How complex is your Classification Problem?: A Survey on Complexity Measures for Classification Problems

Ana Carolina Lorena
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo

Abstract. Extracting characteristics from the training datasets of Machine Learning (ML) algorithms has proven effective in a number of meta-analyses. One type of such characteristics is devoted to understand the intrinsic complexity of the underlying classification problem. This information can support the formulation of new preprocessing and pattern recognition techniques, in a data-driven approach. Descriptors of the spatial distribution of the data within the classes and estimates of the boundary needed to separate the classes are among the measures employed in this characterization. This report surveys measures that can characterize the complexity of a classification problem which are computed by simple indexes extracted from the learning datasets. Their main advantages and criticisms are also discussed, allowing to prospect opportunities for future work in the area.

São José dos Campos
May/2016

Contents

Chapter 1

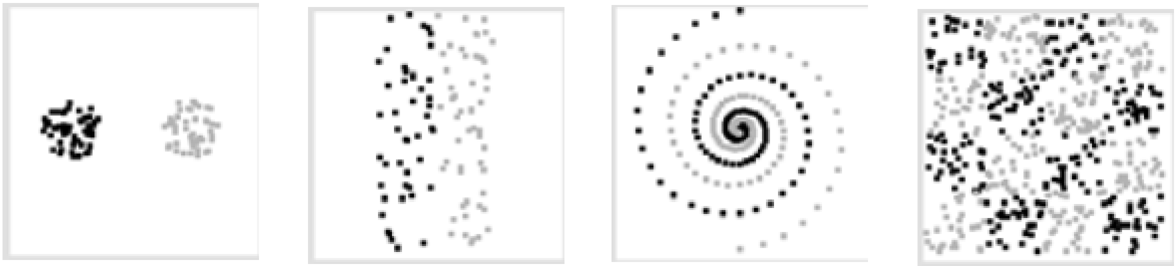
Introduction

The work from Ho and Basu (2002) was seminal in analyzing the difficulty of a classification problem by using descriptors extracted from the learning datasets. Given that, according to the *No-free lunch theorem* (Wolpert, 1996), no Machine Learning (ML) technique can consistently obtain the best performance for every classification problem, this type of analysis allows understanding when each technique succeeds and fails (Ali and Smith, 2006; Flores et al., 2014; Luengo and Herrera, 2015). Furthermore, it also supports the development of new data pre-processing and pattern recognition techniques that are data-driven, as done in (Dong and Kothari, 2003; Smith et al., 2014a; Mollineda et al., 2005; Hu et al., 2010; Garcia et al., 2015).

According to Basu and Ho (2006), the complexity of a classification problem can be attributed to three main factors: the ambiguity of the classes, the complexity of the boundary separating the classes and the sparsity and dimensionality of the data. Often there is a combination of these three factors.

The ambiguity of the classes is present in scenarios where the classes can not be distinguished using the data provided, regardless of the classification algorithm employed. This is the case of poorly defined concepts and the use of data features that are not discriminative enough. These problems are known to have non-zero Bayes error (García et al., 2009). Faced with this problem, the solution is usually to pre-process the data so that its ambiguity is resolved or to collect new data items/features. As an example of poor class definition, Basu and Ho (2006) cite distinguishing the characters 1 (one) and l (lower case L) is some specific text fonts. An example of using insufficient features would be to diagnose patients disregarding their prognostic.

The complexity of the classification boundary is related to the size of the smallest description needed to represent the classes and is native of the problem (Antolínez, 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 capable of describing the relationships between the data (Li 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, most compact descriptions can be obtained. In practice, the Kolmogorov complexity is incomputable and approximations are made based on specific learning algorithms (Gamberger and Lavrac, 1997; Li and Abu-Mostafa, 2006) or on indicators and geometric features drawn from the learning datasets (Ho and Basu, 2002; Singh, 2003a; Mollineda et al., 2005; Basu and Ho, 2006). Examples of datasets with classification boundaries of increasing complexities (from left to right) are shown in Figure 1.1 (Basu and Ho, 2006).



(a) Linear with wide margins

(b) Linear with narrow margins

(c) Spirals

(d) Checker board

Figure 1.1: Examples of classification datasets of different complexities (Basu and Ho, 2006).

Finally, an incomplete or sparse dataset also hinders a proper data analysis. This shortage makes some input space regions to be classified arbitrarily. The sparsity is frequent, for example, in the domain of gene expression data analysis (Lorena et al., 2012).

This report surveys the main complexity measures that can be estimated directly from the data available for learning. This data-driven approach enables to better understand the peculiarities of a given domain that can be exploited to get better prediction results. Their usage through recent literature is reviewed, highlighting various domains where an advantageous use of the measures can be done. The main strengths and weakness of each measure are also presented and analyzed. As a side result, this analysis provides insights into adaptations needed to some of the measures and also evidences new unexplored areas where the complexity measures can succeed.

Chapter 2

Standard 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 necessary boundary for separating binary classification problems, later extended to multiclass scenarios in (Mollineda et al., 2005; Sotoca et al., 2006; Orriols-Puig et al., 2010). These indexes can be divided into three main groups: feature overlapping measures, measures of the separability of classes and geometry, topology and density of manifolds measures.

The feature overlapping measures characterize how informative are the available features to separate the classes. In many of them each feature is evaluated individually. If there is at least a very discriminative feature in the data, the problem can be considered simpler than if no such attribute exists.

The class separability measures try to quantify how separate are the classes and to estimate the shape of the boundary between them. The intuition is that more separated classes can be distinguished more easily, requiring simpler descriptions for such. Some of them also assume that a linear separable problem can be considered simpler than a problem requiring a non-linear decision boundary.

Finally, the category of geometry, topology and density of manifolds measures compute information on how the examples are distributed within the classes. For instance, a dense class can probably be distinguished more easily than one that is sparse or has many manifolds.

For defining the measures, we consider that they are estimated from a 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 $y_i \in \{1, \ldots, C\}$. That is, each example $\mathbf{x}_i$ is described by m predictive features and has a label y_i out of C classes.

This chapter describes the original measures from Ho and Basu (2002) and possible generalizations presented in (Orriols-Puig et al., 2010), which provides a tool with C ++ implementations of all these indexes.

2.1 Feature Overlapping Measures

The feature overlapping measures from (Ho and Basu, 2002) are denoted by the letter "F" (from feature), followed by a number.

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:

$$F1 = \max_{i=1}^{m} r_{f_i}, \tag{2.1}$$

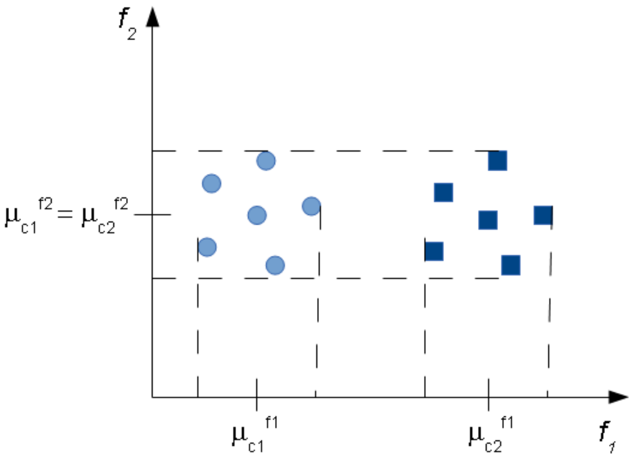
where r_{f_i} is a discriminant ratio for each feature f_i. That is, F1 takes the value of the largest discriminant ratio among all the available features. This is consistent with the definition that, if at least one feature discriminates the classes, the dataset can be considered simpler than if no such attribute exists.

Orriols-Puig et al. (2010) present different equations for calculating r_{f_i}. For binary classification problems ($C = 2$), r_{f_i} is given by:

$$r_{f_i} = \frac{(\mu_{c_1}^{f_i} - \mu_{c_2}^{f_i})^2}{(\sigma_{c_1}^{f_i})^2 + (\sigma_{c_2}^{f_i})^2}, \tag{2.2}$$

where $\mu_{c_j}^{f_i}$ and $(\sigma_{c_j}^{f_i})^2$ represent, respectively, the mean and the variance of feature f_i within class c_j. Taking, for instance, the dataset shown in Figure 2.1, 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 overlaps between the two classes, which show the same mean and variance.

The r_{f_i} values in Equation 2.2 can be calculated for numerical features only. For symbolical



(a)

Figure 2.1: Example of F1 calculation for a two-class dataset

features, Orriols-Puig et al. (2010) first map each symbolic value into an integer. Herewith, $\mu_{c_j}^{f_i}$ corresponds to the median value of feature f_i in class c_j, and $(\sigma_{c_j}^{f_i})^2$ is calculated as variance of the binomial distribution and is given by:

$$\sigma_{c_j}^{f_i} = \sqrt{p_{\mu_{c_j}^{f_i}}(1 - p_{\mu_{c_j}^{f_i}}) * n_{c_j}}, \tag{2.3}$$

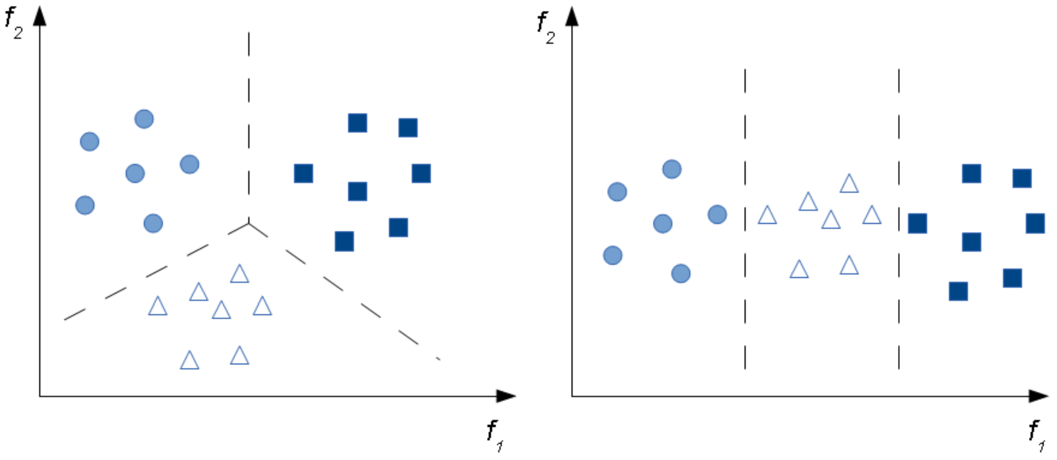
where $p_{\mu_{c_j}^{f_i}}$ is the frequency of the median $\mu_{c_j}^{f_i}$ and n_{c_j} is the number of examples in class c_j. As observed in (Cummins, 2013), mapping symbolic values into integers is viable in the case of ordinal features only. This is not always the case for all symbolic (or mixed) datasets, where nominal features can be present.

For multiclass classification problems, where $C > 2$, r_{f_i} in (Orriols-Puig et al., 2010) is calculated for each feature f_i as:

$$r_{f_i} = \frac{\sum_{c_j=1}^{C} \sum_{c_l=c_j+1}^{C} p_{c_j} p_{c_l} (\mu_{c_j}^{f_i} - \mu_{c_l}^{f_i})^2}{\sum_{c_j=1}^{k} p_{c_j} \sigma_{c_j}^2}, \tag{2.4}$$

where p_{c_j} is the proportion of examples in the class c_j. Therefore, the discriminant of the means or medians for pairs of classes is calculated and the final result is weighted by the proportion of examples from each class. The results are then added together, which will make the measure usually higher for multiclass problems than for binary problems. Due to the larger number of classes, a multiclass problem tends to be more complex than a binary problem. For instance, a binary classification problem for which a balanced learning dataset is available has a majority error rate of 0.50. Therefore, one expects to obtain classifiers that have an expected error rate lower than 0.5 using this dataset. For a balanced three-class classification problem, the majority error rate decreases to 0.33. This makes the work more difficult for the classification technique, which should now reach an error rate bellow 0.33. This is not reflected by the F1 generalization for multiclass problems presented. Therefore, some caution is required when comparing F1 values measured for binary and multiclass classification problems. Gunal and Edizkan (2008) suggest using instead, for each feature, the minimum value among all class pairs, since this would reflect the worst case scenario.

Equation 2.4 also implicitly assumes a pairwise analysis of the classes (One-Versus-One or OVO approach (Lorena et al., 2008)). In the solution of the original multiclass problem the border may not have to separate all pairs of classes, as illustrated in the example from Figure 2.2b. In the case of Figure 2.2a, on the other hand, a pairwise analysis of the classes makes more sense.



(a) (b)

Figure 2.2: Examples of multiclass classification datasets

An alternative for r_{f_i} for any classification problem (binary or multiclass) is given in (Mollineda et al., 2005):

$$r_{f_i} = \frac{\sum_{j=1}^{C} n_{c_j} (\mu_{c_j}^{f_i} - \mu^{f_i})^2}{\sum_{j=1}^{C} \sum_{l=1}^{n_{c_j}} (x_{li}^{j} - \mu_{c_j}^{f_i})^2}, \tag{2.5}$$

where n_{c_j} is the number of examples in class c_j, μ^{f_i} is the mean of the f_i values, despite of the classes, and x_{li}^{j} denotes the individual values of the feature f_i for examples from class c_j. The generalization for symbolic features is possible by replacing the mean by the median, but it is only valid for ordinal features which are first mapped into integers.

High values of the F1 measure indicate that there is at least one feature whose values overlap little among the different classes. That is, it indicates the existence of a feature for which a straight line perpendicular to its axis can separate the classes fairly (or pairs of them in the case of multiclass problems according to Equation 2.4). Ho and Basu (2002) argue that linearly separable problems can be considered simpler than classification problems requiring non-linear boundaries. Nonetheless, Figure 2.3 presents a dataset containing two features that isolately overlap between the two existent classes, although the problem is still linearly separable. The problem is that the linear border has to be oblique to the two feature axes and not perpendicular, as required by F1. Finally, Hu et al. (2010) note that the F1 measure assumes that the distributions of probabilities of the classes are (or approach) a normal, which is specially false when the classification border is irregular. Ho (2004) also point that F1 would not be able to indicate the separation in the case of two classes forming non-overlapping concentric rings one inside the other (Figure 1.1b), for example.

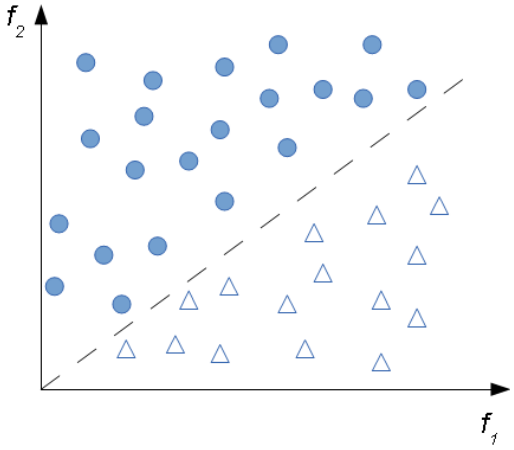


Figure 2.3: Example of oblique linear problem.

Orriols-Puig et al. (2010) propose a F1 variant called F1 Directional Vector (F1V), computed only for binary classification problems. The difference is that high values for this measurement indicate that there is a vector that can separate the two classes after the examples have been projected into it. However, the use of this measure is not common in the relate literature.

2.1.2 Volume of Overlapping Region (F2)

The volume of the overlapping region (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 length of the overlap region is then calculated, normalized by the range of the values in both classes. Finally, the obtained values are multiplied, as shown in Equation 2.6.

$$F2 = \prod_{i}^{m} \frac{overlap(f_i)}{range(f_i)} = \prod_{i}^{m} \frac{\min\max(f_i) - \max\min(f_i)}{\max\max(f_i) - \min\min(f_i)}, \tag{2.6}$$

where:

$$\min\max(f_i) = \min(\max(f_i, c_1), \max(f_i, c_2)), \tag{2.7}$$

$$\max\min(f_i) = \max(\min(f_i, c_1), \min(f_i, c_2)), \tag{2.8}$$

$$\max\max(f_i) = \max(\max(f_i, c_1), \max(f_i, c_2)), \tag{2.9}$$

$$\min\min(f_i) = \min(\min(f_i, c_1), \min(f_i, c_2)). \tag{2.10}$$

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 higher the F2 value, the greater the amount of overlap between the problem classes. Therefore, its complexity is also higher. And if there is at least one non-overlapping feature, the F2 value should be null. Figure 2.4 illustrates the region F2 tries to capture in shaded for a dataset with two features and two classes. For qualitative features, Orriols-Puig et al. (2010) suggest mapping the feature values to integer values, which is inadequate if they are not ordinal.

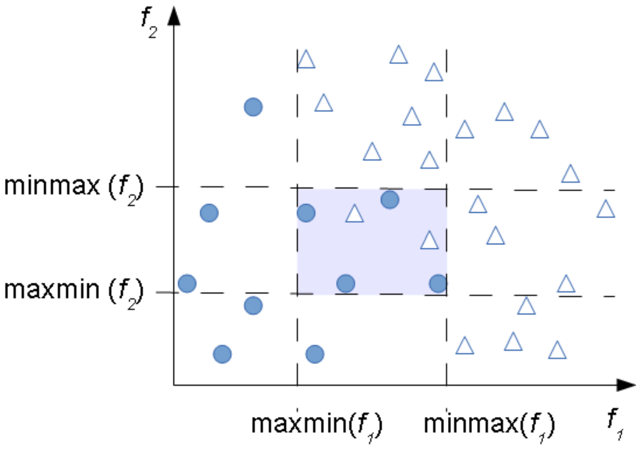


Figure 2.4: Example of overlapping region.

As noted in (Souto et al., 2010; Cummins, 2013), Equation 2.6 can result in negative values in some cases where there is no overlap among the features values. This happens, for example, in the case of the feature illustrated in Figure 2.5a, where the value F2 is $-\frac{1}{3}$, although it had to be null. Orriols-Puig et al. (2010) take the absolute value of the ratio in the product. However, the result would still be incorrect ($\frac{1}{3}$). The solution given in (Souto et al., 2010; Lorena et al., 2012) is to perform the following modification into the numerator of Equation 2.6:

$$overlap(f_i) = \max\{0, \min\max(f_i) - \max\min(f_i)\} \tag{2.11}$$

Cummins (2013) also point a problem in the case illustrated in Figure 2.5b. There is overlap at one point, but the resulting F2 value is null. This can be partially solved by adding a small amount ϵ to the numerator when it is not null, although the choice of the ϵ value would be rather arbitrary. Two other situations where Equation 2.6 results in spurious values are illustrated in: Figure 2.5c, where the attribute is discriminative but the minimum and maximum values overlap in the different classes; and Figure 2.5d, where there is a noisy example. Cummins (2013) proposes changes to deal with all these situations, by counting the number of feature values where there is overlap, which is only suitable for discrete-valued features. Instead, we could take the proportion of examples that are in the overlapping region. In the case of Figure 2.4, the resulting value would be $F2 = \frac{6}{30} = 0.2$. However, this would not be a volume measure anymore, but the proportion of examples in the overlapping region.

It should be noted that the situation shown in Figure 2.5c can be also harmful for the F1 measure, as well as the presence of many noisy cases. 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 the features' axes.

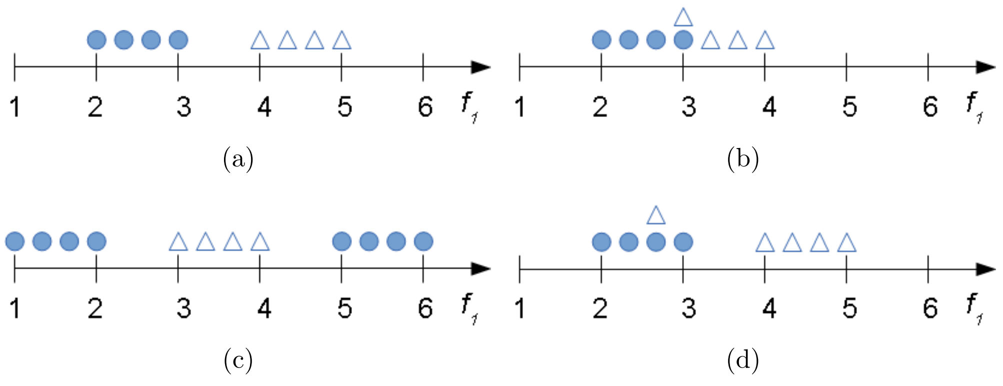


Figure 2.5: Problematic situations for F2.

Finally, the F2 value can become very small depending on the number of operands in

Equation 2.6. That is, it is highly dependent on the number of features a dataset has. This can lead into spurious situations, like that presented in the following example. Consider two two-class datasets. One of them has two features with a low normalized overlap value of 0.2, while the other dataset has 15 features, but all of them show a high region of overlap of 0.8. The final F2 values for each of these datasets are 0.04 and 0.035, respectively. Therefore, the second dataset would be considered simpler than the first regarding F2, which does not reflect the actual situation. 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); Lorena et al. (2012) use a sum instead of the product, which partially solves the problems identified. In the previous example, the new F2 values would now be of 0.4 and 12, respectively. However, the result is not an overlapping volume, but the amount or "size" of the overlapping region. In addition, the measure can still take greater values for problems that have more features when compared to problems that are more complex, but have fewer features. For instance, a 11-dimensional problem where all features show a low overlapping value of 0.1 has a resulting F2 value greater than that obtained for a one dimensional problem where the overlapping is complete (1.1 compared to 1).

For multiclass problems, F2 is calculated for each pair of classes and the sum of the values obtained for all pairs is returned (Orriols-Puig et al., 2010; Mollineda et al., 2005). Again this generalization presupposes a pairwise analysis of the classes, which may not reflect the actual difficulty of the problem, as discussed for F1. It also makes it difficult to compare F2 values between binary and multiclass problems.

2.1.3 Maximum Individual Feature Efficiency (F3)

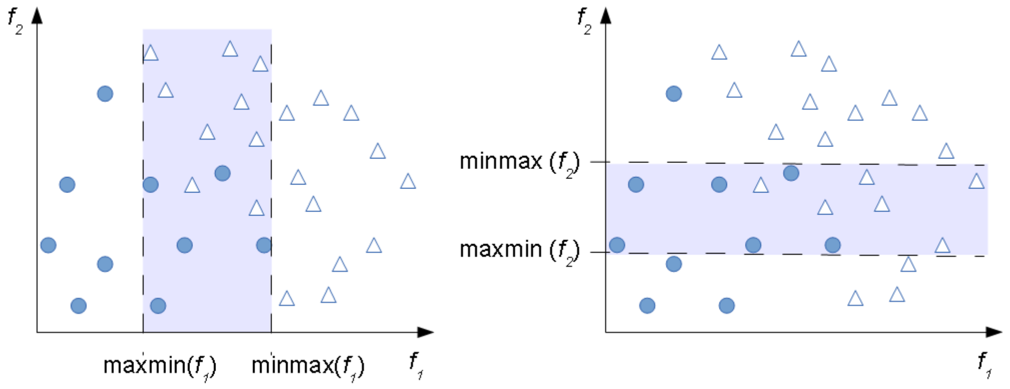
F3 estimates the individual efficiency of each feature in separating the classes, and returns the maximum value found among the m features. For each feature, it checks if there is overlap of values between examples of different classes. If there is overlap, the classes are considered to be ambiguous 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 is then given by:

$$F3 = \max_{i=1}^{m} \frac{n_o(f_i)}{n}, \tag{2.12}$$

where $n_o(f_i)$ gives the number of examples that are not in the overlapping region for feature f_i and can be expressed by Equation 2.13. High values of F3 indicate simpler problems, where few examples overlap in at least one dimension.

$$n_o(f_i) = \sum_{j=1}^{n} I(x_{ji} < \max\min(f_i) \vee x_{ji} > \min\max(f_i)) \tag{2.13}$$

In Equation 2.13, $I(a)$ is the indicator function, which returns 1 if a is true and 0 otherwise, while $\max\min(f_i)$ and $\min\max(f_i)$ are the same as defined for F2. Figure 2.6 presents the computation of F3 for the same dataset from Figure 2.4. While for feature f_1 the proportion of examples in the overlapping region is $\frac{16}{30}$ (Figure 2.6a), for f_2 this proportion is $\frac{18}{30}$ (Figure 2.6b), resulting in a F3 value of $\frac{18}{30}$. As can be seem from Equation 2.12, F3 can be applied only to two-class datasets.



(a) (b)

Figure 2.6: Calculating F3 for the dataset from Figure 2.4.

Since the way $n_o(f_i)$ 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 2.5c), susceptibility to noise (Figure 2.5d) and the fact that it is assumed that in linearly separable problems, the boundary is perpendicular to an axis, which is not always true (Figure 2.3).

2.1.4 Collective Feature Efficiency (F4)

This measure was proposed in (Orriols-Puig et al., 2010) to get an overview of how all the features work together, in opposition to the previous indexes. 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 are removed from the dataset and the above 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 was discriminated. Larger values of F4 indicate that it is possible to discriminate more examples and, therefore, the problem is simpler. The idea is to get the number of examples that can be correctly classified if hyperplanes perpendicular to the axes of the features are used in their separation. Therefore, it has the same drawbacks of F1, F2 and F3 as not to admit oblique hyperplanes. Its equation is denoted by:

$$F4 = \frac{\sum_{i=1}^{m} n_o(f_i)_{T_i}}{n}, \tag{2.14}$$

where $n_o(f)_{T_i}$ measures the number of points out of the overlapping region of feature f_i for the dataset T_i. At each round, f_i is the current most discriminative feature in T_i according to Equation 2.15, adapted from F3.

$$f_i = \{f_j | \max_{j=1}^{m} n_o(f_j)\}_{T_i} \tag{2.15}$$

The recursion expression for T_i is:

$$T_1 = T, \tag{2.16}$$

$$T_i = T_{i-1} - \{\mathbf{x}_j | x_{ji} < \max\min(f_{i-1}) \vee x_{ji} > \min\max(f_{i-1})\} \tag{2.17}$$

T_i is continuously reduced by removing all examples that are already discriminated by the previous considered feature f_{i-1}. Therefore, F4 computation is similar to that of F3, except that it is applied to increasingly reduced datasets. Since the overlapping measure applied is similar from that used for F3, they share the same problems in some estimates (as discussed for Figures 2.5c and 2.5d).

Figure 2.7 shows the F4 operation for the dataset from Figure 2.4. Feature f_2 is the most discriminative in the first round. Figure 2.7a shows the resulting dataset after all examples correctly discriminated by f_2 are disregarded. Figure 2.7b shows the final dataset after feature

f_1 has been analyzed. The F4 value for this dataset is $\frac{25}{30}$.

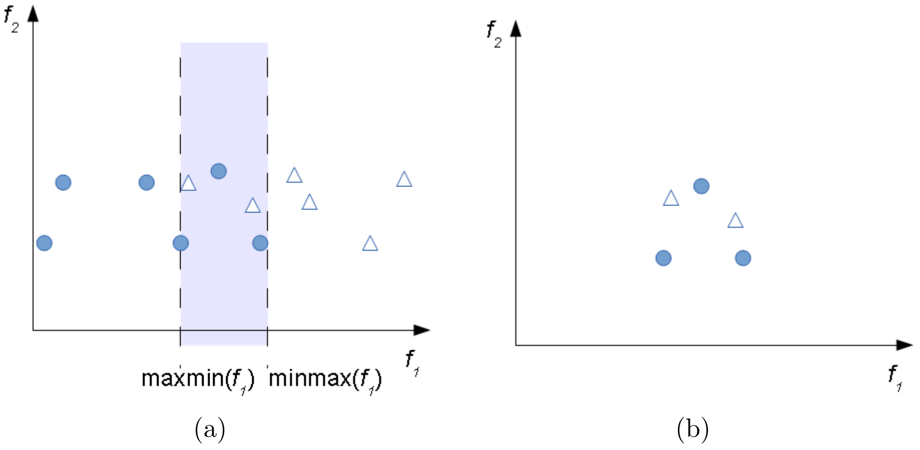


Figure 2.7: Calculating F4 for the dataset from Figure 2.4.

2.2 Measures of Class Separability

The first measures of this category check whether a problem is linearly separable. In this case, it can then be considered simpler than a problem in which a non-linear boundary is required. These measures are initiated by the letter "L". The next measures are based on the distances between nearest neighbors, and are initiated by the letter "N".

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

This measure assesses if the data are linearly separable, that is, if it is possible to separate the classes by an hyperplane. L1 computes, for a dataset, the sum of the error distances between the predictions given by a linear classifier and the actual labels of the examples. 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. For obtaining the linear classifier, Ho and Basu (2002) suggest to solve the following optimization problem proposed by Smith (1968):

$$\text{Minimize } \mathbf{1} \cdot \mathbf{e} \tag{2.18}$$

$$\text{Subject to: } \begin{cases} Z^t\mathbf{w} + \mathbf{e} \geq \mathbf{1} \\ \mathbf{e} \geq 0 \end{cases} \tag{2.19}$$

In the presented formulation, $\mathbf{1}$ are identity vectors (composed by ones), $\mathbf{w}$ is a weight vector, $\mathbf{e}$ is an error vector and Z is a data matrix. The columns $\mathbf{z}_i$ from Z correspond to the input vectors $\mathbf{x}_i$ (with one added dimension of value 1), weighted according to their class y_i according to Equation 2.20.

$$\begin{cases} \mathbf{z}_i = +\mathbf{x}_i, & \text{if } y_i = c_1 \\ \mathbf{z}_i = -\mathbf{x}_i, & \text{if } y_i = c_2 \end{cases} \tag{2.20}$$

Therefore, the linear classifier that best separates the data, admitting some errors (modeled by $\mathbf{e}$), is found. L1 is given by the value of the objective function, as in Equation 2.21.

$$L1 = \mathbf{1} \cdot \mathbf{e}^*, \tag{2.21}$$

where $\mathbf{e}^*$ is obtained by solving the optimization problem previously outlined.

Orriols-Puig et al. (2010) propose that the linear classifier is a Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000) with a linear Kernel. In this case the hyperplane sought 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:

$$\underset{\mathbf{w},b,\mathbf{e}}{\text{Minimize}} \ \frac{1}{2} \|\mathbf{w}\|^2 + c \left(\sum_{i=1}^{n} e_i \right) \tag{2.22}$$

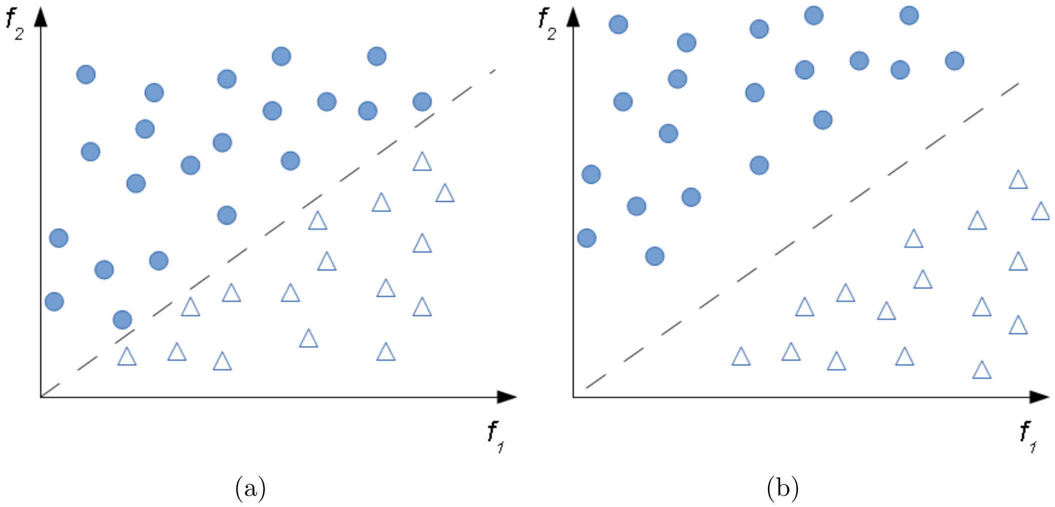
$$\text{Subject to:} \begin{cases} y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \geq 1 - e_i, \\ \mathbf{e} \geq 0, \end{cases} \tag{2.23}$$

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 $\mathbf{e}$. This is quite similar to what was performed in the previous optimization problem, but here the structural error of the model is also taken into account by the margin maximization principle. 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. The c value used in (Orriols-Puig et al., 2010) is 0.05, although nothing is discussed about why this particular value was chosen. For obtaining L1, Orriols-Puig et al. (2010) compute:

$$L1 = \frac{\sum_{i=1}^{n} |(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - y_i|}{n}, \tag{2.24}$$

In Equation 2.24, $\mathbf{w}^*$ and b^* are found by solving the SVM optimization problem. This equation sums the difference between the raw predictions obtained for the training examples to their actual label, that is, it gives a rough estimate of how distant the examples are from the correct side of the decision hyperplane. It must be pointed that this is not equivalent to summing the error vector as done for the original L1 measure. For obtaining a similar measure, the **e** values should be employed instead.

Considering the original definition of Ho and Basu (2002), low values for L1 indicate that the problem is linearly separable, that is, simpler. It is not possible, however, to check if a linearly separable problem is simpler than other also linearly separable. Figure 2.8 presents two linearly separable datasets. In both of the cases, the L1 measure will be null. Nonetheless, the dataset from Figure 2.8b can be considered simpler than the one shown in Figure 2.8a, where data from the different classes are more distant from each other. The same happens for the datasets in Figures 1.1a and 1.1b. This could be measured by taking the margin of the closest examples (Support Vectors) to the separating hyperplane when SVMs are used, but would lead to a new measure of different interpretation.



(a) (b)

Figure 2.8: Examples of linearly separable datasets of different complexities.

L1 can also be estimated for binary datasets and problems with numerical attributes only. For multiple classes Orriols-Puig et al. (2010) suggest to first decompose the problem into multiple binary sub-problems, and compute an average L1 measure. Different strategies can be used in this decomposition, such as One-Versus-All (OVA) and OVO (Lorena et al., 2008). And symbolic attributes must first be converted into numeric values according to some strategy (such as using a bit for each symbolic value in the case of nominal attributes). Another disadvantage of L1 is its cost. In fact, it involves the induction of a linear classifier and can not be estimated at a lower cost.

2.2.2 Error Rate of Linear Classifier (L2)

L2 measures the error rate of the same linear classifier described previously. Let $h(\mathbf{x})$ denote the linear classifier obtained. L2 is then given by:

$$L2 = \frac{\sum_{i=1}^{n} I(h(\mathbf{x}_i) \neq y_i)}{n}, \tag{2.25}$$

where $I(a)$ will be one whenever the argument a is true and it will be null otherwise. Higher L2 values denote more errors and therefore a greater complexity (the data cannot be separated linearly). L2 disadvantages are similar to those of L1.

2.2.3 Fraction of Borderline Points (N1)

For the calculation of N1, first a Minimum Spanning Tree (MST) is built from data as illustrated in Figure 2.9. Herewith, each vertex corresponds to an example and the edges are weighted according to the distance between them. N1 calculates the percentage of vertices connecting examples of opposite classes in the generated MST. These examples are in the border or in overlapping areas between the classes. They can also be noisy examples surrounded by examples from another class. N1 then estimates the size and complexity of the required decision border 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 frontiers to separate the classes and/or that there is a lot amount of overlap between the classes. N1 can be expressed as:

$$N1 = \frac{\sum_{i=1}^{n} I(\mathbf{x}_j \in 1NN(\mathbf{x}_i) \ \wedge \ y_i \neq y_j)}{n}, \tag{2.26}$$

where the condition in the numerator holds for examples $\mathbf{x}_i$ whose nearest neighbors (1NN) $\mathbf{x}_j$ are from another class.

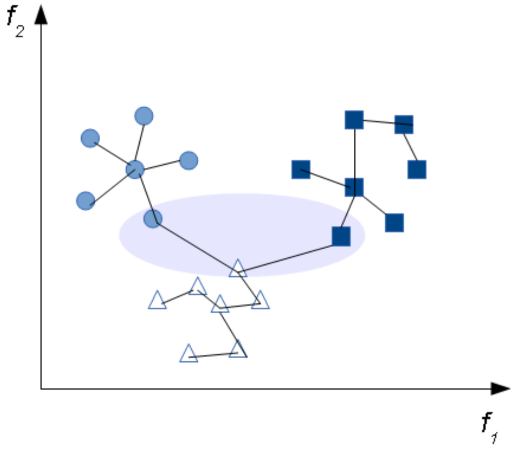


Figure 2.9: Example of MST and points in the border.

For calculating the distances between the examples, typically the Euclidean distance is employed. However, it is only applicable to numeric attributes. It is advisable to employ an heterogeneous distance measure as the Euclidean-overlap (Hu et al., 2010). Data must also be normalized, so that attributes at higher numerical scales do not dominate the distance value improperly. In fact, this normalization is required for all measures which consider distances between examples.

N1 is also sensitive to noise because the closest neighbors of noisy examples will usually have a different class of their own. Since a noisy dataset can be considered more complex than a clear counterpart, this can be interesting and N1 will correctly reflect this fact. This was observed experimentally in (Lorena et al., 2012; Garcia et al., 2015), for instance.

Another issue discussed in (Cummins, 2013) is that there can be multiple MSTs valid for the same set of points. In particular, if examples farthest from the separation boundary are examined before borderline examples, the N1 value will be lower than if the reverse order is used. Cummins (2013) propose to generate 10 MSTs by presenting the data points in different orderings and reporting an average N1 value. Basu and Ho (2006) also point that the N1 value can be large even for a linearly separable problem. This happens when the distance between borderline examples are smaller that the distance between examples from the same class, as shown in Figure 2.10. On the other hand, Ho (2002) point that a problem with a complicated nonlinear class boundary can still have few edges among examples from different classes as long as the data points are compact within each class.

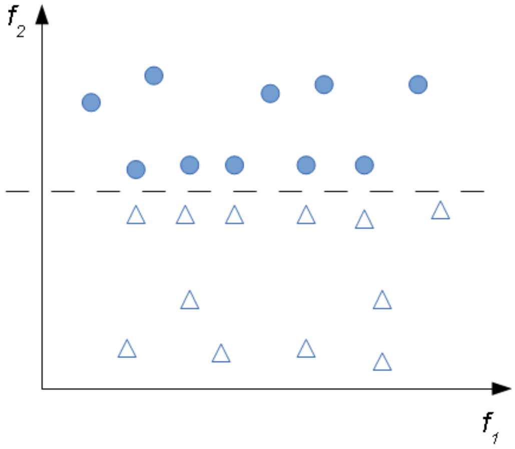


Figure 2.10: Example of linearly separable dataset with large N1 value.

2.2.4 Ratio of Intra/Extra Class Nearest Neighbor Distance (N2)

N2 averages the ratio of the distances between each example and its closest neighbor from the same class (intra-class) and from another class (extra-class), as shown in Equation 2.27.

$$N2 = \sum_{i=1}^{n} \frac{d_{intra}(\mathbf{x_i}, NN(\mathbf{x}_i))}{d_{inter}(\mathbf{x_i}, NN(\mathbf{x}_i))}, \tag{2.27}$$

where $d_{intra}(\mathbf{x_i}, NN(\mathbf{x}_i))$ corresponds to the distance of example $\mathbf{x}_i$ to its nearest neighbor from its class and $d_{inter}(\mathbf{x_i}, NN(\mathbf{x}_i))$ represents the distance of $\mathbf{x}_i$ to the closest neighbor from another class. Figure 2.11 illustrates the intra and inter distances for a particular example. The distance measure should be heterogeneous to make possible using N2 for both symbolic and numeric features.

Low N2 values are indicatives of simpler problems, in which the distance between examples of different classes exceeds the distance between examples from the same class. Therefore, N2 reflects how data are distributed within classes and not only how is the boundary between the classes. It can also be sensitive to noise in the data, as 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 (as in the case of Figure 1.1b).

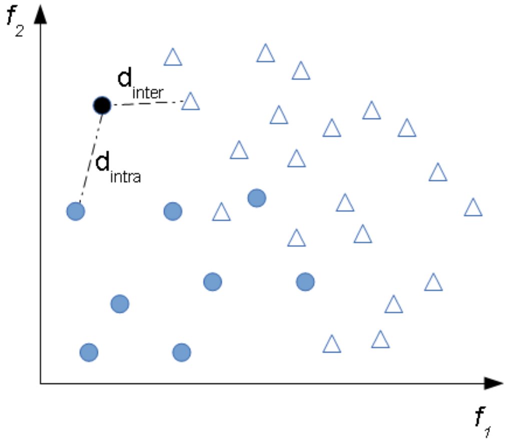


Figure 2.11: Example of intra and inter class distances for a particular example.

2.2.5 Error Rate of the Nearest Neighbor Classifier (N3)

N3 returns the error rate of a 1NN (Nearest Neighbor) classifier using leave-one-out. The following equation provides this measure:

$$N3 = \frac{\sum_{i=1}^{n} I(1NN(\mathbf{x}_i) \neq y_i)}{n}, \tag{2.28}$$

where $1NN(\mathbf{x}_i)$ represents the nearest neighbor prediction for example $\mathbf{x}_i$. For instance, using the Euclidean distance, the 1NN prediction for the black circle in Figure 2.12 would be "triangle", which accounts for an error in N3 computation. High N3 values indicate that many examples are close to examples of other classes, making the problem more complex. This measure is costly, as it consults a NN classifier n times.

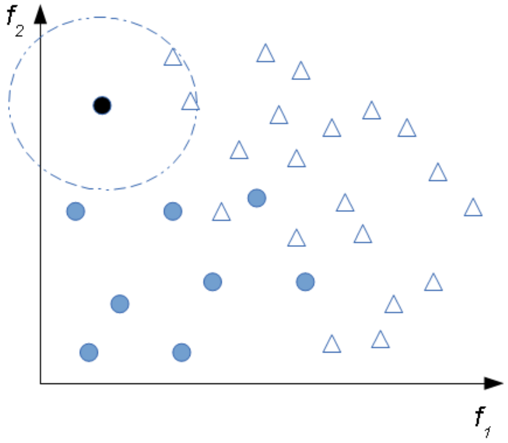


Figure 2.12: Example of 1NN prediction for a particular example.

2.3 Measures of Geometry, Topology and Density of Manifolds

This category contains measures that to try to estimate the density and distribution (topology) of data within different classes. They assume that each class is made up of one or mode manifolds and try to characterize their shape and overlap (Cummins, 2013).

2.3.1 Non Linearity of Linear Classifier (L3)

This measure is based on a method proposed by Hoekstra and Duin (1996). It first creates a new dataset by interpolating 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 2.13 illustrates the generation of six new examples (in black) 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. In (Orriols-Puig et al., 2010) a linear SVM is used. 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 $h(\mathbf{x})$ denote the linear classifier induced from the training data T, the L3 measure is given by:

$$L3 = \frac{\sum_{i=1}^{l} I(h(\mathbf{x}'_i) \neq y'_i)}{l}, \tag{2.29}$$

where l is the number of interpolated examples $\mathbf{x}'_i$ and their corresponding classes is denoted by y'_i. It is unclear how many points must be generated. In (Orriols-Puig et al., 2010) $l = 2n$.

Since L3 uses a linear classifier induced by a SVM or by solving the optimization problem illustrated in L1 description, it can only be applied to binary classification problems. Another issue is that, since a linear interpolation of the attribute values is made, L3 is only applicable to numeric attributes.

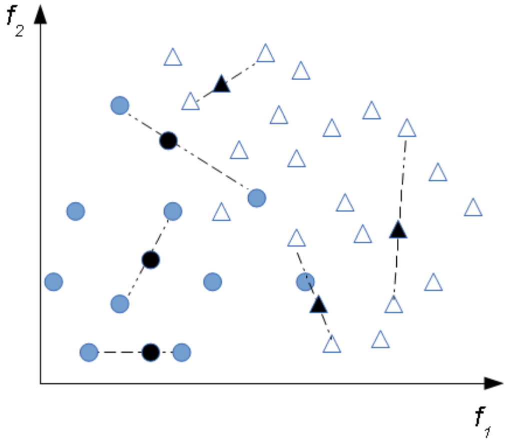


Figure 2.13: Example of how new points are generated in measures L3 and N4.

2.3.2 Non Linearity of the Nearest Neighbor Classifier (N4)

N4 is similar to L3, but uses the 1NN classifier instead of the linear predictor. It can be expressed as:

$$L3 = \frac{\sum_{i=1}^{l} I(1NN(\mathbf{x}_i) \neq y_i)}{l}, \tag{2.30}$$

Higher N4 values are indicative of problems of greater complexity. A heterogeneous distance must be employed to compute the distance between examples. In contrast to L3, N4 can be applied directly to multiclass classification problems, without the need to decompose them into binary subproblems.

2.3.3 Fraction of Hyperspheres Covering Data (T1)

This measure builds hyperspheres centered at each one of the examples. Their radius is increased until they reach an example of another class. Smaller hyperspheres contained in larger hyperspheres are eliminated. T1 then returns the ratio between the number of these hyperspheres and the total number of examples in the dataset:

$$T1 = \frac{\sharp Hyperspheres(T)}{n} \tag{2.31}$$

where $\sharp Hyperspheres(T)$ gives the number of hyperspheres that can be formed in the dataset. Figure 2.14 illustrates the hyperspheres formed in a given dataset. For building the hyperspheres, the adherence subset concept defined by Equation 2.32 is used (Leyva et al., 2015).

$$ad(T) = \{\}, \text{ if } T = \{\} \tag{2.32}$$

$$\mathbf{x} \cup \Gamma(\mathbf{x}), \text{ if } T = \{\mathbf{x}\} \tag{2.33}$$

$$\cup_{\mathbf{x} \in T}\, ad(T), \text{ if } \sharp T > 1 \tag{2.34}$$

$$\tag{2.35}$$

where:

$$\Gamma(\mathbf{x}_i) = \{\mathbf{x}_j | dist(\mathbf{x}_i, \mathbf{x}_j) < \epsilon\} \tag{2.36}$$

Ho and Basu (2002) recommend using $\epsilon = 0.55\delta$, where δ is the distance between the two closest points from different classes. It is possible to compose these subsets, obtaining adherence subsets of different orders:

$$ad^0(X) = X, \tag{2.37}$$

$$ad^1(X) = ad(X), \tag{2.38}$$

$$\dots, \tag{2.39}$$

$$ad^n(X) = ad(ad^{n-1}(X)) \tag{2.40}$$

Therefore, for each example from the training dataset T an adherence subset of maximum order is built such that it includes only examples from the same class. Subsets included in other ones are discarded. Each resulting subset corresponds to a hypersphere.

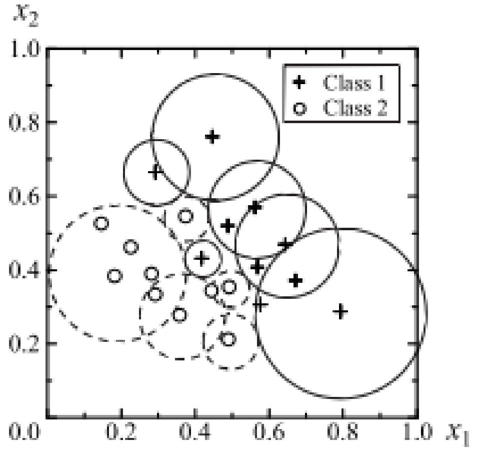


Figure 2.14: Example of hyperspheres formed in a dataset (Nojima et al., 2011).

Fewer hyperspheres are obtained for simpler datasets. This reflects the fact that 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 on the border.

2.3.4 Average Number of Examples per Dimension (T2)

T2 divides the number of examples in the dataset by their dimensionality (number of predictive attributes), as follows:

$$T2 = \frac{n}{m} \tag{2.41}$$

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.

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.

2.4 Discussion

Table 2.1 summarizes the characteristics of the complexity measures from (Ho and Basu, 2002; Orriols-Puig et al., 2010). It presents to which type of problem the measures can be applied (binary - Bin - and/or multiclass - Multi), the type of attributes they are are able to deal with (numeric and/or symbolic) and the limit values 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 "Complexity" 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). This table was built partially based on information reported in (Garcia et al., 2015). Some observations are noted with superscript indices, which should be interpreted as follows:

1. Although the measure has been generalized for multiclass problems, it still needs adaptations. It should also be noted that the multiclass problem can be decomposed into

multiple binary subproblems and the measure can then be averaged between them.

2. Although the measure has been generalized to symbolic attributes, it is only valid if they are ordinal.

3. The measure can be used for symbolic attributes, if they are appropriately mapped into numerical values previously.

4. An heterogeneous distance function must be used to enable the application of the measure to both numeric and symbolic values simultaneously.

5. The measure performs a linear interpolation of the data, which is only applicable to numeric attributes.

Table 2.1: Characteristics of the complexity measures.

Category	**Measure**	**Problem**	**Feature**	**Minimum**	**Maximum**	**Complexity**
Feature Overlapping	F1	Bin/Multi[1]	Num/Simb[2]	0	$+\infty$	↓
	F2	Bin/Multi[1]	Num/Simb[2]	0	$+\infty$	↑
	F3	Bin	Num/Simb[2]	0	1	↓
	F4	Bin	Num/Simb[2]	0	1	↓
Separability of Classes	L1	Bin	Num/Simb[3]	0	$+\infty$	↑
	L2	Bin	Num/Simb[3]	0	1	↑
	N1	Bin/Multi	Num/Simb[4]	0	1	↑
	N2	Bin/Multi	Num/Simb[4]	0	$+\infty$	↑
	N3	Bin/Multi	Num/Simb[4]	0	1	↑
Geometry, Topology, Density	L3	Bin	Num[5]	0	1	↑
	N4	Bin/Multi	Num[5]	0	1	↑
	T1	Bin/Multi	Num/Simb[4]	0	1	↑
	T2	Bin/Multi	Num/Simb	≈ 0	$+\infty$	↓

Taking the measure F1 as an example, according to Table 2.1 it can be applied to binary and multiclass problems (although there are restrictions on its formulation in this case), to datasets with both numeric and ordinals features, its lower limit is 0 (when the mean/median values of the attributes are the same for all classes), its upper value is dependent on the dataset, and 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 indicative into the complexity of the problem according to some characteristics of its learning dataset, a single interpretation of their values is not indicated. For example, 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. Furthermore, 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 2.5c) and must then be considered an estimate of the problem complexity, which may have associated errors. Since they are estimated from a dataset T, they also give 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 adaptations should be made, such as in the case of F2, whose final values dependent on the number of predictive attributes in the dataset.

Finally, while some measures need to induce classification models, others only use statistics derived from the data. Those that use classification models, either linear or NN, are: L1, L2, L3, N2, N3, N4. This makes these measures dependent on the classifiers decisions they are based in (Mansilla and Ho, 2005). The other measures are based on characteristics extracted from data only, although the N1 index involves pre-assembling a graph from the dataset.

Chapter 3

Other Measures

This chapter gives an overview of some other measures that can be used to characterize the complexity of classification problems found in the relate literature. For instance, in a previous paper Ho and Basu (2000) had already introduced their data complexity measures. In that paper they also defined another measure which iteratively employes a supervised k-means clustering procedure to the dataset. At each round, the centroids of each class are calculated and all points lying close to the centroid of their own class than that of another class are removed. The centroids for the remaining points are updated and the previous procedure is repeated until either no more points can be removed or one of the classes becomes empty. The number of iterations required for a dataset is output as a complexity measure. Ho and Basu (2000) report this measure is "sensitive to the difference in variances of the classes and to the amount of overlap of the convex hulls of the classes". It is unclear why this measure was disregarded in (Ho and Basu, 2002).

3.1 Simple Generalizations

Some work has performed some simple generalizations into the measures of Ho and Basu (2002). It is the case of (Van Der Walt and Barnard, 2007), which use the complexity measures as meta-features in a meta-learning setup designed to predict the accuracy performance of some ML techniques. In particular, they present some variations of the T1 measure. The first one generates a MST connecting the hyperspheres centers and count the number of vertices that connect examples from different classes. Another measure is the average size of the retained subsets in T1. There is also a measure that computes the density of the hyperspheres. A mutual information between classes and features is used to obtain an intrinsic dimensionality of the datasets too, which gives origin to another measure of data sparseness.

3.2 Density Measures

Sotoca et al. (2006) present another general meta-learning framework based on some data complexity measures. Besides those measures from (Ho and Basu, 2002), they also include new density measures. The first one, named D1, gives the average number of examples per unit of volume. This volume is given by the product of the lengths of all feature ranges within the classes. 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. If the majority of the k neighbors of an example belong to another class, this point is considered to be in an overlap region.

3.3 Partitioning Strategies

Some of the measures found in the literature propose to analyze the dataset using a divisive approach or in multiple resolutions. Singh (2003a) reports some of such measures. Their partitioning algorithm generates hypercuboids in the space, at different resolutions (increasingly numbers of partitions 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. If a cell is composed of data from a single class, then it is totally pure. It is totally impure if it contains mixed classes, with the same number of examples per class. 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 for obtaining a single estimate, where the weight of a cell is proportional to the number of elements it contains. The overall measurement across all cells at a given resolution is exponentially weighted, giving more weight to partitions with less cells. Afterwards, the area under the curve defined by one separability measure versus the resolution defines the overall data separability, which is bounded within the $[0, 1]$ range, such that higher values are obtained for more separable data. 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. The author claims his measures are uncorrelated to the number of features, classes, or data points in a dataset, so that they remain comparable for datasets with different values for these parameters.

In (Gong and Huang, 2012) a decision tree method based on the KolmogorovSmirnov statistic (KS tree) segments the training dataset, dividing the problem into easier sub-problems where class imbalance is relieved. The authors state that decomposing a complex imbalanced problem into easier sub-problems can mitigate the impact of class imbalance in classification results.

In (Armano and Tamponi, 2016) a method named Multiresolution Complexity Analysis (MRCA) is used to partition a dataset into regions of different classification complexity. Hyperspheres of different amplitudes are drawn around the examples and their 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).

3.4 Graph-based measures

In Garcia et al. (2015) the authors analyze how label noise affects the complexity of classification problems. This is performed by monitoring the sensitivity of several indices of data complexity in the presence of different label noise levels. This paper also presents structural measures of the dataset, which are obtained by representing the dataset as a graph. Next graph-based measures as number of edges, degree, density, closeness, hubs and average path are extracted. Low correlation values were observed between the basic complexity measures of Ho and Basu (2002) and the graph-based measures, stressing the relevance of exploring different views and representations of the data structure. Some similar graph-based measures for dataset characterization are also exploited in Morais and Prati (2013) in a meta-learning setup.

Zighed et al. (2002) present a statistical approach for characterizing the class separability degree. First a neighborhood structure is built using models like the Relative Neighborhood

Graph of Toussaint. After that, the number of edges that must be removed from the neighborhood graph to obtain homogeneous clusters (containing points from a same class) is counted. They have then established a law of the edge proportion that must be removed under the null hypothesis of a random distribution of the labels. Using this concept and a statistical test they are able to state if classes are separable or not.

3.5 Neighborhood Information

In (Van Der Walt and Barnard, 2007) neighborhood information is used to obtain a new definition of classifiability. If the class label of the examples are used as a $d+1$ input data dimension, a class label surface can be obtained. The smoothness or roughness of this surface gives an indicative of classification complexity, where rough surfaces are considered more complex. They use the probabilities of going from class c_i to class c_j within a distance d to estimate such information. This involves consulting the labels of neighborhood examples within a distance d. If they are from a same class, the class label surface will be smooth. This index is used in (Van Der Walt and Barnard, 2007) for feature selection purposes.

Hu et al. (2010) also propose a novel feature evaluation measure, named neighborhood decision error rate (NDER). It estimates classification complexity in different feature subspaces. They claim that only the boundary samples in the minority classes will be misclassified according to the Bayes rule and therefore they represent the real source of classification complexity. A neighborhood rough-set model is used to identify the boundary regions. The boundary examples are further grouped into recognizable and misclassified subsets. NDE is given as the percentage of misclassified examples identified.

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 neighbour with the same label. Another measure named Hypothesis margin (HM) considers the distance between an object's nearest neighbour of the same class and a nearest neighbour of another class. They combine these two measures in order to complement each other.

Anwar et al. (2014) introduce a complexity measure which also focuses on local information for each example by employing the nearest neighbor algorithm. Their study is devoted to obtain a complexity measure sensitive to class imbalance. 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, but it can also be decomposed by class in order to give an estimate of the complexity within each class.

Leyva et al. (2015) define some measures based on the concept of Local Sets, which employ neighborhood information. The proposed measures are used as meta-features in the design of a meta-learner able to perdict the expected performance of some popular instance selection methods. The Local-Set (LS) of an example $\mathbf{x}_i$ is defined as the set of instances from T whose distance to $\mathbf{x}_i$ is smaller than the distance to $\mathbf{x}_i$'s nearest enemy (nearest neighbor from a different class). The first measure is the Local Set Cardinality Average (LSCAvg), which averages the cardinality of the LS of all examples in T. Leyva et al. (2015) also propose to cluster the data in the local sets and counting 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 $N3$. According to the authors, this is first set of complexity measures specifically designed for characterizing instance selection problems.

3.6 Instance Hardness

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 in an instance level, rather than in the higher dataset level. 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. The Disjunct Size (DS) corresponds to the size of a disjunct that covers an example divided by the largest disjunct produced, where 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 Class Likelihood (CL) index estimates the likelihood of an instance belonging to a class. 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 measure the class skew. Their aggregate hardness measures have shown to be competitive with those from Ho and Basu (2002).

3.7 Data Sparsity

The study presented in (Lorena et al., 2012) makes use of the data complexity measures to analyze which particular characteristics of cancer gene expression data mostly impact the predictive ability of Support Vector Machine classifiers, often used in this area. Cancer gene expression data often present challenging characteristics, such as data sparsity and an skewed class distribution. Three new indexes are proposed in the paper: ratio of the principal component dimensionality to the number of instances, ratio of the principal component dimensionality to the real dimensionality of the problem and class balance (normalized class entropy). The first two indexes first perform a Principal Component Analysis (PCA) of the dataset, obtaining an intrinsic dataset dimensionality after correlation among features is minimized. They can be also regarded as data sparsity measures, being alternatives for T2.

Kamath et al. (2008) present a study on classification complexity of gene expression datasets too. They use three measures: proportion of genes identified as significant by a Students t-test; Fishers discriminant ratio; and the proportion of features whose values do not change between the classes. They report the Fishers discriminant ratio provides a good measure of the complexity of the classification problem in this particular domain.

Williams and Wagner (2009) also study measures for characterizing sparse datasets of large dimensionality and small sample sizes. One of them, named Intrinsic Dimensionality (ID), measures the smallest number of dimensions needed to represent a data set using the k nearest neighbor algorithm with multiple k values. Henze-Penrose divergence (HPD) and Bayes Error Estimator (BEE) are presented as multivariate analysis tools which measure class separability and classification error rates, respectively.

3.8 Linear Separability

Elizondo et al. (2012) focus their study on the relationship between linear separability and the level of complexity of classification datasets. Their method use Recursive Deterministic Perceptron (RDP) models and count the number of hyperplanes needed to transform the original problem, possibly non-linear separable, into a linearly separable problem. This is given by the number of intermediate neurons of the RDP network. Although they use a classification model, the authors state their method do not focus on the generalization performance of this classifier.

3.9 Class Separability

In (Skrypnyk, 2011) various class separability measures are presented. They focus on measures that can be employed in feature selection. Some parametric measures are the Mahalanobis and the Bhattacharyya distances between the classes and the Normal Information Radius. All of them take into accont the centroids of the classes and their covariance matrices. These measures are computationally intensive due to the need to compute covariance matrices and their inverse. An information theorectical measure is the Kullback-Leibler distance. It quantifies the discrepance between two probability distributions. Based on discriminant analysis, a number of class separability measures can also be defined. In this analysis, three scatter matrices are obtained: intra-class, inter-class and total. The ratio of the traces of the intra-class and the inter-class scatter matrices is one example. N2 is slighly related to this concept. Concerning feature selection, Skrypnyk (2011) report most encouraging results for the Normal Information Radius, Kullbak-Leibler distance and N2 measures.

In (Okun and Priisalu, 2009) a scheme for generating ensembles of classifiers that selects subsets of features of low complexity is proposed. Linear discriminants are used to evaluate the quality of the features. Data is projected into a linear discriminant axis using subsets of the features. The projected coordinates are submitted to a Wilcoxon rank sum test, whose result is used to estimate the classification complexity. Their scheme has obtained good results in a set of experiments on gene expression datasets, which show a high number of features and a low number of examples.

3.10 Complexity Map

Cummins (2013) employ the dataset complexity measures to evaluate case base editing algorithms. Therein, they also define some alternative measures. The first, named N5, consists of multipling N1 by N2. According to Fornells et al. (2007), the multiplication of N1 and N2 emphasizes extreme behaviors concerning class separability. Another measure (named Case Base Complexity Profile - C1) 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 accounted. The obtained values are then averaged and subtracted from one (high values indicates a higher complexity).

The N5 measure was inspired by the complexity map from Fornells et al. (2007). It combines F3 versus the product of N1 and N2. This map allows evaluating the dataset properties according to both the discriminant power of the features and the class separability, which can complement each other. Figure 3.1 presents an example from (Fornells et al., 2007) where a number of datasets are ploted as points in the complexity map according to their F3 and N1.N2 values. The map contains three regions, defined according to their distance to the point of minimum complexity (1,0): A (less complex), B (medium complexity) and C (more complex).

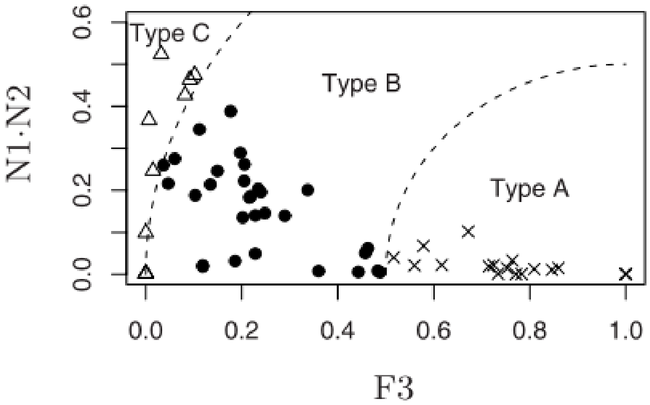


Figure 3.1: Complexity map (Fornells et al., 2007).

3.11 Discussion

As noted in this chapter, there are many alternative measures in the literature for characterizing the complexity of classification problems. Many of them are quite similar, so as D3 from (Sotoca et al., 2006), kDN from (Smith et al., 2014a) and the measure from (Anwar et al., 2014). Various also employ the nearest neighbor algorithm in their computation (ex. D2 and D3 from (Sotoca et al., 2006); nearest neighbor separability measure from (Singh, 2003a) and (Anwar et al., 2014); SI from (Mthembu and Marwala, 2008); C1 from (Cummins, 2013); ID from (Williams and Wagner, 2009); LS-based from (Leyva et al., 2015); and even the graph-based measures from (Morais and Prati, 2013; Garcia et al., 2015), which are based on an ϵ-nearest neighbor graph built from data). The general intuition is that in simple datasets the examples from one class will be closer to other examples from the same class while distant from examples from other classes.

Smith et al. (2014a) also highlight a worthwhile point: that some measures are unable to provide an instance-level hardness estimate. Understanding which instances are hard to classify may be a valuable information, since more efforts can be devoted to them. However, it should be noted that many of the complexity measures originally proposed for a dataset-level analysis can be 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.

There are no empirical or theorectical studies relating all measures presented in this chapter. It would be interesting to: (i) identify those measures with most distinct concepts, since many of them heve similar computation; (iii) summarize their main characteristics, as performed for the basic complexity measures in Table 2.1; and (iii) compare their ability in revealing the complexity of a diverse set of classification problems.

Chapter 4

Application Areas

The data complexity measures have been applied to support of various ML tasks. This chapter discusses some of the main applications of the complexity measures found in the relate literature. They are roughly divided into the following categories: (i) domain analysis, where the measures are used to understand the peculiarities of a particular domain; (ii) data selection and generation, where the measures are employed in dataset selection and generation; (iii) data pre-processing, where the measures are employed in data-preprocessing tasks such as feature selection and noise identification; (iv) learning algorithms, where the measures are employed for understanding or in the design of ML algorithms; (v) meta-learning, where the measures are employed as meta-features to characterize learning datasets.

4.1 Domain Analysis

This section describes the use of the data complexity measures in some specific application domains. The general intent is to get a better understanding on how the main characteristics from the datasets available for learning in these domains affect the classification performance achieved in their analysis.

In (Lorena et al., 2012) the complexity measures are employed to analyze the characteristics of cancer gene expression data which mostly impact the predictive performance in their classification. The paper also shows that, by applying a proper feature selection procedure to the data, the influence of those characteristics in the error rates of the classifiers induced is reduced. Cancer gene expression data often present challenging characteristics, such as data sparsity and a skewed class distribution. The indices from Ho and Basu (2002) and three other indices measuring class balance and data sparsity are employed. The most important complexity aspects in the experiments were related to data sparsity and class separability, measured by N1 and N3.

In (Kamath et al., 2008) three measures of classification complexity are proposed to explore limits on the predictive accuracy that can be achieved in a gene expression dataset. In their study the Fishers discriminant ratio (F1) proved to be a good measure of the complexity of the gene expression classification problem.

Angulo and Godo (2007) present a methodology to reduce the complexity of the breast cancer diagnosis problem. Therein, they search for the best problem transformation minimizing geometrical complexity, where the data complexity measures are employed to find proper transformations of the classification problem.

4.2 Data Selection and Generation

An interesting use of the data complexity measures has been carried out to generate artificial datasets with controlled characteristics. Some data repositories containing data complexity information has also been devised. These approaches are described next.

4.2.1 Data Repositories

Smith et al. (2014b) present the Machine Learning Results Repository (MLRR), which maintains and provides easy access to data for meta-learning tasks. For all datasets, it presents the predictions obtained by various classifiers per instance. The algorithms' hyperparameters and the training set partitions used are also made available. The complexity measures are employed as meta-features to describe the datasets. The instance hardness measures from (Smith et al., 2014a) are included too.

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). For expanding the diversity of the repository, Macià and Bernadó-Mansilla (2014) suggest to include artificial datasets carefully designed to spam the complexity space. This gives rise to the UCI+ repository.

4.2.2 Data Generation

In (Macia et al., 2008) the N1 measure is used for generating artificial datasets. A MST is built from random data and the labels of the connected vertices are assigned such as to meet a user specified boundary length. In (Macià et al., 2008) a Genetic Algorithm is used to generate the synthetic datasets with a desired N1 value.

Macià et al. (2013) propose using the data complexity measures for guiding which datasets should be included in an experimental design and also for generating artificial datasets. The artificial data sets were designed to cover all regions of the complexity measurement space. Multi-objective Genetic Algorithms were used to optimize the complexity measures in the dataset generation process, taking as base examples from real datasets.

4.3 Data-Proprocessing

This section describes the use of the data complexity measures in the support of data pre-processing tasks. The first one is Feature Selection (FS) (Liu et al., 2010). The objective of FS is to reduce the dimensionality of a dataset by keeping its most relevant features only, while discarding irrelevant and redundant features. Another related pre-processing step is the selection of the most relevant data points in a dataset, a task named instance or prototype selection (Garcia et al., 2012). This allows reducing the size of the learning dataset, which is particularly interesting for instance-based learning algorithms such as k-NN.

The third pre-processing step discussed in this section is noise identification (Frenay and Verleysen, 2013). In classification problems, noise can be present in either the predictive features or in the labels of the examples (Zhu and Wu, 2004). There are different strategies to deal with noise (Frenay and Verleysen, 2013): to filter the data in order to identify and remove or correct the noisy examples; to devise noise tolerant classifiers; to embed a data cleansing step into the learning algorithm.

Another issue in classification problems is data unbalance (He and Garcia, 2009). It happens when two or more of the classes in a dataset have a large difference in their number of examples. When confronted to this situation, most of the ML techniques will favor the majority class, harming the identification of the minority class examples. There are various strategies for dealing with class imbalance, which usually involve undersampling the majority class or oversampling the minority class. It is also possible to employ cost-sensitive learning procedures or modify the learning algorithm to deal with data imbalance.

4.3.1 Feature Selection

One of the first works to employ data complexity measures in FS was (Singh, 2003b). In his work Singh (2003b) define some new measures, based on partitioning the feature space into multiple resolutions (described in Section 3.3). He then proposes to perform feature selection by maximizing a neighbourhood separability measure. Dong and Kothari (2003) also present a preliminary work on FS by using a definition of classifiability introduced in their work (described in Section 3.5). Their method systematically adds features which increase the classifiability the most. In (Hu et al., 2010) the NDER complexity measure is proposed (Section 3.5) and embed into a FS algorithm. Their algorithm was effective for datasets containing both discrete and continuous features.

Baumgartner et al. (2006) use the data complexity measures to understand feature selection effects in magnetic resonance spectrum data classification. Using N1, N2, T1, F2, F1, L3 and N4 they find that a GA-averaged feature set makes the classification problem easier. Herewith, there are less borderline examples and the classes become more concentrated and spherical.

Pranckeviciene et al. (2006) propose to quantify whether FS effectively changes the complexity of the original classification problem. For such they monitor the values of the measures N1, N2 and T1 in datasets with dimensionality reduced by two popular FS techniques. In both cases, they found that FS was able to increase class separability in the reduced spaces.

Skrypnyk (2011) perform a similar study, by monitoring changes in class separability due to the presence of irrelevant features in a dataset. All of the measures from Ho and Basu (2002), except from F2, F3 and T2, are used. In a controlled set of experiments, they observed that class separability has changed after the elimination of irrelevant features. They report as the most encouraging results in reflecting the changes those obtained by the measures: Normal Information Radius, Kullbak-Leibler, and N2.

4.3.2 Instance Selection

Instance (or prototype) selection (IS) has been the theme of various work involving the data complexity measures. Mollineda et al. (2005) tries to predict which instance selection algorithm must be applied to a new dataset. They report the F1 measure is more suitable to indicate when IS should be clearly performed. García et al. (2009) employ the F1 measure to predict when evolutionary IS algorithms will be effective for a particular problem.

Leyva et al. (2015) investigate the IS problem using meta-learning. They first relate the performance of some IS techniques and the values of some of data complexity measures, namely those from Ho and Basu (2002) and some developed in their work (local-set based measures, described in Section 3.5). Their measures perform better as meta-features for the IS recommendation problem.

Cummins and Bridge (2011) use the data complexity measures to evaluate case base editing algorithms, which aim to reduce the case base by maintaining the most relevant cases only. One of the algorithms evaluated, for instance, was able to reduce complexity (measured by F3, N1 and T1), but at the expense of classification accuracy when the edited base is used in case retrieval.

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 for the original datasets. The motivation is that some complexity measures are costly to calculate, such that they may benefit from dataset reduction. They report IS can be succesfully used in the case of all overlap measures, except from F1.

4.3.3 Noise Handling

Under his partitioning framework, Singh (2003b) discussed that any point $\mathbf{x}$ surrounded by cells that only contain data from other classes can be identified as outliers. Smith et al. (2014a) identify as noisy those instances harder to classify by a diverse set of classifiers. This is related to their instance-hardness concept (Section 3.6).

In (Saz et al., 2013) the complexity measures are used to predict when the use of a noise filter will statistically improve the predictive results of the NN classifier. A methodology that extracts a rule set based on the data complexity measures is devised. All measures from Ho and

Basu (2002) are used, except from T2. The obtained rule set was fairly accurate in predicting the efficacy of the noise filters. Moreover, the best complexity measures for such were F2, N2 and F3, followed by T1 and F1. All of them provide information about the shape of the classes and their overlapping.

Garćia et al. (2013) use the complexity measures as features for describing noisy and noiseless datasets. Thereafter classifiers are induced to identify the presence of noise in new datasets. This identification can support a decision on the need for noise filtering. Usually, this identification was easier for datasets with higher noise levels.

In (Garcia et al., 2015) the authors investigate how different label noise levels affect the values of the complexity measures. Herewith, structural complexity measures are also used (Section 3.4). Measures able to capture characteristics such as the separability of the classes, alterations in the class boundary and the densities within the classes were most affected by the introduction of label noise in the data. The two measures most sensitive to noise imputation were then combined to develop a new noise filter, named GraphNN.

Garcia et al. (2016) develops a meta-learning recommendation system able to predict the expected performance of some filters in noise identification. The complexity measures are used as meta-features for characterizing the datasets. The recommendation system devised can support the choice of a particular filter for a new dataset. Two of the complexity meta-features that account for classes separability were among the most informative in the obtained recommendations.

4.3.4 Class Imbalance

Gong and Huang (2012) find that the underlying data complexity level of a dataset is more determinant in a model's performance than class imbalance and that class imbalance amplifies the effects of data complexity. When the data is easy to separate, all models perform well despite of extreme class imbalance. They employ a Decision Tree based on the Kolmogorov-Smirnov statistic (K-S tree) to segment the training data, dividing the original problem into several easier sub-problems where class imbalance is relieved.

Vorraboot et al. (2012) adapt the back-propagation (BP) algorithm to take into account the class overlap (measured by F1) and the imbalance ratio of a dataset. Modifications are made into the optimized MSE so as to measure the error rate for imbalanced datasets more properly.

López et al. (2012) uses the F1 measure to analyze the differences between preprocessing techniques and cost-sensitive learning for addressing imbalanced data classification. They also present a procedure for combining both approaches.

In (Xing et al., 2013) the original complexity measures from Ho and Basu (2002) are evaluated faced to different data distributions. The authors report that most of the measures need to be adapted in the case of imbalanced data, since only F1 kept unchanged when data skeweness was increased.

Anwar et al. (2014) introduce in their work a new complexity measure sensitive to class imbalance. It uses nearest neighbor information and was described in Section 3.5. According to these authors, among the original measures from Ho and Basu (2002), only F1, L3 and N1 were informative in the imbalanced scenario.

4.4 Learning Algorithms

This section describes works which employed the data complexity measures at algorithmic level analysis. These analysis can be either for devising, tuning or understanding the behavior of different learning algorithms.

A 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 identifying problems' characteristics for which a given technique will probably succeed or fail. While improving the understanding of each technique's capabilities and limitations, 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.

The last activity shown in this section is parameter tuning. Most of the ML classification techniques have parameters that need to be specified prior to their application (Lorena and De Carvalho, 2008). Usually a trial-and-error approach is adopted for such, which is usually incomplete, since a limited number of parameter values are tested, and labor intensive. The data complexity measures can be regarded as heuristics to guide this process.

4.4.1 Domains of Competence

Bernadó-Mansilla and Ho (2005) analyze the domain of competence of the XCS classifier. XCS is a classifier system which combines reinforcement learning and genetic algorithms. The authors report high correlations between the XCS's performance and the values of the measures N1, N2 and N3. This work is extended in (Ho and Mansilla, 2006) by adding more classifiers, namely NN, Linear Classifier, Decision Tree, Subspace Decision Forest and Subsample Decision Forest. In this case, the most relevant metrics for discriminating between the domains of competence of the classifiers were N1, L3, N4 and N2.

Flores et al. (2014) use the complexity measures to find datasets that fit for a semi-naive Bayesian Network Classifier (BNC). They are also used to recommend the best semi-naive BNC to use for a new dataset.

In (Luengo and Herrera, 2015) an 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. They show how N1, N3, L1 and L2 can properly characterize the behavior of three classification techniques (DT, SVM and k-NN).

Trujillo et al. (2011) use the complexity indices to estimate the expected performance of a classifier based on Evolutionary Algorithms (Genetic Programming - GP classifier). Their objective is to determine when a problem can be considered difficult for the GP classifier. The measures employed were F1, F2, F3 and N2, one measure extracted from the GP population (named SD) and some standard measures for data characterization, like the average correlation of the features. It was possible to accurately predict the expected performance of the GP classifier using such indexes. Moreover, the measures SD, F2, F3 and N2 were the most informative in this task.

Ciarelli et al. (2013) try to identify characteristics from a dataset that make it suitable for incremental learning. In particular, the model should be able to accommodate new knowledge without loosing crucial past information. The shape of the class boundary and the spatial distribution of the samples have shown to influence more this trade-off. Datasets with simpler and well-defined class boundaries and low feature overlap were considered more favorable. On the other hand, incremental learning can be impaired for datasets with complex and poorly defined class boundaries.

Garcia-Piquer et al. (2012) verify if a CBR approach based on multiobjective Evolutionary Algorithms is affected by data complexity. They report the tested algorithm was not affected for the measures F3, N1 and N2 and it was also more stable than a baseline.

Fornells et al. (2007) analyze the performance of case retrieval when the case memory is clustered. Using F3, N1 and N2 in a complexity map (Section 3.10), they find that a SOM clustering algorithm allied to CBR works better in complex domains.

Ho (2000) related the values of the complexity measures to the performance of two ensemble methods that use DTs as base classifiers and found strong correlations between classifier accuracies and measures N1 and T1. She also discuss data characteristics for which each technique performs better. In a similar study, Ho (2002) compare two methods for building decision forests: bootstrapping and random subspaces. In this case, strong correlations between the measures N1, T1, N4 and the classifier accuracies are found. Bootstrapping showed better results for sparse data, while the subspace method was better for datasets with compact classes and smooth boundaries.

Britto Jr et al. (2014) explore the concept of Dynamic Selection (DS) of classifiers in ensembles, in which specific classifiers are selected for each test example. They verified a relation between the performance gain obtained by DS and the complexity of the classification problem, measured by F1, F2, N2, N3 and T2. In particular, DS performed better in problems with low F1 values combined to high N3 and T2 values.

4.4.2 Modifications into Learning Algorithms

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.

The work from Vorraboot et al. (2012), described in Section 4.3.4, also adapted the BP algorithm. Their objective was to make the NN more robust to imbalanced datasets.

Campos et al. (2012) explore the idea of computing data complexity locally rather than globally. They present a classifier competence estimation method which is based on local data complexity for weighting decision forests constructed with a Random Subspace Method (RSM). The F4 and N1 measures were used. Their approach outperformed the standard non-weighted average combination, while consuming less time.

4.4.3 New Techniques

Okun and Priisalu (2009) propose a method for generating ensembles of k-NN classifiers that selects subsets of features of low complexity. This method was briefly described in Section 3.9. It employes a linear discriminant analysis combined to a hypothesis test. For experiments on gene expression datasets, they report predictive results superior to that obtained by the single best classifier in the ensemble and by a traditional ensemble construction scheme.

Lorena and Carvalho (2010) use measures F1 and F2 as splitting criteria for building binary-tree-based multiclass classifiers. The ideia is positioning easier binary separations at higher levels of the tree. Suitable hierarchical structures were obtained in experiments involving several benchmark multiclass data sets.

Garcia et al. (2015) present a new technique for label noise identification which is based on two complexity measures highly correlated to noise level in classification datasets: N3 and degree (graph-based measure). This technique showed comparable results to some popular filtering techniques from relate literature.

4.4.4 Parameter Tuning

In (He et al., 2015) the data complexity measures are applied to describe the leak quantification problem. In addition, a parameter tuning procedure based on the leak quantification is proposed. It maximizes data complexity under some domain-specific constraints. The parameters selected by employing N1 and T1 outperformed those using other quantification measures.

Nojima et al. (2011) use the complexity measures to specify the parameter values of fuzzy classifiers. Therein, a meta-classifier is induced to select the appropriate fuzzy classifier along with its parameters for new problems. The data complexity measures are employed as meta-features. They report some conclusions regarding the values of the complexity measures and their relationship to the performance of the fuzzy classifiers.

4.5 Meta-Learning

In Meta-learning (MTL) meta-knowledge about the solution of previous problems is used to aid the solution of new problems (Vilalta and Drissi, 2002). For this, a meta-dataset composed of datasets for which the solutions are known is built. They must be described by meta-features, which is how the complexity measures are mainly used in this area. Some work previously described have made use of meta-learning and also fit in this category (ex. (Smith et al., 2014b; Leyva et al., 2015; Garcia et al., 2016; Nojima et al., 2011)).

Sotoca et al. (2006) was one of the first works to present a general meta-learning framework based on a number of the data complexity measures. Van Der Walt and Barnard (2007) employes 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. As metafeatures they use both the cross-validation errors and other measures characterizing the datasets, among them T2. Their results suggest that meta-learning can be more accurate than cross-validation is classifier selection, while taking less time.

Cavalcanti et al. (2012) use the data complexity measures F1, F2, F3, N1, N2, T1 and T2 to predict the behavior of the NN classifier. While no single measure was good enough in the task, their combination provided good results.

4.6 Discussion

This chapter presented some of the main applications of the data complexity measures found in the literature. It can be observed throughout the review that these indices 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. Another common use of the measures is as meta-features for describing datasets in meta-learning studies.

However, some relevant work has also been done in devising new learning schemes and pre-processing techniques. This highlights the potential of these measures, which still remains poorly explored.

Chapter 5

Conclusion

This report reviewed the main data complexity measures from the literature. These indexes allow to characterize the difficulty of a classification problem. They were preliminary proposed and analyzed in Ho and Basu (2002) and have been extensively used so far in the analysis and development of classification and pre-processing techniques.

The original complexity measures along with some direct generalizations were presented in details along with some of their main limitations. Next new measures found in relate literature were briefly presented. Despite the presence of many indexes for measuring the complexity of classification problems, many of them have similar concepts. There is not a study comparing them neither revealing which ones can extract more distinct aspects regarding data complexity.

Lastly, the main domains and problems where the measures have been applied were presented. The most common use of the measures is to both characterize: datasets in meta-learning studies; or the domain of competence of learning and pre-processing techniques. Nonetheless, more contributions remain open regarding employing the conclusions of these studies in the adaptation and proposal of new learning and pre-processing schemes.

Bibliography

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

Angulo, C. and Godo, L. (2007). Modeling problem transformations based on data complexity. *Artificial Intelligence Research and Development*, 163:133.

Antolínez, N. M. (2011). *Data Complexity in Supervised Learning: a far-reaching implication.* PhD thesis, La Salle, Universitat Ramon Llull.

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

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

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

Baumgartner, R., Ho, T. K., Somorjai, R., Himmelreich, U., and Sorrell, T. (2006). Complexity of magnetic resonance spectrum classification. In *Data Complexity in Pattern Recognition*, pages 241–248. Springer.

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

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

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

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

Ciarelli, P. M., Oliveira, E., and Salles, E. O. (2013). Impact of the characteristics of data sets on incremental learning. *Artificial Intelligence Research*, 2(4):p63.

Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

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

Cummins, L. and Bridge, D. (2011). On dataset complexity for case base maintenance. In *Case-Based Reasoning Research and Development*, pages 47–61. Springer.

Dong, M. and Kothari, R. (2003). Feature subset selection using a new definition of classificability. *PRL*, 24:1215–1225.

Elizondo, D. A., Birkenhead, R., Gamez, M., Garcia, N., and Alfaro, E. (2012). Linear separability and classification complexity. *Expert Systems with Applications*, 39(9):7796–7807.

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

Fornells, A., Golobardes, E., Martorell, J. M., Garrell, J. M., Macià, N., and Bernadó, E. (2007). A methodology for analyzing case retrieval from a clustered case memory. In *Case-Based Reasoning Research and Development*, pages 122–136. Springer.

Frenay, B. and Verleysen, M. (2013). Classification in the presence of label noise: a survey. *Neural Networks and Learning Systems, IEEE Transactions on*, in press(99):1–25.

Gamberger, D. and Lavrac, N. (1997). Conditions for Occam's razor applicability and noise elimination. In *Proc European Conf on Machine Learning*, pages 108–123.

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

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

Garćia, L. P. F., de Carvalho, A. C., and Lorena, A. C. (2013). Noisy data set identification. In *Hybrid Artificial Intelligent Systems*, pages 629–638. Springer.

García, S., Cano, J.-R., Bernadó-Mansilla, E., and Herrera, F. (2009). Diagnose effective evolutionary prototype selection using an overlapping measure. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(08):1527–1548.

Garcia, S., Derrac, J., Cano, J. R., and Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):417–435.

Garcia-Piquer, A., Fornells, A., Orriols-Puig, A., Corral, G., and Golobardes, E. (2012). Data classification through an evolutionary approach based on multiple criteria. *Knowledge and information systems*, 33(1):35–56.

Gong, R. and Huang, S. H. (2012). 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.

Gunal, S. and Edizkan, R. (2008). Subspace based feature selection for pattern recognition. *Information Sciences*, 178(19):3716–3726.

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

He, Z.-M., Chan, P. P., Yeung, D. S., Pedrycz, W., and Ng, W. W. (2015). 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.

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

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

Ho, T. K. (2004). Geometrical complexity of classification problems. *arXiv preprint cs/0402020*.

Ho, T. K. and Basu, M. (2000). Measuring the complexity of classification problems. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, pages 43–47. IEEE.

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

Ho, T. K. and Mansilla, E. B. (2006). Classifier domains of competence in data complexity space. In *Data complexity in pattern recognition*, pages 135–152. Springer.

Hoekstra, A. and Duin, R. P. (1996). On the nonlinearity of pattern classifiers. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 4, pages 271–275. IEEE.

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

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

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

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

Leyva, E., Gonzalez, A., and Perez, R. (2015). A set of complexity measures designed for applying meta-learning to instance selection. *Knowledge and Data Engineering, IEEE Transactions on*, 27(2):354–367.

Li, L. and Abu-Mostafa, Y. S. (2006). Data complexity in machine learning. Technical Report CaltechCSTR:2006.004, Caltech Computer Science.

Liu, H., Motoda, H., Setiono, R., and Zhao, Z. (2010). Feature selection: An ever evolving frontier in data mining. *JMLR: Works/Conf Proc of The 4th Works on Feature Selection in Data Mining*, 10:4–13.

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

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

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

Lorena, A. C., Costa, I. G., Spolar, N., and Souto, M. C. P. (2012). Analysis of complexity indices for classification problems: Cancer gene expression data. *Neurocomputing*, 75:33–42.

Lorena, A. C. and De Carvalho, A. C. (2008). Evolutionary tuning of svm parameter values in multiclass problems. *Neurocomputing*, 71(16):3326–3334.

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

Macià, N. and Bernadó-Mansilla, E. (2014). Towards uci+: A mindful repository design. *Information Sciences*, 261:237–262.

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

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

Macià, N., Orriols-Puig, A., and Bernadó-Mansilla, E. (2008). Genetic-based synthetic data sets for the analysis of classifiers behavior. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*, pages 507–512. IEEE.

Mansilla, E. B. and Ho, T. K. (2005). Domain of competence of xcs classifier system in complexity measurement space. *IEEE Transactions on Evolutionary Computing*, 9(1):82–104.

Ming, L. and Vitanyi, P. (1993). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag.

Mollineda, R. A., Snchez, J. S., and Sotoca, J. M. (2005). Data characterization for effective prototype selection. In *Proc 2nd Iberian Conf on Pattern Recognition and Image Analysis*, pages 27–34. Springer.

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

Mthembu, L. and Marwala, T. (2008). A note on the separability index. *arXiv preprint arXiv:0812.1107*.

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

Okun, O. and Priisalu, H. (2009). Dataset complexity in gene expression based cancer classification using ensembles of k-nearest neighbors. *Artif Intell in Medicine*, 45(2-3):151–162.

Orriols-Puig, A., Maci, N., and Ho, T. K. (2010). Documentation for the data complexity library in c++. Technical report, La Salle - Universitat Ramon Llull.

Pranckeviciene, E., Ho, T. K., and Somorjai, R. (2006). Class separability in spaces reduced by feature selection. In *Proc 18th Inte Conf Pattern Recognition*, volume 2, pages 254–257.

Saz, J. A., Luengo, J., and Herrera, F. (2013). Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. *Pattern Recognition*, 46:355–364.

Singh, S. (2003a). Multiresolution estimates of classification complexity. *IEEE Trans PAMI*, 25:1534–1539.

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

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

Smith, F. W. (1968). Pattern classifier design by linear programming. *Computers, IEEE Transactions on*, 100(4):367–372.

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

Smith, M. R., White, A., Giraud-Carrier, C., and Martinez, T. (2014b). An easy to use repository for comparing and improving machine learning algorithm usage. *arXiv preprint arXiv:1405.7292*.

Sotoca, J., Mollineda, R., and Sanchez, J. (2006). A meta-learning framework for pattern classification by means of data complexity measures. *Intel Artif*, 10(29):31–38.

Souto, M. C. P., Lorena, A. C., Spolar, N., and Costa, I. G. (2010). Complexity measures of supervised classification tasks: a case study for cancer gene expression data. In *Proc IJCNN*, pages 1352–1358.

Trujillo, L., Martnez, Y., Galvn-Lpez, E., and Legrand, P. (2011). Predicting problem difficulty for genetic programming applied to data classification. In *Proc Gecco*, pages 1355–1362.

Van Der Walt, C. and Barnard, E. (2007). Measures for the characterisation of pattern-recognition data sets. 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA).

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

Vorraboot, P., Rasmequan, S., Lursinsap, C., and Chinnasarn, K. (2012). A modified error function for imbalanced dataset classification problem. In *Computing and Convergence Technology (ICCCT), 2012 7th International Conference on*, pages 854–859. IEEE.

Williams, A. and Wagner, G. (2009). Error estimation procedure for large dimensionality data with small sample sizes. In *SPIE Defense, Security, and Sensing*, pages 73350N–73350N. International Society for Optics and Photonics.

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

Xing, Y., Cai, H., Cai, Y., Hejlesen, O., and Toft, E. (2013). Preliminary evaluation of classification complexity measures on imbalanced data. In *Proceedings of 2013 Chinese Intelligent Automation Conference*, pages 189–196. Springer.

Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210.

Zighed, D. A., Lallich, S., and Muhlenbach, F. (2002). Separability index in supervised learning. In *Principles of Data Mining and Knowledge Discovery*, pages 475–487. Springer.