IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 24, 2021, accepted November 13, 2021, date of publication November 25, 2021,
date of current version December 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3131150

Adjacent Inputs With Different Labels and Hardness in Supervised Learning

SEBASTIÁN A. GRILLO[1], JULIO CÉSAR MELLO ROMÁN[1,2], JORGE DANIEL MELLO-ROMÁN[2], JOSÉ LUIS VÁZQUEZ NOGUERA[1], MIGUEL GARCÍA-TORRES[1,3], FEDERICO DIVINA[1,3], AND PEDRO ESTEBAN GARDEL SOTOMAYOR[1]

[1]Computer Engineering Department, Universidad Americana, Asunción 1029, Paraguay
[2]Facultad de Ciencias Exactas y Tecnológicas, Universidad Nacional de Concepción, Concepción 8700, Paraguay
[3]Division of Computer Science, Universidad Pablo de Olavide, 41013 Seville, Spain

Corresponding author: Sebastián A. Grillo (sebastian.grillo@ua.edu.py)

This work was supported by the Consejo Nacional de Ciencia y Tecnología, CONACYT, Paraguay, under Grant PINV18-1199.

ABSTRACT An important aspect of the design of effective machine learning algorithms is the complexity analysis of classification problems. In this paper, we propose a study aimed at determining the relation between the number of adjacent inputs with different labels and the required number of examples for the task of inducing a classification model. To this aim, we first quantified the adjacent inputs with different labels as a property, using a measure denoted as Neighbour Input Variation (NIV). We analyzed the relation that NIV has to random data and overfitting. We then demonstrated that a threshold of NIV may determine if a classification model can generalize to unseen data. We also presented a case study aimed at analyzing threshold neural networks and the required first hidden layer size in function of NIV. Finally, we performed experiments with five popular algorithms analyzing the relation between NIV and the classification error on problems with few dimensions. We conclude that functions whose similar inputs have different outputs with high probability, considerably reduce the generalization capacity of classification algorithms.

INDEX TERMS Classification, data complexity, machine learning, overfitting, supervised learning.

I. INTRODUCTION

Supervised learning is the task of mapping inputs to the corresponding output, where there is a previous set of input-output pairs given as examples. Supervised learning is highly successful in automating classification problems in the most diverse areas, from subatomic particle detection to melanoma diagnosis [1]–[6]. However, supervised learning is not a fully understood process, since its development requires a lot of empirical work [7].

On the other hand, Computational Learning Theory (CLT) is the field that studies the success of machine learning (ML) algorithms, and in particular of classification algorithms [8]. CLT offers several mathematical approaches to formalize classification problems [9]–[14] and represents an important advance, although with some limitations. For instance, there is a limited understanding of overfitting and generalization, for supervised classification models with a large number of parameters [15], [16]. A second problem is that supervised classification algorithms must deal with real problems, with little or no prior knowledge. Thus, the convergence between theory and practice is limited by the uncertainty from data [7].

An alternative strategy for analyzing classification algorithms is to perform experiments aimed at measuring the computational costs for learning data with some specific properties. There are several measures for experimental analysis of data complexity for classification algorithms [17]. Some data measures evaluate the influence of single variables in class separability, for example, Fisher's discrimination ratio, volume overlap region and attribute efficiency [18]. Other data measures consider the separability of classes, for instance, the minimal error by a hyperplane classification [19] or the distance between classes [20].

Other measures consider the geometry of manifolds spanned by classes, like the amount of space covering adherence subsets [17], non-linearity [21] or density [18], [20]. Finally, Kolmogorov Complexity is based on the minimum computer program that can replicate a given pattern. Kolmogorov Complexity has high theoretical importance, however it is difficult to calculate for real world problems [22], [23]. By using data complexity measures, we can define the capacity of models based on the complexity of the data that

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Wang.

they can learn. In this sense, analyzing data complexity can give alternatives to capacity measures for models, e.g. the Vapnik–Chervonenkis (VC) dimension [24], for assessing the effectiveness of classification algorithms.

In this paper, we study the particular problem of how different labels between neighboring inputs influence error in supervised classification problems. Considering that data is usually represented by bits, we model classification problems as finite discrete functions that are learned from a sample. From this formulation, the hypothesis is the following. For difficult classification problems, random inputs have neighboring inputs whose values vary with high probability. In other words, complex data is characterized by the fact that input value does not provide much information of the neighboring inputs. In this sense, we propose a data complexity measure called *Neighbour Input Variation* (NIV), for classification problems using categorical labels. NIV counts the times that neighboring inputs have a different value according to the function and divides such sum by the number of all possible inputs. From the study presented in this paper, we can highlight the following results. First, we show a strong relationship between high NIV values and a randomly generated function. However, we show that functions with regular patterns and high NIV can be found as well. That implies that high values of NIV in a learned function may be a signal of overfitting. Second, we show that classification error has a lower bound that has linear dependency on a given value V, under the condition that we only know that our classification problem has a NIV value less or equal to V. The interpretation is that low classification error occurs for high NIV, only if the problem has good prior knowledge. Third, we formulate an algorithm whose error has an upper bound, under the hypothesis that our classification problem has a NIV value less or equal to V. For these results, the error depends on the maximum number of inputs with different labels that we can find between two functions, where such functions have NIV values bound by V and their labels match with the classification on the sample.

We show that the expectation value of adjacent inputs with different outputs by the algorithm, is an alternative for measuring capacity in supervised learning models. As a case study, we consider feed-forward neural networks with threshold units. NIV shows to be a measure that allows straightforward results given its simplicity. We find a necessary and sufficient condition depending on the first hidden layer size, for computing any function with a fixed NIV value.

Finally, we present experiments showing that higher expectation of neighboring inputs with different value, tends to produce higher error in classification problems. For such experiments, we selected 2,3 and 4-dimensional problems. We apply K-Nearest Neighbours [25], K* [26], Random Forests [27], RIPPER algorithm [28], bagging [29] with RepTree [30] and artificial neural networks [31]. The experiments are compared by classification error.

Thus, this paper contributes towards a better understanding of the consequences of neighboring inputs with different values on classification problems. Such results are formulated by the NIV measure. We summarize our contributions as follows:

- A positive relation between noise and expectation of neighboring inputs with different values, in classification problems.
- A lower bound showing that a high expectation of neighboring inputs with different values implies a high classification error.
- An upper bound showing that a low expectation of neighboring inputs with different values implies a low classification error, for an appropriate classification algorithm.
- A tight bound of the number of first hidden layer units for computing a function f by a feed-forward neural network with threshold units, given the expectation of neighboring inputs with different values in f.
- Experimental evidence that a high expectation of neighboring inputs with different values implies high classification error.

This paper is structured as follows. Section II defines error on classification problems and the NIV measure. This section also describes the properties of the proposed measure. Section III analyzes feed-forward threshold neural networks in relation to NIV of data. Section IV presents experiments that relate NIV to classification error, while Section V discusses the relation of the studied property in relation to existing complexity measures. Finally, in Section VI we draw the main conclusions and identify possible future works.

II. MEASURING COMPLEXITY IN LABELED DATA

In this section, we start by introducing the notation needed in the following of the paper. Let $E = \{e_i\}$ be a finite sequence of real numbers, such that $e_{i+1} - e_i = \Delta > 0$ for all i, some constant Δ and $|E| = n$. We denote as $A = E^k$ a set of $k-$dimensional inputs, where each term from input belongs to E. For example, each term x_i can model a pixel from $x \in A$, which can be an image or intensity in a discrete-time signal. In classification problems, each input is assigned a label. We denote Y as the set of possible labels. Thereby, we model a classification problem as a function $f : A \to Y$, whose value is known in some set $S \subset A$ denoted as the sample. The results are restricted to classification problems satisfying the following hypothesis:

- The set Y has no order defined. Thus we only consider categorical classification problems.
- The classification problem f is defined for all input $x \in S$. We consider problems where we may find inputs with nonsense, which have a special label $\varnothing \in Y$.

A classification algorithm must infer the values of f for inputs outside S. However, the classification problem f must have some regularity or restriction, on its instances. If each instance is completely independent from the other, then the only way to know $f(x)$ is by querying x itself. In this sense, we define a hypothesis, which is a set of functions

TABLE 1. **Example for definition 1.**

Input	f	g	$\hat{f}$	$\tilde{f}$
0	1	1	1	1
1	-1	1	1	1
2	1	1	-1	-1
3	1	1	-1	1

H containing f. Such hypothesis H represents a previous knowledge about f analogous to the Probably Approximately Correct learning theory [32]. The following definition formalizes the notion of classification error, that we consider in this paper.

Definition 1: Let $\mathcal{C}$ be a classification algorithm which is trained on some sample S from A. Let H be a set of functions of the form $f : A \to Y$. We say that $\mathcal{C}$ has an error ε for H and S, if $\mathcal{C}$ outputs some function g with non-zero probability and there is some $f \in H$, such that all the following properties are satisfied:

1) *The function g is equal in S to f.*
2) *The function g differs in $\varepsilon\left(|A| - |S|\right)$ inputs from $A - S$ in relation to f.*
3) *There is no function in H satisfying item 1) that differs in more inputs in relation to g than f.*

For example, take $A = \{0, 1, 2, 3\}$, $g : A \to \{1, -1\}$, $f : A \to \{1, -1\}$, $\hat{f} : A \to \{1, -1\}$ and $\tilde{f} : A \to \{1, -1\}$; such that Table 1 summarizes the values of f, g, $\hat{f}$ and $\tilde{f}$ on A. Suppose that $H = \left\{f, \hat{f}, \tilde{f}\right\}$ and $S = \{0\} \subset A$. Let C be a classifier trained on a sample S on f that infers that the correct function must be g. Then C has an error $\frac{2}{3}$ over H, because g differs in two values from $A - S = \{1, 2, 3\}$ in relation to $\hat{f}$.

Notice that Definition 1 considers randomized algorithms, but $\mathcal{C}$ becomes a deterministic algorithm by taking a single function g with probability 1 from H. All information about f is given to $\mathcal{C}$ from S and H, which are restrictions to the candidate functions.

The following definition introduces the proposed measure for different labels between adjacent inputs, where function f models the classification problem. The measure depends on the values of f over pairs of adjacent inputs. We say that x and y are adjacent if $x_i = y_i$ for all i, except a unique k such that, if $x_k = e_j$ and $y_k = e_h$ then $|j - h| = 1$.

Definition 2: Let $f : A \to Y$ be a function. Consider the function $\delta : A \times A \to \{0, 1\}$, where $\delta(x, y) = 1$ if and only if, (i) x and y are adjacent and (ii) $f(x) \neq f(y)$. The Neighbour Input Variation (NIV) of f is defined as

$$\nu(f) = \frac{1}{2|A|} \sum_x \sum_y \delta(x, y). \tag{1}$$

We present a very simple example, with

$$A = \{0, 1, 2, 3\}$$

and

$$f : A^2 \to \{1, -1\};$$

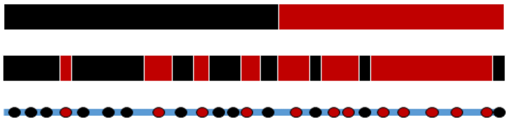


FIGURE 1. **The blue line with red and black dots represents a unidimensional domain with some labeled data, where each dot is red or black. The first bar represents a simple prediction that generalizes the labeled data. However, the second bar represents an over-fitting behavior.**

where we take

$$f(i, j) = -1^{(\lfloor i/2 \rfloor + \lfloor j/2 \rfloor)}.$$

Notice that there are eight pairs of adjacent inputs with different value in f, then $\delta([0, 1], [0, 2]) = \delta([1, 0], [2, 0]) = \delta([1, 1], [2, 1]) = \delta([1, 1], [1, 2]) = \delta([2, 1], [2, 2]) = \delta([1, 2], [2, 2]) = \delta([3, 1], [3, 2]) = \delta([1, 3], [2, 3]) = 1$. As $\delta(x, y) = \delta(y, x)$, then $\sum_x \sum_y \delta(x, y) = 16$ and $\nu(f) = 1/2$.

It is important to mention that both Definitions 1 and 2 represent quantities not intended to be estimated experimentally. As we will see later, these definitions are used to specify hypotheses in the analysis of algorithms or classification problems.

We can see that the NIV measure is the number of times that two adjacent points have different categories on f, divided by the size of the domain of f. The idea behind NIV is that a high number of adjacent inputs with different outputs is a signal of randomness and noise in data. In real classification problems, two adjacent inputs have similar values with high probability. Thus, methods preventing over-fitting tend to limit the difference between outputs in neighboring inputs [33], [34]. For example, Fig. 1 shows a classification problem over a line. The classification with over-fitting shows higher NIV than a classification generalizing on density. The following theorem shows that if we generate a random function we may expect a high NIV. This theorem implies that high NIV tends to occur when there is no restriction in the hypothesis.

Theorem 1: Suppose that we generate a random function $f : A \to Y$, by selecting a label from Y following a uniform distribution, for each input in A. Then

$$\mathbb{E}[\nu(f)] = \frac{k(n-1)(|Y|-1)}{n|Y|}. \tag{2}$$

Proof: By Definition 1 we have

$$\mathbb{E}[\nu(f)] = \frac{1}{2|A|} \sum_x \sum_y \mathbb{E}[\delta(x, y)]. \tag{3}$$

If x and y are adjacent then $\mathbb{E}[\delta(x, y)] = \frac{(|Y|-1)}{|Y|}$, otherwise $\mathbb{E}[\delta(x, y)] = 0$. Notice that each input has $2k$ potential neighbours and there are n^k inputs in total, however there are $2kn^{k-1}$ missing neighbours for inputs with extreme coordinates. Then, we have $2kn^{k-1}(n-1)$ permutations of adjacent pairs (x, y), and Equation (3) implies Equation (2). □

It is worth noticing that a high NIV does not imply noisy functions without regular predictable patterns, for example we have a function that generalizes the chessboard pattern. Let $p: E \to \{1, -1\}$ be a function such that $p(e_i) = (-1)^i$. We denote a function $\Phi: A \to \{0, 1\}$, defined as

$$\Phi(x) = \frac{1 + \prod_i p(x_i)}{2}. \tag{4}$$

Notice that Φ reaches the maximum NIV value for functions of domain A and range $\{0, 1\}$. The reason is that $\delta(x, y) = 1$ for all pair x, y of adjacent inputs, thus

$$\nu(\Phi) = \frac{k(n-1)}{n}.$$

The following theorem relates classification error with NIV. The theorem supposes that the only thing that we know about f is that its NIV cannot surpass a fixed value. That hypothesis H can be seen as a regularization where we limit the number of inputs that can have different outputs from their neighborhood. As we see in Fig. 1, overfitting can be associated with models of high NIV. That is because noisy data is 'memorized' by a classifier regardless of the values of the neighbors.

Theorem 2: Let $m, V > 0$. Let $\mathcal{C}$ be a classification algorithm trained on a sample S of size m from A. Denote the hypothesis H as the set of functions of the form $f: A \to Y$ with NIV equal or less than V. If $V \geq \frac{2km}{n^k}$ then all $\mathcal{C}$ has an error ε for f and S that satisfies

$$\varepsilon \geq \frac{1}{2}. \tag{5}$$

Proof: Let g be the function that $\mathcal{C}$ outputs given the sample S. The function f has the same values given by S as g, by definition. Notice that there are at most $2km$ elements from A that are adjacent to elements from S, but disjoint to S. Then the NIV value allows the construction of the following function g. We choose a unique value $y \in Y$ for all $x \in A - S$ in f, such that this class y has the lower cardinality over the set $A - S$ in g. This implies that at least $\frac{n^k}{2}$ inputs in $A - S$ have a different value for f and g. □

Consider the example for Definition 2. As $\nu(f) = \frac{1}{2}$, if we have a dataset that can be represented by any function with NIV equal or less than $\frac{1}{2}$ then we need a sample with more than 4 elements. Otherwise, the classification algorithm can be wrong at least half the time.

Theorem 2 presents $V \geq \frac{2km}{n^k}$ as a critical condition. We can see that if the condition is satisfied then V assigns enough freedom to g for ignoring the information from the sample. We say that g ignores S because the best algorithm just guarantees the values that we already know from S. Notice that equation (5) implies a high error because S is much smaller than A in practical situations. This theorem comes from that excessive randomness in datasets disables the prediction capacity of the sample. That is because randomness produces adjacent inputs with different labels with high probability, as Theorem 1 showed. The following theorem analyzes error in a distinct case when the sample is large enough in relation to the NIV value.

Theorem 3: Consider the same variables from Theorem 2. If $\frac{2km}{n^k} > V$ and n is multiple of $\sqrt[k]{m}$, there is a classification algorithm $\mathcal{C}$ that has an error ε for H and S, such that

$$\varepsilon \leq \frac{Vn^k}{km}. \tag{6}$$

Proof: We consider an algorithm that can choose its sample S. This algorithm considers a partition of A in m sets of the form $\Delta_{i_1} \times \Delta_{i_2} \times \ldots \times \Delta_{i_k}$, where

$$\Delta_i = \left\{ e_{(i\sqrt[k]{m}+j)} : 0 \leq j < \sqrt[k]{m} \right\}.$$

Thus $\mathcal{C}$ chooses a sample S such that there is an element $s \in S$ on each set of the partition and s is not adjacent to any element from other sets of the partition. Finally, suppose that $x \notin S$ and $s, x \in p$ for some set p of the partition, then $\mathcal{C}$ just assumes that $g(x) = g(s)$. Notice that (i) we cannot have more than Vn^k adjacent pairs of inputs with a different value in f and (ii) if $\mathcal{C}$ gives a wrong answer in some set of the partition, there are at least k adjacent pairs of inputs with a different value in such set. Then $\mathcal{C}$ can give a wrong answer in no more than Vn^k/k sets from the partition. We also have that each partition has $\left(\frac{n^k}{m} - 1\right)$ inputs whose class is unknown. Then $\left(\frac{n^k}{m} - 1\right) Vn^k/k$ is an upper bound for the inputs with wrong value by $\mathcal{C}$ and dividing by $\left(n^k - m\right)$ which is the total number of inputs outside the sample, we have equation (6). □

As example, take a generic function f whose inputs have a same value, excepting p inputs that are not adjacent between each other and whose coordinates do not take extreme values of A. Notice that $\nu(f) = \frac{2pk}{n^k}$ and choosing an appropriate $\sqrt[k]{m}$ that divides n, we need m satisfying $\frac{2km}{n^k} > \frac{2pk}{n^k}$ or $m > p$. Thus we upper bound error by $\varepsilon \leq \frac{p}{m}$, or in other words, if the dataset can be any function with NIV equal or less than $\frac{2pk}{n^k}$ then there is an algorithm that fails with probability no more than $\frac{2p}{m}$ on inputs outside S.

Choosing the sample is unusual in realistic situations. However, Theorem 3 can be interpreted as an error from a very well distributed sampling.

III. NIV AND THRESHOLD NEURAL NETWORKS

In this section, we propose an analysis of feed-forward neural networks using the NIV measure. Let us first introduce the following definitions. The *Heaviside step function* $th(x)$ is defined by $th(x) = 1$ for $x > 0$ and $th(x) = 0$ otherwise. If some unit applies the activation function $th(x)$, then such unit is a *threshold unit*. If a neural network only has threshold units, then it is a *threshold neural network*. The number of units in the first hidden layer of a feed-forward neural network is denoted as the *base*. The following theorem relates the base and NIV.

Theorem 4: Let $f: A \to Y$ be a function. Then a threshold neural network $\mathcal{N}$ with feed-forward architecture that computes f, has a base

$$b \geq \nu(f)\, n/k. \tag{7}$$

Proof: Let x, y be two adjacent inputs from A, such that $f(x) \neq f(y)$. The segment line $\overline{xy}$ is cut by a hyper-plane from a neural unit of the first hidden layer and such segment line is denoted as *critic*. Otherwise, x and y would have the same output in the first hidden layer, and they would be indistinguishable for $\mathcal{N}$. Therefore, for computing f by $\mathcal{N}$, all critic line segments are cut. The cardinality of the set of critic line segments is $\nu(f)\, n^k$. The hyper-plane defined by a first hidden layer unit can cut no more than kn^{k-1} line segments, therefore we have (7). □

Theorem 4 implies that limiting the base can be a regularization method for threshold neural networks. That is because a regularization method must limit NIV on the learned function for noisy data. The following theorem shows that there are functions such that equation (7) is an asymptotically tight bound.

Theorem 5: Let k be an even number. There is a threshold neural network $\mathcal{N}$ with base $k(n-1)$ that computes function Φ using two layers.

Proof: We denote (i) w^i_{jh} as the weight for unit j on layer $i-1$ to unit h on layer i and (ii) b^i_h as the bias of unit h on layer i, where input is considered layer 0. We divide the units of the first hidden layer in two groups of identical size. If (i) $0 \leq h < t = \frac{k(n-1)}{2}$ then $w^1_{jh} = 1$ and $b^1_h = -\Delta(2h+1) - e_0 k$, for $t \leq h < 2t$ $w^1_{jh} = -1$ and $b^1_h = \Delta 2(h-t) + \Delta + e_0 k$. Layer two has just one unit, then $w^2_{j1} = 1$ and $b^2_1 = \frac{1-k(n-1)}{2}$.

Notice that if

$$\sum_i x_i = L\Delta + e_0 k$$

for some natural number L. Then there are $\left\lfloor \frac{L}{2} \right\rfloor$ first hidden layer units with output 1, such that $h < t$ and there are $\frac{k(n-1)}{2} - \left\lceil \frac{L}{2} \right\rceil - 1$ first hidden layer units with output 1 such that $h \geq t$. Thus, there are $\left\lfloor \frac{L}{2} \right\rfloor + \frac{k(n-1)}{2} - \left\lceil \frac{L}{2} \right\rceil - 1$ units with value 1 in the layer one. This is equivalent to $\frac{k(n-1)}{2} - 1$ units with value 1 if L is even and $\frac{k(n-1)}{2} - 2$ units with value 1 if L is odd.

Take $w^2_{i1} = 1$ and $b^2_i = -\frac{k(n-1)}{2} - 3/2$ for all i. Considering that layer two is an output layer with a single neural unit, then if L is even then $\mathcal{N}$ outputs 1, otherwise $\mathcal{N}$ outputs 0. □

Notice that replacing $f = \Phi$ in equation (7) we find that Theorem 5 implies that $b = \nu(\Phi)\, n = n - 1$. This is asymptotically tight to Theorem 5 as n tends to infinity.

IV. EXPERIMENTAL EVALUATION OF CLASSIFICATION PROBLEMS DEPENDING ON NIV

This section presents experimental evidence that in classification problems, a higher NIV is related to a higher error. We analyzed the behavior of the error of five algorithms on classification problems, with different NIV values, but the same dimension. The problems have just 2, 3 and 4 attributes. However, the proposed analysis can be extended to any number of features. We chose a generalization of function Φ as rule for the classification problems. The function $\Phi^k_{n,h}: E^k \rightarrow \{0, 1\}$ is defined as

$$\Phi^k_{n,h}(x) = \frac{1 + \prod_i p_h(x_i)}{2}, \tag{8}$$

where $p_h(e_i) = (-1)^{\left\lfloor \frac{i}{h} \right\rfloor}$ and $|E| = n$. We chose this family of functions because their NIV is easily calculated. It is not difficult to prove that $\nu\left(\Phi^k_{n,h}\right) = k\frac{(n-h)}{hn}$.

We separated the classification problems into 3 groups depending on dimension k:

(i) Problems of two dimensions $\Phi^2_{60,20}$, $\Phi^2_{100,20}$, $\Phi^2_{40,10}$, $\Phi^2_{100,10}$, and $\Phi^2_{80,4}$;

(ii) Three dimensions $\Phi^3_{20,10}$, $\Phi^3_{21,7}$, $\Phi^3_{20,5}$, $\Phi^3_{16,4}$, and $\Phi^3_{18,3}$;

(iii) Four dimensions $\Phi^4_{10,5}$, $\Phi^4_{9,3}$, $\Phi^4_{8,2}$, $\Phi^4_{10,2}$, and $\Phi^4_{14,2}$.

For each function, we generated datasets that represent 3%, 6%, 9%, 12% and 15% of the domain for each function. The records in each dataset are unique and were selected uniformly from the function domain. The functions have different size domains depending on their parameters, thus the generated datasets have different sizes. For $\Phi^i_{j,k}$ we have a size domain of j^i. The selected functions are difficult to learn due to patterns that are not linearly separable. Thus, we selected supervised classification algorithms with good performance on complex patterns, but different classification approaches. In particular, we have selected K-Nearest Neighbours, K*, Random Forest, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Bagging applied on RepTree and Artificial Neural Networks. K-Nearest Neighbours is a classifier that assigns labels, depending on the distance of some instance to previously classified examples. K* is an instance-based classifier as KNN, however differs from other instance-based classifiers in the use of an entropy-based distance function. Random Forest is an ensemble approach based on decision trees. RIPPER is a rule learner that applies a divide-and-conquer strategy. RepTree is an algorithm from the family of decision trees. Finally, the artificial neural network applied consists of a single hidden layer of 10 ReLu units [35].

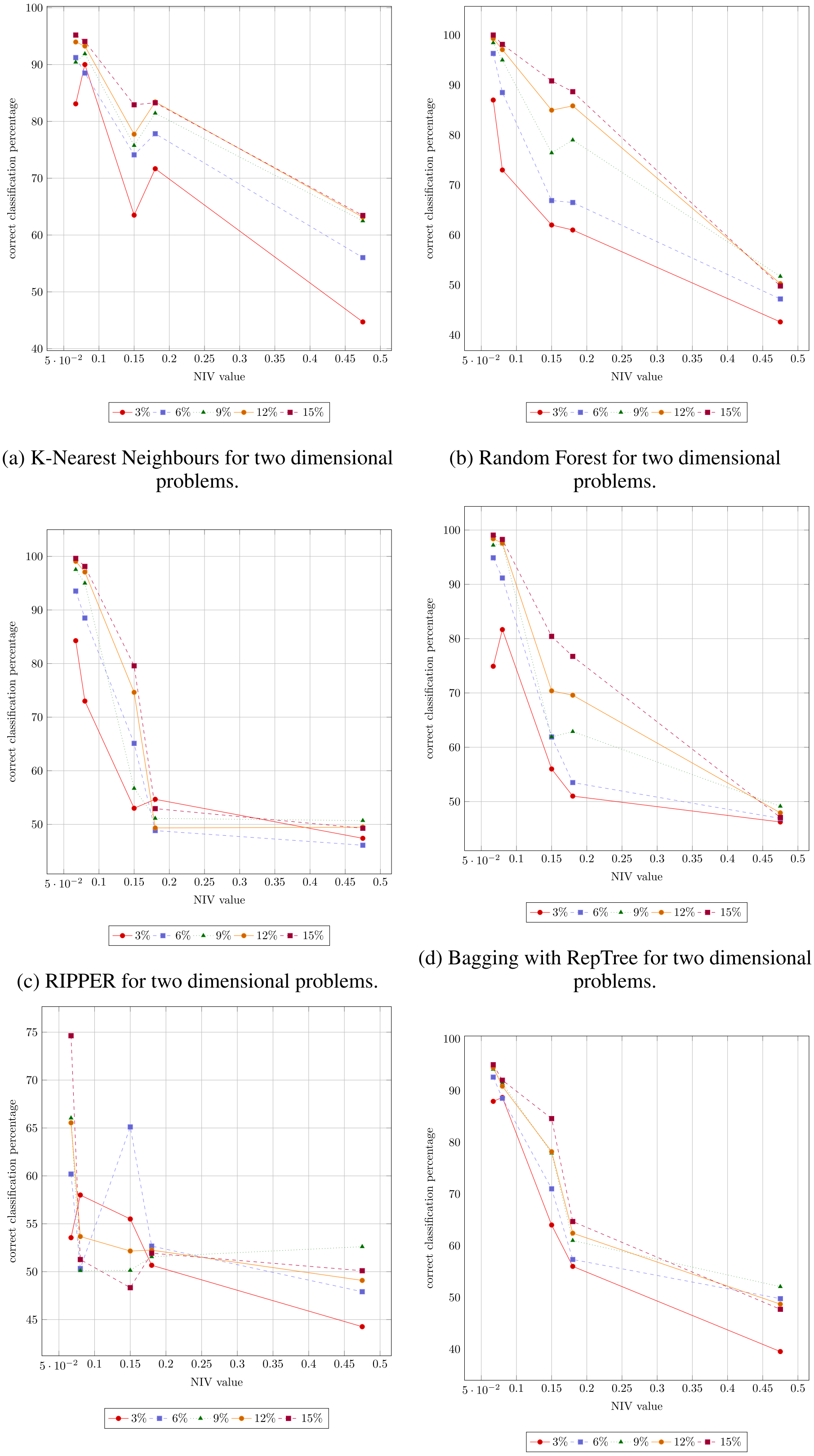
In order to facilitate the interpretation, results are visualized using 2-dimensional plots. In the graphs, each curve corresponds to the results obtained by an algorithm on a dataset of a given size. The points represent the results of an algorithm on a specific classification problem $\Phi^k_{n,h}$, using the percentage sample size defined by its curve. Thus, for each point, the y-axis represents the correct classification percentage and the x-axis represents the NIV value.

Figure 2 show the results for 2-dimensional classification problems $\Phi^2_{60,20}$, $\Phi^2_{100,20}$, $\Phi^2_{40,10}$, $\Phi^2_{100,10}$, and $\Phi^2_{80,4}$.

Figure 3 show the results for 3-dimensional classification problems $\Phi^3_{20,10}$, $\Phi^3_{21,7}$, $\Phi^3_{20,5}$, $\Phi^3_{16,4}$, and $\Phi^3_{18,3}$.

Figure 4 show the results for 4 dimensional classification problems $\Phi^4_{10,5}$, $\Phi^4_{9,3}$, $\Phi^4_{8,2}$, $\Phi^4_{10,2}$, and $\Phi^4_{14,2}$.

We can notice that not all the curves present a steady decreasing behavior, as far as NIV is concerned. However, the



(a) K-Nearest Neighbours for two dimensional problems.

(b) Random Forest for two dimensional problems.

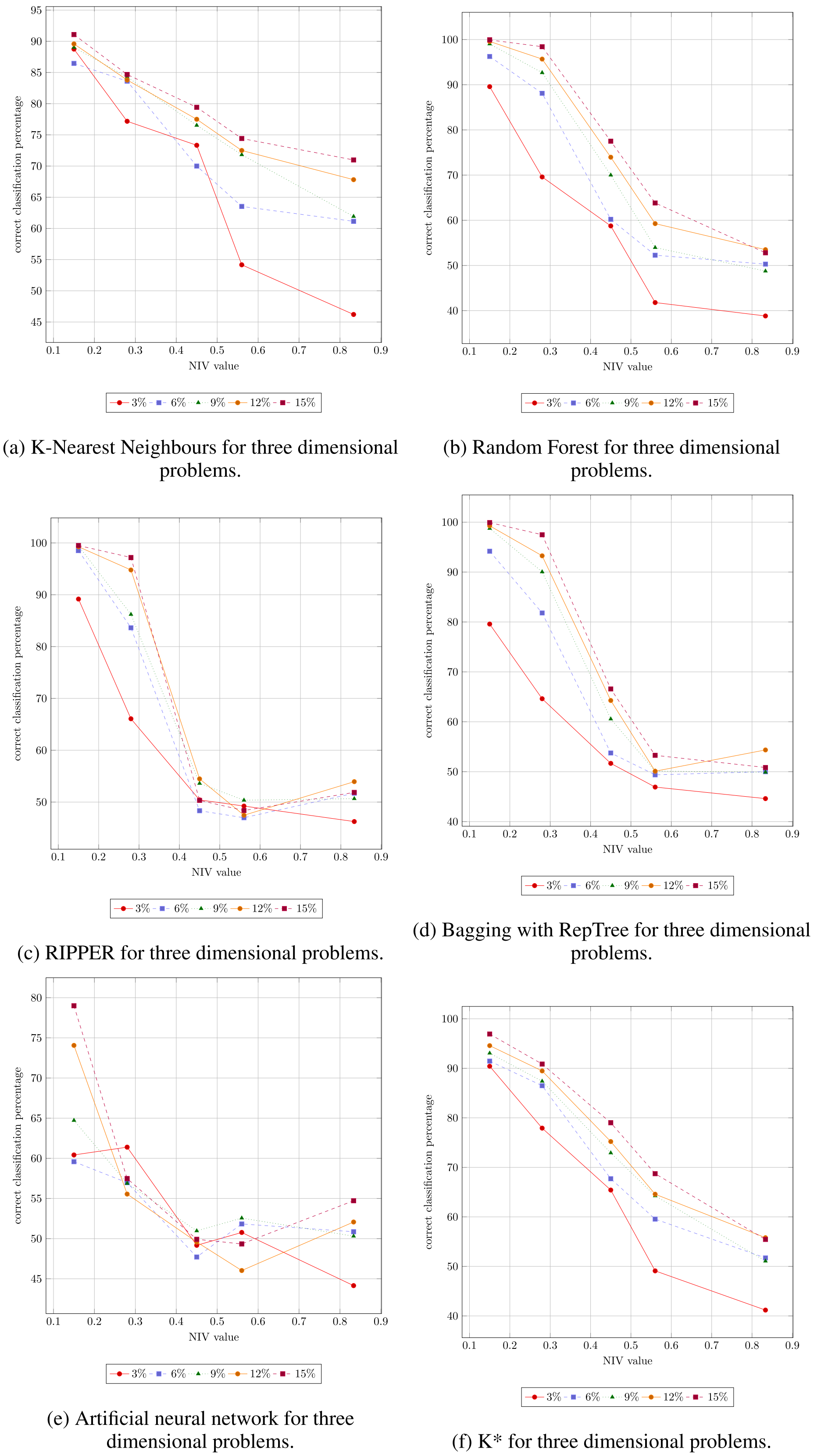
(c) RIPPER for two dimensional problems.

(d) Bagging with RepTree for two dimensional problems.

(e) Artificial neural network for two dimensional problems.

(f) K* for two dimensional problems.

FIGURE 2. **Results for 2-dimensional classification problems.**



(a) K-Nearest Neighbours for three dimensional problems.

(b) Random Forest for three dimensional problems.

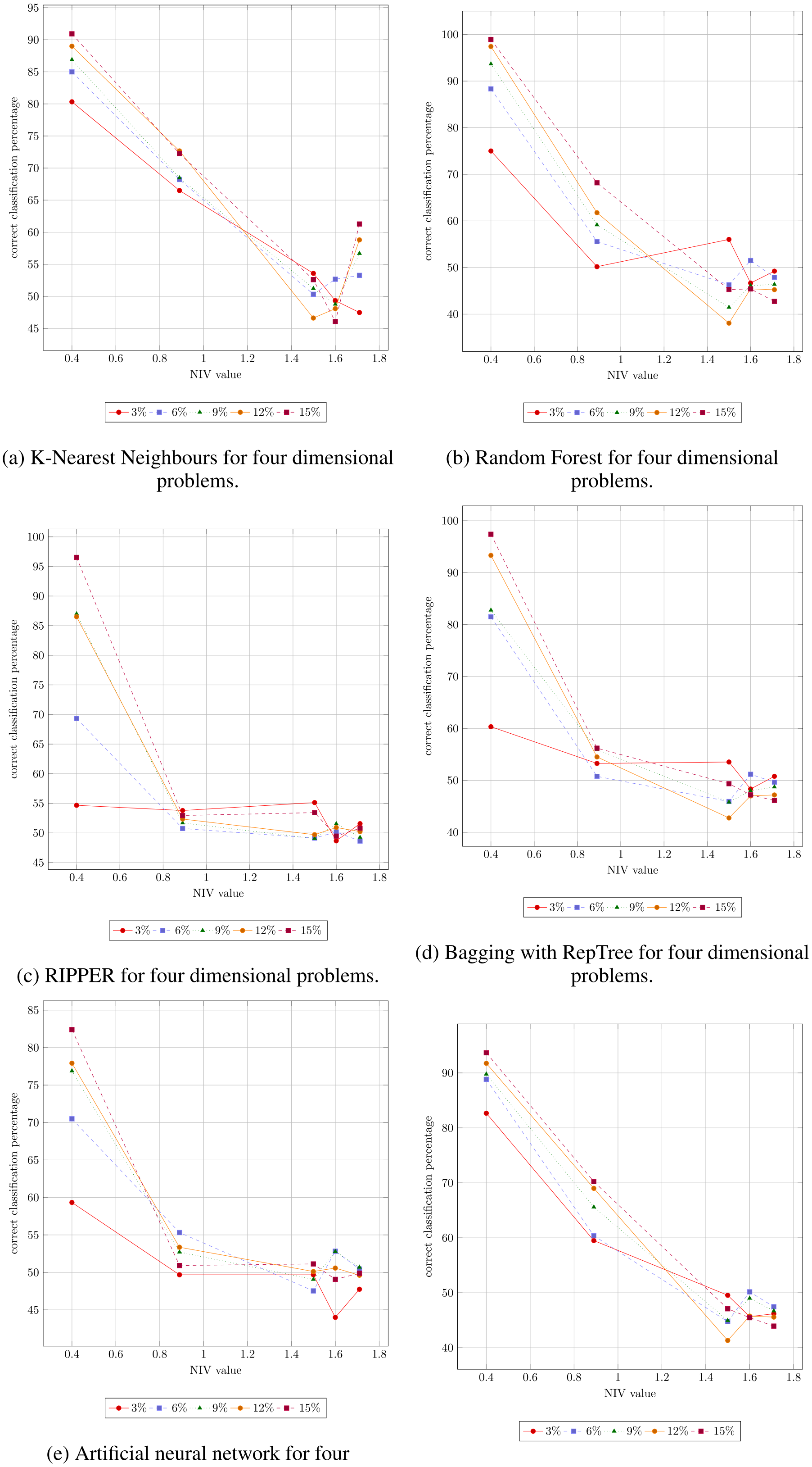
(c) RIPPER for three dimensional problems.

(d) Bagging with RepTree for three dimensional problems.

(e) Artificial neural network for three dimensional problems.

(f) K* for three dimensional problems.

FIGURE 3. **Results for 3-dimensional classification problems.**



(a) K-Nearest Neighbours for four dimensional problems.

(b) Random Forest for four dimensional problems.

(c) RIPPER for four dimensional problems.

(d) Bagging with RepTree for four dimensional problems.

(e) Artificial neural network for four dimensional problems.

(f) K* for four dimensional problems.

FIGURE 4. Results for 4-dimensional classification problems.

TABLE 2. Pearson correlation between NIV and correct classification percentage for the 2-dimensional classification problems.

%	KNN	K*	RIPPER	RF	BWR	ANN
3	-0.916	-0.893	-0.745	-0.901	-0.767	-0.908
6	-0.957	-0.843	-0.777	-0.895	-0.785	-0.53
9	-0.932	-0,87	-0.687	-0.959	-0.808	-0.283
12	-0.94	-0,908	-0.766	-0.995	-0.905	-0.636
15	-0.979	-0,928	-0.807	-0.998	-0.976	-0.428

TABLE 3. Pearson correlation between NIV and correct classification percentage for the 3-dimensional classification problems.

%	KNN	K*	RIPPER	RF	BWR	ANN
3	-0.963	-0.972	-0.86	-0.935	-0.906	-0.926
6	-0.939	-0.966	-0.813	-0.91	-0.872	-0.712
9	-0.998	-0.993	-0.866	-0.95	-0.904	-0.843
12	-0.98	-0.98	-0.824	-0.952	-0.879	-0.686
15	-0.968	-0.994	-0.818	-0.969	-0.917	-0.656

TABLE 4. Pearson correlation between NIV and correct classification percentage for the 4-dimensional classification problems.

%	KNN	K*	RIPPER	RF	BWR	ANN
3	-0.997	-0.968	-0.567	-0.801	-0.88	-0.872
6	-0.974	-0.941	-0.86	-0.897	-0.832	-0.898
9	-0.95	-0.973	-0.844	-0.934	-0.919	-0.847
12	-0.923	-0.976	-0.849	-0.94	-0.897	-0.882
15	-0.916	-0.994	-0.845	-0.983	-0.91	-0.839

curves have a general decreasing tendency. In most cases, the lower the NIV values, the higher the number of correct classifications. Some figures do not present a clear descendant tendency at high NIV values, as Figures 2c, 2e, 3c, 3e and 4a. For those figures, the classification percentage collapses on an interval around 50 and fluctuates on curves for higher NIV values. This implies that the algorithm's capacity is surpassed in some threshold of the NIV. At this point, the algorithm simply guesses the output, since it cannot extract enough information from the sample. Such threshold of the NIV varies depending on the algorithm.

We find a decreasing tendency in curves using Pearson correlation, between NIV and correct classification percentage. We measured the correlation for fixed sample percentage, algorithm and problem dimension. Tables 2, 3 and 4 show high negative correlation values for most curves obtained from a fixed sample percentage. Notice that each curve on each figure has a corresponding value on some tables.

We can conclude that fixing the number of dimensions and increasing NIV tends to increase classification error. However, the sensitivity of classification error to NIV depends on the algorithm and error itself, because such sensitivity seems to be reduced as error reaches values near 50%. The error also seems to be less affected by NIV values as dimension increases.

V. DISCUSSION

In this section we discuss the difference between the properties measured by pre-existing complexity metrics and the property studied in this work. For that we will consider a dataset of k attributes represented by the function f, where the value of the class only depends on an attribute such that contiguous values (e^i and e^{i+1}) must have a different class. Notice that this dataset is constant in the other attributes and tends to take smaller NIV values in relation to the maximum as k grows.

The property studied is related to measures of complexity that measure the overlap of classes [36]. A dataset that maximizes the NIV value will also tend to maximize the Maximum Fisher's discriminant ratio (MFDR), the Volume of overlap region (VOR), and the Maximal (individual) feature efficiency (MFE) [17]. However, it is not necessary to maximize the NIV value to obtain a maximum overlap between classes, since we can get very similar means, maximums and minimums between classes with the dataset defined by f, because the classes occupy almost the same space.

There are also the complexity metrics based on geometry and density. The e-neighborhood (EN) metric measures the number of balls of maximum size and centered in points that are necessary to cover each class without covering other classes, normalized by the number of points. Like other metrics, the dataset f maximizes the measure since all the points are at a minimum distance from another point of different class. The Local Set Average Cardinality (LSAC) [37] is based on the average for each instance $a \in A$, of the number of instances of the same class of a that are closer than any other instance of another class. For the function f the LSAC takes a value of maximum complexity because each instance is at a minimum distance from another instance of the opposite class. The Average number of points per dimension (ANPD) is the quotient between the number of points and the number of attributes. This measure is completely independent to the NIV value, since the measure is independent to the classes.

In the case of separability measures such as Error rate of linear classifier by linear programming (ERLCLP) [38], we can take datasets as defined by f, where a plane is incapable of minimally separating the classes like, but with limited values of NIV. Therefore linear separability is a much less strict property. For the Ratio of average intra/inter class nearest neighbor distance (RAICNND) [39] we see a relationship similar to measures of overlap, because in f taking a single attribute that produces a lot of overlap of classes equals the distance between instances of a same class and different classes, but with a limited number of neighboring instances of different classes. In the case of the Fraction of points on class boundary (FPCB) [40], a minimum spanning tree is constructed joining points by their distance, such that the number of neighboring points of different class is counted in a similar way to the property studied. The difference is that the property is defined on a grid where you can define a huge number of spanning trees that can give different values. Therefore it can be seen as a random measure of the difference between neighboring instances of distinct classes. This makes it the measure of complexity most related to the property studied in this work. Nevertheless, its ambiguity in grids

TABLE 5. Comparison of the different complexity measures discussed for the functions *f* and Φ.

Measure	Value for f	Value for Φ
NIV	$\frac{(n-1)}{n}$	$\frac{k(n-1)}{n}$
MFDR	$\frac{3}{2(n^2-1)}$	0
VOR	$\frac{(n-2)}{n}$	1
MFE	$\frac{(n-2)}{n}$	1
EN	n^k	n^k
ANPD	n^{k-1}	n^{k-1}
ERLCP	$\approx \frac{(n-1)}{2}$	$\approx \frac{n}{2}$
RAINNCD	$\frac{1}{2}$	$\frac{1}{2}$
LSAC	$\frac{(n-1)}{n}$	$\frac{(n-1)}{n}$
FPCB	Undefined	1

means that it cannot be used to adequately study the property analyzed in this work.

Table 5 compares the behavior of the analyzed complexity measures for the function f and the function Φ from equation (4), where we know that the latter is the binary function that maximizes the value of NIV. We can notice that unlike NIV, all metrics tend to the same value for both functions if n tends to infinity. Unlike FPCB, where the value for f is indeterminate because it depends on a tree that does not have a specific shape for total functions. This shows that the measures analyzed are only partially influenced by the property studied in this work.

Notice that the revised complexity metrics are oriented to generator functions whose domain is not fully defined. On the other hand, the property studied in this work is defined in the entire domain. This implies that to study this property in real datasets, this generating rule must be known, which is not possible without making assumptions.

VI. CONCLUSION

In this paper, we have shown a positive relation between classification error and the probability of finding adjacent inputs with different labels.

First, with Theorem 1, we proved that noisy data tends to produce adjacent inputs with different labels. This implies that bounding the number of adjacent inputs with different outputs is important in order to prevent overfitting in classification algorithms. We can also conclude that the NIV measure can provide valuable insights for the development of regularization methods.

Theorem 2 shows that too many adjacent inputs with different labels, cause the algorithms not to be able to produce good predictions on unseen data. Thus, high NIV values imply a classification problem that can be considered unlearnable. Theorem 3 shows that if we have freedom on sampling and an appropriate algorithm, then few adjacent inputs with different labels imply a low classification error.

We also show that NIV can be applied in the analysis of specific models. Theorems 4 and 5 relate NIV to the number of units in the first hidden layer of a threshold feed-forward neural network.

The theoretical results are complemented with experiments, and the results obtained show that the number of adjacent inputs with a different label is a variable whose increment causes the classification error to grow, when other properties as dimension are constant. We would like to stress out that even if the experiments are performed on problems with few attributes, the theoretical results are valid for problems of any dimension.

It is important to note that this property can only be properly measured in data sets where the rule that decides the classes is known. Despite this drawback, it is a property closely related to over-fitting and therefore helps us to better understand how classification algorithms work. In this sense, this type of analysis occupies an intermediate and complementary place, between purely mathematical approaches and purely experimental approaches where nothing is known about the data set. Although, the property studied is difficult to measure in real problems, this does not imply that it is not an important property.

Finally, we identify some possible extensions of this paper:

- An error analysis considering a statistical or expectation evaluation. Thus, a continuation of this work may be an analysis using CLT frameworks that analyze the mean error concerning NIV.
- We may avoid the hypothesis that the categories are defined for any possible input. Then, future work may generalize to partial classification problems.
- Existing data-complexity measures do not seem to relate to the number of adjacent inputs with a different label. However, a relation may exist and such possibility can be explored. We can say the same thing to VC-dimension.
- The basic ideas in this paper can motivate studies of other machine learning tasks. An open question is how to generalize the property studied to regression problems, since it is only defined in classification problems. In the case of unsupervised learning, it is not trivial to identify a similar property, since it is defined on labels.

REFERENCES

[1] J. Zhang, W. Xiao, Y. Li, and S. Zhang, "Residual compensation extreme learning machine for regression," *Neurocomputing*, vol. 311, pp. 126–136, Oct. 2018.

[2] D. D. Adhikary and D. Gupta, "Applying over 100 classifiers for churn prediction in telecom companies," *Multimedia Tools Appl.*, vol. 80, pp. 35123–35144, 2021.

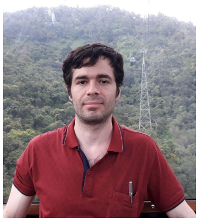
[3] N. Q. K. Le, Q.-T. Ho, E. K. Y. Yapp, Y.-Y. Ou, and H.-Y. Yeh, "DeepETC: A deep convolutional neural network architecture for investigating and classifying electron transport chain's complexes," *Neurocomputing*, vol. 375, pp. 71–79, Jan. 2020.

[4] J. N. Sua, S. Y. Lim, M. H. Yulius, X. Su, E. K. Y. Yapp, N. Q. K. Le, H.-Y. Yeh, and M. C. H. Chua, "Incorporating convolutional neural networks and sequence graph transform for identifying multilabel protein lysine PTM sites," *Chemometric Intell. Lab. Syst.*, vol. 206, Nov. 2020, Art. no. 104171.

[5] N. K. Mishra and M. E. Celebi, "An overview of melanoma detection in dermoscopy images using image processing and machine learning," 2016, *arXiv:1601.07843*.

[6] B. P. Roe, H.-J. Yang, J. Zhu, Y. Liu, I. Stancu, and G. McGregor, "Boosted decision trees as an alternative to artificial neural networks for particle identification," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 543, nos. 2–3, pp. 577–584, May 2005.

[7] Z. C. Lipton and J. Steinhardt, "Troubling trends in machine learning scholarship," *ACM Queue*, vol. 17, no. 1, p. 80, 2019.
[8] M. Anthony and N. Biggs, *Computational Learning Theory*, vol. 30. Cambridge, U.K.: Cambridge Univ. Press, 1997.
[9] D. Angluin, "Computational learning theory: Survey and selected bibliography," in *Proc. 24th Annu. ACM Symp. Theory Comput.*, 1992, pp. 351–369.
[10] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of Complexity*. Cham, Switzerland: Springer, 2015, pp. 11–30.
[11] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.
[12] M. Kearns and M. Li, "Learning in the presence of malicious errors," *SIAM J. Comput.*, vol. 22, no. 4, pp. 807–837, 1993.
[13] L. Pitt and M. K. Warmuth, "Prediction-preserving reducibility," *J. Comput. Syst. Sci.*, vol. 41, no. 3, pp. 430–467, Dec. 1990.
[14] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie, "On the learnability of discrete distributions," in *Proc. STOC*, vol. 94. Princeton, NJ, USA: Citeseer, 1994, pp. 273–282.
[15] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," 2016, *arXiv:1611.03530*.
[16] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," 2017, *arXiv:1710.05468*.
[17] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 289–300, Mar. 2002.
[18] R. A. Mollineda, J. S. Sánchez, and J. M. Sotoca, "Data characterization for effective prototype selection," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.* Berlin, Germany: Springer, 2005, pp. 27–34.
[19] F. W. Smith, "Pattern classifier design by linear programming," *IEEE Trans. Comput.*, vol. C-17, no. 4, pp. 367–372, Apr. 1968.
[20] E. Leyva, A. González, and R. Pérez, "A set of complexity measures designed for applying meta-learning to instance selection," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 354–367, Feb. 2015.
[21] A. Hoekstra and R. P. W. Duin, "On the nonlinearity of pattern classifiers," in *Proc. 13th Int. Conf. Pattern Recognit.*, vol. 4, 1996, pp. 271–275.
[22] L. Li, "Data complexity in machine learning and novel classification algorithms," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 2006.
[23] M. Li *et al.*, *An Introduction to Kolmogorov Complexity and Its Applications*, vol. 3. New York, NY, USA: Springer, 2008.
[24] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the Vapnik–Chervonenkis dimension," *J. ACM*, vol. 36, no. 4, pp. 929–965, Oct. 1989.
[25] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, 1992.
[26] J. G. Cleary and L. E. Trigg, "K*: An instance-based learner using an entropic distance measure," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 108–114.
[27] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[28] W. W. Cohen, "Fast effective rule induction," in *Proc. 25th Int. Conf. Mach. Learn.*, A. Prieditis and S. J. Russell, Eds. Tahoe City, CA, USA: Morgan Kaufmann, Jul. 1995, pp. 115–123.
[29] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[30] J. R. Quinlan, "Simplifying decision trees," *Int. J. Man-Mach. Stud.*, vol. 27, no. 3, pp. 221–234, Sep. 1987.
[31] B. Yegnanarayana, *Artificial Neural Networks*. Delhi, India: PHI Learning, 2009.
[32] L. Valiant, *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. New York, NY, USA: Basic Books, 2013.
[33] J. Zhang, Y. Li, W. Xiao, and Z. Zhang, "Robust extreme learning machine for modeling with unknown noise," *J. Franklin Inst.*, vol. 357, no. 14, pp. 9885–9908, Sep. 2020.
[34] D. Gupta, H. J. Sarma, K. Mishra, and M. Prasad, "Regularized universum twin support vector machine for classification of EEG signal," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 2298–2304.
[35] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*.
[36] J.-R. Cano, "Analysis of data complexity measures for classification," *Expert Syst. Appl.*, vol. 40, no. 12, pp. 4820–4831, 2013.
[37] A. C. Lorena, L. P. F. Garcia, J. Lehmann, M. C. P. Souto, and T. K. Ho, "How complex is your classification problem? A survey on measuring classification complexity," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–34, Oct. 2019.
[38] M. Basu and T. Kam Ho, "The learning behavior of single neuron classifiers on linearly separable or nonseparable input," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 2, 1999, pp. 1259–1264.
[39] T. K. Ho, "Data complexity analysis for classifier combination," in *Proc. Int. Workshop Multiple Classifier Syst.* Berlin, Germany: Springer-Verlag, 2001, pp. 53–67.
[40] S. P. Smith and A. K. Jain, "A test to determine the multivariate normality of a data set," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-10, no. 5, pp. 757–761, Sep. 1988.



SEBASTIÁN A. GRILLO was born in San Martín, Argentina, in 1985. He received the bachelor's degree in pure mathematics and the master's degree in computer science from the Universidad Nacional de Asunción, Paraguay, in 2008 and 2011, respectively, and the Ph.D. degree in computer engineering from the Federal University of Rio de Janeiro, Brazil, in 2017. He collaborated with the Computer Engineer Department of the Universidad Americana in the Grant PINV18-1199. He has authored articles ranging from theory of computing to artificial intelligence.

JULIO CÉSAR MELLO ROMÁN was born in Concepción, Paraguay, in 1981. He received the degree in computer systems analysis from the Universidad Católica Nuestra Señora de la Asunción, Paraguay, in 2012, and the M.Sc. and Ph.D. degrees in computer sciences from the National University of Asuncion, Paraguay, in 2017 and 2021, respectively. He collaborated with the Computer Engineer Department of the Universidad Americana in the Grant PINV18-1199. His research interests include digital image processing and analysis, mathematical morphology, and machine learning.

JORGE DANIEL MELLO-ROMÁN received the Ph.D. degree from the Mathematical Engineering, Statistics and Operations Research Program, Complutense University of Madrid, and the master's degree in advanced mathematics—statistics and operations research from the National University of Distance Education (UNED), Spain. He is currently a Professor and the Dean of the Faculty of Exact and Technological Sciences, National University of Concepción, Paraguay.

JOSÉ LUIS VÁZQUEZ NOGUERA was born in Asunción, Paraguay, in 1985. He received the degree in computer systems engineering from the Instituto Tecnológico de León, México, in 2008, and the M.Sc. and Ph.D. degrees in computer sciences from the Universidad Nacional de Asunción, Paraguay, in 2012 and 2018, respectively. Since 2019, he has been a Researcher with the Computer Engineer Department, Universidad Americana. He has authored over 40 papers in the field of image processing, mathematical morphology, and computer vision.

MIGUEL GARCÍA-TORRES received the B.S. degree in physics and the Ph.D. degree in computer science from the Universidad de La Laguna, Tenerife, Spain, in 2001 and 2007, respectively. After obtaining the doctorate, he held a postdoctoral position at the Laboratory for Space Astrophysics and Theoretical Physics, National Institute of Aerospace Technology (INTA). There, he joined the Gaia mission from the European Space Agency (ESA) and started to participate in the Gaia Data Processing and Analysis Consortium (DPAC) as a member of astrophysical parameters at the Coordination Unit (CU8). Since then, he has been involved at the Object Clustering Analysis (OCA) Development Unit. He is currently an Associate Professor with the Escuela Politécnica Superior, Universidad Pablo de Olavide. He collaborated with the Computer Engineer Department of the Universidad Americana in the Grant PINV18-1199. His research interests include machine learning, metaheuristics, big data, time series forecasting, bioinformatics, and astrostatistics.

FEDERICO DIVINA received the Ph.D. degree in artificial intelligence from Vrije Universiteit Amsterdam. After that, he worked as a Postdoctoral Researcher at the University of Tilburg, within the European Project NEWTIES. In 2006, he moved to the Pablo de Olavide University. He has been working on knowledge extraction since his Ph.D. thesis at Vrije Universiteit Amsterdam. He collaborated with the Computer Engineer Department of the Universidad Americana in the Grant PINV18-1199. His main research interests include machine learning, in particular on techniques based on soft computing, bioinformatics, and big data.

PEDRO ESTEBAN GARDEL SOTOMAYOR was born in Asunción, Paraguay, in 1980. He received the degree in electromechanical engineering from the National University of Asuncion, Paraguay, in 2006, and the Ph.D. degree in energy and fluid mechanics engineering from the University of Valladolid, Spain, in 2013. He collaborated with the Computer Engineer Department of the Universidad Americana in the Grant PINV18-1199. His main research interests include machine learning, multiobjective optimization, and electric power systems.