjai. 2006. Class separability in spaces reduced by feature selection. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 2. 254–257.

Thaise M. Quiterio and Ana C. Lorena. 2018. Using complexity measures to determine the structure of directed acyclic graphs in multiclass classification. *Appl. Soft Comput.* 65 (2018), 428–442.

George D. C. Cavalcantiand Tsang I. Ren and Breno A. Vale. 2012. Data complexity measures and nearest neighbor classifiers: A practical analysis for meta-learning. In *Proceedings of the 24th International Conference on Tools with Artificial Intelligence (ICTAI'12)*, Vol. 1. 1065–1069.

Anandarup Roy, Rafael M. O. Cruz, Robert Sabourin, and George D. C. Cavalcanti. 2016. Meta-learning recommendation of default size of classifier pool for META-DES. *Neurocomputing* 216 (2016), 351–362.

José A. Saéz, Julián Luengo, and Francisco Herrera. 2013. Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. *Pattern Recog.* 46, 1 (2013), 355–364.

Miriam Seoane Santos, Jastin Pompeu Soares, Pedro Henrigues Abreu, Helder Araujo, and Joao Santos. 2018. Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches. *IEEE Comput. Intell. Mag.* 13, 4 (2018), 59–76.

Borja Seijo-Pardo, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. 2019. On developing an automatic threshold applied to feature selection ensembles. *Inform. Fus.* 45 (2019), 227–245.

Rushit Shah, Varun Khemani, Michael Azarian, Michael Pecht, and Yan Su. 2018. Analyzing data complexity using metafeatures for classification algorithm selection. In *Proceedings of the Prognostics and System Health Management Conference (PHM-Chongqing'18).* 1280–1284.

Sameer Singh. 2003a. Multiresolution estimates of classification complexity. *IEEE Trans. Pattern Anal. Machine Intell.* 25, 12 (2003), 1534–1539.

Sameer Singh. 2003b. PRISM: A novel framework for pattern recognition. *Pattern Anal. Appl.* 6, 2 (2003), 134–149.

Iryna Skrypnyk. 2011. Irrelevant features, class separability, and complexity of classification problems. In *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI'11).* 998–1003.

Fred W. Smith. 1968. Pattern classifier design by linear programming. *IEEE Trans. Comput.* C-17, 4 (1968), 367–372.

Michael R. Smith, Tony Martinez, and Christophe Giraud-Carrier. 2014a. An instance level analysis of data complexity. *Machine Learn.* 95, 2 (2014), 225–256.

Michael R. Smith, Andrew White, Christophe Giraud-Carrier, and Tony Martinez. 2014b. An easy to use repository for comparing and improving machine learning algorithm usage. *Arxiv Preprint Arxiv:1405.7292* (2014).

Kate A. Smith-Miles. 2009. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Comput. Surv.* 41, 1 (2009), 1–26.

José M. Sotoca, José Sánchez, and Ramón A. Mollineda. 2005. A review of data complexity measures and their applicability to pattern classification problems. In *Actas Del III Taller Nacional de Minería de Dados y Aprendizaje (TAMIDA'05).* 77–83.

Marcilio C. P. Souto, Ana C. Lorena, Newton Spolaôr, and Ivan G. Costa. 2010. Complexity measures of supervised classification tasks: A case study for cancer gene expression data. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'10).* 1352–1358.

MengXin Sun, KunHong Liu, QingQiang Wu, QingQi Hong, BeiZhan Wang, and Haiying Zhang. 2019. A novel ECOC algorithm for multiclass microarray data classification based on data complexity analysis. *Pattern Recog.* 90 (2019), 346–362.

Ajay K. Tanwani and Muddassar Farooq. 2010. Classification potential vs. classification accuracy: A comprehensive study of evolutionary algorithms with biomedical datasets. *Learn. Class. Syst.* 6471 (2010), 127–144.

Leonardo Trujillo, Yuliana Martínez, Edgar Galván-López, and Pierrick Legrand. 2011. Predicting problem difficulty for genetic programming applied to data classification. In *Proceedings of the 13th Conference on Genetic and Evolutionary Computation (GECCO'11).* 1355–1362.

Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artif. Intell. Rev.* 18, 2 (2002), 77–95.

Piyanoot Vorraboot, Suwanna Rasmequan, Chidchanok Lursinsap, and Krisana Chinnasarn. 2012. A modified error function for imbalanced dataset classification problem. In *Proceedings of the 7th International Conference on Computing and Convergence Technology (ICCCT'12).* 854–859.

Christiaan V. D. Walt and Etienne Barnard. 2007. Measures for the characterisation of pattern-recognition data sets. In *Proceedings of the 18th Symposium of the Pattern Recognition Association of South Africa (PRASA'07).*

D. Randall Wilson and Tony R. Martinez. 1997. Improved heterogeneous distance functions. *J. Artific. Intell. Res.* 6 (1997), 1–34.

David H. Wolpert. 1996. The lack of a priori distinctions between learning algorithms. *Neural Comput.* 8, 7 (1996), 1341–1390.

Yan Xing, Hao Cai, Yanguang Cai, Ole Hejlesen, and Egon Toft. 2013. Preliminary evaluation of classification complexity measures on imbalanced data. In *Proceedings of the Chinese Intelligent Automation Conference: Intelligent Information Processing.* 189–196.

Xueying Zhang, Ruixian Li, Bo Zhang, Yunxiang Yang, Jing Guo, and Xiang Ji. 2019. An instance-based learning recommendation algorithm of imbalance handling methods. *Appl. Math. Comput.* 351 (2019), 204–218.

Xingmin Zhao, Weipeng Cao, Hongyu Zhu, Zhong Ming, and Rana Aamir Raza Ashfaq. 2018. An initial study on the rank of input matrix for extreme learning machine. *Int. J. Machine Learn. Cyber.* 9, 5 (2018), 867–879.

Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. 2005. *Semi-supervised Learning with Graphs.* Ph.D. Dissertation. Carnegie Mellon University, Language Technologies Institute, School of Computer Science.

Julian Zubek and Dariusz M. Plewczynski. 2016. Complexity curve: A graphical measure of data complexity and classifier performance. *PeerJ Comput. Sci.* 2 (2016), e76.

Received December 2017; revised February 2019; accepted July 2019