Applied Intelligence (2023) 53:8073–8096
https://doi.org/10.1007/s10489-022-03793-w

ORIGINAL SUBMISSION

# Hostility measure for multi-level study of data complexity

Carmen Lancho[1] · Isaac Martín De Diego[1] · Marina Cuesta[1] · Víctor Aceña[1,2] · Javier M. Moguerza[1]

Accepted: 21 May 2022 / Published online: 26 July 2022

## Abstract

Complexity measures aim to characterize the underlying complexity of supervised data. These measures tackle factors hindering the performance of *Machine Learning* (*ML*) classifiers like overlap, density, linearity, etc. The state-of-the-art has mainly focused on the dataset perspective of complexity, i.e., offering an estimation of the complexity of the whole dataset. Recently, the instance perspective has also been addressed. In this paper, the *hostility measure*, a complexity measure offering a multi-level (instance, class, and dataset) perspective of data complexity is proposed. The proposal is built by estimating the novel notion of *hostility*: the difficulty of correctly classifying a point, a class, or a whole dataset given their corresponding neighborhoods. The proposed measure is estimated at the instance level by applying the $k$-means algorithm in a recursive and hierarchical way, which allows to analyze how points from different classes are naturally grouped together across partitions. The instance information is aggregated to provide complexity knowledge at the class and the dataset levels. The validity of the proposal is evaluated through a variety of experiments dealing with the three perspectives and the corresponding comparative with the state-of-the-art measures. Throughout the experiments, the *hostility measure* has shown promising results and to be competitive, stable, and robust.

**Keywords** Hostility measure · Complexity measures · Data complexity · Classification · Supervised problems

## 1 Introduction

For several years now, *Machine Learning* (*ML*) is in the spotlight and supervised problems account for an important part of it. For classification problems and, indeed, for all analysis involving data, a first step of data exploration is essential, providing the user with the knowledge and understanding of the data. This is extremely useful for the following tasks involving data, and skipping it could lead to wrong decisions and results. However, on a daily basis, this exploratory phase is rarely focused on the underlying complexity of the dataset. As a matter of fact, during the modeling stage, the selection of the best classifier ordinarily follows a trial-and-error approach. Several classifiers are tested and the one offering the best performance is finally selected. No information is drawn about why some classifiers perform better or which characteristics of the data are causing the final results. Different factors can disturb the performance of classifiers [3]. For instance, the distribution of classes, the sparsity of data, the type of decision boundary, the overlap among classes or the noise. The purpose of complexity measures is to identify and quantify this type of data characteristics as a way of understanding the complexity of the data and its impact in the classification [14].

Complexity measures have mainly focused on a global perspective, quantifying the complexity of the whole dataset from different points of view: linearity, overlap, balance of classes, etc. In the last years, a new approach building complexity measures at the instance level and averaging them to get the global dataset perspective has emerged

✉ Carmen Lancho
carmen.lancho@urjc.es

Isaac Martín De Diego
isaac.martin@urjc.es

Marina Cuesta
marina.cuesta@urjc.es

Víctor Aceña
victor.acena@urjc.es

Javier M. Moguerza
javier.moguerza@urjc.es

[1] Data Science Laboratory, Rey Juan Carlos University,
C/ Tulipán, s/n, 28933, Móstoles, Spain

[2] Madox Viajes, C/ de Cantabria, 10, 28939,
Arroyomolinos, Spain

[33]. In fact, some of the classic measures were originally constructed from the instance level [14]. The instance approach is fruitful since it provides the global complexity estimation by identifying the critical and problematic individual points that are actually causing that complexity. Thus, data complexity is analyzed from two points of view: the instance and the dataset level.

In this paper, the two-level perspective is taken a step further presenting, to our knowledge for the first time, a multi-level study of data complexity through the here defined concept of *hostility* and the proposed complexity measure, the *hostility measure*, that estimates it. The notion of *hostility* refers to the difficulty of correctly classifying a point, a class, or a dataset according to their surroundings. For example, a point fully surrounded by instances of its own class will have an *hostility* of zero. On the other hand, the more instances from other classes are around it, the more *hostility* the point will have. This concept is intuitive, it naturally embraces the various perspectives of data: point, class, and dataset, and it offers an interpretable value in terms of the probability of how difficult it is to classify them regarding their neighborhoods.

The estimation of the notion of *hostility* is carried out by building a complexity measure at the instance level, aggregating it at the class level to get a complexity value for each class and also, aggregating it at the dataset level to have a global quantification of the complexity. This is the proposed multi-level complexity measure which is called the *hostility measure*. It analyzes the distribution of classes in neighborhoods of increasing size. By doing this, it detects critical points. That is, those points that are in overlapping areas (a really detrimental factor for classifiers [31, 33, 36]). These can also be borderline points, which are near the decision boundary, or noisy points that can be faded among points from other classes. In contrast with most complexity measures, the here proposed *hostility measure* combines different layers of information since it is calculated by applying the well-known algorithm $k$-means in a recursive and hierarchical way. This increases the robustness and adaptability of the method. Also, tracking the results from the different layers of the procedure provides useful information. Some promising results of a preliminary version of the proposal have been presented in [20].

The main contributions of the present paper are:

1. To revisit the main state-of-the-art complexity measures clarifying its levels of definition.
2. To introduce the concept of *hostility*.
3. Based on the notion of *hostility*, to propose a new complexity measure called the *hostility measure* that addresses a multi-level perspective of the data complexity.
4. To evaluate the performance of the *hostility measure* and compare it with the state-of-the-art.
5. To present the *hostility* tracking graph and the overlapping tracking graph that are able to offer exploratory information about a dataset.

The paper is structured as follows. Section 2 recapitulates the state-of-the-art of complexity measures with special emphasis on the ones considering more than one level of information. The concept of *hostility* is formally presented in Section 3 and the proposed measure of complexity is described in Section 4. In Section 5, experiments comprising the three perspectives are expounded. Section 6 describes the research opportunities that have emerged along the current research. Finally, Section 7 concludes.

## 2 State-of-the-art

Complexity measures gained more attention as a result of the work from Ho and Basu [14]. Ever since, these measures have been further studied and applied for several purposes: imbalanced problems [2, 18, 30, 39], meta-learning [21, 24], analysis of learning algorithms [4, 27], automatic recommendation of classifiers [6], and hyper-parameter optimization [7]. They have also been implemented in different fields like genetics, medicine, and human-computer interaction [3]. A detailed summary with applications of complexity measures as well as a recapitulation of the existing complexity measures can be found in [25].

Regarding complexity measures that address more than one level of information, the work in [33] is seminal providing the instance perspective. The aim is to detect which instances are harder to classify and to calculate the individual contribution of the instances to the global complexity. To this end, a range of complexity measures, called *hardness measures*, were defined to tackle the instance perspective specifically. They can be later averaged to get the dataset level. In accordance with this new perspective, some of the classical measures have been adapted to the instance level [1].

Following [25], complexity measures are grouped in six main categories: feature-based, linearity, neighborhood, network, dimensionality, and class imbalance. Next, a brief explanation and the main measures of each category are described.

**Feature-based measures** focus on the overlap of features between classes to assess the discriminant ability of the features:

- $F1$, the maximum Fisher's discriminant ratio measures the capacity of every feature to separate the classes in terms of the overlap that the values of each feature present [14].

- $F2$, the volume of overlapping region calculated as the length of the overlapping zones among classes [14].
- $F3$, the maximum individual feature efficiency is defined as the maximum discriminant capacity of the features. This is calculated as the maximum number of points from different classes in overlap, for the set of features, divided by the total number of points [13].
- $F4$, the collective feature efficiency is similar to $F3$ but analyzing jointly the discriminatory power of the features [29].
- $F1v$, the directional-vector maximum Fisher's discriminant ratio was created as a version of $F1$ able to overcome the major drawback of feature-based measures: they assume the discriminative hyperplane is perpendicular to the axis of one input feature [29]. $F1v$ projects data for maximizing class separation and, in that projection, looks for a vector able to separate the classes.
- In [1], the idea of getting the overlap of features within the two classes is extended to the instance level with four *hardness measures*. $F1_{HD}$ captures the number of features for which an instance lies in an overlapping area. The distance of each instance to the overlapping region is gauged for each feature and then transformed to obtain higher values for instances placed on the middle of the overlapping region. This is calculated for each feature and $F2_{HD}$ is defined to be the minimum of them, $F3_{HD}$ the mean, and $F4_{HD}$ the maximum.

**Linearity measures** check the linear separateness in a problem:

- $L1$, the sum of the error distance by linear programming [14]. It evaluates if the data is linearly separable by adding the distances of the incorrectly classified instances to the linear boundary. Note that although $L1$ detects if a problem is linearly separable, it is not able to distinguish which one is the simpler linear problem. The instance version of $L1$ is $L1_{HD}$ [1] which multiplies, for each instance, its distance to the linear frontier by its label $y_i \in \{-1, +1\}$.
- $L2$, the error rate of a linear classifier [14].
- $L3$, non-linearity of a linear classifier [15]. It starts generating test points by linear interpolation between random pairs of points of the same class. Then, the linear classifier is trained on the original points and tested on the new points. $L3$ is the test error.

**Neighborhood measures** are based on the distance among points. They study the distribution of classes, how they intertwine with each other, the presence of overlapped and borderline points in neighborhoods:

- $N1$, the fraction of borderline instances obtained from a *Minimum Spanning Tree* (*MST*) built from the data [14]. Each vertex of the tree corresponds to one instance and the edges are weighted according to the distance between them. $N1$ is the percentage of vertices connected to instances of other classes. The instance version of $N1$, $N1_{HD}$, is defined as the number of connections the instance holds with instances from other classes [1]. Both measures are sensitive to noisy instances.
- $N2$ is the ratio of intra/extra class nearest neighbor distance [14], defined as $r/(1+r)$, where $r$ is the ratio of the sum of the distances between each point and its closest neighbor (intra) and the sum of the distances between every point and its closest neighbor from other class (extra). Its instance version $N2_{HD}$ takes the ratio value for each point [1]. $N2$ is influenced by the data distribution, the shape of the boundary, and noisy points.
- $N3$ is the error rate of the *k-Nearest Neighbour* (*kNN*) classifier with $k = 1$ and using a leave-one-out procedure [14].
- $N4$, non-linearity of the *kNN* classifier, is similar to $L3$ but using the *kNN* classifier with $k = 1$ [14].
- $T1$ is the fraction of hyperspheres covering the data [14]. This measure starts building a hypersphere centered at each point, whose radius is determined by the distance between the point and its nearest enemy (i.e., the nearest point from other class). Then, all the hyperspheres completely included in bigger hyperspheres are eliminated. Finally, $T1$ is the ratio between the final number of retained hyperspheres and the number of points.
- $LSC$: local-set average cardinality. Following [21], the local-set of an instance (originally defined in [5]) is the set of instances closer to that instance than its nearest enemy. The cardinality of the local-set indicates its proximity to the decision boundary and also the space between classes.
- In [1], four measures related to $T1$ and local-set concept are added:
    - – $LSC(\mathbf{x}_i)$, the local-set cardinality of each point.
    - – $LS_{radius}$ is the radius of the local-set of each point, showing how close every point is to the other class.

  - The *usefulness* index $U$ of an instance is the number of instances containing it in its local-set.
  - The *harmfulness* index $H$ of an instance is the number of instances for which it is the nearest enemy.

- The *k-Disagreeing Neighbors* (*kDN*) of a point is the percentage of its nearest neighbors from other classes [33]. It is averaged for the dataset level.
- *R-value* [28] measures the existing overlap among classes on the dataset. It examines, using *kNN*, if a point is in an overlapping area. If more than a parameter $\theta$ of its $k$ nearest neighbors are from other class, the point is considered to be in overlap. This is averaged to have an overlapping ratio per class and an overall overlapping ratio for the whole dataset.

**Network measures** are gauged based on a graph built from the data preserving the original similarities among instances. Each instance is represented as a node in the graph and is connected with undirected edges to instances distancing from it less than a threshold $\epsilon$. In the final graph, nodes from different classes are not connected. The main measures [11] are the average density of the network (*Density*), the clustering coefficient (*ClsCoef*) and the hub score (*Hubs*).

**Dimensionality measures** focus on the sparsity of the data and measure the relationship between the number of points and the number of features. The principal measures are: the average number of features per dimension $T2$ [14], the average number of *Principal Component Analysis* (*PCA*) dimensions per points $T3$ [23], and the ratio of the *PCA* dimension to the original dimension $T4$ [23].

**Class imbalance measures** assess the balance between the class sizes. Two common metrics are the imbalance ratio $C2$ and the entropy of class proportions $C1$.

Furthermore, in this work, the category **model-inspired measures** is proposed to be added to the previous taxonomy. These complexity measures are inspired by the learning mechanism of different classifiers. The *hardness measures* from [33] framed in this category are listed below. The dataset level value of these measures is just the average of the instance values.

- *Disjunct Size* (*DS*). The *DS* of an instance is the number of instances in its disjunct (i.e., leaf node where it is classified in a *Decision Tree* (*DT*)) divided by the number of instances in the largest disjunct. Disjuncts are created with a version of C4.5 algorithm: not pruned and allowing one instance per node.
- The *Disjunct Class Percentage* (*DCP*) of a instance is the proportion of instances from its class in its belonging disjunct.
- The *Tree Depth* (*TD*) is the depth of the leaf node in a *DT* where the point is classified. It uses a C4.5 decision tree in its pruned version, *Tree Depth Pruned* (*TDP*), and unpruned version *Tree Depth Unpruned* (*TDU*). Note that if a point is misclassified in a shallow split of the pruned tree, the resulting complexity information of that point is not trustworthy.
- Based on the philosophy of the Naïve Bayes, the *Class Likelihood* (*CL*) estimates the likelihood of an instance belonging to a class deeming independent features. For continuous variables, likelihood is gauged with a kernel density estimation.
- *Class Likelihood Difference* (*CLD*) offers the difference between the *CL* of an instance and its maximum likelihood for the rest of classes.

Table 1 summarizes the complexity measures of this section and details the level of definition of each measure.

## 3 Hostility

In this section, the formal notion of *hostility* at each one of the considered levels (point, class, and dataset) is presented. For this purpose, the notation used for the formal definitions of the *hostility* and the preliminary concept of the neighborhood of a point are first addressed.

Let $\boldsymbol{\mathcal{X}} = (\mathcal{X}_1, \dots, \mathcal{X}_p)^T$ be the input vector of $p$ random variables, $\mathcal{Y}$ the random output variable and assume that $(\boldsymbol{\mathcal{X}}, \mathcal{Y})$ is the corresponding joint distribution. Suppose that $D = \{(\mathbf{x}_i, y_i) \,|\, i = 1, \dots, n\}$ is the dataset containing $n$ independent and identically distributed observations from $(\boldsymbol{\mathcal{X}}, \mathcal{Y})$ where $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ is the $i$th observed value of $\boldsymbol{\mathcal{X}}$ and $y_i$ is the $i$th observed value of $\mathcal{Y}$, $i = 1, \dots, n$. Now, let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ be the set of input observations from $D$ and $\mathbf{Y} = \{y_i\}_{i=1}^n$ the set of the corresponding labels from $D$, where $y_i \in \mathcal{C} = \{1, \dots, c\}$ being $\mathcal{C}$ the set of class labels.

In these terms and following [17], the neighborhood of a point $\mathbf{x}_i \in \mathbf{X}$ is a subset of $\mathbf{X}$ containing an open ball with center $\mathbf{x}_i$ and radius $r > 0, r \in \mathbb{R}$. This is, containing the set of all points $\mathbf{x}_j \in \mathbf{X}$ such that $d(\mathbf{x}_i, \mathbf{x}_j) < r$, being $d(\cdot, \cdot)$ a distance function. The neighborhood of a point $\mathbf{x}_i \in \mathbf{X}$ will be denoted as $N(\mathbf{x}_i)$.

**Definition 1** The ***hostility* of an instance** is the difficulty of correctly classifying the instance given its neighborhood. That is, the *hostility* of an instance $(\mathbf{x}_i, y_i) \in D$, denoted as $H(\mathbf{x}_i, y_i)$, is the opposite of the probability of identifying

**Table 1** Levels of definition of each complexity measure

| Category | Instance | Class | Dataset |
|---|---|---|---|
| Feature | – | – | $F1$ [14] |
| | – | – | $F1v$ [29] |
| | $F1_{HD}, F2_{HD}, F3_{HD}, F4_{HD}$ [1] | – | $F2$ [14] |
| | $F1_{HD}, F2_{HD}, F3_{HD}, F4_{HD}$ [1] | – | $F3$ [13] |
| | – | – | $F4$ [29] |
| Linearity | $L1_{HD}$ [1] | – | $L1$ [14] |
| | – | – | $L2$ [14] |
| | – | – | $L3$ [15] |
| Neighborhood | $N1_{HD}$ [1] | – | $N1$ [14] |
| | $N2_{HD}$ [1] | – | $N2$ [14] |
| | – | – | $N3$ [14] |
| | – | – | $N4$ [14] |
| | $LSC(\mathbf{x}_i)$ [1], $LS_{radius}$ [1], $U$ [22], $H$ [22] | – | $T1$ [14] |
| | $LSC(\mathbf{x}_i)$ [1], $LS_{radius}$ [1], $U$ [22], $H$ [22] | – | $LSC$ [21] |
| | – | *R-value* [28] | *R-value* [28] |
| | *kDN* [33] | – | *kDN* [33] |
| Network | – | – | $Density$ [11] |
| | – | – | $ClsCoef$ [11] |
| | – | – | $Hubs$ [11] |
| Dimensionality | – | – | $T2$ [14] |
| | – | – | $T3$[23] |
| | – | – | $T4$ [23] |
| Class imbalance | – | – | $C1$ [23] |
| | – | – | $C2$ [34] |
| Model-inspired | *DS* [33] | – | *DS* [33] |
| | *DCP* [33] | – | *DCP* [33] |
| | *TDP*[33] | – | *TDP* [33] |
| | *TDU* [33] | – | *TDU* [33] |
| | *CL* [33] | – | *CL* [33] |
| | *CLD* [33] | – | *CLD* [33] |

its class $y_i$ given the distribution of classes of all the points that belongs to its neighborhood:

$$H(\mathbf{x}_i, y_i) = 1 - P(\mathcal{Y} = y_i \mid \{(\mathbf{x}_j, y_j) \in D \mid \mathbf{x}_j \in N(\mathbf{x}_i)\}), \tag{1}$$

being $\{(\mathbf{x}_j, y_j) \in D \mid \mathbf{x}_j \in N(\mathbf{x}_i)\}$ the instances pertaining to the neighborhood of the point $\mathbf{x}_i$, that is, to $N(\mathbf{x}_i)$.

**Definition 2** The ***hostility* of a class** $c$ is the difficulty of adequately identifying all the points of the class $c$, as belonging to class $c$, given their neighborhoods. That is, the *hostility* of a class $c$ is the opposite of the probability of correctly classifying the complete class $c$ given the points that belong to the neighborhood of the set of the points from class $c$. Let $D_c = \{(\mathbf{x}_i, y_i) \in D \mid y_i = c\}$ be the restricted dataset $D$ to class $c$. Then, the *hostility* of a class $c$, denoted as $H(D_c)$, is:

$$H(D_c) = 1 - P(\mathcal{Y} = c \mid \{(\mathbf{x}_j, y_j) \in D \mid \mathbf{x}_j \in N(\mathbf{X}_c)\}), \tag{2}$$

where $\mathbf{X}_c = \{\mathbf{x}_i \mid (\mathbf{x}_i, y_i) \in D, y_i = c\}$ is the set of all instances from class $c$ and $N(\mathbf{X}_c) = \bigcup_{\{(\mathbf{x}_i, y_i) \in D_c\}} N(\mathbf{x}_i)$ is the neighborhood of the set $\mathbf{X}_c$.

**Definition 3** The ***hostility* of a dataset** $D$ is the difficulty of correctly classifying all the points of $D$ given their neighborhoods. In other words, the *hostility* of a dataset $D$ is the opposite of the probability of identifying the class of each point of the dataset given the neighborhood of the set

$\mathbf{X}$ of input observations from $D$. Then, given a dataset $D$, its *hostility* $H(D)$ is:

$$H(D) = 1 - P((\mathcal{Y} = y_1, \ldots, \mathcal{Y} = y_n) \,|\, \{(\mathbf{x}_j, y_j) \in D \,|\, \mathbf{x}_j \in N(\mathbf{X})\}), \tag{3}$$

where $N(\mathbf{X}) = \bigcup_{i=1}^{n} N(\mathbf{x}_i)$ is the neighborhood of the set $\mathbf{X}$.

## 4 Proposed method

In this section, the proposed *hostility measure* to estimate the previously defined *hostility* concept is described. The *hostility measure* is able to provide knowledge in three different levels: instance, class, and dataset. It is initially calculated for every single point, offering an *hostility* estimation value for every instance $\widehat{H}(\mathbf{x}_i, y_i)$. These instance values are used for two further aggregations. First, an *hostility* value for every class $\widehat{H}(D_c)$ and second, a global value for the whole dataset $\widehat{H}(D)$. The dataset value goes hand-in-hand with complexity measures from the state-of-the-art. However, the perspective per class is quite novel and offers prior knowledge about which class is more affected by the distribution of others and, hence, will be harder to classify.

The calculation of the *hostility measure* starts by applying the $k$-means clustering algorithm with the *Euclidean* distance. If this unsupervised algorithm is applied to a supervised dataset and, for each cluster, the probability of every class is extracted, an informative class data structure map can be achieved. Not only this map will reveal where a class is dominant, but it will also point out the most uncertain areas where classifiers tend to fail. Nevertheless, if the parameter $k$ is not correctly selected, any exploratory analysis derived from the resulting partition would be worthless. To avert this situation the $k$-means algorithm is here hierarchically and recursively performed following [37] (see Fig. 1). The $k$-means is a simple method that allows to easily analyze the data in the natural groups they form according to their similarities and to select a good representative for each cluster. In addition, applying them in a hierarchic and recursive way guarantees robust partitions capturing the structure of the data and interactions among classes. Also, it enables to efficiently track the evolution of these partitions in the different iterations. In this recursive process, the different clustering iterations will be denoted as "layers" from now on. In the first layer, $k$-means is implemented using the whole set $\mathbf{X}$ and, in the next iterations, the data input is the set of centroids gathered from the previous step. Thus, the data input will be denoted as $\mathbf{X}'$, being $\mathbf{X}' = \mathbf{X}$ in the first iteration. Every time the algorithm is performed, a cluster partition $\mathcal{B} = \{B_1, \ldots, B_k\}$ of the current data input $\mathbf{X}'$ is achieved. $\mathcal{B}$ is a crisp partition matching:

$$\begin{gathered} \bigcup_{r=1}^{k} B_r = \mathbf{X}', \\ B_r \cap_{r \neq s} B_s = \emptyset,\ r \neq s,\ r, s \in \{1, 2, \ldots, k\}, \\ \emptyset \subset B_r \subset \mathbf{X}',\ r \in \{1, 2, \ldots, k\}. \end{gathered} \tag{4}$$

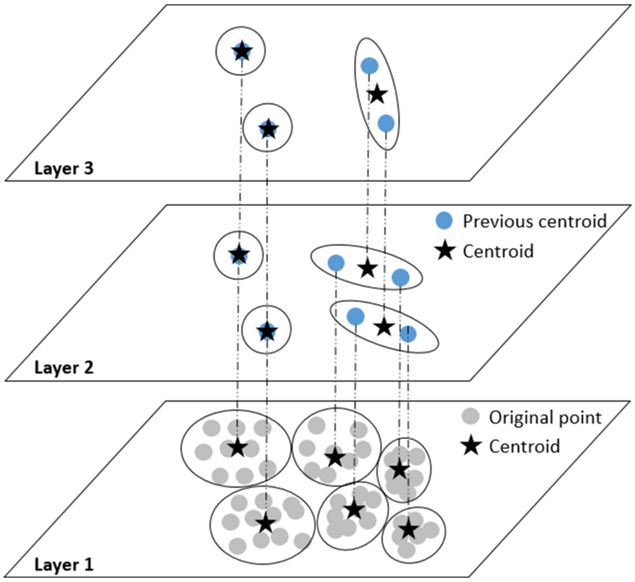


**Fig. 1** Hierarchical and recursive application of $k$-means based on [37]

Note that the larger the number of layers, the smaller the number of clusters and, consequently, more points are grouped together. The objective is to capture the behavior of classes through recursive partitions and to get successive clusters revealing how data from different classes are grouped.

In every layer $l \in \mathbb{N}$, for any original point $\mathbf{x}_i \in \mathbf{X}$, the probability of its class $y_i$ in the cluster $B_r \in \mathcal{B}$ it pertains to is stored. The probability is denoted as $p_{li}$, with $l$ indicating the number of the layer and $i = 1, \ldots, n$ referring to the particular instance $\mathbf{x}_i$. This probability is the proportion of the class $y_i$ in the specific cluster $B_r$ based on the original points that belong to it:

$$p_{li} = \frac{|\{\mathbf{x}_j \in \mathbf{X} \,|\, \mathbf{x}_j, \mathbf{x}_i \in B_r \wedge y_j = y_i\}|}{|\{\mathbf{x}_j \in \mathbf{X} \,|\, \mathbf{x}_j \in B_r\}|}, \tag{5}$$

where $|\cdot|$ represents the cardinal of a set. As the procedure is hierarchical, it is straightforward to get to which cluster a point belongs to at any layer. Thus, in every layer $l$, a probability vector $\mathbf{p}_l = (p_{l1}, \dots, p_{ln})^T \in \mathbb{R}^n$ is gathered and it is averaged with the probability vector from previous layers:

$$\mathbf{p} = (\mathbf{p}_l + \mathbf{p}_{l-1} + \cdots + \mathbf{p}_1) \cdot 1/l. \tag{6}$$

This probability vector $\mathbf{p} = (p_1, \dots, p_n)^T \in \mathbb{R}^n$ summarizes for every point the dominance of its class through the variety of clusters where the point has been grouped. This average probability vector is the key for the *hostility measure* calculation since it reflects, for each point, the presence of its class. Consequently, its opposite $1 - \mathbf{p}$ shows the absence of its class, that is, the dominance of the others and how harmful they are. In other words, the estimated *hostility* value for all points is just the opposite of this average probability vector, that is,

$$\widehat{H}(\mathbf{x}_i, y_i) = 1 - p_i. \tag{7}$$

These *hostility measure* values at the instance level estimate the probability of Definition 1. Finally, all points have, for every layer, an *hostility measure* value in the range [0, 1]. A high *hostility measure* value means that the point is surrounded by points from other class. Medium values imply the point lies in an overlapping area where both classes are quite equally dominant. Low values are for points in zones where its class is the dominant one.

To estimate the *hostility* for the class and the dataset levels meeting Definition 2 and 3, a probability threshold $\delta$ is applied to binarize the instance values. If the *hostility measure* is equal or higher than $\delta$, its binary value will be 1. Otherwise, it will be binarized to 0 as the point lies in areas where its class is better represented and is less harmful for the classification task. This binary information is averaged to achieve the *hostility* estimation per class and for the total dataset:

- The *hostility measure* of a class $c$ is calculated by averaging the binarized *hostility measure* of the instances belonging to that specific class:

  $$\widehat{H}(D_c) = \frac{\sum_{(\mathbf{x}_i, y_i) \in D_c} I(\widehat{H}(\mathbf{x}_i, y_i) \geq \delta)}{n_c}, \tag{8}$$

  being $n_c$ the number of instances in class $c$ and $I(\cdot)$ the indicator function that takes value 1 when its argument is true and 0 otherwise. This estimation gives an indication of how complex it is to identify each class within the dataset and allows a ranking of the complexity of the different classes.
- The global *hostility measure* for the dataset is calculated as the average of the binarized *hostility measure* of all points in dataset:

  $$\widehat{H}(D) = \frac{\sum_{(\mathbf{x}_i, y_i) \in D} I(\widehat{H}(\mathbf{x}_i, y_i) \geq \delta)}{n}. \tag{9}$$

  Similarly to the *hostility measure* per class, it is estimated as the proportion of critical points in the whole dataset. That is, points expected to be erroneously identified as from other class.

For both cases, the maximum value is reached when the *hostility measure* of all points is higher or equal than $\delta$, that is, when $\widehat{H}(\mathbf{x}_i, y_i) \geq \delta$, $\forall (\mathbf{x}_i, y_i) \in D_c$ for the class level and $\forall (\mathbf{x}_i, y_i) \in D$ for the dataset one. Notice that all the proposed *hostility measures* are defined in the range [0, 1] to ease its interpretation and comparison. Besides, it supplies, in both levels, an estimation of the classification complexity.

The proposed method has three parameters:

- The probability threshold ($\delta$) aforementioned.
- The proportion of grouped points per cluster ($\sigma$). This parameter automatically determines the number of clusters $k$ in every layer. The purpose of $\sigma$ is to set the pace of grouping in the recursive $k$-means process.
- The minimum number of clusters allowed ($k_{min}$). It cannot be lower than the number of classes. The iterative process stops when the following $k$ is going to be lower than $k_{min}$. The final results come from this last layer.

Algorithm 1 presents the pseudocode to obtain the *hostility measure*.[1] The inputs of the algorithm are the three parameters $\delta$, $\sigma$ and $k_{min}$, the set of points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ and the set of the corresponding binary labels $\mathbf{Y} = \{y_i\}_{i=1}^n$ with $y_i \in \{-1, +1\}$. The algorithm returns an *hostility* vector with the respective *hostility measure* per instance $\widehat{\mathbf{h}} = (\widehat{H}(\mathbf{x}_1, y_1), \dots, \widehat{H}(\mathbf{x}_n, y_n))$, the *hostility measure* per class $\widehat{H}_{-1} = \widehat{H}(D_{-1})$ and $\widehat{H}_{+1} = \widehat{H}(D_{+1})$, and the *hostility measure* of the whole dataset $\widehat{H}_D = \widehat{H}(D)$. It can also return the estimated *hostility* of each point per layer $\widehat{\mathbf{H}}_m$. Note that in all layers except for the first one, the hierarchical structure has to be used to extract the belonging cluster of every original point.

Since in every layer the *hostility measure* values are obtained, their evolution across partitions can be tracked. This tracking is used to select, by seeking for changes in the *hostility measure* behavior, the best layer to stop and, consequently, the $k_{min}$ parameter. A pattern change in this resulting *hostility* tracking graph will point out where the partition of clusters starts to lose the structure of the data, which is usually when the number of clusters is low. The final selected layer must be the one before the pattern changes to ensure that data structure is captured and stable results.

Notice that, even though the *hostility measure* uses $k$-means as the base of the method, it is not affected by its main drawbacks. The $k$-means method depends

[1] The code is available at https://github.com/URJCDSLab/Hostility_measure/tree/main/Algorithm_code.

**Algorithm 1** *Hostility measure* algorithm.

```
Input: $\mathbf{X}, \mathbf{Y}, \sigma, \delta, k_{min}$
Output: $\widehat{\mathbf{h}}, \widehat{H}_{-1}, \widehat{H}_{+1}, \widehat{H}_D, \widehat{\mathbf{H}}_m$
$min_k = \max(2, k_{min})$
$\mathbf{p} = (0, \ldots, 0)$    ▷ Initialization probability vector $\mathbf{p}$
$\widehat{\mathbf{H}}_m$    ▷ Matrix $\widehat{\mathbf{H}}_m$ to save *hostility* from all layers
$num\_layers = 1$
$k = \lfloor n/\sigma \rfloor$    ▷ $k$: number of clusters
$\mathbf{X}' = \mathbf{X}$    ▷ $\mathbf{X}$: original points without labels
while $k \geq min_k$ do
    $kmeans(\mathbf{X}', k) \rightarrow \mathbf{X}_k, B_1, \ldots, B_k$    ▷ $\mathbf{X}_k$: centroids, $B_1, \ldots, B_k$: clusters
    $\forall \mathbf{x}_i \in \mathbf{X} : p_i = p_i + \frac{|\{\mathbf{x}_j \in \mathbf{X} \,|\, \mathbf{x}_j, \mathbf{x}_i \in B_r \wedge y_j = y_i\}|}{|\{\mathbf{x}_j \in \mathbf{X} \,|\, \mathbf{x}_j \in B_r\}|}$    ▷ proportion of class $y_i$ in $B_r$
            $\widehat{\mathbf{H}}_m[i, num\_layers] = 1 - p_i/num\_layers$
    $\mathbf{X}' = \mathbf{X}_k$    ▷ New data $\mathbf{X}'$ are centroids
    $k = \lfloor k/\sigma \rfloor$    ▷ Update $k$
    $num\_layers += 1$
end while
$last\_layer = num\_layers - 1$
$\widehat{\mathbf{h}} = \widehat{\mathbf{H}}_m[, last\_layer]$    ▷ *Hostility measure* for all points
$\widehat{\mathbf{H}}_m\_binarized = I(\widehat{\mathbf{H}}_m \geq \delta)$    ▷ Binarization for all layers
$\widehat{H}_{-1} = mean(\widehat{\mathbf{H}}_m\_binarized[\mathbf{Y} = -1, last\_layer])$    ▷ *Hostility measure* for class -1
$\widehat{H}_{+1} = mean(\widehat{\mathbf{H}}_m\_binarized[\mathbf{Y} = +1, last\_layer])$    ▷ *Hostility measure* for class +1
$\widehat{H}_D = mean(\widehat{\mathbf{H}}_m\_binarized[, last\_layer])$    ▷ *Hostility measure* for dataset level
```

on the initialization and cannot form non-convex shapes [9]. Nevertheless, the *hostility measure* overcomes these problems thanks to its initialization with a high value for $k$ to maximize the number of layers and, consequently, the resulting information to combine. When the number of clusters $k$ is high, the chance of having a bad initialization decreases. Also, in the first layer, the method starts with a local perspective and the quantity of points per cluster is small which saves the problem of the inability to form non-convex clusters. As for the rest of the layers, thanks to tracking it is possible to detect when the behavior of the partition becomes different and, thus, to keep the previous results.

For the sake of simplicity and clarity, the method has been expounded for the binary case but its calculation for multi-class problems is straightforward. In fact, in the multi-class case, more information can be extracted: the estimated *hostility* that every single class received from the rest of classes and the estimated *hostility* that a class received from a specific class or a group of classes.

## 5 Experiments

This section is devoted to evaluate the *hostility measure* through a variety of experiments involving artificial data and benchmark real datasets. The section begins with the description of the datasets and the selection of parameters for the rest of experiments. Later, the performance of the proposed measure is analyzed and compared with the state-of-the-art measures. Since a multi-level approach is presented, these experiments are divided into instance, class, and dataset levels. After this, two more experiments are presented highlighting other abilities of the *hostility measure*: an experiment showing the explanatory power of the *hostility* and overlapping tracking graphs derived from the method as well as the extension of the proposal to multi-class problems. The section ends with the lessons learned throughout the experiments. Notice that all the results from this section related to *hostility* come from the *hostility measure* which estimates the formal concept of *hostility*. However, for the sake of simplicity, both terms will be used interchangeably.

### 5.1 Set up

A total of 27 datasets have been considered: 11 are artificial datasets specifically created to assess the behavior of the *hostility measure* and the remaining 16 are binary real datasets from [8, 35].[2]

[2] The real data and the generator code of the artificial data can be found at https://github.com/URJCDSLab/Hostility_measure/tree/main/Data.

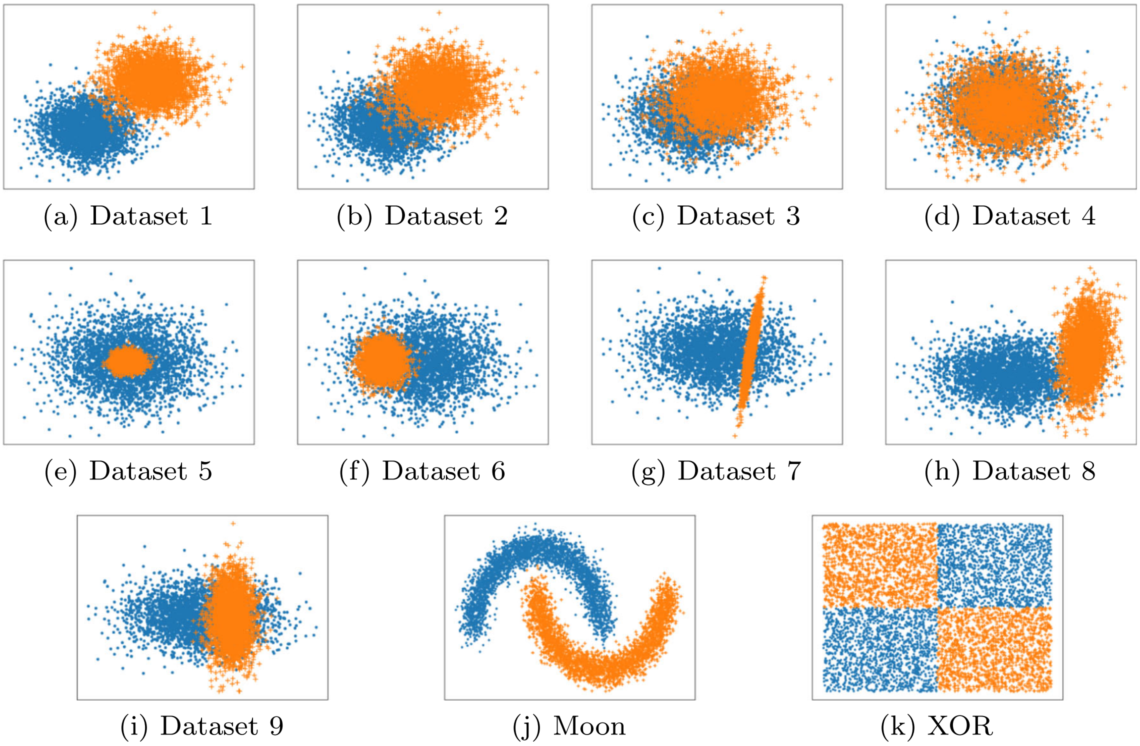

(a) Dataset 1
(b) Dataset 2
(c) Dataset 3
(d) Dataset 4
(e) Dataset 5
(f) Dataset 6
(g) Dataset 7
(h) Dataset 8
(i) Dataset 9
(j) Moon
(k) XOR

**Fig. 2** Artificial datasets. The class $-1$ is the blue one represented with $\cdot$ and the class $+1$ is the orange one represented with $+$

The 11 artificial datasets (see Fig. 2) are 9 sets of Normal distributions, the moon dataset and the XOR dataset. Each of the simulated Normal datasets is formed by 2 bivariate Normal distributions with different degrees of overlap, density, and a variety of shapes. Datasets 1,2,3, and 4 are formed by classes with equal variance and with an increasing symmetric overlap between classes. Datasets 5,6,7,8, and 9 present different dispersion for each pair of classes and various asymmetric types of overlap. In these last datasets, there is a sparse class that is always less overlapped and a denser class that can be fully or partially concentrated inside the sparse class. For all cases, each class has 4500 points.

The artificial data have been mainly generated using the Normal distribution so as to have a theoretical overlap reference value. Given two Normal probability density functions $f(x)$ and $g(x)$, their overlap [38] is defined as:

$$overlap(f, g) = \int_{-\infty}^{\infty} \min\{f(x), g(x)\}dx. \tag{10}$$

Table 2 contains the overlap of artificial datasets. For Moon and XOR datasets, the overlap is 0.

The main features of the 16 real datasets are presented in Table 3. Notice that the Wine and Yeast datasets are originally multi-class problems but the two more balanced classes have been chosen. Besides, as a reference of the complexity of each real and artificial dataset, a set of *ML* algorithms have been considered: *SVM* with linear kernel, *SVM* with *RBF* kernel, *Random Forest* (*RF*), *MLP*, *XGBoost*, *kNN*, *DT* and *LR*. The respective parameters have been selected through a 5-fold cross validation and a grid search maximizing the balanced accuracy. For the artificial datasets, 6000 points are destined to training and 3000 to testing. The real datasets are split into training (70%) and testing (30%). Finally, the best model for each dataset is the one maximizing the balanced accuracy while avoiding overfitting, that is, the model matching $\max(Test\ Balanced\ Accuracy\ -$

**Table 2** Artificial datasets: overlap, best classifier and corresponding error

| Dataset | Overlap | Best classifier | Error |
|---|---|---|---|
| 1 | 0.034 | *kNN* | 0.020 |
| 2 | 0.157 | *Multiple Layer Perceptron* (*MLP*) | 0.084 |
| 3 | 0.479 | *Logistic Regression* (*LR*) | 0.241 |
| 4 | 1.000 | *DT* | 0.495 |
| 5 | 0.160 | *MLP* | 0.085 |
| 6 | 0.227 | *Support Vector Machine* (*SVM*) *RBF* | 0.124 |
| 7 | 0.053 | *MLP* | 0.039 |
| 8 | 0.077 | *Linear SVM* | 0.034 |
| 9 | 0.360 | *MLP* | 0.184 |
| Moon | 0.000 | *MLP*, *XGBoost*, *kNN* | 0.001 |
| XOR | 0.000 | *SVM RBF* | 0.005 |

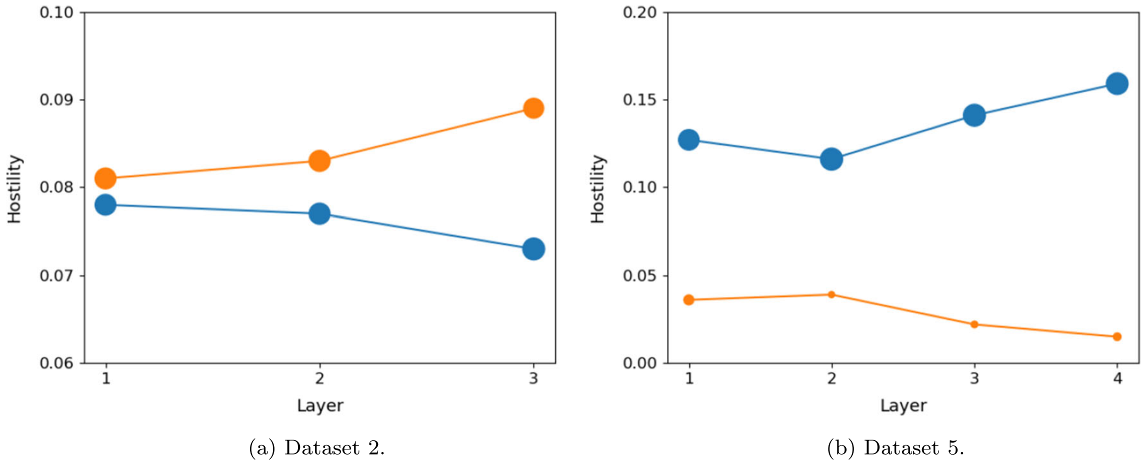
**Table 3** Real datasets: characteristics, best classifier and corresponding error

| Dataset | Instances | Features | Best classifier | Error |
|---|---|---|---|---|
| Bands | 365 | 20 | *LR* | 0.390 |
| Banknote | 1372 | 5 | *SVM RBF* | 0.000 |
| Bupa | 345 | 7 | *LR* | 0.367 |
| Haberman | 306 | 4 | *SVM RBF* | 0.350 |
| Hill Valley without noise | 606 | 101 | *LR* | 0.361 |
| Ionosphere | 351 | 34 | *SVM RBF* | 0.041 |
| Magic | 19020 | 11 | *MLP* | 0.141 |
| Mammographic | 830 | 6 | *LR* | 0.166 |
| Phoneme | 5404 | 6 | *XGBoost* | 0.148 |
| Pima | 768 | 9 | *MLP* | 0.251 |
| Sonar | 208 | 61 | *SVM RBF* | 0.101 |
| Spambase | 4597 | 58 | *SVM RBF* | 0.065 |
| WDBC | 569 | 31 | *kNN* | 0.047 |
| Wine (WineQualityRed 5vs6) | 1319 | 12 | *LR* | 0.309 |
| Wisconsin | 683 | 10 | *SVM RBF* | 0.030 |
| Yeast (Yeast CYTvsNUC) | 892 | 9 | *MLP* | 0.340 |

$\max(0, Train\ Balanced\ Accuracy - (Test\ Balanced\ Accuracy)))$.

In all cases, complexity measures are only applied to the training set [14]. All results, except the classification error, are computed on the training set and, for all experiments, the datasets are previously standardized. Tables 2 and 3 contain the best model for each dataset and its corresponding test error. Notice that the error is, in all cases, $1 - Balanced\ Accuracy$.

Regarding the parameters of the *hostility measure*, the selection of $k_{min}$, $\sigma$ and $\delta$ is required. The parameter $\delta$ is a threshold for a probability vector and, as such, is fixed to 0.5 as the default value for class probabilities. For the $\sigma$ parameter, values between 4 and 8 are recommended by the authors. Smaller $\sigma$ values are discarded because they are not able to capture the data structure in the first layer. On the other hand, higher values minimize the number of layers and can lose the data structure in intermediate and last layers due to the high number of clusters that they assemble. To maximize the number of layers, lower $\sigma$ values are preferred in general. For large datasets, higher $\sigma$ values can be used to save computational cost. Throughout the paper, results for these $\sigma$ values in $\{4, 5, 6, 7, 8\}$ will be shown to point out their validity. The parameter $k_{min}$ is, by default, the number of classes but it can be selected using the *hostility* tracking graph. As an example, the *hostility*



(a) Dataset 2.

(b) Dataset 5.

**Fig. 3** *Hostility* tracking graph. The negative class is represented in blue and the positive one in orange. For each layer and each class, the size of each dot indicates the proportion of clusters in which each class is the majority one. The *hostility measure* is obtained with $\sigma = 8$ for the (a) dataset 2 and $\sigma = 5$ for the (b) dataset 5

tracking graphs for the datasets 2 and 5 are displayed in Fig. 3. The $\sigma$ values are equal to 8 and 5, respectively. The size of the dots of the graph represents, for every layer and every class, the proportion of clusters in which the class is the majority class. Figure 3a reveals that both classes have a similar low *hostility* caused by the opposite class. This behavior is maintained across the three layers as the steady *hostility* values per class reflect. Note that in the last layer, there is a slight change in the *hostility* patterns. Therefore, the best layer to stop could be the layer 2 or 3. Moreover, in each layer, the size of the dots for both classes is similar, which means that they are the most representative class in a similar number of clusters. In Fig. 3b, the class $-1$ clearly has more *hostility* than the class $+1$. The trend of *hostilities* change from the layer 2 ($k = 240$) and starts to widen. Hence, the best layer to stop is the layer 2 to avoid instability. Regarding the dot sizes, the negative class is the most representative in most of the clusters through all layers. Given the low *hostility* of the class $+1$, this also reflects that it is less sparse than the class $-1$. To ease and automatize the user work, the rest of the experiments are all obtained following the next criterion: selecting the last layer that offers *hostilities* per class that do not vary more than 25% from the *hostility* results from the first layer.

Concerning measures from the state-of-the-art, the parameters have been chosen according to authors' recommendations: for *kDN*, $k = 5$ following [1, 33] and *R-value* is obtained with $k = 7$ and $\theta = 3$ following [28]. In the case of *CL* and *CLD* the Gaussian kernel density estimation is selected. The C4.5 algorithm from RWeka [16] is used to calculate the *hardness measures* based on C4.5 *DT*. For *DS* and *TDU*, parameters are chosen to avoid pruning and with a minimum number of instances per node equal to 1. For *DCP* and *TDP*, default parameters are taken. In particular, the complexity measures $F1$, $F1v$, $F2$, $F3$, $F4$, $N1$, $N2$, $N3$, $N4$, $T1$, $LSC$, $Density$, $ClsCoef$, $Hubs$, $L1$, $L2$, $L3$, $T2$, $T3$, $C1$, and $C2$ are obtained from the R package 'ECoL' [10].

Moreover, for all experiments, all complexity measures have been correspondingly re-scaled to behave accordingly to the error, i.e., lower values imply simpler instances and higher values more complex instances.

## 5.2 Instance level

This subsection is dedicated to the instance perspective of complexity measures. It is, in turn, divided into two different experiments. First, a graphical study and comparison of the behavior of several complexity measures is presented. This experiment shows the relation among complexity values at the instance level and the predicted probabilities from the best classifier. The second experiment aims to verify if each complexity measure is actually able to identify the most complex points. The complexity measures considered in this section are all the measures covering the instance level in Table 1.

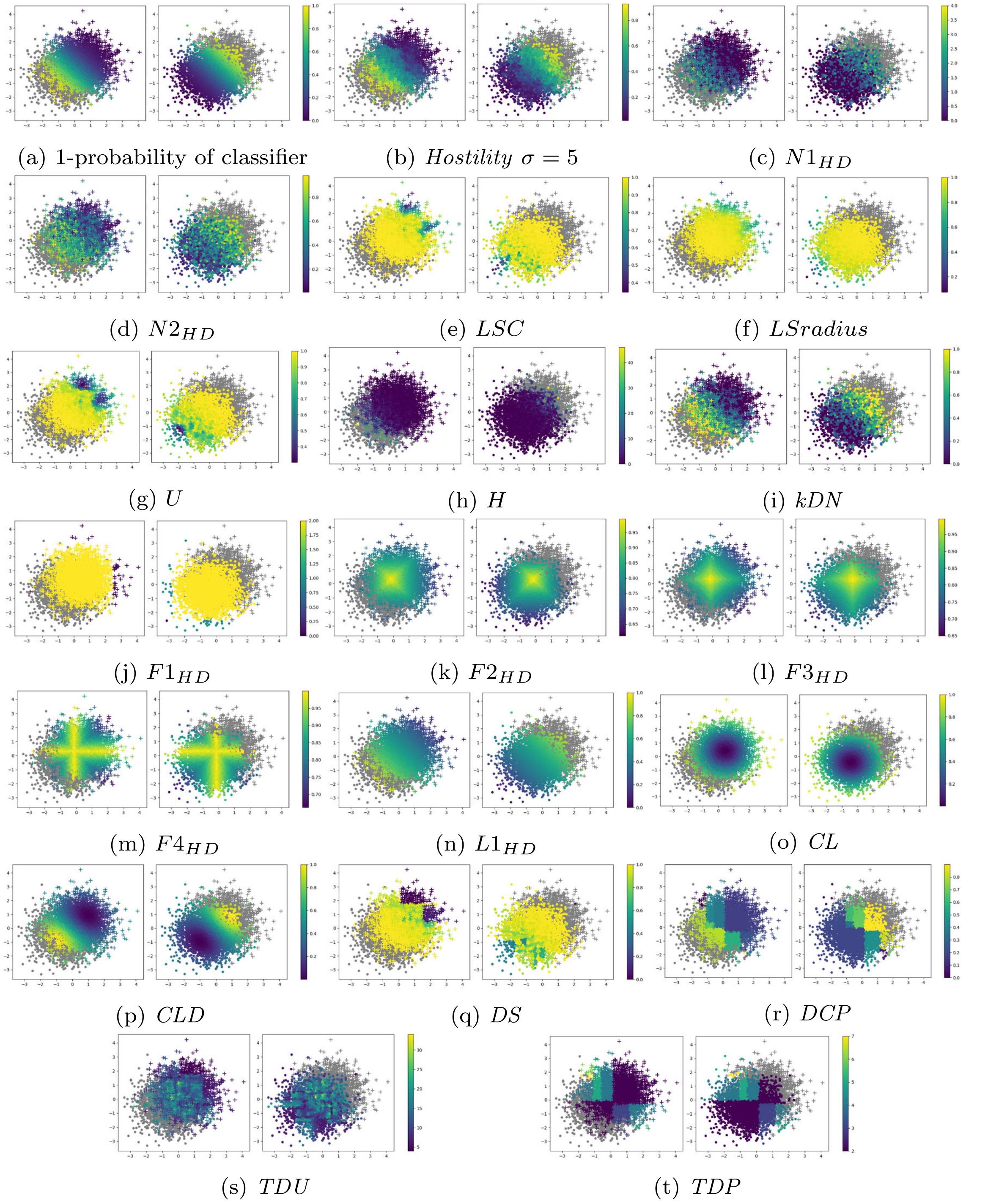
In this experiment, the predicted probabilities offered by the best classifier of each point belonging to its correct class are obtained following the cross validation scheme in [33]. The opposite of these predicted probabilities serve as a complexity reference for all instances.

### 5.2.1 Graphical analysis and correlation with classification error

For this experiment, the datasets 3 and 6 have been chosen[3] as a representative sample of the artificial datasets: two Normal distributions with similar density and overlap and other two with a great difference in density and the positive class fully inside the negative one. Figures 4 and 5 contain the complexity measures values for instances in the datasets 3 and 6, respectively. Since some classes are in overlap, a graph per class and per complexity measure is generated. As detailed before, it is also presented a graph with the predicted probability that each point has of belonging to the opposite class according to the corresponding best classifier. In the graphs, yellow colors imply more complexity and blue colors less.

As expected, the feature-based measures ($F1_{HD}$, $F2_{HD}$, $F3_{HD}$ and $F4_{HD}$) fail in the task of determining the difficulty of each point since they appraise overlapping perpendicular to axes. *CL* and *CLD* are calculated with a kernel density estimation using the Gaussian distribution. This assumption is reflected in the results. *CL* is informing about how far are points from the center of the distribution but not about its complexity. Even though *CLD* detects better the hardest instances, the captured complexity distribution is biased by the Gaussian assumption. The measures based on decision trees (*DS*, *DCP*, *TDP* and *TDU*) show sharp cuts associated with the hyperplanes generated by the trees instead of degraded complexity values. This behavior does not comprise the complexity distribution of points. Although *TDU* provides richer information not so characterized by hyperplanes, it considers that the overlapping region of the dataset 6 (see Fig. 5s) is equally complex for both classes even when the class $-1$ is clearly less present in the specific area (recall that both classes have the same number of samples). The linearity measure $L1_{HD}$ performs well for the dataset 3 (Fig. 4n) but fails for the dataset 6 that it is not linearly separable (Fig. 5n).

[3] The visual analysis for the rest of artificial datasets can be found at https://github.com/URJCDSLab/Hostility_measure/tree/main/Visual_analysis



(a) 1-probability of classifier

(b) *Hostility* $\sigma = 5$

(c) $N1_{HD}$

(d) $N2_{HD}$

(e) $LSC$

(f) $LSradius$

(g) $U$

(h) $H$

(i) $kDN$

(j) $F1_{HD}$

(k) $F2_{HD}$

(l) $F3_{HD}$

(m) $F4_{HD}$

(n) $L1_{HD}$

(o) $CL$

(p) $CLD$

(q) $DS$

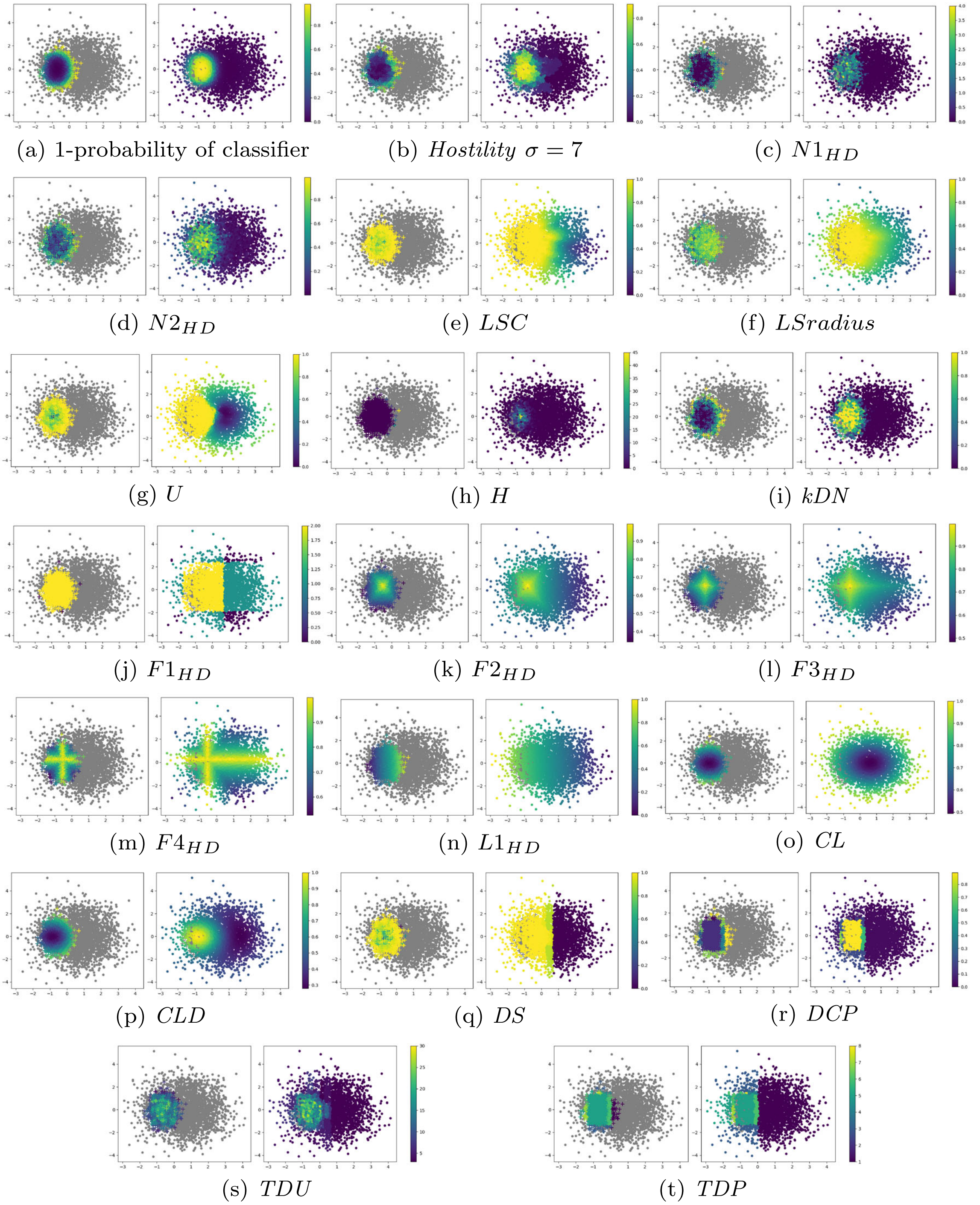
(r) $DCP$

(s) $TDU$

(t) $TDP$

**Fig. 4** Visual analysis of the complexity measures for the dataset 3 at the instance level. For each complexity measure, the left graph is for the class +1 and the right graph is for the class −1. Yellow colors indicates more complex instances

Concerning measures based on the local-set concept, the *harmfulness* index $H$ is the less informative since it only assigns high complexity values (yellow points) to points absolutely surrounded by points of other class (i.e., the point is the nearest enemy of all of them). Except for these few points, it considers similar low levels of complexity for the rest of the dataset points. $LSC$, $LSradius$ and the *usefulness* index $U$ perform better in detecting the most complex areas but, inside those areas, they do not generate a clear complexity degradation (in contrast with the classifier behavior in Figs. 4a and 5a). In addition, they overestimate the complexity and identify as complex some points that are not even in overlap (see Fig. 4a, f and g). The measures *kDN*, $N1_{HD}$ and $N2_{HD}$ reveal the best results among measures



(a) 1-probability of classifier (b) $Hostility\ \sigma = 7$ (c) $N1_{HD}$

(d) $N2_{HD}$ (e) $LSC$ (f) $LSradius$

(g) $U$ (h) $H$ (i) $kDN$

(j) $F1_{HD}$ (k) $F2_{HD}$ (l) $F3_{HD}$

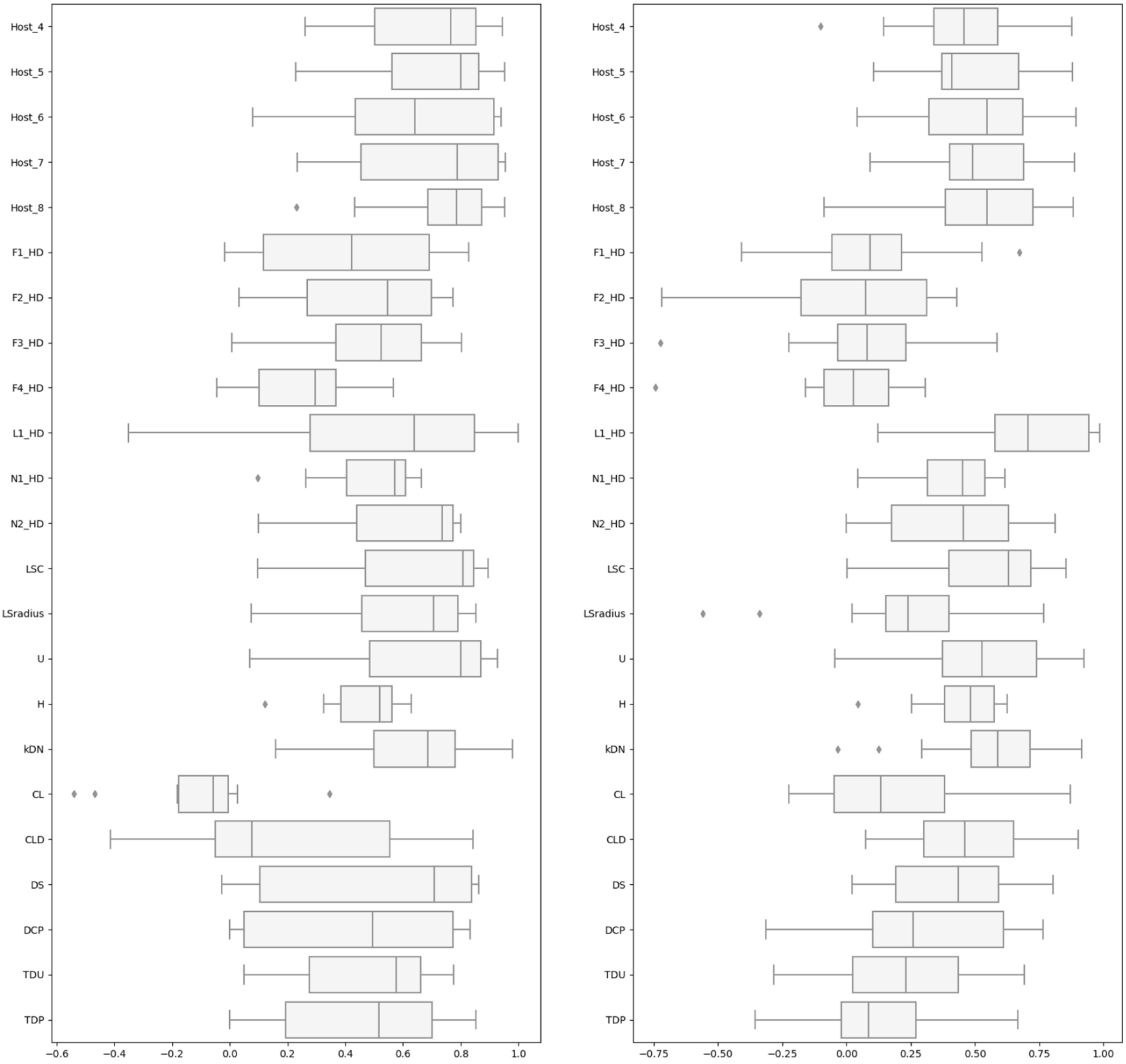
(m) $F4_{HD}$ (n) $L1_{HD}$ (o) $CL$

(p) $CLD$ (q) $DS$ (r) $DCP$

(s) $TDU$ (t) $TDP$

**Fig. 5** Visual analysis of the complexity measures for the dataset 6 at the instance level. For each complexity measure, the left graph is for the class $+1$ and the right graph is for the class $-1$. Yellow colors indicates more complex instances

from the state-of-the-art. They capture the distribution of the complexity for the two datasets and reflect the same patterns as the classifier. Nevertheless, none of them accomplishes a smooth complexity degradation as the classifier. The only measure that achieves it is the here proposed *hostility measure* thanks to its construction. The combination of information from different layers produces richer results. Also, as the number of clusters is smaller in each layer, their size is larger and this enables to study the points and the distribution of classes from a local to a global perspective. For both datasets, the *hostility measure* is the complexity measure that visually most closely resembles the results from the classifier (see Fig. 4a and b for the dataset 3 and Fig. 5a and b for the dataset 6). Points receiving more



(a) Artificial data

(b) Real data

**Fig. 6** Boxplots of the Spearman correlations, at the instance level, between each complexity measure and the predicted probabilities from the best model for the artificial (a) and real datasets (b). Axis $y$ shows the complexity measures and axis $x$ is the correlation. The numbers accompanying the *hostility measure* are the $\sigma$ value

*hostility* (yellow) are those surrounded by points from the other class since they lie in areas where its class is not dominant and will be easily mixed up with the opposite class. Medium *hostility* values (green-light blue) are in the boundary between classes, where both classes are equally present so they harm each other in a balanced way. Low values (dark blue) are for points placed in areas where its class is the dominant one.

To evaluate analytically if the complexity values per instance are consistent with the predicted probabilities from the corresponding best classifier, the Spearman correlation is gauged between the complexity values of each measure at the instance level and those predicted probabilities of the classifier for each dataset. These correlations ease the comparison of the capacity of the complexity measures to correctly rank the points given their complexity.

**Table 4** The train errors of each real dataset when the 10% (first row) and the 50% (second row) more complex points are eliminated using each of the complexity measures. The train error achieved with the whole train set is shown as a reference

| Data | Train error | *Hostility* | *CLD* | $L1_{HD}$ | *LSC* | *U* | *kDN* |
|---|---|---|---|---|---|---|---|
| Bands | 0.301 | 0.258 | 0.250 | 0.218 | 0.255 | 0.233 | 0.244 |
| | | 0.107 | 0.092 | 0.000 | 0.109 | 0.127 | 0.006 |
| Bupa | 0.299 | 0.245 | 0.252 | 0.210 | 0.267 | 0.278 | 0.259 |
| | | 0.160 | 0.028 | 0.000 | 0.180 | 0.233 | 0.057 |
| Haberman | 0.275 | 0.043 | 0.165 | 0.201 | 0.000 | 0.000 | 0.078 |
| | | 0.000 | 0.000 | – | – | 0.009 | 0.010 |
| Hill Valley | 0.201 | 0.220 | 0.217 | 0.190 | 0.226 | 0.230 | 0.206 |
| | | 0.141 | – | 0.149 | 0.258 | 0.306 | 0.249 |
| Magic | 0.115 | 0.010 | 0.082 | 0.064 | 0.006 | 0.000 | 0.001 |
| | | 0.009 | 0.000 | 0.000 | 0.000 | 0.000 | – |
| Mammo. | 0.172 | 0.090 | 0.095 | 0.076 | 0.073 | 0.073 | 0.059 |
| | | 0.003 | 0.000 | 0.000 | 0.004 | 0.000 | 0.005 |
| Phoneme | 0.076 | 0.013 | 0.006 | 0.008 | 0.000 | 0.002 | 0.001 |
| | | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | – |
| Pima | 0.261 | 0.000 | 0.171 | 0.170 | 0.000 | 0.000 | 0.000 |
| | | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Wine | 0.281 | 0.207 | 0.212 | 0.202 | 0.215 | 0.213 | 0.215 |
| | | 0.081 | 0.002 | 0.000 | 0.079 | 0.090 | 0.088 |
| Yeast | 0.338 | 0.283 | 0.228 | 0.262 | 0.000 | 0.000 | 0.000 |
| | | 0.000 | 0.000 | 0.000 | 0.040 | 0.000 | 0.026 |

The *hostility measure* is obtained with $\sigma = 5$. The symbol '–' means there is not enough data to train a model

Then, to easily compare all results, a boxplot based on these correlations is generated per each complexity measure. High and positive Spearman correlations mean that the complexity measure is able to order the points adequately according to their complexity and matching the predicted probabilities from the best model. On the other hand, low and negative values imply that the complexity ranking established by the complexity measure behaves differently than the results from the best model. That is, there is no agreement about which points are more complex. Note that, for this experiment, the *hostility measure* is computed for $\sigma \in \{4, 5, 6, 7, 8\}$.

For the artificial data (see Fig. 6a), the measures revealing higher correlation with the classifier are the *hostility measure*, some of the local-set concept based measures like $LSC$, $LSradius$, and $U$ and also $N2_{HD}$ and *kDN*. In the case of the real datasets (see Fig. 6b), the outstanding measures are the *hostility measure*, $LSC$, $U$, *CLD*, *kDN* and $L1_{HD}$. The only measures keeping its behavior for both types of datasets are the *hostility measure*, $LSC$, $U$ and *kDN*. Taking into account the performance of $LSC$ and $U$ in the visual study, it can be concluded that the *hostility measure* and *kDN* are the two complexity measures performing better in estimating the complexity of each point.

### 5.2.2 Complexity points detection verification

The purpose of the current experiment is to prove that the instances pointed out as complex by complexity measures are indeed harder to classify. To that aim, the evolution of the train error when using all points and when filtering a proportion of the most complex ones is analyzed. In particular, two subsets of the train data are considered: the first subset removes the 10% of most complex points and the second subset the 50%. Since, in every subset, the samples are simpler according to the complexity measures, the error is expected to decrease.

In this experiment, results are shown for the 10 real datasets with higher classification error and for the highlighted measures in the former experiment: the *hostility measure*, *kDN*, $L1_{HD}$, $LSC$, $U$ and *CLD*.[4]

Table 4 reveals that, in general, all the considered measures are detecting the most complex points since they achieve an error reduction when filtering the 10% and the 50% of those points. Another common and expected pattern is that the more complex points are removed, the

[4] The results for the remaining complexity measures can be found at https://github.com/URJCDSLab/Hostility_measure/tree/main/Filter_experiment

lower the error. The case of Hill Valley is noteworthy: only the *hostility measure* and $L1_{HD}$ have managed to reduce the initial error. When retaining the 90% of the simplest points, the *hostility measure* slightly increases the error of the Hill Valley data set. Note that, due to its construction, when filtering the most complex points with the proposed measure, at the beginning, only noise points or outliers will be extracted. However, in an intermediate stage, the points lying in the most uncertainty areas will be removed, only remaining the simpler ones. The behavior of $LSC$ and *kDN* in the Yeast data set is also remarkable. They increase the error when training with the second and simpler subset. Despite this, in general, the six considered complexity measures correctly identify which points are harder to classify and they could be used to reduce the size of train data. Nevertheless, the *hostility measure*, $L1_{HD}$, and *CLD* outshine presenting a more stable performance.

### 5.3 Class level

This section of experiments is devoted to the approach of data complexity through the class level perspective. To assess the capacity of the proposal to estimate the complexity of each class, a comparison among the results of the *hostility measure* for each class with the best classifier's errors is addressed in this section. The comparison includes also the *R-value* since it offers the class perspective.

For the artificial data (Table 5), the *hostility measure* (with all the $\sigma$ values considered) and *R-value* show values similar to the error committed in both classes. Hence, both measures allow anticipating what to expect in the classification task. Similar results are found for the real data (Table 6). The two measures perform well in capturing the proportion of critical points that really harm the classifier. Nevertheless, *R-value* fails in determining which class is more difficult to classify for the Hill Valley dataset and the proposal fails for the Mammographic one. In this case, the *hostility* results are less stable for the different values of $\sigma$ due to the small size of some real datasets (see Table 3). In these cases, the recommended $\sigma$ among the possible options is the one maximizing the number of layers. For the small datasets, $\sigma$ should be 4 or 5.

These results are accompanied by the Spearman correlation among the error, the *hostility measure* and *R-value*, to evaluate if they are able to rank classes according to the real complexity. Results are presented in Table 7a for

**Table 5** The error, the *hostility measure* and *R-value* for the artificial data at the class level

| Data | Error | *Host.* 4 | *Host.* 5 | *Host.* 6 | *Host.* 7 | *Host.* 8 | *R-value* |
|---|---|---|---|---|---|---|---|
| D1 | 0.015 | 0.014 | 0.012 | 0.014 | 0.014 | 0.012 | 0.020 |
| | 0.025 | 0.019 | 0.021 | 0.021 | 0.025 | 0.024 | 0.025 |
| D2 | 0.079 | 0.066 | 0.083 | 0.065 | 0.091 | 0.073 | 0.087 |
| | 0.088 | 0.091 | 0.070 | 0.089 | 0.070 | 0.089 | 0.095 |
| D3 | 0.241 | 0.236 | 0.231 | 0.239 | 0.250 | 0.235 | 0.271 |
| | 0.241 | 0.204 | 0.211 | 0.225 | 0.215 | 0.220 | 0.277 |
| D4 | 0.393 | 0.327 | 0.324 | 0.356 | 0.352 | 0.385 | 0.509 |
| | 0.598 | 0.317 | 0.355 | 0.353 | 0.375 | 0.349 | 0.521 |
| D5 | 0.130 | 0.150 | 0.116 | 0.126 | 0.133 | 0.136 | 0.133 |
| | 0.041 | 0.032 | 0.039 | 0.034 | 0.035 | 0.031 | 0.045 |
| D6 | 0.196 | 0.181 | 0.168 | 0.169 | 0.177 | 0.170 | 0.177 |
| | 0.051 | 0.048 | 0.056 | 0.060 | 0.052 | 0.057 | 0.076 |
| D7 | 0.074 | 0.053 | 0.057 | 0.065 | 0.063 | 0.065 | 0.072 |
| | 0.005 | 0.009 | 0.009 | 0.013 | 0.012 | 0.010 | 0.005 |
| D8 | 0.055 | 0.059 | 0.060 | 0.059 | 0.056 | 0.063 | 0.064 |
| | 0.013 | 0.019 | 0.026 | 0.021 | 0.021 | 0.020 | 0.024 |
| D9 | 0.298 | 0.252 | 0.264 | 0.254 | 0.255 | 0.271 | 0.267 |
| | 0.069 | 0.122 | 0.107 | 0.088 | 0.108 | 0.070 | 0.136 |
| Moon | 0.001 | 0.001 | 0.001 | 0.001 | 0.000 | 0.000 | 0.001 |
| | 0.000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| XOR | 0.004 | 0.012 | 0.009 | 0.013 | 0.017 | 0.018 | 0.008 |
| | 0.007 | 0.016 | 0.012 | 0.015 | 0.016 | 0.017 | 0.012 |

For each dataset, the first row shows the results for the class $-1$ and the second one for the class $+1$. The numbers accompanying the *hostility measure* indicate the corresponding $\sigma$ value

**Table 6** The error, the *hostility measure* and *R-value* for the real data at the class level

| Data | Error | *Host.* 4 | *Host.* 5 | *Host.* 6 | *Host.* 7 | *Host.* 8 | *R-value* |
|---|---|---|---|---|---|---|---|
| Bands | 0.145 | 0.217 | 0.174 | 0.161 | 0.130 | 0.093 | 0.224 |
| | 0.634 | 0.532 | 0.511 | 0.457 | 0.468 | 0.606 | 0.553 |
| Bank. | 0.000 | 0.000 | 0.000 | 0.002 | 0.002 | 0.004 | 0.002 |
| | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Bupa | 0.500 | 0.386 | 0.436 | 0.505 | 0.386 | 0.545 | 0.505 |
| | 0.233 | 0.186 | 0.314 | 0.250 | 0.300 | 0.179 | 0.307 |
| Hab. | 0.074 | 0.076 | 0.051 | 0.121 | 0.089 | 0.115 | 0.146 |
| | 0.625 | 0.561 | 0.702 | 0.825 | 0.754 | 0.632 | 0.825 |
| Hill V. | 0.022 | 0.207 | 0.188 | 0.211 | 0.249 | 0.300 | 0.493 |
| | 0.700 | 0.498 | 0.536 | 0.564 | 0.545 | 0.507 | 0.488 |
| Iono. | 0.029 | 0.025 | 0.051 | 0.057 | 0.025 | 0.045 | 0.025 |
| | 0.053 | 0.136 | 0.125 | 0.136 | 0.261 | 0.205 | 0.420 |
| Magic | 0.057 | 0.077 | 0.074 | 0.068 | 0.074 | 0.079 | 0.057 |
| | 0.224 | 0.288 | 0.314 | 0.326 | 0.331 | 0.342 | 0.374 |
| Mammo. | 0.109 | 0.151 | 0.151 | 0.161 | 0.201 | 0.214 | 0.191 |
| | 0.223 | 0.128 | 0.124 | 0.142 | 0.145 | 0.131 | 0.216 |
| Phon. | 0.103 | 0.064 | 0.066 | 0.084 | 0.086 | 0.089 | 0.076 |
| | 0.193 | 0.218 | 0.240 | 0.226 | 0.262 | 0.276 | 0.260 |
| Pima | 0.120 | 0.163 | 0.123 | 0.189 | 0.146 | 0.137 | 0.197 |
| | 0.383 | 0.406 | 0.380 | 0.390 | 0.353 | 0.385 | 0.444 |
| Sonar | 0.173 | 0.191 | 0.221 | 0.132 | 0.132 | 0.250 | 0.324 |
| | 0.029 | 0.078 | 0.104 | 0.273 | 0.273 | 0.260 | 0.104 |
| Spamb. | 0.040 | 0.047 | 0.036 | 0.063 | 0.047 | 0.072 | 0.069 |
| | 0.090 | 0.160 | 0.200 | 0.164 | 0.192 | 0.200 | 0.153 |
| WDBC | 0.000 | 0.036 | 0.024 | 0.052 | 0.028 | 0.016 | 0.012 |
| | 0.094 | 0.061 | 0.027 | 0.054 | 0.068 | 0.108 | 0.068 |
| Wine | 0.275 | 0.195 | 0.197 | 0.182 | 0.222 | 0.216 | 0.302 |
| | 0.344 | 0.269 | 0.280 | 0.330 | 0.256 | 0.327 | 0.325 |
| Wisc. | 0.045 | 0.035 | 0.051 | 0.042 | 0.058 | 0.071 | 0.045 |
| | 0.014 | 0.066 | 0.084 | 0.066 | 0.066 | 0.096 | 0.102 |
| Yeast | 0.223 | 0.275 | 0.293 | 0.191 | 0.284 | 0.204 | 0.315 |
| | 0.457 | 0.290 | 0.423 | 0.380 | 0.393 | 0.450 | 0.463 |

For each dataset, the first row shows the results for the class $-1$ and the second one for the class $+1$. The numbers accompanying the *hostility measure* are the $\sigma$ value

the artificial datasets and in Table 7b for the real datasets. Both measures achieve correlations between 0.79 and 1. Hence, the *hostility measure* is able to correctly identify which class is harder and will need more attention during the classification task. Also, it offers interpretable values.

**Table 7** Spearman correlation at the class level of both the *hostility measure* and *R-value* with the error

| Measure | Error | Measure | Error |
|---|---|---|---|
| *Hostility* $\sigma = 4$ | 0.984 | *Hostility* $\sigma = 4$ | 0.872 |
| *Hostility* $\sigma = 5$ | 0.983 | *Hostility* $\sigma = 5$ | 0.879 |
| *Hostility* $\sigma = 6$ | 0.990 | *Hostility* $\sigma = 6$ | 0.826 |
| *Hostility* $\sigma = 7$ | 0.979 | *Hostility* $\sigma = 7$ | 0.823 |
| *Hostility* $\sigma = 8$ | 0.985 | *Hostility* $\sigma = 8$ | 0.789 |
| *R-value* | 0.978 | *R-value* | 0.792 |
| (a) Artificial data | | (b) Real data | |

## 5.4 Dataset level

The final experiments deal with the classic dataset perspective to study the performance of the *hostility measure* and to compare it with measures from the state-of-the-art.

Since complexity measures should be well-correlated with the classification error, the Spearman correlation among complexity measures and the error from best classifiers are presented in Table 8 for the artificial data and in Table 8b for the real data. The correlations with the theoretical overlap are also shown for the artificial case. All measures have been previously normalized so that a positive correlation with the error is expected. Both tables

**Table 8** Spearman correlation at the dataset level. In the artificial case, the correlation is gauged between the complexity measures and both the error and the theoretical overlap. For the real case, only for the error

| Measure | Error | Overlap | Measure | Error |
|---|---|---|---|---|
| *Hostility* $\sigma = 4$ | 0.991 | 0.998 | *Hostility* $\sigma = 4$ | 0.938 |
| *Hostility* $\sigma = 5$ | 0.991 | 0.998 | *Hostility* $\sigma = 5$ | 0.935 |
| *Hostility* $\sigma = 6$ | 0.991 | 0.998 | *Hostility* $\sigma = 6$ | 0.935 |
| *Hostility* $\sigma = 7$ | 0.991 | 0.998 | *Hostility* $\sigma = 7$ | 0.921 |
| *Hostility* $\sigma = 8$ | 0.989 | 0.993 | *Hostility* $\sigma = 8$ | 0.906 |
| $F1$ | 0.309 | 0.246 | $F1$ | 0.824 |
| $F1v$ | 0.409 | 0.337 | $F1v$ | 0.918 |
| $F2$ | 0.218 | 0.228 | $F2$ | −0.079 |
| $F3$ | 0.500 | 0.446 | $F3$ | 0.653 |
| $F4$ | 0.536 | 0.483 | $F4$ | 0.330 |
| $L1$ | 0.209 | 0.118 | $L1$ | 0.768 |
| $L2$ | 0.564 | 0.515 | $L2$ | 0.903 |
| $L3$ | 0.564 | 0.515 | $L3$ | 0.897 |
| $T2$ | – | – | $T2$ | 0.076 |
| $T3$ | – | – | $T3$ | −0.076 |
| $T4$ | – | – | $T4$ | 0.295 |
| $C1$ | – | – | $C1$ | −0.162 |
| $C2$ | – | – | $C2$ | −0.162 |
| $Density$ | 0.664 | 0.601 | $Density$ | 0.832 |
| $ClsCoef$ | 0.782 | 0.793 | $ClsCoef$ | 0.365 |
| $Hubs$ | 0.755 | 0.711 | $Hubs$ | −0.112 |
| *DS* | 0.918 | 0.920 | *DS* | 0.565 |
| *DCP* | 0.624 | 0.573 | *DCP* | 0.712 |
| *TDU* | 0.945 | 0.907 | *TDU* | 0.621 |
| *TDP* | −0.059 | −0.094 | *TDP* | −0.043 |
| *CL* | 0.291 | 0.232 | *CL* | −0.088 |
| *CLD* | 0.436 | 0.360 | *CLD* | 0.021 |
| $N1$ | 0.991 | 0.998 | $N1$ | 0.924 |
| $N2$ | 1.000 | 0.989 | $N2$ | 0.729 |
| $N3$ | 0.991 | 0.998 | $N3$ | 0.888 |
| $N4$ | 0.736 | 0.720 | $N4$ | 0.791 |
| $T1$ | −0.991 | −0.998 | $T1$ | −0.388 |
| $LSC$ | 0.927 | 0.902 | $LSC$ | 0.603 |
| *kDN* | 0.982 | 0.989 | *kDN* | 0.924 |
| *R-value* | 0.991 | 0.998 | *R-value* | 0.924 |
| (a) Artificial data | | | (b) Real data | |

The symbol “–” means that Spearman correlation cannot be computed due to constant values

reveal that the *hostility measure* and the other neighborhood measures are more correlated with the error and the overlap. For both the artificial and real datasets, the correlation values related to *hostility measure* are higher than 0.9. Other measures showing a similar behavior are: $N1$, $N3$, *kDN*, and *R-value* and, not so close, $N2$ and $N4$. On the one hand, $LSC$, *DS*, *TDU*, $ClsCoef$, and $Hubs$ have a satisfactory behavior for the artificial datasets but not for the real datasets. On the other hand, $F1$, $F1v$, $L1$, $L2$, $L3$, and $Density$ show high correlations only for real datasets. *DCP* presents medium-high correlation values and $F2$, $F3$, and $F4$ medium and low values revealing that they are not able to rank the datasets according to the error. Finally, $T1$, *TDP*, *CL*, and *CLD* have negative or really low correlations not matching at all the ranking of the error. Dimensionality measures and class imbalanced measures return the same value for the artificial datasets since they have the same number of instances per class and features. In the case of the real datasets, they have low and negative correlations.

**Table 9** The *hostility measure* values at the dataset level for the artificial and real datasets

| Dataset | Error | Overlap | *Hostility* |
|---|---|---|---|
| 1 | 0.020 | 0.034 | 0.018 |
| 2 | 0.084 | 0.157 | 0.081 |
| 3 | 0.241 | 0.479 | 0.227 |
| 4 | 0.495 | 1.000 | 0.367 |
| 5 | 0.085 | 0.160 | 0.084 |
| 6 | 0.124 | 0.227 | 0.113 |
| 7 | 0.039 | 0.053 | 0.037 |
| 8 | 0.034 | 0.077 | 0.041 |
| 9 | 0.184 | 0.360 | 0.171 |
| Moon | 0.001 | 0.000 | <0.001 |
| XOR | 0.005 | 0.000 | 0.018 |

(a) Artificial data. The values of the *hostility measure* are shown with $\sigma = 8$

| Dataset | Error | *Hostility* |
|---|---|---|
| Bands | 0.390 | 0.333 |
| Banknote | 0.000 | 0.000 |
| Bupa | 0.367 | 0.270 |
| Haberman | 0.350 | 0.206 |
| Hill Valley | 0.361 | 0.351 |
| Ionosphere | 0.041 | 0.065 |
| Magic | 0.141 | 0.151 |
| Mammogra. | 0.166 | 0.139 |
| Phoneme | 0.148 | 0.109 |
| Pima | 0.251 | 0.248 |
| Sonar | 0.101 | 0.131 |
| Spambase | 0.065 | 0.091 |
| WDBC | 0.047 | 0.045 |
| Wine | 0.309 | 0.231 |
| Wisconsin | 0.030 | 0.046 |
| Yeast | 0.340 | 0.282 |

(b) Real data. The values of the *hostility measure* are shown with $\sigma = 4$

To enable the comparison, the error, and the theoretical overlap for the artificial case, are also presented

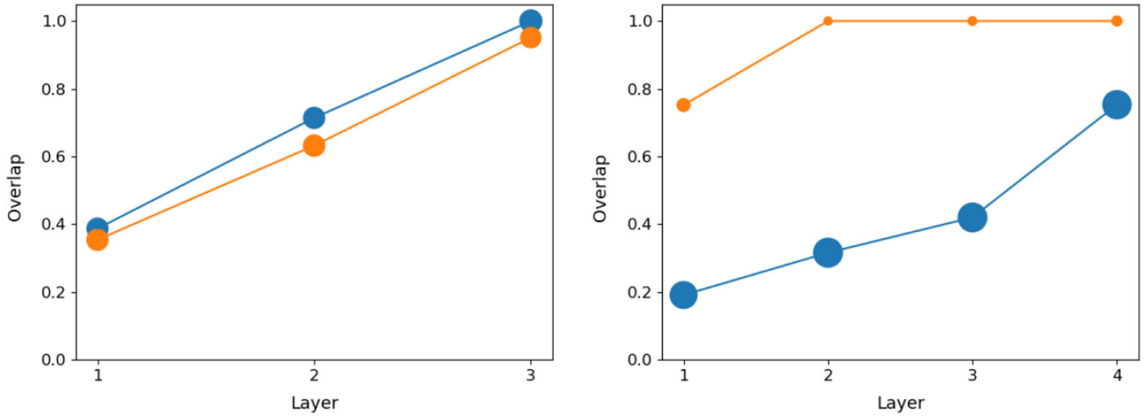
As a way to proof the clarity and explainability of the *hostility measure*, Table 9a and b show the dataset *hostility* value and the classification error for the artificial and real datasets, respectively. In this case, $\sigma = 8$ has been chosen for the artificial datasets due to the high number of instances (6000). For the real datasets, to maximize the number of layers, $\sigma = 4$ has been selected. Recall that, at the dataset level, the proposed measure is an estimation of the proportion of critical points. It is shown that it differs slightly from the error. That is, the *hostility measure* offers a good estimation of the critical points of a dataset.

## 5.5 Enrichment of the *hostility* tracking graph

The *hostility* tracking graph, besides its utility to select the best layer to stop, offers information about the interaction among classes. The *hostility* tracking graph of the dataset 2 (see Fig. 3a) showed a close and constant *hostility* behavior centered around 0.08. This means that the global complexity for the dataset is low and that both classes are equally harmed by the other class. There is an 8% of critical points in each class. In the case of the dataset 5, Fig. 3b presented a different situation. Until layer 2, the *hostility* of each class remains pretty steady, but from layer 3 both *hostilities* start to widen. This means that the behavior of the data when analyzed in small neighborhoods is not the same than in bigger ones. Moreover, the class $+1$ always shows lower *hostility* than the class $-1$. Hence, the class $-1$ is expected to be harder to classify.

Previous figures were obtained from the binarization of the *hostility measure* using the threshold $\delta = 0.5$. Furthermore, if it is binarized with $\delta > 0$, insights about the overlap of each class are achieved. With this binarization, 1 means that the point is in an overlapping area. Thus, any point that shares cluster with a point from other class is considered to be in overlap. In the first layer, this offers an estimation of the overlapping per class. As the number of clusters increases, the overlapping is obviously tending to 1. Despite this, tracking this overlapping estimation provides information about the density of the classes and how they interact. Besides, the dot sizes of both graphs inform about the distribution of the classes among the clusters of the different partitions and how this evolves across layers.

As an example, the overlapping tracking graph of the datasets 2 and 5 are presented in Fig. 7. In the case of the dataset 2, the same behavior in both classes across layers is detected. That is, as the number of clusters increases, a similar amount of points from both classes share cluster with points from the other class. If this information is combined with the one from the *hostility* tracking graph (Fig. 3a) and the balanced size of the dots (similar distribution of both classes in the partitions), it is concluded that they have the same density, and they overlap in a symmetric way. The overlapping tracking graph of the dataset 5 is quite different: the class $+1$ begins with a really high value of overlap and quickly reaches the maximum (i.e., in every cluster with a point from the class $+1$ there is at least one point from the class $-1$) and the class $-1$



(a) Dataset 2. (b) Dataset 5.

**Fig. 7** Overlapping tracking graph. The negative class is represented in blue and the positive one in orange. For each layer and each class, the size of each dot indicates the proportion of clusters in which each class is the majority one. The overlapping is obtained with the *hostility measure* with $\sigma = 8$ for the dataset 2 and $\sigma = 5$ for the dataset 5

starts with a low overlap value and increases uniformly but keeping distance with the class $+1$. This pattern highlights an asymmetric overlap. The class $+1$ is totally in overlap but not the class $-1$. That is, the class $+1$ is fully (or almost fully) inside the class $-1$. This complements the information of the *hostility* tracking graph (Fig. 3b). The class $-1$ has more *hostility* than the class $+1$ but the class $+1$ is totally in overlap. The dot sizes reflect that the class $+1$ represents the majority class in few clusters but, given its low *hostility*, the class is well covered by these clusters. Notice that, in the first layer, the class $+1$ is the majority class in a bigger proportion of clusters than in the rest of layers. This means that when clusters are smaller, it is easier for the class $+1$ to be the majority one. Therefore, the class $+1$ is clearly denser and, even though it is totally in overlap, all its points are concentrated. This also explains why the behavior of the dataset 5 when analyzing more local clusters or more global clusters was different. Since one class is so dense, when the number of cluster starts to be low, it is eminently being grouped in the same cluster. Thus, as mostly all its class is grouped together and is the densest class, it has lower *hostility*. Consequently, all points from the sparse class that are grouped in the same cluster as the dense one have more *hostility*.

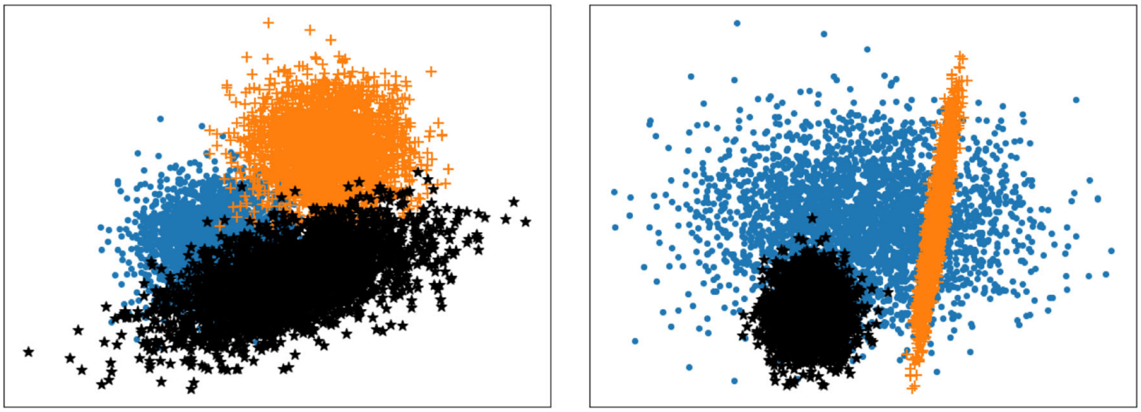
As a conclusion, these tracking graphs serve to provide rich information about classes and guide the user's next steps. For example, knowing that one class is entirely inside the other let the user to filter the non-overlapping ones and to focus the classification model in the complex areas. Another strong aspect is that these tracking graphs can be made for high dimensional data offering some exploratory insights for data hard to visualize.

### 5.6 *Hostility measure* for multi-class problems

The *hostility measure* is expounded for the binary case but its extension to multi-class is straightforward. In this experiment, two artificial datasets composed of 3 classes have been generated and presented in (Fig. 8).[5] The first one is similar to the dataset 1 with a new more dispersed class that overlaps with the two former ones. The second one is similar to the dataset 7 with a new dense class that only overlaps with the black one (originally the negative class). For both datasets, the total *hostility* per class and the *hostility* that each class presents due to each one of the other classes are contained in Table 10a and b, respectively. Regarding the first multi-class dataset, the *hostility* reflects that the class 2 is the one with more *hostility* (0.139). This *hostility* comes mostly from the class 0 (0.119) while the class 1 brings practically no disturb to it (0.018). Similarly, the most perturbing class to the class 0 is the class 2 (0.120). Concerning the class 1, its total *hostility* is 0.045 which is equally provoked by the classes 0 and 1 (0.021). The second multi-class dataset is specially interesting since the *hostility* results reveal that the classes 1 and 2 do not generate *hostility* towards each other. The class 0 is the one that has more *hostility* (0.122) and that is equally caused by the classes 1 and 2. The *hostility* of the classes 1 and 2 due to the class 0 is pretty low revealing that these two classes are better differentiated than the class 0.

Thus, the *hostility measure* is useful for multi-class problems since it shows how hostile is every class for

[5] Its generator code can also be found at https://github.com/URJCDSLab/Hostility_measure/tree/main/Data/Artificial_data



(a) Dataset multi-class 1.

(b) Dataset multi-class 2.

**Fig. 8** Multi-class data. The class 0 is represented in blue, the class 1 in orange and the class 2 is the black one

each of the other classes and the total *hostility* that a class is suffering. Thus, it let the user divide the data space according to its *hostility* and tackle every area with a different strategy.

### 5.7 Lessons learned

The main lessons learned from the proposal along the experiments are summarized in the current section. These lessons include from the great stability of the method and its good performance in all evaluated datasets to the validity of the recommended parameters.

Some of the alternative complexity measures have performed properly, offering really good results for some of these experiments. However, they have revealed poor performance in other cases. For instance, $L1$ is weak in estimating the global complexity but stands out in filtering the complex points. *kDN* performs well in general but in the graphical analysis showed closest points with opposite complexity values and was not so balanced when tackling artificial or real data. The *hostility measure* is clearly the most stable always offering good and satisfactory results for both artificial and real datasets. This is because the combination of layers and the use of $k$-means to analyze the points in their inherent groups, instead of in fixed structures, provides adaptability and robustness to the method.

It is remarkable that the *hostility measure*, using the *Euclidean* distance, has performed well in all the considered types of datasets. The datasets involved in the experiments have contemplated extreme scenarios regarding overlap, density, and decision boundary shape, including linear and non-linear data. Hence, the method is able to detect the interactions among classes independently of the linearity of the data and of the shape of the decision boundary.

The results of the *hostility measure* have also revealed the validity of the recommended $\sigma$ values: $\{4, 5, 6, 7, 8\}$. Taking into account the construction of the method, the validity of a range of values for the $\sigma$ parameters reflects that the *hostility measure* is capturing the data structure. Smaller and higher $\sigma$ values are discarded since they group few points in the first layer or too many points in the intermediate and final layers, respectively. Note that the weakest part of the *hostility measure* is the number of parameters. Despite the fact that this should be further revised by the authors, at the moment the selection of parameters has been quite simplified to the user.

**Table 10** The *hostility measure* values for the multi-class datasets

| | | Give | | | |
|---|---|---|---|---|---|
| | *Host.* | $C_0$ | $C_1$ | $C_2$ | Total |
| Receive | $C_0$ | - | 0.013 | 0.120 | 0.131 |
| | $C_1$ | 0.021 | - | 0.021 | 0.045 |
| | $C_2$ | 0.119 | 0.018 | - | 0.139 |

(a) Dataset multi-class 1.

| | | Give | | | |
|---|---|---|---|---|---|
| | *Host.* | $C_0$ | $C_1$ | $C_2$ | Total |
| Receive | $C_0$ | - | 0.060 | 0.068 | 0.122 |
| | $C_1$ | 0.014 | - | 0.000 | 0.014 |
| | $C_2$ | 0.020 | 0.000 | - | 0.023 |

(b) Dataset multi-class 2.

The tables contain the *hostility measure* between each pair of classes and the total value for each class. The *hostility measure* is obtained with $\sigma = 6$ for both datasets 1 and 2. $C_q, q = 0, 1, 2$ refers to each one of the classes

## 6 Research opportunities

The research ways that have appeared throughout the construction of this work are expounded in this section:

- **Finding the decision boundary.** Medium *hostility* values mean that the point resides in an uncertain area where both classes are equally present. This is normally the decision boundary and it could be detected by identifying those uncertain areas. One way of doing this is by translating the *hostility* in terms of uncertainty (for example, using Rényi's entropy) since its maximum value reflects the worst case, that is, when all classes are equally present. On the other hand, its minimum value is the best scenario in which there is only one class. This can be used for classification: applying weights to points depending on their uncertainty, dividing the data in different sets and building different classifiers for each one of them, etc.
- **Feature selection.** The *hostility measure* can be used to select the subset of features that minimizes the *hostility* of the dataset. This can also be tackled from the class level to discover which features are more harmful for each class.
- **Imbalanced data.** In the present work, the parameter $\delta$ has been set to 0.5 as the default threshold for probabilities. However, when classes are not balanced, this value might favor the majority class. The idea will be to set the $\delta$ parameter equal to the proportion of the minority class. The resulting *hostility measure* would allow the user to know whether the classification model used is harming the minority class or not. This version of the proposed measure could be compared with complexity measures specifically created for the imbalanced case that offer values per class [2, 26, 32].
- **The choice of metric.** In this paper, the *Euclidean* distance has been selected. It is worth analyzing the effect of the metric in the performance of the proposal and checking if the *hostility measure* with, for example, a Radial Basis Function Kernel achieves better results in the case of non-linear data.
- **Methods to estimate the concept of *hostility*.** The $k$-means algorithm performed hierarchically and recursively has been considered to estimate the *hostility*. However, alternative estimation methods like applying hierarchical clustering in every layer with an automatic selection of the number of clusters and selection of prototypes or using a dendrogram as the basis for the calculation of the concept of *hostility* could be also contemplated.
- **The applicability of the method.** As reflected in Section 2, complexity measures have been applied to different fields [3]. Besides the application of the *hostility measure* in those fields, it could also perform in different domains like, for example, Big Data, to check if the condensation of data is rich enough to substitute the original data [12]. Also, an adapted version of the *hostility* could be applied to obtain useful information to enrich the modeling phase in target-environment networks that arises in fields like genetics and economics [19].

## 7 Conclusions

In the present work, the concept of *hostility* has been introduced for the instance, the class, and the dataset level. The *hostility* is the damage, in terms of probability, that a point, a class or a dataset suffers from the presence of points of other classes in its surroundings when being classified. To estimate it, the *hostility measure*, a neighborhood measure offering a multi-level data complexity perspective, has been presented. The measure is constructed at the instance level by means of a hierarchical and recursive application of the $k$-means algorithm. After this procedure, an *hostility measure* value between 0 and 1 is obtained for every point. These values per point are aggregated to get an *hostility measure* value per class, indicating how hard is to identify each class, and for the whole dataset, illustrating the difficulty of separating the classes. As the method is hierarchical and recursive, the neighborhoods analyzed are of increasing size which allows the method to combine a local and a more global perspective.

To evaluate the proposed complexity measure, several experiments for each one of the perspectives have been carried out. In them, the performance of the *hostility measure* has been compared with complexity measures from the state-of-the-art. The *hostility measure* has generally stood out, showing a good and stable performance in all of them. It is easy to understand, to interpret and is suitable for binary and multi-class classification problems. Also, the proposal does not make assumptions about data nor it is based on a supervised classifier which ensures that there is no relation between results from the complexity measure and the posterior classification. In addition, to the best of the authors' knowledge, the *hostility measure* is the only complexity measure that offers a multi-level analysis of data complexity and combines different layers of information which increases the robustness of the method. Moreover, the complexity class perspective is deeply tackled in the present work. Not only an estimation of the complexity of each class is offered, but also two exploratory graphs (*hostility* and overlapping tracking graph) are presented providing information about the density and the relation between classes.

**Acknowledgments** This research has been supported by grants from Rey Juan Carlos University (Ref: C1PREDOC2020), Madrid Autonomous Community (Ref: IND2019/TIC-17194) and the Spanish Ministry of Science and Innovation, under the Retos-Investigación program: MODAS-IN (Ref: RTI-2018-094269-B-I00). We would like to thank Antonio Alonso Ayuso and Emilio López Cano from the Data Science Laboratory at Rey Juan Carlos University for the checking of mathematics.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

## References

1. Arruda JL, Prudêncio RB, Lorena AC (2020) Measuring instance hardness using data complexity measures. In: Brazilian conference on intelligent systems. Springer, pp 483–497
2. Barella VH, Garcia LP, de Souto MC, Lorena AC, de Carvalho AC (2021) Assessing the data complexity of imbalanced datasets. Inf Sci 553:83–109
3. Basu M, Ho TK (2006) Data complexity in pattern recognition. Springer Science & Business Media
4. Bernadó-Mansilla E, Ho TK (2005) Domain of competence of xcs classifier system in complexity measurement space. IEEE Trans Evol Comput 9(1):82–104
5. Brighton H, Mellish C (2002) Advances in instance selection for instance-based learning algorithms. Data Min Knowl Discov 6(2):153–172
6. Brun AL, Britto AS Jr, Oliveira LS, Enembreck F, Sabourin R (2018) A framework for dynamic classifier selection oriented by the classification problem difficulty. Pattern Recogn 76:175–190
7. Cai Z, Long Y, Shao L (2019) Classification complexity assessment for hyper-parameter optimization. Pattern Recogn Lett 125:396–403
8. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed 9 June 2022
9. Fahim A (2021) K and starting means for k-means algorithm. J Comput Sci 55:101445
10. Garcia L, Lorena A (2019) ECoL: complexity measures for supervised problems. https://CRAN.R-project.org/package=ECoL, r package version 0.3.0. Accessed 9 June 2022
11. Garcia LP, de Carvalho AC, Lorena AC (2015) Effect of label noise in the complexity of classification problems. Neurocomputing 160:108–119
12. Hariri RH, Fredericks EM, Bowers KM (2019) Uncertainty in big data analytics: survey, opportunities, and challenges. J Big Data 6(1):1–16
13. Ho TK, Baird HS (1998) Pattern classification with compact distribution maps. Comput Vis Image Underst 70(1):101–110
14. Ho TK, Basu M (2002) Complexity measures of supervised classification problems. IEEE Trans Pattern Anal Mach Intell 24(3):289–300
15. Hoekstra A, Duin RP (1996) On the nonlinearity of pattern classifiers. In: Proceedings of 13th international conference on pattern recognition, vol 4. IEEE, pp 271–275
16. Hornik K, Buchta C, Zeileis A (2009) Open-source machine learning: R meets Weka. Comput Stat 24(2):225–232. https://doi.org/10.1007/s00180-008-0119-7
17. Kaplansky I (2020) Set theory and metric spaces, vol 298. American Mathematical Society
18. Koziarski M (2021) Potential anchoring for imbalanced data classification. Pattern Recogn 120:108114
19. Kropat E, Weber GW, Tirkolaee EB (2020) Foundations of semialgebraic gene-environment networks. J Dyn Games 7(4):253
20. Lancho C, Martín de Diego I, Cuesta M, Aceña V, Moguerza JM (2021) A complexity measure for binary classification problems based on lost points. In: International conference on intelligent data engineering and automated learning. Springer, pp 137–146
21. Leyva E, González A, Perez R (2014) A set of complexity measures designed for applying meta-learning to instance selection. IEEE Trans Knowl Data Eng 27(2):354–367
22. Leyva E, González A, Pérez R (2015) Three new instance selection methods based on local sets: a comparative study with several approaches from a bi-objective perspective. Pattern Recogn 48(4):1523–1537
23. Lorena AC, Costa IG, Spolaôr N, De Souto MC (2012) Analysis of complexity indices for classification problems: cancer gene expression data. Neurocomputing 75(1):33–42
24. Lorena AC, Maciel AI, de Miranda PB, Costa IG, Prudêncio RB (2018) Data complexity meta-features for regression problems. Mach Learn 107(1):209–246
25. Lorena AC, Garcia LP, Lehmann J, Souto MC, Ho TK (2019) How complex is your classification problem? A survey on measuring classification complexity. ACM Comput Surv (CSUR) 52(5):1–34
26. Lu Y, Cheung YM, Tang YY (2019) Bayes imbalance impact index: a measure of class imbalanced data set for classification problem. IEEE Trans Neural Netw Learn Syst 31(9):3525–3539
27. Luengo J, Herrera F (2015) An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. Knowl Inf Syst 42(1):147–180
28. Oh S (2011) A new dataset evaluation method based on category overlap. Comput Biol Med 41(2):115–122
29. Orriols-Puig A, Macia N, Ho TK (2010) Documentation for the data complexity library in c++. Universitat Ramon Llull La Salle 196(1–40):12
30. Pascual-Triana JD, Charte D, Arroyo MA, Fernández A, Herrera F (2021) Revisiting data complexity metrics based on morphology for overlap and imbalance: snapshot, new overlap number of balls metrics and singular problems prospect. Knowl Inf Syst 1–29
31. Sáez JA, Galar M, Krawczyk B (2019) Addressing the overlapping data problem in classification using the one-vs-one decomposition strategy. IEEE Access 7:83396–83411
32. Singh D, Gosain A, Saha A (2020) Weighted k-nearest neighbor based data complexity metrics for imbalanced datasets. Stat Anal Data Min: the ASA Data Science Journal 13(4):394–404
33. Smith MR, Martinez T, Giraud-Carrier C (2014) An instance level analysis of data complexity. Mach Learn 95(2):225–256
34. Tanwani AK, Farooq M (2009) Classification potential vs. classification accuracy: a comprehensive study of evolutionary algorithms with biomedical datasets. In: Learning classifier systems. Springer, pp 127–144
35. Triguero I, González S, Moyano JM, García S, Alcalá-Fdez J, Luengo J, Fernández A, del Jesús MJ, Sánchez L, Herrera F (2017)

Keel 3.0: an open source software for multi-stage analysis in data mining. Int J Comput Intell Syst 10(1):1238–1249

36. Vuttipittayamongkol P, Elyan E (2020) Neighbourhood-based undersampling approach for handling imbalanced and overlapped data. Inf Sci 509:47–70
37. Wan S, Zhao Y, Wang T, Gu Z, Abbasi QH, Choo KKR (2019) Multi-dimensional data indexing and range query processing via voronoi diagram for internet of things. Futur Gener Comput Syst 91:382–391
38. Weitzman MS (1970) Measures of overlap of income distributions of white and Negro families in the United States, vol 22. US Bureau of the Census
39. Zhang X, Li R, Zhang B, Yang Y, Guo J, Ji X (2019) An instance-based learning recommendation algorithm of imbalance handling methods. Appl Math Comput 351:204–218

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Carmen Lancho** Bachelor of Mathematics and Statistics at the Complutense University of Madrid (UCM). Master in Statistical and Computational Data Processing at the UCM and at the Technical University of Madrid (UPM). She is currently a PhD candidate at the Rey Juan Carlos University (URJC) under a pre-doctoral grant. At the URJC, she is part of the Data Science Laboratory (DSLAB) research group. Her main research interests are complexity measures, classification of imbalanced data and clustering.

**Isaac Martín De Diego** Tenured Professor at the Rey Juan Carlos University (URJC). Coordinator of the high-performance research group, DSLAB: Foundations and Applications of Data Science at the URJC. His CV includes numerous publications related to research projects, technology transfer to the productive sector through patents and collaborative projects, extensive undergraduate and postgraduate teaching experience and experience in university management. His scientific production includes more than 50 articles indexed in JCR, more than 70 papers presented at national and international conferences, and two books for the dissemination of knowledge. Contributions in methods and techniques for data cleaning and cleaning, information representation, feature fusion from diferent information sources, new machine learning methods, novel metrics for evaluating artifcial intelligence models, explainability techniques and visualisation of learning model results. Intense activity of technology transfer to the private sector: healthcare, video surveillance, computer vision, cybersecurity, livestock, telecommunications, and energy.

**Marina Cuesta** is a PhD student in Information and Communication Technology at the Rey Juan Carlos University (URJC), for which she received a pre-doctoral grant. Since 2019, she is a data scientist researcher in the Data Science Laboratory (DSLAB) research group at the URJC. She graduated from the Bachelor's Degree in Mathematics and Statistics at the Complutense University of Madrid (UCM) in 2015. In 2016, she completed the Master's Degree in Statistical and Computational Data Processing at the UCM and the Technical University of Madrid (UPM). Her main interests in Data Science research includes visualization, dimensionality reduction techniques and explainable Machine Learning.

**Víctor Aceña** Bachelor of Mathematics at the National Distance Education University (UNED). Master in Computational Statistical Treatment of Information at the Complutense University of Madrid and at the Technical University of Madrid (UCM-UPM). Researcher at Data Science Laboratory (DSLAB) of the URJC. Data Scientist at MADOX VIAJES (JOF ASSOCIATES INT S.L.U.). His main area of work is the design of advanced sampling methodologies for fitting statistical models to dynamic and longitudinal data in the domain of incremental learning and ensemble learning.

**Javier M. Moguerza** PhD in Mathematical Engineering at University Carlos III of Madrid (UC3M). Full Professor at Rey Juan Carlos University. Previously he has worked at Carlos III University of Madrid and at Pontificia Comillas University of Madrid (ICAI-ICADE). His research interests are focused on Operations Research (Six Sigma Quality, Optimization of Resources), the design of Machine Learning methods and Data Science. He has been responsible for the Ericsson Institutional Chair on Data Science applied to 5G at Rey Juan Carlos University from September 2016 to September 2019. He has been an academician of the Global Young Academy (GYA) since December 2010 until May 2016, and currently he belongs to the Alumni of the Global Young Academy. He is founder academician of the Young Academy of Spain, created by the Spanish Government in 2019.