How Complex is your classification problem? A survey on measuring classification complexity

Preprint · August 2018

CITATIONS
0

READS
520

5 authors, including:

Ana Carolina Lorena
Instituto Tecnologico de Aeronautica
119 PUBLICATIONS **1,431** CITATIONS
SEE PROFILE

Luís Paulo Faina Garcia
University of Brasília
17 PUBLICATIONS **195** CITATIONS
SEE PROFILE

Jens Lehmann
University of Bonn
284 PUBLICATIONS **12,106** CITATIONS
SEE PROFILE

Marcilio Carlos Pereira de Souto
Université d'Orléans
94 PUBLICATIONS **1,161** CITATIONS
SEE PROFILE

How Complex is your classification problem? A survey on measuring classification complexity

Ana C. Lorena
Instituto Tecnológico de Aeronáutica
São José dos Campos, SP, Brazil
aclorena@ita.br

Luís P. F. Garcia
Universidade de São Paulo
São Carlos, SP, Brazil
lpgarcia@icmc.usp.br

Jens Lehmann
University of Bonn
Bonn, Germany
jens.lehmann@cs.uni-bonn.de

Marcilio C. P. Souto
University of Orléans
Orleans, France
marcilio.desouto@univ-orleans.fr

Tin K. Ho
IBM Watson
Yorktown Heights, USA
tho@us.ibm.com

August 13, 2018

ABSTRACT

Extracting characteristics from the training datasets of classification problems has proven effective in a number of meta-analyses. Among them, measures of classification complexity can 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 existent 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 challenging characteristics of the problems. This paper surveys and analyzes measures which can be extracted from the training datasets in order 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.

***K*eywords** Supervised Machine Learning, Classification, Complexity Measures

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) 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]. Furthermore, it guides the development of new data-driven pre-processing and pattern recognition techniques, as done in [Dong and Kothari, 2003, Smith et al., 2014a, Mollineda et al., 2005, Hu et al., 2010, Garcia et al., 2015]. This data-driven approach enables a better understanding of the peculiarities of a given application domain that can be explored in order 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 tose regions are classified arbitrarily. Finally, Ho and Basu [2002] focus on the complexity of the classification boundary, and presents 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 which 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 paper surveys the main complexity measures that can be estimated 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 extracted by the measures proposed in their work. The usage of the complexity measures through recent literature is also reviewed, 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 reimplemented in R, and a set of novel measures from the related literature. The added measures were chosen in order to complement the concepts assessed by the original complexity measures. Some corrections into the existent measures are also discussed and detailed in the paper. The ECol package is publicly available at `https://github.com/SmartDataAnalytics/ECoL`.

This paper is structured as follows: Section 2 presents the complexity measures. Section 3 presents some of their applications in the ML literature. Section 4 presents and analyzes the complexity measures included in the ECol package. 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 like [Mollineda et al., 2005, Ho et al., 2006, Mollineda et al., 2006, Orriols-Puig et al., 2010]. Ho and Basu [2002] divide their measures into three main groups: (i) feature overlapping measures; (ii) measures of the separability of classes; and (iii) geometry, topology and density of manifolds measures. In this paper, we adopt a more granular grouping of the measures, as follows:

1. **Feature overlapping 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 balance 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 assertion 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 be first 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.

2.1 Feature Overlapping 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 an 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.

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:

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

where r_{f_i} is a discriminant ratio for each feature f_i. That is, 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.

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]. An alternative for r_{f_i} computation which can be employed for both binary and multiclass classification problems is given in [Mollineda et al., 2005]. Here we adopt this formulation:

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

where n_{c_j} is the number of examples in class c_j, $\mu_{c_j}^{f_i}$ is the mean of feature f_i across examples of class c_j, 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, on the other hand, is non-discriminative, since its values for the two classes overlap, with the same mean and variance.

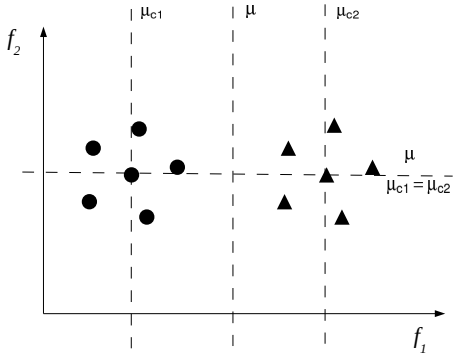


Figure 1: Example of F1 computation for a two-class dataset

The denominator in Equation 2 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. It is also similar to the Caliński and Harabasz [1974] clustering validation index. High 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. In order to deal with this issue, Orriols-Puig et al. [2010] proposes to use a F1 variant based on a *Directional Vector*, to be discussed next. Finally, Hu et al. [2010] notes 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 small value for F1.

2.1.2 The Directional-vector Maximum Fisher's Discriminant Ratio (F1v)

The **directional-vector maximum Fisher's discriminant ratio** measure (F1v) is used in Orriols-Puig et al. [2010] as a complement to F1. This measure searches for a vector which can separate the two classes after the examples have been projected into it. It implements a directional Fisher criterion defined in Malina [2001] as:

$$F1v = \frac{\mathbf{d}^t \mathbf{B} \mathbf{d}}{\mathbf{d}^t \mathbf{W} \mathbf{d}}, \tag{3}$$

where $\mathbf{d}$ is the directional vector onto which data are projected in order 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 4, 5 and 6, respectively.

$$\mathbf{d} = \mathbf{W}^{-1}(\mu_{c_1} - \mu_{c_2}), \tag{4}$$

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

$$\mathbf{B} = (\mu_{c_1} - \mu_{c_2})(\mu_{c_1} - \mu_{c_2})^t \tag{5}$$

$$\mathbf{W} = p_{c_1}\Sigma_{c_1} + p_{c_2}\Sigma_{c_2}, \tag{6}$$

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. 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 $O(n_c^2)$ 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 $O(m \cdot n \cdot n_c + m^3 \cdot n_c^2)$.

Higher values in F1v 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.

2.1.3 Volume of Overlapping Region (F2)

The **volume of the overlapping region** measure (F2) calculates the overlap of the distributions of the features values within the classes. F2 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 7.

$$F2 = \prod_i^m \frac{overlap(f_i)}{range(f_i)} = \prod_i^m \frac{\max\{0, \min\max(f_i) - \max\min(f_i)\}}{\max\max(f_i) - \min\min(f_i)}, \tag{7}$$

where:

$$\min\max(f_i) = \min(\max(f_i^{c_1}), \max(f_i^{c_2})), \tag{8}$$
$$\max\min(f_i) = \max(\min(f_i^{c_1}), \min(f_i^{c_2})), \tag{9}$$
$$\max\max(f_i) = \max(\max(f_i^{c_1}), \max(f_i^{c_2})), \tag{10}$$
$$\min\min(f_i) = \min(\min(f_i^{c_1}), \min(f_i^{c_2})). \tag{11}$$

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, respectively. The numerator becomes zero when the per-class value ranges are disjoint for at least one feature. This 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. And 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.

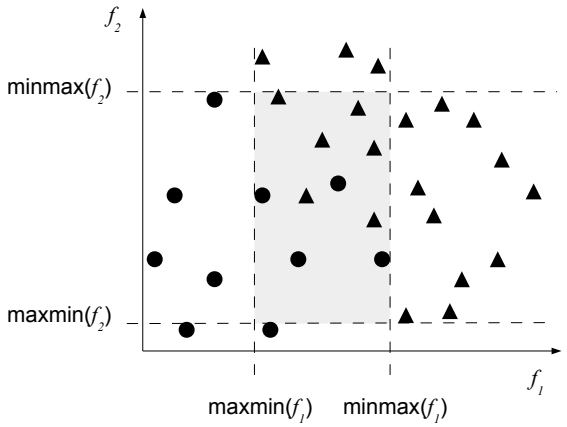
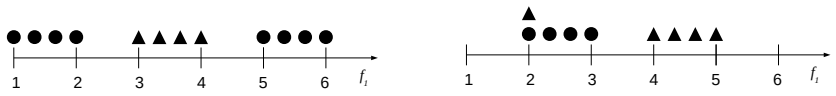


Figure 2: Example of overlapping region.

Cummins [2013] points to an issue with F2 for the cases illustrated in Figure 3. In Figure 3a, the attribute is discriminative but the minimum and maximum values overlap in the different classes; and in Figure 3b, there is one noisy example which 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.



(a) (b)

Figure 3: Problematic situations for F2.

It should be noted that the situations shown in Figure 3 can be also 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 7. That is, it is highly dependent on the number of features a dataset has. This worsens for problems with many features, so that their F2 values may not be comparable to those of other problems with few features. Souto et al. [2010] and Lorena et al. [2012] use a sum instead of the product in Equation 7, which partially solves the problems identified. Nonetheless, the result is not an overlapping volume, but 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)

The **maximum individual feature efficiency** measure (F3) estimates the individual efficiency of each feature in separating the classes, and returns the maximum value found among the m features. 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 efficiency of each feature is given by the ratio between the number of examples that are not in the overlapping region and the total number of examples:

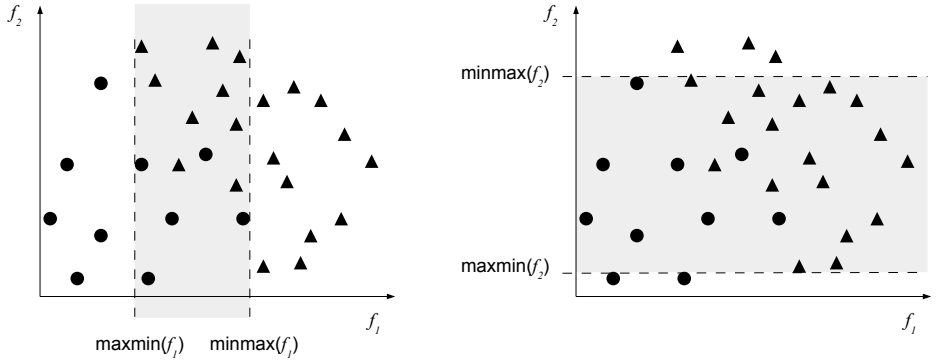
$$F3 = \max_{i=1}^{m} \frac{n - n_o(f_i)}{n}, \tag{12}$$

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 13. High values of F3 indicate simpler problems, where few examples overlap in at least one dimension.

$$n_o(f_i) = \sum_{j=1}^{n} I(x_{ji} > \max\min(f_i) \wedge x_{ji} < \min\max(f_i)) \tag{13}$$

In Equation 13, 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. As with F2, the asymptotic cost of the F3 measure is $O(m \cdot n \cdot n_c)$.

Figure 4 presents the computation of F3 for the same dataset from Figure 2. While for feature f_1 the proportion of examples that are not in the overlapping region is $\frac{16}{30}$ (Figure 4a), for f_2 this proportion is $\frac{5}{30}$ (Figure 4b), resulting in a F3 value of $\frac{16}{30}$.



(a) (b)

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

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 3a), susceptibility to noise (Figure 3b) 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 **collective feature efficiency** measure (F4) was proposed in Orriols-Puig et al. [2010] to get an overview of how all 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 is, the feature which 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 returns the ratio of examples that has been discriminated. Larger values of F4 indicate that it is possible to discriminate more examples and, therefore, that 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.

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, whilst it can get up to m in the case all features have to be considered. Its equation can be denoted by:

$$F4 = \frac{n - n_o(f_{max}(T_l))}{n} \tag{14}$$

where $n_o(f_{max}(T_l))$ measures the number of points in the overlapping region of feature f_{max} for the dataset from the l-th round (T_l). This is the current most discriminative feature in T_l. Taking the i-th iteration of F4, the most discriminative feature in dataset T_i can be found using Equation 15, adapted from F3.

$$f_{max}(T_i) = \{f_j | \max_{j=1}^{m}(n - n_o(f_j))\}_{T_i} \tag{15}$$

where $n_o(f_j)$ is computed according to Equation 13. While the dataset at each round can be defined as:

$$T_1 = T, \tag{16}$$

$$T_i = T_{i-1} - \{\mathbf{x}_j | x_{ji} < \max\min(f_{max}(T_{i-1})) \vee x_{ji} > \min\max(f_{max}(T_{i-1}))\} \tag{17}$$

That is, the dataset at the i-th round is reduced by removing all examples that are already discriminated by the previous considered feature $f_{max}(T_{i-1})$. Therefore, F4 computation is similar to that of F3, except that it can be applied to reduced datasets. Since the overlapping measure applied is similar to that used for F3, they share the same problems in some estimates (as discussed for Figures 3a and 3b). 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 4a). Figure 5a shows the resulting dataset after all examples correctly discriminated by f_1 are disregarded. Figure 5c shows the final dataset after feature f_2 has been analyzed in Figure 5b. The F4 value for this dataset is $\frac{26}{30}$.

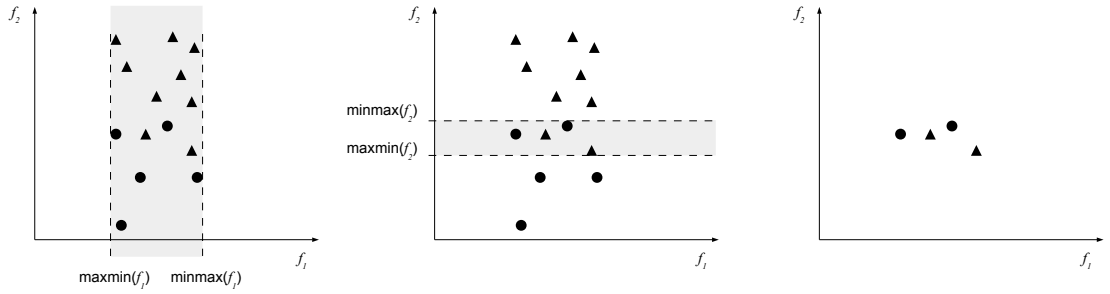


Figure 5: Calculating F4 for the dataset from Figure 2.

From the discussions above, one can see two common issues in designing a feature based complexity measure. The first is that it involves an expectation that a feature has a certain atomic contribution to the discrimination task, and that the axis representing the feature can be interpreted as 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 a certain 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 so for other cases. For categorical features, the notion of value ranges degenerates into specific values and several measures in this family have difficulties.

2.2 Measures of Linearity

These measures try to quantify to 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 which separates the examples from different classes with a maximum margin while minimizing training errors. This hyperplane is obtained by solving the following optimization problem:

$$\underset{\mathbf{w},b,\epsilon}{\text{Minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{n}\varepsilon_i\right) \tag{18}$$

$$\text{Subject to: } \begin{cases} y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \varepsilon_i, \\ \varepsilon_i \geq 0, i = 1, \ldots, n \end{cases} \tag{19}$$

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 be first decomposed.

2.2.1 Sum of the Error Distance by Linear Programming (L1)

The **sum of the error distance by linear programming** measure (L1) 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, whist it indicates the distance of the example to the linear boundary otherwise. This is expressed in Equation 20. The ε_i values are determined in the SVM optimization process.

$$L1 = \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i, \tag{20}$$

Low values for L1 indicate that the problem is close to being linearly separable, that is, simpler. Figure 6 presents an example of L1 application. After a linear boundary is obtained, the ϵ_i values of the misclassified examples (gray circles) are summed up.

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.

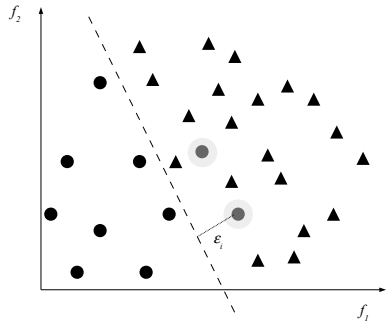


Figure 6: Example of L1 and L2 computation. The examples misclassified by the linear SVM are highlighted in gray.

2.2.2 Error Rate of Linear Classifier (L2)

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

$$L2 = \frac{\sum_{i=1}^{n} I(h(\mathbf{x}_i) \neq y_i)}{n} \tag{21}$$

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)

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

$$L3 = \frac{1}{l}\sum_{i=1}^{l} I(h_T(\mathbf{x}'_i) \neq y'_i), \tag{22}$$

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 $O(n^2 + m \cdot l \cdot n_c)$.

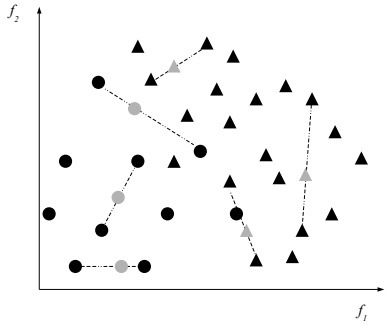


Figure 7: Example of how new points are generated in measures L3 and N4.

The several measures in this 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 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, 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 measures 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.

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 [Gower, 1971]. For symbolic features, the Gower metric computes if the compared values are equal, whilst for numerical features, a normalized difference of values is taken.

2.3.1 Fraction of Borderline Points (N1)

In the **fraction of borderline points** measure (N1), 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 that have opposite 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:

$$N1 = \frac{1}{n}\sum_{i=1}^{n} I((\mathbf{x}_i, \mathbf{x}_j) \in MST \ \wedge \ y_i \neq y_j) \tag{23}$$

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

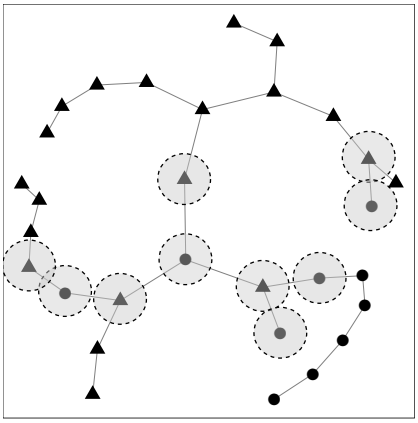


Figure 8: Example of MST generated for the dataset from Figure 2 and the detected points in the decision border.

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 ten 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. On the other hand, 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)

The **ratio of intra/extra class nearest neighbor distance** measure (N2) 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.

$$N2 = \frac{\sum_{i=1}^{n} d(\mathbf{x_i}, NN(\mathbf{x}_i) \in y_i)}{\sum_{i=1}^{n} d(\mathbf{x_i}, NN(\mathbf{x}_i) \in y_j \neq y_i)}, \tag{24}$$

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). 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 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 be also observed that N2 is related to F1 and F1v, since they all assess intra and inter class variabilities. However, unlike F1 and F1v where

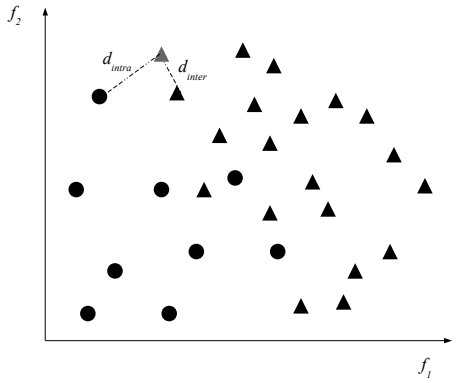


Figure 9: Example of intra and inter class distances for a particular example.

features are evaluated individually, 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 **error rate of the nearest neighbor classifier** (N3) 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)

The **non-linearity of the nearest neighbor classifier** measure (N4) 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)

The **fraction of hyperspheres covering data** measure (T1) 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:

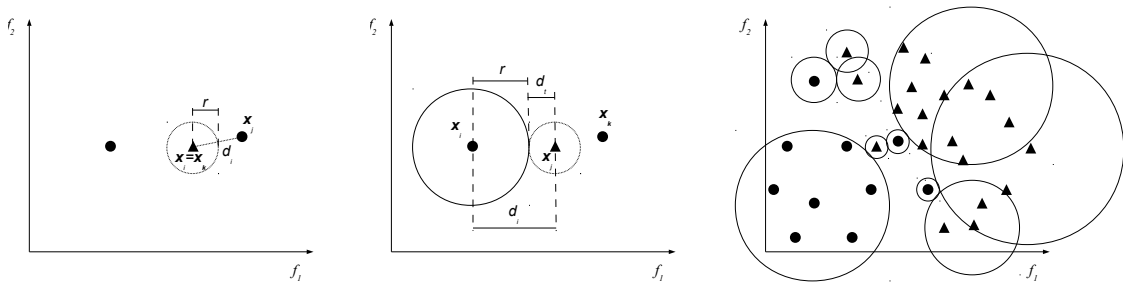
$$T1 = \frac{\sharp Hyperspheres(T)}{n} \tag{27}$$

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

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 paper 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 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 radii of their hyperspheres correspond to half of the distance between them (lines 4 and 5, also see Figure 10a). The radii of the hyperspheres around other examples can be determined recursively (lines 7 to 9), as illustrated in Figure 10b.



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

(b) Radius r obtained recursively

(c) Final hyperspheres for a dataset

Figure 10: Calculating T1 for a dataset.

Once the radii 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 10c. 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.

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;
1: $\mathbf{x}_j$ = $ne(\mathbf{x}_i)$;
2: d_i = distance of $\mathbf{x}_i$ to $\mathbf{x}_j$;
3: $\mathbf{x}_k$ = $ne(\mathbf{x}_j)$;
4: **if** ($\mathbf{x}_i = \mathbf{x}_k$) **then**
5: **return** $\frac{d_i}{2}$;
6: **else**
7: d_t = radius(D, $\mathbf{y}$, j);
8: **return** $d_i - d_t$;
9: **end if**

2.3.6 Local Set Average Cardinality (LSCAvg)

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 of low-cardinality local sets in a dataset suggests

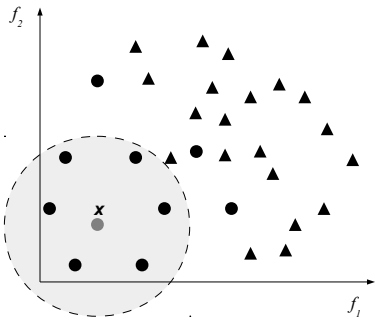


Figure 11: Local set of an example $\mathbf{x}$ in a dataset.

that the space between classes is narrow and irregular, that is, the boundary is more complex. The **local set average cardinality** measure (LSCAvg) is calculated as:

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

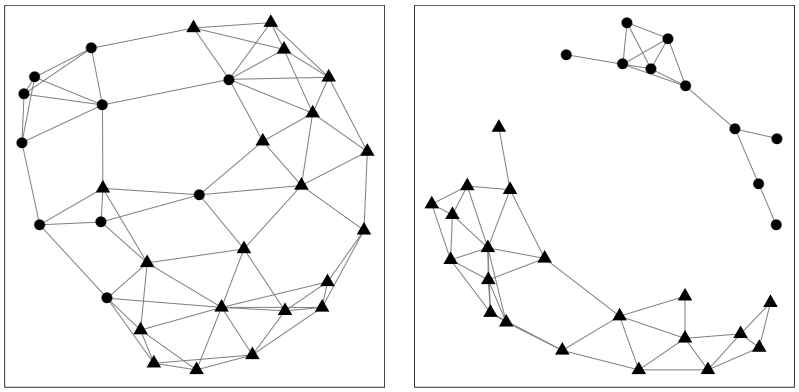
where $|LS(\mathbf{x}_i)|$ is the cardinality of the local set for example $\mathbf{x}_i$. This measure can complement N1 and L1 by also revealing the narrowness of the between-class margin. Lower values are expected for more complex datasets, in which each example is nearest to an enemy than to other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

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 sophisticate distance functions could be used instead [Wilson and Martinez, 1997]. In addition, 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.

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 the current discussion, 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, whilst undirected edges connect pairs of examples and are weighted by the distances between the examples. As in the neighborhood measures, the Gower distance is employed. Pairs of 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 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 12a 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 12b 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.



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

Figure 12: Building a graph using ϵ-NN.

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 i-th 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)

The **average density of the network** (Density) measure gives 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.

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

Higher 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. On the other hand, 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** measure (ClsCoef) 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:

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

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. It will be larger 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** measure (Hubs) 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 which are also connected to highly

connected vertexes will have a larger hub score. This is a measure of the influence of each node of the graph.

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

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 vertexes.

In complex datasets, in which a high overlapping of the classes is observed, strong vertexes will tend to be less connected to strong neighbors. On the other hand, for simple datasets there will be dense regions within the classes and higher hub scores. Therefore, larger Hubs values are expected for simpler datasets.

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 built the graph from a learning dataset considers both the proximity of the examples (ϵ-NN) and the data label information (pruning step). Throughout, we expect to get an overview of both intra and inter-class relationships.

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 points per dimension (T2)

The **average number of points per dimension** measure (T2) divides the number of examples in the dataset by their dimensionality, as follows:

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

T2 can be computed at $O(m + n)$. In some work the logarithmic function is applied to the measure (ex. 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 features is larger than the number of examples.

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

2.5.2 Average number of points per PCA dimension (T3)

The **average number of points per PCA dimension** 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:

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

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, larger values will be obtained for simpler datasets, which will be less sparse. Since this measure requires performing a PCA analysis of the dataset, its cost is $O(m^2 \cdot n + m^3)$.

2.5.3 Ratio of the PCA Dimension to the Original Dimension (T4)

The **ratio of the PCA dimension to the original dimension** measure (T4) 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 $O(m^2 \cdot n + m^3)$.

All measures from this section 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 paper 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 that simple classification models may not generalize well to new data points that occupy regions formerly underrepresented in the training dataset.

2.6 Class Balance Measures

These measures try to capture one aspect that may largely influence the predictive performance of ML techniques when solving data classification problems: class balance, that is, the differences 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 balance. If the problem has a high class 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 **entropy of class proportions** measure (C1) was used in Lorena et al. [2012] to capture the imbalance in a dataset. It can be expressed as:

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

where $p_i = n_i/n$ 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 **imbalance ratio** measure (C2) 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]:

$$C2 = \frac{n_c - 1}{n_c} \sum_{i=1}^{n_c} \frac{n_i}{n - n_i}, \tag{37}$$

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

The measures of the class balance category regard on the number of examples per class only. 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 which may influence the performance of many ML classification techniques, which is the underrepresentation of one or more classes in relation to others.

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 relate literature. Part of these measures was not formally included 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 LSCAvg measure, with a difference on the normalization used by LSCAvg. Another variation first generates an MST connecting the hyperspheres 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 LSCAvg 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).

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 which 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 LSCAvg, 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 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. The *k-Disagreeing Neighbors* (kDN) 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 Sotoca et al. [2005], Mthembu and Marwala [2008], Anwar et al. [2014]. 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 relate 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.

3 Application Areas

The data complexity measures have been applied to support of various supervised ML tasks. This section discusses some of the main applications of the complexity measures found in the relate 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.

3.1 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.

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 challenge conditions [Smith et al., 2014b, Macià and Bernadó-Mansilla, 2014]. In Macià and Bernadó-Mansilla [2014] the UCI repository is analyzed. 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. This gave rise to the UCI+ repository.

3.2 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 [He and Garcia, 2009].

In FS, the measures have been used to both guide the search for the best featues in a dataset [Singh, 2003b, Okimoto et al., 2017] 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.

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. Other works include: Leyva et al. [2014] and Cummins and Bridge [2011]. 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.

Under his partitioning framework, Singh [2003b] identifies potential outliers in a dataset. Other uses of the complexity measures in the noise identification context are: [Smith et al., 2014a, Saéz et al., 2013, Garćia et al., 2013, Garcia et al., 2015, 2016]. Garcia et al. [2015], for instance, investigate how different label noise levels affect the values of the complexity measures. The two measures most sensitive to noise imputation are 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. López et al. [2012] uses 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] and Anwar et al. [2014].

3.3 Learning Algorithms

The 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. [2016] use the complexity measures to understand the data transformations performed by *Extreme Learning Machines* at each of their layers.

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; Garcia-Piquer et al. [2012], Fornells et al. [2007] for CBR; and Britto Jr et al. [2014] for the *Dynamic Selection* (DS) of classifiers in ensembles.

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 into the *back-propagation* algorithm for training NNs which 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; Campos et al. [2012] in DT ensembles. 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.

On the other hand, 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 [2016] also work on the decomposition of multiclass problems, using the complexity measures to place the binary classifiers in *Directed Acyclic Graph* structures.

Other 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. In addition, a parameter tuning procedure which minimizes data complexity under some domain-specific

constraints is proposed. Nojima et al. [2011] use the complexity measures to specify the parameter values of fuzzy classifiers.

3.4 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., Smith et al. [2014b], Leyva et al. [2014], Nojima et al. [2011]).

The work 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 to predict the behavior of the NN classifier. In Garcia et al. [2016] an MtL recommendation 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. More recent works on meta-learning include: das Dôres et al. [2016], Roy et al. [2016], Parmezan et al. [2017].

3.5 Discussion

A summary of the main applications of the data complexity measures found in the literature is presented in Table 1. 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.

Table 1: Some work applying the data complexity measures.

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

However, relatively few works have been done in devising new learning schemes and pre-processing techniques based on the complexity measures. This highlights the potential of these measures, which still remains 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.

4 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 `https://github.com/SmartDataAnalytics/ECoL`. Table 2 summarizes the characteristics of the complexity measures included in the package. It presents the category, name, acronym ("Acron."), and the limit values (minimum and maximum) assumed by these measures. If some limit is data-dependent, that is, there is no clear limit, this is denoted by the $+\infty$ symbol. Column "Compl." presents the relation of the measures values to the complexity of the classification problem, where $\uparrow$ denotes a direct relation (higher values of the measure

imply in a higher complexity of the underlying problem) and $\downarrow$ indicates the opposite relation (a negative correlation). We also present the asymptotic time complexity cost for computing the measures ("Asymp." column).

Table 2: Characteristics of the complexity measures.

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

Taking the measure F1 as an example, according to Table 2 its lower limit is 0 (when the average values of the attributes are the same for all classes), its value has no upper bound and is dependent on the dataset. The higher the value found for F1, the lower the complexity of the problem regarding the maximum separability of the features contained in the dataset.

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 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 - Figure 3a) and must then be considered only as an estimate of the problem complexity, which may have associated errors. Since the measures are estimated from 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.

For instance, a linearly separable problem with an oblique hyperplane will have a low F1, indicating that it is complex, and also a low L1, denoting that it is simple. LSCAvg, on the other hand, will assume a high 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 LSCAvg interpretation, but data imbalance should be considered too. In the particular case of class balance 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, whist some measures are defined 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. It should be noticed too 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.

5 Conclusion

This paper 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 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 relate 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. Lastly, 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.

In the recent literature there are 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.

Finally, this work provides an R package with an implementation of a set of 22 complexity measures. 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.

Acknowledgements

The first author would like to thank the financial support of the foundations FAPESP (grant 2012/22608-8) and CNPq (grant 308858/2014-0).

References

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

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

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

Mitra Basu and Tin K Ho. *Data complexity in pattern recognition*. Springer, 2006.

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

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

Ester Bernadó-Mansilla and Tin K Ho. Domain of competence of xcs classifier system in complexity measurement space. *IEEE Transactions on Evolutionary Computation*, 9(1):82–104, 2005.

Léon Bottou and Chih-Jen Lin. Support vector machine solvers. *Large scale kernel machines*, 3(1):301–320, 2007.

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

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

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

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

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

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

Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

Lisa Cummins. *Combining and Choosing Case Base Maintenance Algorithms*. PhD thesis, National University of Ireland, Cork, 2013.

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

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

Ming Dong and Rishabh P Kothari. Feature subset selection using a new definition of classificability. *Pattern Recognition Letters*, 24:1215–1225, 2003.

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

María J Flores, José A Gámez, and Ana M Martínez. Domains of competence of the semi-naive bayesian network classifiers. *Information Sciences*, 260:120–148, 2014.

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

Benoit Frenay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845 – 869, 2014.

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

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

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

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

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

Rongsheng Gong and Samuel H Huang. A kolmogorov–smirnov statistic based segmentation approach to learning from imbalanced datasets: With application in property refinance prediction. *Expert Systems with Applications*, 39 (6):6192–6200, 2012.

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

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

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

Tin K Ho. Complexity of classification problems and comparative advantages of combined classifiers. In *Multiple Classifier Systems (MCS)*, pages 97–106. 2000.

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

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

Tin K Ho and Ester Bernadó-Mansilla. Classifier domains of competence in data complexity space. In *Data complexity in pattern recognition*, pages 135–152, 2006.

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

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

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

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

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

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

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

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

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

Li Ling and Yaser S Abu-Mostafa. Data complexity in machine learning. Technical Report CaltechCSTR:2006.004, California Institute of Technology, 2006.

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

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

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

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

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

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

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

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

Núria Macià. *Data Complexity in Supervised Learning: a far-reaching implication*. PhD thesis, La Salle, Universitat Ramon Llull, 2011.

Núria Macià and Ester Bernadó-Mansilla. Towards uci+: A mindful repository design. *Information Sciences*, 261: 237–262, 2014.

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

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

Witold Malina. Two-parameter fisher criterion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(4):629–636, 2001.

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

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

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

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

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

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

Linda Mthembu and Tshilidzi Marwala. A note on the separability index. *arXiv preprint arXiv:0812.1107*, 2008.

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

Lucas C Okimoto, Ricardo M Savii, and Ana C Lorena. Complexity measures effectiveness in feature selection. In *6th Brazilian Conference on Intelligent Systems (BRACIS)*, 2017.

Albert Orriols-Puig, Núria Macià, and Tin K Ho. Documentation for the data complexity library in c++. Technical report, La Salle - Universitat Ramon Llull, 2010.

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

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

Thaise M Quiterio and Ana C Lorena. Determining the structure of decision directed acyclic graphs for multiclass classification problems. In *5th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 115–120, 2016.

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

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

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

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

Sameer Singh. Prism: A novel framework for pattern recognition. *Pattern Analysis and Applications*, 6(2):134–149, 2003b.

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

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

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

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

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

José M Sotoca, José Sánchez, and Ramón A Mollineda. 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)*, pages 77–83, 2005.

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

Ajay K Tanwani and Muddassar Farooq. Classification potential vs. classification accuracy: a comprehensive study of evolutionary algorithms with biomedical datasets. *Learning Classifier Systems*, 6471:127–144, 2010.

Leonardo Trujillo, Yuliana Martínez, Edgar Galván-López, and Pierrick Legrand. Predicting problem difficulty for genetic programming applied to data classification. In *13th annual conference on Genetic and evolutionary computation (GECCO)*, pages 1355–1362, 2011.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18 (2):77–95, 2002.

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

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

D Randall Wilson and Tony R Martinez. Improved heterogeneous distance functions. *Journal of artificial intelligence research*, pages 1–34, 1997.

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

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

Xingmin Zhao, Weipeng Cao, Hongyu Zhu, Zhong Ming, and Rana Aamir Raza Ashfaq. An initial study on the rank of input matrix for extreme learning machine. *International Journal of Machine Learning and Cybernetics*, pages 1–13, 2016.

Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science, 2005.

Julian Zubek and Dariusz M Plewczynski. Complexity curve: a graphical measure of data complexity and classifier performance. *PeerJ Computer Science*, 2:e76, 2016.