IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 May 2022, accepted 23 June 2022, date of publication 25 July 2022, date of current version 4 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192514

RESEARCH ARTICLE

Why Is Multiclass Classification Hard?

PABLO DEL MORAL, SŁAWOMIR NOWACZYK, AND SEPIDEH PASHAMI
Center for Applied Intelligent Systems Research (CAISR), Halmstad University, 301 18 Halmstad, Sweden

Corresponding author: Pablo Del Moral (pabmor@hh.se)

This work was supported by Stiftelsen för kunskaps- och kompetensutveckling.

ABSTRACT In classification problems, as the number of classes increases, correctly classifying a new instance into one of them is assumed to be more challenging than making the same decision in the presence of fewer classes. The essence of the problem is that using the learning algorithm on each decision boundary individually is better than using the same learning algorithm on several of them simultaneously. However, why and when it happens is still not well-understood today. This work's main contribution is to introduce the concept of heterogeneity of decision boundaries as an explanation of this phenomenon. Based on the definition of heterogeneity of decision boundaries, we analyze and explain the differences in the performance of state of the art approaches to solve multi-class classification. We demonstrate that as the heterogeneity increases, the performances of all approaches, except one-vs-one, decrease. We show that by correctly encoding the knowledge of the heterogeneity of decision boundaries in a decomposition of the multi-class problem, we can obtain better results than state of the art decompositions. The benefits can be an increase in classification performance or a decrease in the time it takes to train and evaluate the models. We first provide intuitions and illustrate the effects of the heterogeneity of decision boundaries using synthetic datasets and a simplistic classifier. Then, we demonstrate how a real dataset exhibits these same principles, also under realistic learning algorithms. In this setting, we devise a method to quantify the heterogeneity of different decision boundaries, and use it to decompose the multi-class problem. The results show significant improvements over state-of-the-art decompositions that do not take the heterogeneity of decision boundaries into account.

INDEX TERMS Classification complexity, heterogeneity of decision boundaries, multi-class classification.

I. INTRODUCTION

Multi-class classification is the task of classifying a new instance into one among at least three classes. Multi-class problems are ubiquitous in many domains such as image classification [1], text classification [2], microarray classification [3], etc. It is widely accepted that these problems become more challenging as the number of classes increases; however, *why they are more difficult* is an unanswered question. We claim that this extra layer of complexity in multi-class classification problems can be explained, at least partially, by the heterogeneity of decision boundaries. This paper formalizes that notion and provides justification for that claim.

A decision boundary is the manifold that separates the region of the feature space labeled with one class from the region of the space labeled with another one. We regard the multi-class problem as the problem of solving each of its classes' one-to-one decision boundaries. As the number of classes increases, so does the number of decision boundaries a learning algorithm has to solve. Intuition and experimental results tell us that as the number of decision boundaries increases, so does the problem's difficulty. However, experiments also show that if we divide the multi-class problem into smaller ones, the performance of our models can increase.

A fundamental question is "why?". Why is multi-class classification hard? What are the characteristics of the decision boundaries that make the classification problem harder or easier? Where does the intrinsic difficulty of multi-class classification problems lay?

The difficulty of classification problems has been extensively studied for the binary case. In [4], the authors characterize the complexity of a binary classification problem based on geometrical features of the decision boundary between the two classes. No similar work exists on multi-class

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

classification. In this paper, we argue that the total difficulty of the multi-class classification problem is not just the combination of the geometrical complexities of the one-to-one decision boundaries; there exists an additional source of difficulty coming from the interaction of different decision boundaries. We identify the heterogeneity of decision boundaries as a source of this interaction and an ultimate cause of the difficulty of multi-class classification problems.

There exist different approaches to solve multi-class classification problems. Some learning algorithms (CART, Neural Networks, K-NN) can naturally handle outputs with more than two classes. In general, when these algorithms are used, little attention is paid to the structure between classes. Some other algorithms can only handle binary outputs natively (SVM, perceptron, logistic regression, AdaBoost,...). In these cases, binary classifiers need to be adapted to a setting that can address multi-class classification problems. The adaptation runs in three steps: the original problem is divided into binary problems, independent models are trained for each one, and the results are aggregated.

There are multiple ways to decompose a multi-class classification problem to use binary classifiers [5]–[7]. A very interesting common conclusion is how, using the same learning algorithm, results can vary significantly depending on the decomposition chosen [8]. In this scenario, finding the best multi-class classifier means finding the best binary classifiers as well as finding the best decomposition.

This fact points back to our claim: there exists an extra layer of difficulty in multi-class classification problems. A layer of difficulty that the learning algorithm often cannot unravel itself, a layer of complexity that can only be dealt with by a smart decomposition. However, [9] shows that some learning algorithms are not affected by how we decompose some particular problems. These results lead us to more questions: What are the characteristics of the learning algorithm that make it sensitive to the different decompositions? How do the characteristics of the learning algorithm and the decision boundaries relate to each other?

There are many works dedicated to finding the best possible decompositions of multi-class classification problems. However, there has not been a similar effort into understanding why some decompositions are better than others. In [10], the authors compare flat and hierarchical approaches. They conclude that grouping decision boundaries is better for balanced problems, whereas for unbalanced problems, separating decision boundaries yields better results.

Our work aims to show how the heterogeneity of decision boundaries can affect the classification performance of different multi-class classification approaches. We will first present a set of illustrative experiments performed on synthetic datasets specifically designed to isolate the effects of the heterogeneity of decision boundaries. We will analyze how different approaches towards solving multi-class classification are affected by the varying degrees of heterogeneity in decision boundaries.

After the examples that build up the intuition, we will present a method to measure the heterogeneity of decision boundaries on a real dataset. We will show how the heterogeneity of decision boundaries affects the classification performance using a state of the art classification algorithm (nonlinear SVM). Finally, we will use the knowledge of the heterogeneity of decision boundaries to decompose the multi-class problem.

Our contributions can be summarized as:

- We define formally the concept of heterogeneity of decision boundaries.
- We quantify the overall negative effect of the heterogeneity of decision boundaries on the performance of classification tasks.
- We show how the heterogeneity of decision boundaries can explain the differences in classification performance of several approaches to solving multi-class problems.
- We propose new decomposition approaches that take the heterogeneity of decision boundaries into account and analyze the improvements with respect to approaches that do not.
- We showcase the above phenomena with both intuitive and educational examples, as well as a real dataset using a state of the art classifier.

II. DEFINITIONS AND BACKGROUND

In this section, we will establish the basic concepts to define heterogeneity of decision boundaries.

Definition: A **learning algorithm** receives as input a training set S, sampled from an unknown distribution $\mathcal{D}$ and labeled by some target function $f : \mathcal{X} \longrightarrow \mathcal{Y}$, and should output a classifier $h : \mathcal{X} \longrightarrow \mathcal{Y}$. The goal of the algorithm is to find the classifier h that minimizes a given loss function with respect to the unknowns $\mathcal{D}$ and f.

Ideally, we would like to find h from the set of all possible functions $\mathcal{H}$ mapping the feature space $\mathcal{X}$ into the label space $\mathcal{Y}$. In practice, we restrict the size of $\mathcal{H}$, by either selecting a smaller set of functions (e.g., support vector machines can only find linear functions); or by choosing a heuristic to reduce the search automatically (e.g., CART algorithm to build decision trees; here we assume that a tree of arbitrary depth can represent any function in a discrete space). Reducing the space of hypothesis $\mathcal{H}$ by either selecting a family of functions or following a heuristic is called inductive bias.

The typical definition of a learning algorithm denotes, in fact, a family of learning algorithms differing in some hyperparameter configuration. With CART, for example, we need to fix hyperparameters related to the depth of the tree, the pruning complexity parameter, the minimal number of instances in each leaf of the tree, and so on; with SVM's we can select the type of kernel, its characteristics, the regularization parameter, and more; with neural networks, we can choose the structure, number of neurons, activation functions, etc.

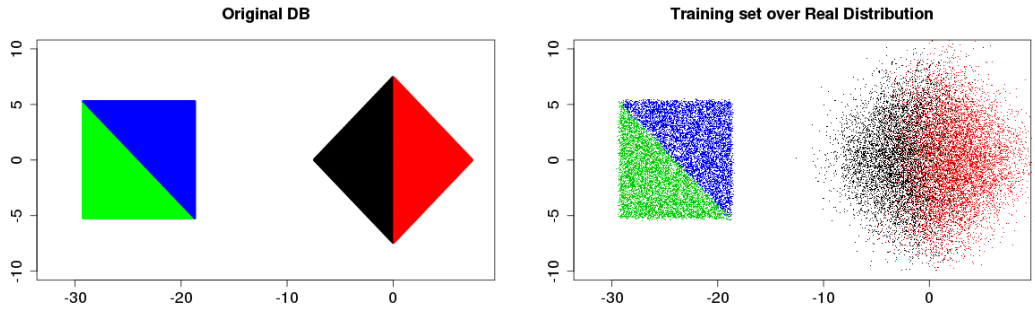

FIGURE 1. On the left, the two underlying decision boundaries are illustrated, and, on the right, the corresponding training set used for this experiment.

When facing a classification task, the standard practice is to choose one (or several) of these families of learning algorithms. Then, through validation schemes like cross-validation, we estimate the performance of each individual learning algorithm with its hyperparameter configuration. Fixing these hyperparameters can be regarded as a second layer of learning, where instead of choosing the classifier that obtains the best performance, we choose from a pool of learning algorithms, the one that produces the classifier that results in the lowest classification error.

Definition: we define a **ground truth decision boundary** as the manifold that delimits regions of the feature space labeled with one class from those labeled with another, as defined by the unknowns distribution $\mathcal{D}$ and labeling function f.

The final output of a classifier can be interpreted as a decision boundary approximating the unknown ground truth one. Given a binary classification problem, the role of learning can be regarded as finding the decision boundary that best matches the unknown ground truth one. How well the classifier matches the decision boundary is computed through the estimated classification performance.

In practice, different learning algorithms can lead to classifiers with the same classification performance. This can be due to those algorithms' output being the same decision boundary or different classifiers finding different decision boundaries that result in the same classification performance.

The inherent characteristics of a ground truth decision boundary will affect the classification performance of the classifiers trained to solve it. Given a family of learning algorithms, we can look at the space of its possible hyperparameter configurations and measure the classification performance of the corresponding classifiers. We can regard every learning algorithm defined by its hyperparameters as a measurement instrument and the classification performance as the final measure: the different classification performances are a reflection of the inherent characteristics of the ground truth decision boundary.

Given a learning algorithm, with its corresponding hyperparameters to tune, in a case with more than one decision boundary, the regions of maximum classification performance in the hyperparameter space of each decision boundary may not overlap. In such a case, there is no single configuration that allows the learning algorithm to learn all decision boundaries as well as if it was solving each decision boundary independently. If, on the other hand, those regions overlap, it means that there exists a parameter configuration that allows the learning algorithm to solve both of them at once and still achieve maximum classification performance.

Definition: given two or more decision boundaries and a family of learning algorithms defined by their hyperparameter configurations, we say that these decision boundaries are **heterogeneous** if their regions of maximum classification performance in the hyperparameter space do not overlap.

Our definition of heterogeneity is dependent on the characteristics of the data and the characteristics of the learning algorithm chosen. For different algorithms, the geometrical characteristics of the decision boundaries that make them heterogeneous will be different.

This paper will study how the heterogeneity of decision boundaries can diminish the classification performance when two or more decision boundaries are solved simultaneously.

III. PROOF OF CONCEPT

We are going to illustrate the concept behind heterogeneous decision boundaries with a toy example. As a family of learning algorithms, we use a decision tree learner, CART [11] (as implemented in the package rpart in R). In our setting, we will only tune the depth of the tree and keep the rest of the parameters constant. The advantage of just tuning the depth of the tree is that we can measure the classifier's complexity – the shallower the tree, the simpler the model.

In the left picture Fig. 1, we have exemplified two decision boundaries. We will solve two different problems with CART. First, we will classify all the classes simultaneously. Then, we will independently solve the diagonal decision boundary and the horizontal one. The CART algorithm can only make splits parallel to the axis. Therefore, the diagonal decision boundary needs a very deep (or complex) tree, while the parallel one needs a very shallow (or simple) tree. In this way, the decision boundaries become heterogeneous for the selected learning algorithm.

In Fig. 1, on the right, we can see an example of the data generated for this proof of concept. To obtain it, we sample

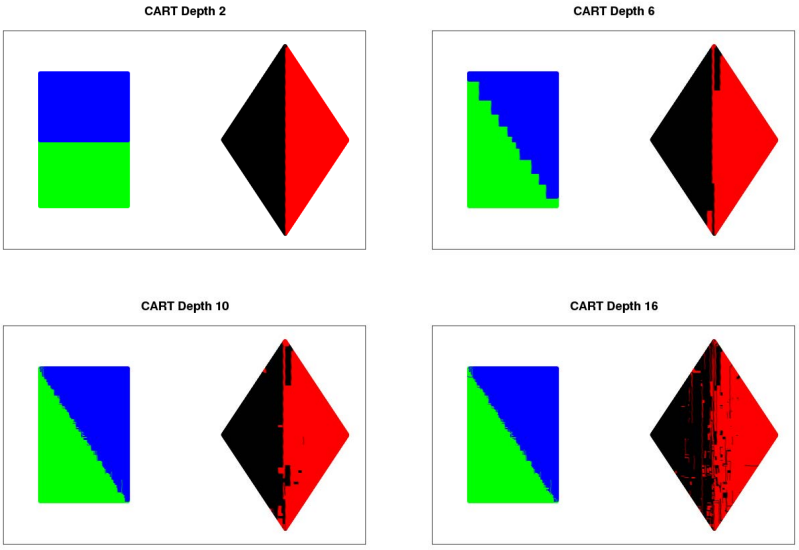

FIGURE 2. Results of a single classifier with tree depths 2, 6, 10 and 16 respectively.

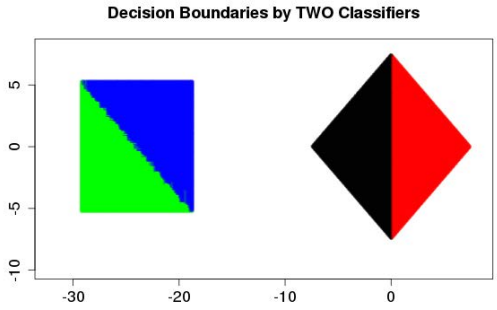

FIGURE 3. Results of learning the two decision boundaries independently.

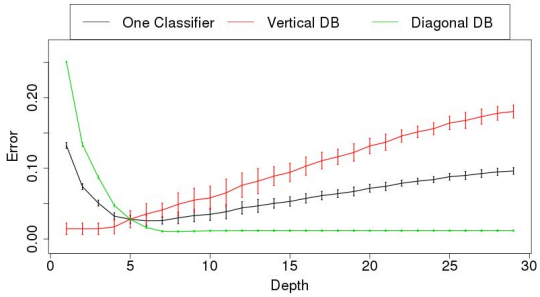

FIGURE 4. Dependence of the error of each classifier as the depth of the tree changes. In red, the error of a classifier solving the diagonal decision boundary. In green, the error of a single solving the horizontal decision boundary. In black, the error of a classifier solving both decision boundaries.

10000 points from each of the classes denoted by the colors. We add Gaussian noise to the x and y components for the vertical decision boundary to force the learning algorithm to be penalized by overfitting. We perform 5-fold cross-validation to evaluate the behavior of our classifiers under different hyperparameter configurations.

In Fig. 2, we can see the behavior of a single classifier solving both decision boundaries simultaneously for different tree depths. The CART algorithm is presented with the four classes denoted by the colors. As the depth of the tree increases, the classifier matches the diagonal decision boundary better. However, we observe the opposite behavior in the horizontal decision boundary.

As a comparison, in Fig. 3, we can see the results of training two different classifiers, separately for each decision boundary. With two classifiers, the decisions match almost perfectly the original distribution of the data. At the same time, for a single classifier, there is no possible depth of the tree where the CART algorithm can achieve optimal performance. By optimal, we mean the best this learning algorithm can do; it is capable of doing better if it solves each decision boundary independently, as demonstrated in Fig. 3.

In a way, the two decision boundaries are interacting with each other through the learning algorithm. The global optimization of the depth of the tree forces the learning algorithms to find a compromise between the two decision boundaries.

In Fig. 4, we can see the details of the error of the different classifiers as a function of the depth of the tree. To obtain an estimation of the error, we have sampled 10 different training sets from the original distribution.

For the vertical decision boundary, the optimal depth of the tree should be 1, and it starts to overfit after depth 4. For the diagonal decision boundary, it underfits until the depth of the tree is 7. The optimal region in the hyperparameter space for the vertical decision boundary is from 1 to 4; the optimal region for the diagonal decision boundary goes from 7 to 30. When training a single classifier, the optimal depth of the tree

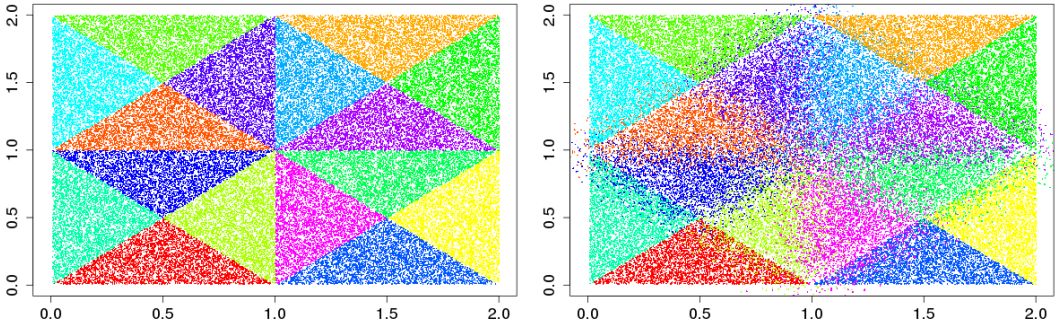

FIGURE 5. On the left, the training set of the first iteration (no noise). On the right, the training for the last iteration with noise in all inner triangles.

is somewhere between 5 and 7. A tree of this depth overfits for the vertical decision boundary.

Theoretically, there clearly exists a tree where the best performance can be achieved – but our family of learning algorithms is unable to find it. The inductive bias rooting from selecting the learning algorithm and its hyperparameters makes it impossible for it to solve both decision boundaries with the best possible performance simultaneously.

Overfitting is of crucial importance to understand this toy experiment. For more advanced algorithms, those able to avoid overfitting, there is no penalization for the most complex trees: in Fig. 4, the red line would be flat, and the optimal region in the hyperparameter space will go from 1 to 30. By design, this experiment has one simple and one complex decision boundaries that do not overlap in the hyperparameter space. However, as we will show later in the paper, model complexity and overfitting are just one possible reason for the heterogeneity; in high dimensional hyperparameter spaces, we can expect more complex sets of heterogeneous decision boundaries.

Obviously, the family of learning algorithms chosen defines the optimal region in the hyperparameter space. Different families will have different parameters to optimize. This illustrative example is specific to decision trees, where we only tune the depth of the tree; other algorithms will exhibit similar behavior, albeit in other scenarios.

IV. EXPERIMENTS

A. MULTI-CLASS CLASSIFICATION AND HETEROGENEITY OF DECISION BOUNDARIES

In this experiment, we are going to evaluate how state-of-the-art approaches are affected by the heterogeneity of decision boundaries. In particular, we are going to evaluate a single classifier, One-vs-All decomposition, and One-vs-One decomposition.

A multi-output classifier can natively deal with more than two classes. CART from [11] is an example of this type of learning algorithm.

One-vs-All trains one binary classifier for each class, versus all the data points from the rest of the classes. All binary

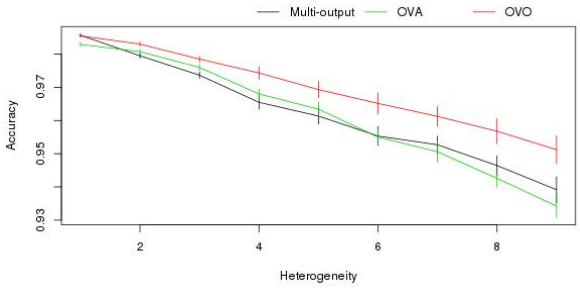

FIGURE 6. Accuracy comparison of multi-output, OVA, and OVO classifiers as the heterogeneity grows.

classifiers need to be evaluated, and the final prediction is built by integrating all the binary predictions through voting. If the output of the classifier is probabilistic, the classifier that outputs the highest probability defines the prediction, cf. [7].

One-vs-One trains one binary classifier for each pair of classes. To output a prediction, all binary classifiers are evaluated, and the final prediction is decided via voting. If the output of the classifier is probabilistic, there are many different approaches to combine the outputs, cf. [7]. In our experiment, for simplicity, we choose to combine the One-vs-One classifiers via voting.

We will again use synthetic datasets based on diagonal and parallel to the axis decision boundaries. Using our definition of heterogeneity of decision boundaries, we will start with a set of homogeneous decision boundaries, and iteratively we will add heterogeneity. Our dataset will consist of 16 classes. In Fig. 5, we can see an example of all the decision boundaries without noise, and all of the inner classes with noise. We control the heterogeneity by adding noise: without it there is no overfitting, parallel to the axis and diagonal decision boundaries are homogeneous; with noise, very complex trees are penalized when solving parallel to the axis decision boundaries. At each iteration, we will add noise to one of the inner triangles. Our learning algorithm will again be CART, and we will tune the pruning parameter through 10-Fold cross-validation.

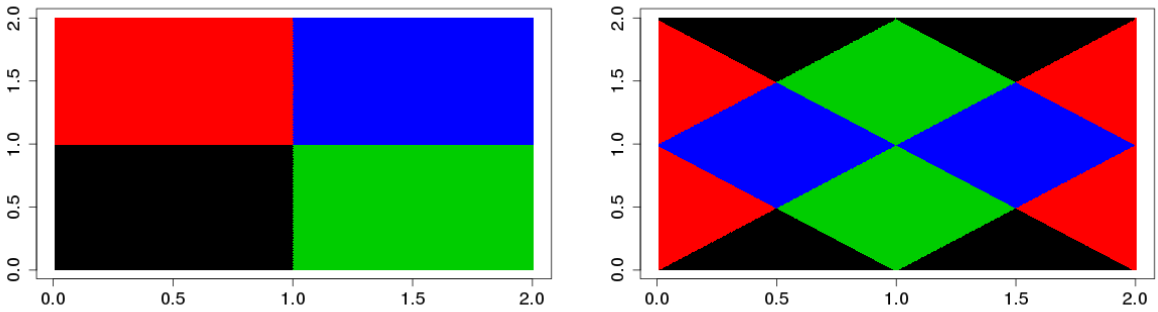

FIGURE 7. Decomposition of the original multi-class problem into two 4-class problems. On the left, parallel to the axis decision boundaries. On the right, diagonal decision boundaries.

For all our experiments, we will sample 50 training sets and report the accuracy on the original distribution without noise. In this way, we make sure that we correctly evaluate how well the classifier matches all underlying decision boundaries.

The results of the comparison of the three approaches can be seen in Fig. 6. In the first iterations, there are no significant differences between the approaches. When decision boundaries are fully homogeneous, the performance of the multi-output classifier is indistinguishable from OVO, and only slightly higher than OVA. However, as we inject more and more heterogeneity through noise, obviously the performance of all the classifiers degrades – however, both OVA and the multi-output classifier suffer more, and perform increasingly worse in comparison to OVO.

The key observation here is that OVO is treating every decision boundary individually, finding the optimal hyper-parameter configuration for each of them. Thus, the only difference in performance comes from the problem becoming inherently harder with more noise. The other two approaches group together decision boundaries, including heterogeneous ones, which necessarily leads to additional performance loss, as demonstrated in the toy example in Section III.

B. EXTENDING THE MULTI-OUTPUT CLASSIFIER TO DEAL WITH HETEROGENEITY

Generally, when a learning algorithm that can deal with more than two classes is chosen, usually no attention is paid to the structure of the classes and possible decompositions of the problem. On the other hand, we have shown how a single multi-output classifier is adversely affected by the heterogeneity of decision boundaries. Based on these finding, however, we will now demonstrate how knowing what type of heterogeneities we are dealing with allows for designing efficient, even if ad-hoc, solutions.

For example, in the case of Fig. 5, since we have two types of decision boundaries, we can decompose our 16-classes classiification problem into two 4-classes classification problems. The decomposition is visualized in Fig. 7. The first classifier distinguishes between the four meta-classes on the left figure, the second classifier distinguishes between the four

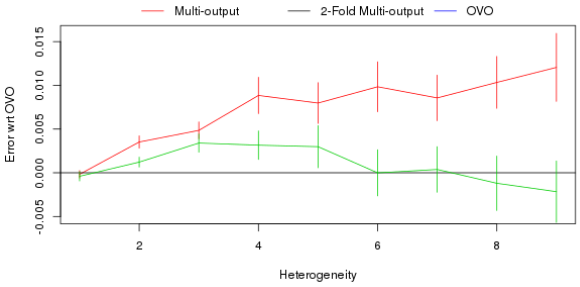

FIGURE 8. Error comparison of multi-output, and 2-fold multi-output with respect to OVO.

meta-classes on the right figure. In this way, every individual class in the figure is identified unambiguously.

In Fig. 8, we can see the comparison of a single multi-output classifier with the 2-fold decomposed multi-output classifier. The graph shows the error compared to OVO, since that is the gold-standard approach that obtains the best results. We knew from before that a single multi-output classifier performs worse and worse as the heterogeneity increases. Here, we are showing that a decomposition that separates the decision boundaries in homogeneous groups obtains results equivalent to those of OVO.

C. EXTENDING BINARY CLASSIFIERS TO DEAL WITH HETEROGENEITY

CART can natively deal with multi-class classification problems; however, other learning algorithms can only deal with binary outputs. Now, we are going to introduce two binary decompositions that incorporate the information about the heterogeneity.

Minimal Decomposition: we decompose the 16-classes classification problem into 4-binary classification problems. A combination of the 4 classifiers will uniquely define each class. The decomposition can be easily understood with help of Fig. 9

Heterogeneity Aware Hierarchy (HAH): we are going to decompose the 16-classes classification problem into a hierarchy of 15 binary problems. At each node of the hierarchy, a binary classifier is trained to discern between the instances

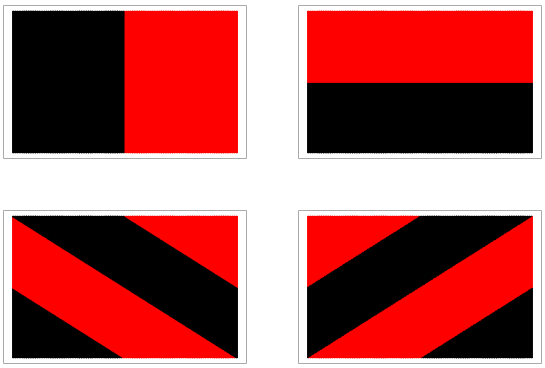

FIGURE 9. Decomposition of the original multi-class problem into 4 binary problems.

labeled with the classes on the left branch, and those labeled with the classes on the right branch. In Fig. 10, we have a description of how the hierarchical decomposition has been made.

In Fig. 11, we can see the comparison of the two approaches, together with OVA, with respect to OVO. Again, those decompositions that separate decision boundaries into homogeneous groups obtain classification performances similar to OVO, and significantly outperform approaches that do not take into account this information, like OVA.

D. COMPARISON OF HAH AND RANDOM HIERARCHIES

Our hierarchy has been manually created. In this subsection, we will compare our handcrafted hierarchy with a state of the art method to find good hierarchies.

The method presented in [6] takes a number of hierarchies sampled uniformly at random. Through cross-validation on the training set, we pick the hierarchy that yields the best performance. Then, we use this hierarchy to train the final models.

For this experiment, we will calculate the best hierarchy from a pool of 1, 5, 10, 30, 50, and 100 randomly generated hierarchies.

The results of the different approaches are presented in Fig. 12. HAH clearly outperforms the rest of the hierarchies. With 16 classes, the number of possible hierarchies is huge compared to the number of hierarchies that effectively separate all heterogeneous decision boundaries.

As the number of random hierarchies to choose from increases, the performance slightly increases. However, even for 100 sampled hierarchies, the performance is significantly worse than that of HAH.

Given our synthetic dataset, finding hierarchies that separate all heterogeneous decision boundaries into homogeneous groups is a complex task. Each decision boundary does not overlap with many of the other decision boundaries; it is very unlikely to find a suitable decomposition randomly. For simpler cases, where the set of decision boundaries that do not overlap in the hyperparameter space is small, it is more likely to find a good decomposition by sampling random hierarchies.

E. COMPARISON OF HAH WITH ECOCS

In the literature, there are two ways to try to improve the multi-class classification performance based on binary classifiers. The first one, shown previously, tries to find a structure between the classes. The second one makes use of ensembles. In this section, we will compare our handcrafted hierarchy with the Error Correcting Output Code (ECOC) method. ECOC, originally presented in [12], trains a number of binary classifiers, each of them separating a group of classes from another.

There are different heuristics to group classes for each binary classifier. The key of ECOC is redundancy and diversity. Several classifiers learn each decision boundary. In this way, if one of the classifiers fails, but the rest is correct, they will recover the error. This phenomenon will happen if the errors of the different classifiers are uncorrelated.

In ECOC, each binary classifier assigns one of two labels to all the classes of the original problem. In this way, each class is defined by a unique code based on the different classifiers.

We will try different lengths of codes (8, 12, 24, 64, 256). To create the codes, we follow the randomized hill-climbing heuristic presented in the original paper.

The results are presented in Fig. 13. When presented with the most homogeneous setup, HAH outperforms some of the simpler ECOC variants, but significantly underperforms compared to the most complex. As we inject heterogeneity, the difference with respect to the simpler ECOCs increases. In fact, some of the ECOCs that outperformed the hierarchy for the homogeneous case, underperform in the more heterogeneous one (ECOC 24). The most complex codes (ECOC 64 and 256), consistently outperform the hierarchical approach.

At the first glance, these results contradict our hypothesis. Heterogeneity of decision boundaries is not being taken into account in the design of the codes; however, ECOC seems to outperform our HAH, when the number of codes is big enough.

To understand this, we will decompose the error in bias and variance, as done in [13]. In Fig. 14, we can see this decomposition. Interestingly, the bias error of the hierarchical decomposition is increasingly lower as the heterogeneity increases when compared to all ECOC versions. However, the more complex ECOC approaches consistently have lower variance error.

These results should remind us about the traditional bias-variance trade-off; however, we can look deeper into what type of bias errors are made by the two different approaches. In Fig. 15, we can track where the bias errors are produced. We already stated that the HAH has a lower bias error. This is obvious in two regions: the union of decision boundaries and the parallel to the axis decision boundaries.

The first case can be explained through the voting and the error correction mechanisms, the same mechanisms that

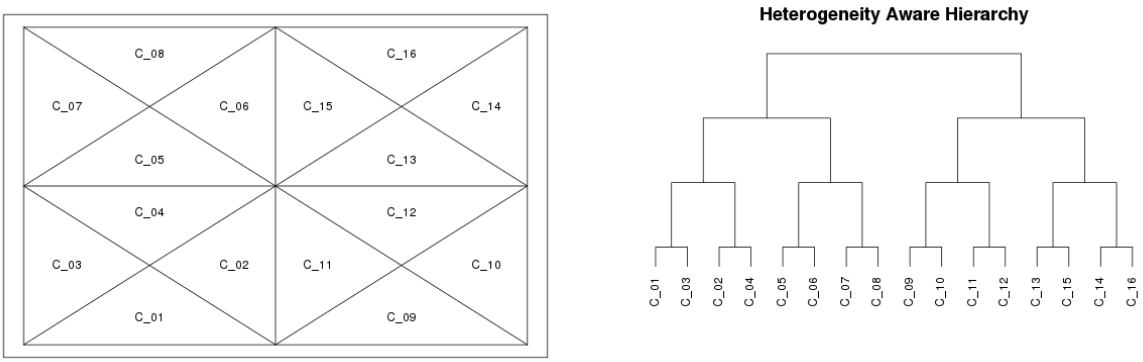

FIGURE 10. Decomposition of the original multi-class problem into a hierarchy of classes.

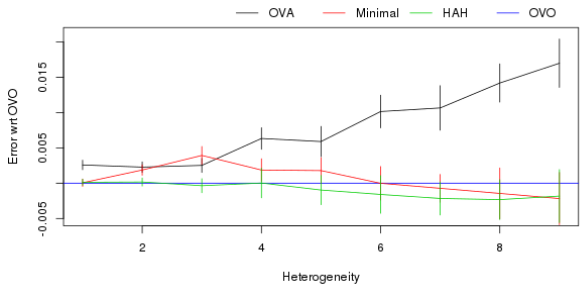

FIGURE 11. Error comparison of OVA, minimal decomposition, and hierarchical decomposition with respect to OVO.

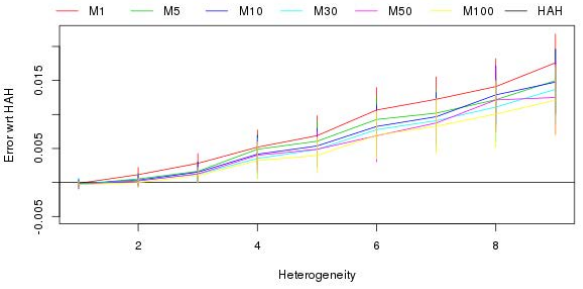

FIGURE 12. Error comparison of Melnikov method, with respect to HAH.

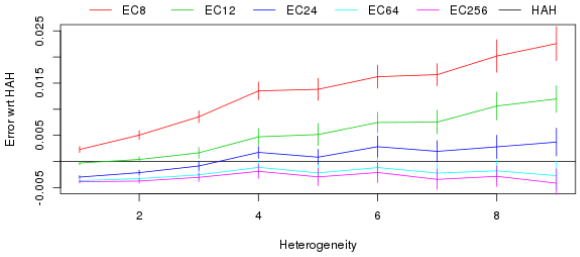

FIGURE 13. Error comparison ECOCs, with respect to the HAH.

lower the variance error. However, the bias error in the parallel to the axis decision boundaries is different. The root of this bias error is in the heterogeneity of decision boundaries. Given our experiment setup, most of the 256 classifiers are mixing diagonal and parallel to the axis decision boundaries; most of them are suffering from the heterogeneity of decision boundaries.

Ensemble approaches like ECOC can significantly increase the classification performance when compared to non-ensemble approaches. However, they are not immune to the problems rooting from the heterogeneity of decision boundaries.

F. A HYBRID SOLUTION COMBINING ECOC AND HETEROGENEITY KNOWLEDGE

For the proposed hybrid approach, we are going to create a variation of ECOC that combines the error-correcting power of ECOC and the decomposition based on the knowledge about the heterogeneity of decision boundaries. We will create specific classifiers to separate the four different regions of the left image in Fig. 7. These four regions combine all parallel to the axis decision boundaries. We will exhaust every possible combination of those 4 regions to train 12 different classifiers.

We will add these 12 classifiers to classifiers created like in ECOC, with the constraint that they are only concerned with diagonal decision boundaries. In total, we are creating 12 codes to separate the parallel to the axis decision boundaries, and 84 to separate the rest.

The results of the comparison can be seen in Fig. 16. The hybrid approach performs increasingly better than the ECOC-256 approach, even if the number of codes is much higher in the latter. The reason behind this can only be the heterogeneity of decision boundaries. By manipulating the codes so that parallel to the axis and diagonal decision boundaries are never mixed, we are decreasing the bias of all the base classifiers. In addition, we are using the error-correcting properties of this approach to reduce the part of the error rooting in the variance, clearly outperforming the hierarchical decomposition.

G. NUMBER OF CLASSES, NUMBER OF CODES, AND TIME

One of the main challenges of multi-class classification problems is the time it takes to train models and output predictions [14]. In this section, we will report the training and

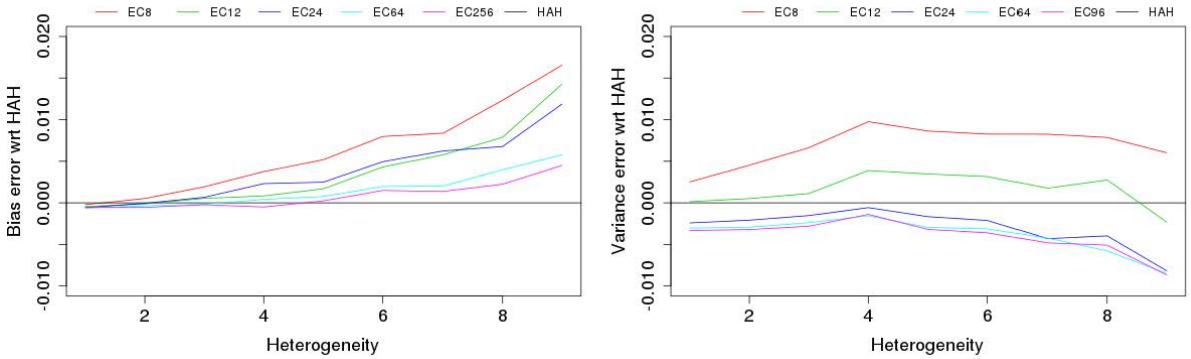

FIGURE 14. On the left, Bias error of ECOC wrt HAH. On the right, Variance error of ECOC wrt HAH.

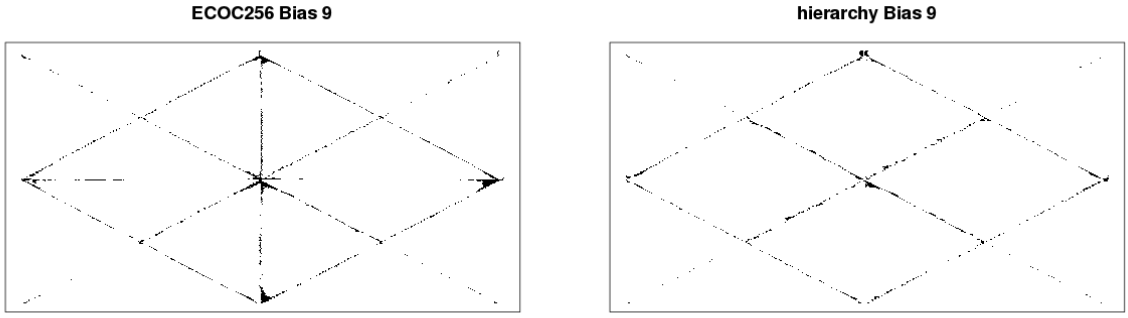

FIGURE 15. On the left, Bias error of ECOC wit 256-long codes. On the right, bias error of hierarchical decomposition.

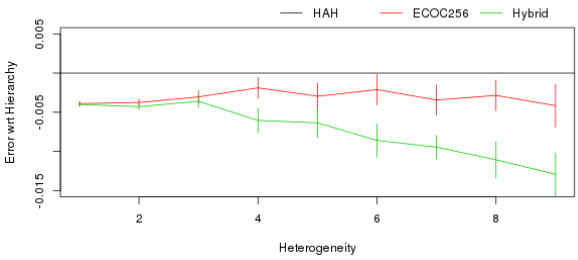

FIGURE 16. Error comparison of the Hybrid approach, and the ECOC approach with 256 codes with respect to the hierarchical decomposition.

cross-validation, and testing times of some of the proposed approaches with a different number of classes: 16, 32, and 64. The total number of instances for training and testing will be the same so that we can isolate the effect of the number of classes. In Table 1, we can see the results of this comparison.

OVO is the standard "brute-force" solution. It achieves higher accuracy than one-vs-all or the multi-output classifier. This increase in accuracy comes at the expense of training and testing time, especially if the number of classes is high. The multi-output classifier is the fastest of them, but also the most affected by the heterogeneity in terms of accuracy, together with OVA.

Minimal decomposition and HAH achieve significantly better results than the multi-output classifier or OVA approaches. In these cases, the benefits of taking into account the heterogeneity of decision boundaries can be seen in the classification performance of our models. When compared to OVO, these approaches achieve similar classification performance. In these cases, the benefit of taking into account the heterogeneity of decision boundaries can be noted in the training and testing times. Note that the testing time of HAH could be further reduced by only evaluating the classifiers connecting the root of the hierarchy and the leaf corresponding to the final prediction [14].

With long enough codes, ECOC achieves the best performances among the off-the-shelf approaches. It does so at the expense of training a high number of classifiers on the whole dataset; it requires more time to both train and test. With the hybrid approach, adding information about the heterogeneity of decision boundaries boosts the classification performance significantly, without further harming the computational time.

H. CONCLUSIONS AND DISCUSSIONS ABOUT THE EXPERIMENT

We have showcased, with ad-hoc decompositions, how a heterogeneity-aware decomposition can improve existing algorithms for multi-class classification. In our case, the problem was explicitly designed so that we have control over the modes of heterogeneity.

TABLE 1. Time comparison in seconds of different approaches and different number of classes. The approaches are sorted (ascending) by the accuracy in classifying the most heterogeneous cases. Double line separation means significant difference in accuracy.

	Training and cv			Testing		
	16 classes	32 classes	64 classes	16 classes	32 classes	64 classes
Multi-output	3.1	4.6	8.5	0.2	0.5	1.2
OVA	9.8	13.6	23.6	1.8	3.6	11.8
2-Fold MO	3.9	4	5.5	1.1	2.1	5.6
OVO	4.8	8	87	5.5	10.8	15.7
HAH	2.4	2.4	3.1	1.7	3.7	15
ECOC 96	103	103	104	19	55	177
Hybrid 96	102	104	102	18	57	177

Another aspect to take into account is the simplicity of our synthetic datasets. We have only three types of decision boundaries: diagonals, perpendicular to the axes, and not contiguous (these do not have any effect on the heterogeneity discussion since they can never overfit). The two "relevant" types of decision boundaries can be easily clustered in homogeneous groups with the minimal or the 2-Fold Multi-output decomposition. Problems with more modes of heterogeneity might not be as easily handled with such straightforward approaches.

However, our synthetic dataset is also, in another sense, quite complex. All classes end up having heterogeneous decision boundaries with respect to the rest of the classes. In this setup, methods based on randomness like the method presented in [6], and the ECOC approach presented in [12] are going to underperform when compared with HAH and the hybrid codes approach respectively. If the total number of heterogeneous decision boundaries is smaller, we expect these methods to be more competitive.

V. MEASURING THE HETEROGENEITY AND APPLICATION TO A REAL DATASET

In this section, we are going to apply our ideas of heterogeneity of decision boundaries to practical use in a real dataset, using state of the art learning algorithms. We are going to use SVM as implemented in the package **liquidsvm**. We define our learning task as the one where we want to find the best SVM model with gaussian kernel, fine tuning the width of the gaussian γ and the regularization parameter C (in the **liquidsvm**, the parameter λ is used instead of C, λ is inversely proportional to C and normalized to the number of instances).

As an example, we are going to use the letter dataset [15], primarily for its interpretability. The dataset has 20 000 instances and 16 features consisting on descriptors of distorted letters from different fonts, corresponding to the 26 letters in English language. Our aim with these experiments is to show that heterogeneity of decision boundaries does indeed happen in real life situations, both in terms of realistic data and realistic learning algorithms. Moreover, this phenomenon can be measured, and can be exploited.

In Figure 17, we show four examples of how the SVM classifier with different hyper-parameter configurations behaves when solving individual decision boundaries. On the top row, we have an example of two heterogeneous decision boundaries, where the regions of maximum performance do not overlap. On the bottom row, we have an example of two homogeneous decision boundaries, where their regions of maximum performance do indeed overlap. In Table 2, we show the results of solving two different decision boundaries with different learning setups. In the first one, we solve the two decision boundaries individually. In the second and third setup, we mix the classes so that the decision boundaries are combined. We run a 10-fold cross-validation and run a paired t-test. In the case of the heterogeneous decision boundaries, we observer a significant difference in favor of solving the decision boundaries individually. In the case of the homogeneous decision boundaries, we observe small differences in favor of solving the decision boundaries individually, however these differences are not significant.

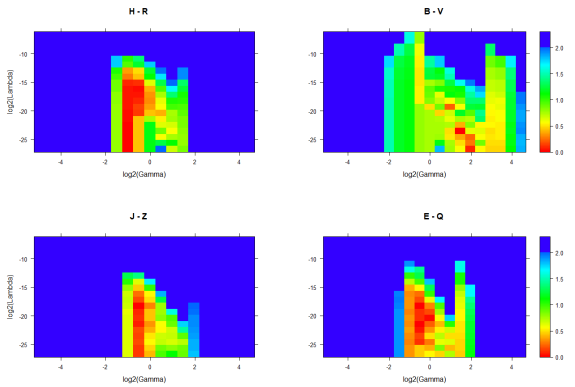

FIGURE 17. Visualization of the relative classification performance of different hyper-parameter configurations for four different decision boundaries. The color gradient corresponds to the relative difference in terms of standard deviations from the best possible performance. On the top row, we have an example of heterogeneous decision boundaries, where the regions of maximum performance do not overlap. On the bottomo row, an example of two decision boundaries that are homogeneous, i.e., their regions of maximum performance overlap.

TABLE 2. The accuracy achievable when solving the decision boundaries of interest individually versus when heterogeneous boundaries are mixed.

	H-R & B-V	HB-RV	HV-BR
Accuracies solving HR and BV	0.991 ± 0.006	0.983 ± 0.007	0.985 ± 0.008
	J-Z & E-Q	JE-ZQ	JQ-EZ
Accuracies solving JZ and EQ	0.999 ± 0.001	0.997 ± 0.003	0.998 ± 0.004

A. MEASURING THE HETEROGENEITY

So far in the paper, we have discussed heterogeneity qualitatively. In our synthetic datasets, we were creating sets

of heterogeneous decision boundaries and we increased or decreased the heterogeneity by increasing or decreasing the number of sets of heterogeneous decision boundaries. In a non-synthetic dataset, we do not have this information a priori, and instead need to find a way to measure the heterogeneity.

For each decision boundary, we are going to compare the accuracy of a classifier with different hyper parameter configurations. Let A_S^i be the array of accuracies of a family of classifiers solving the decision boundary i over the hyper-parameter space S. The plots in Figure 17 are in fact the visualization of different A_S^i, where we have sampled a grid of hyper-parameter configurations. We measure the amount, or degree, of heterogeneity between two decision boundaries i and j with the equation:

$$HDB(i,j) = max(A_S^i) + max(A_S^j) - max(A^i + A^j)_S \quad (1)$$

The last term of the equation corresponds to the selection of the hyper-parameter configuration that produces the highest accuracy of both decision boundaries, as they are combined and solved together. If the two decision boundaries share a region of the hyper-parameter space where they achieve the maximum accuracy, the value of *HDB* is 0, i.e., the decision boundaries are homogeneous. If they do not share the region of maximum performance, the maximum value of the combined $(A^i + A^j)_S$ will not coincide with the individual maxima of A_S^i and A_S^j. In this case, the value of *HDB* would be bigger than 0, indicating that the decision boundaries are heterogeneous, and capturing the degree of the heterogeneity. This measure can be easily expanded to more than two decision boundaries:

$$HDB(1,\ldots,n) = \sum_{i=1}^{n} \max(A_S^i) - \max(\sum_{i=1}^{n} A^i)_S \quad (2)$$

In the next set of experiments, we are going to demonstrate that the heterogeneity of decision boundaries, as measured using HDB, indeed has an effect on classification performance. This will establish that our intuitions and results from Section 4 also apply in this more realistic scenario. For the first experiment, we are going to repeatedly split randomly the letters of the alphabet into two groups. Using a cross-validation scheme, we are going to evaluate the HDB of every split using the training set, then we are going to train a classifier (SVM with the best hyper-parametrization) to distinguish between the two groups of letters. We will name this classifier ''split classifier''. On the validation set, we are going to measure the accuracy of the split classifier and compare it with the accuracy of an OVO classifier trained with all 26 classes, but evaluated on the two groups only. We use OVO as a benchmark, since it treats every decision boundary independently, and we have shown before that it is not affected by the presence of heterogeneous decision boundaries.

On the left of Fig. 18, we can see the relationship between HDB and the difference in accuracy between the OVO classifier and split classifier. HDB is an absolute measure, i.e., it is not normalized to the number of decision boundaries. To explore the effect of the number of classes, we have sampled 30 different splits of the letters, with different number of classes in each split. In the right plot of Fig. in 18, we see the difference in accuracies as a function of the number of classes (counting the majority group in every split).

There is a clear correlation between the HDB and the difference in accuracy. The more heterogeneous the decision boundaries in the split are, the higher the difference in accuracies between an approach that deals with the heterogeneous boundaries separately (OVO) and the approach that deals with them simultaneously (split classifier). There is also an obvious correlation between how heterogeneous the decision boundaries are and the number of classes in the majority group of each split. This is an expected result, since we assume that the set of heterogeneous decision boundaries is uniformly distributed. The more decision boundaries are present in the split, the higher is the expected number of heterogeneous decision boundaries, and thus the higher value of HDB. The extreme case is when we have one class against the rest, where the difference in accuracies approaches 0.

In principle, our notion of heterogeneity of decision boundaries seems to contradict the conclusions of [9]: that there are no significant differences in performance between OVA and OVO. However, this last result points to a phenomenological reason that can reconcile both stances. In the Letters dataset, classes are quite compact. Every individual class is ''equally'' dissimilar from the rest of the classes. It is only when we merge decision boundaries that do not share classes, that we see heterogeneous decision boundaries. In datasets like letters, the OVA decomposition clusters the decision boundaries into homogeneous groups, therefore OVO and OVA can achieve a similar performance in many real world problems. However, we believe that this situation is not universal.

According to this experiment, the difference in accuracy between the OVO classifier and the split classifier is not only correlated to the measure of HDB, but also to the different number of classes in each side of the split. For the next experiment, we want to isolate the effect of the heterogeneity of decision boundaries independently of the number of classes in the split. To this effect, we are going to select the splits that separate the 26 letters into two groups of 13. There are 10 400 600 different possible splits, so exhaustive analysis is clearly infeasible, and we will sample just 150 randomly. Again, we will measure the HDB value and compare it with the difference in accuracies between OVO and the split classifier.

In Fig. 19, we can see the relation between the HDB measured and the difference between the accuracies of approaches that take into account the heterogeneity of decision boundaries and those that not. We can confirm our hypothesis: independently of the number of classes in the split, the more heterogeneity, the bigger the difference between the accuracies of approaches that take into account the heterogeneity of decision boundaries and those that not. We measure the Pearson correlation between these two

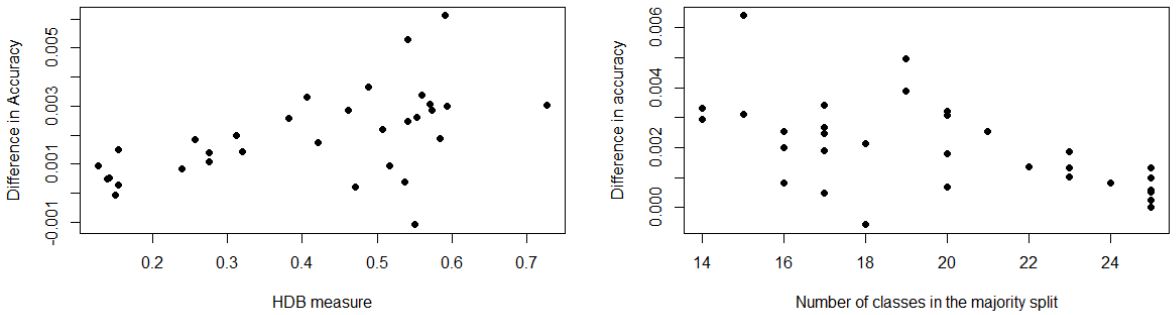

FIGURE 18. On the left, the difference in the accuracy between an OVO classifier and a classifier trained on the two groups of letters as a function of the HDB measure. On the right, the difference in the accuracy between an OVO classifier and a classifier trained on the two groups of letters as a function of the number of classes in the majoritary group.

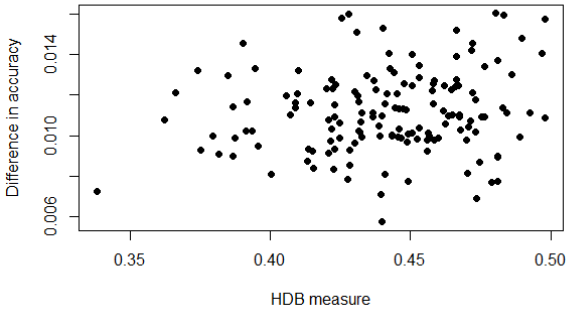

FIGURE 19. Difference in the accuracy between an OVO classifier and a classifier trained on the two groups of letters as a function of the HDB measure.

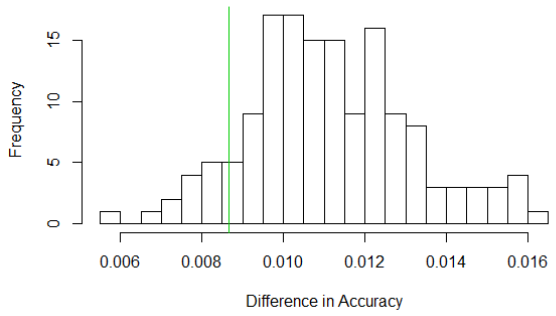

FIGURE 20. Histogram of the difference in accuracies between OVO and the split classifier. Highlighted in green, the split found using the genetic algorithm.

variables and obtain a value of 0.3. We test against the null hypothesis that the two variables are uncorrelated and obtain a p-value 0.01. There exist a clear relation between the HDB measured and the difference in accuracies.

B. EXPLOITING HDB

In this section, we want to find the split of the 26 classes into two groups of 13 where the performance of the split classifier is as close as possible to the performance of the OVO classifier. Evaluating every possible split is not feasible, so we will use the correlation between HDB and the difference in accuracies presented in the previous experiment. Still, we need to evaluate the HDB of every possible split. To avoid this process we are going to use an off-the-shelf genetic algorithm. As a fitness function, we use the HDB of the split measured with the training set in each fold of the cross-validation scheme. We weigh this value with the number of classes, so that the best splits have 13 classes in each group, but the rest of the splits are not discarded in the search process and help the algorithm to find the best possible split.

To evaluate our approach, we will benchmark our candidate split against 150 random splits. In Fig. 20, we can observe the distribution of the difference in the performance of the 150 random splits we have selected. Highlighted in green is the position of the split found by our genetic algorithm. The split optimized for HDB ranks 14th out of 150 in terms of accuracy difference.

Using this methodology of finding the best split, we are going to create a binary hierarchy by greedily selecting the best possible splits in a top-bottom manner. We decide to keep the hierarchy as balanced as possible. Given that, for 26 classes, we need a hierarchy with 5 levels, we will allow the first level to be unbalanced, as long as the splits at the rest of the levels are balanced.

To have a benchmark, we compare the results against a randomly sampled hierarchy and OVO. Results are presented in Table 3. We perform paired t-test comparisons between the different approaches. The hierarchical approach using HDB measurements to create the hierarchy is significantly better than the hierarchical approach using a random hierarchy, with a p-value of 0.04. There is no significant difference between the One-vs-One approach and the hierarchy using HDB (with a p-value of 0.22).

TABLE 3. Accuracies of One-vs-One, hierarchical classification using a random hierarchy, and hierarchical classification using HDB measure to create the hierarchy.

One-vs-One	Random hierarchy	HDB hierarchy
0.9753 ± 0.0054	0.9692 ± 0.0039	0.9737 ± 0.0040

VI. LITERATURE REVIEW

What makes classification hard has been extensively studied, but only in the binary case. In [4], the difficulty of binary classification problems is described by several factors, studying geometrical properties of the decision boundary. It is indicated that the study of the difficulty of multi-class classification problems could be done through the summary of individual class-vs-class decision boundaries. The findings of our work imply that the total complexity of the multi-class problem is not simply the addition of the complexities of the individual decision boundaries but also depends on the heterogeneity of decision boundaries. The total complexity can be more than the sum of the parts, i.e., of the complexities of the individual problems.

In [10], the authors provide a theoretical explanation of whether to choose "flat" or "hierarchical" approaches in multi-class classification. In their formalism, this means eliminating or creating intermediate nodes in the hierarchy, i.e., whether to group or not decision boundaries. Their discussion is based on the study of error generalization bounds. Analyzing the unbalance of classes, the authors suggest to use shallow hierarchies for well-balanced problems and use deeper hierarchies for unbalanced cases, i.e., grouping decision boundaries when the classes have similar number of instances and separate them otherwise.

Although not directly, the unbalance of classes can be related to the heterogeneity of decision boundaries as we have described it in this paper. Many hyperparameters can be related to the number of instances per class (number of neighbors in KNN, number of instances per leaf in CART, slack variables in support vector machines, and so on). On the other hand, in our experiments, we have shown an example of a completely shallow hierarchy (multi-output classifier) performing significantly worse than a deep hierarchy (HAH) for a completely balanced problem.

In [16], the authors observe that the performance of C4.5 on multi-class classification problems decreases if the classes are grouped. The authors attribute this behavior to the fact that the instances of the grouped classes may lie in different areas of the feature space. It would be very interesting to analyze this problem from the point of view of heterogeneity of decision boundaries. In addition, the datasets where this effect is more acute happen to be those with higher imbalance of classes.

Reference [17] presents a novel method to train linear classifier. Instead of learning a model on the data space, they propose to learn models in the weight space. There is a clear correspondence between the weight space in their work and our hyper-parameter space. In their case, the weights are directly related to the final model; in our case, the hyper-parameters define the classifier created by a learning algorithm.

In the literature, ensemble methods like bagging, stacking, or boosting are always among the most competitive methods. The core idea behind the success of ensemble methods is to train diverse classifiers and ensemble their outputs. ECOC [12], is an ensemble method tailored specifically for multi-class classification. A number of binary classifiers are trained to separate a group of classes from the rest, i.e., grouping decision boundaries in different ways.

In our work, we have shown that ensemble approaches have benefits that are independent of the heterogeneity of decision boundaries; however, ECOC is not immune to the adverse effects of the heterogeneity of decision boundary. If most of the binary classifiers are affected, the final ensemble will as well.

Ensemble methods are one of the ways to improve multi-class classification performance. The other direction is transforming the multi-class problem into a collection of binary classifier. The simplest approaches are OVO or OVA. In [7], the authors compare these approaches using different base learners. In general, OVO outperforms OVA, and this in turn outperforms the multioutput classifier. In [18], OVA is proven to outperform the multioutput classifier using random forest as base learner. In [19], OVO is proven to outperform OVA when training depp neural networks from scratch.

Other approaches try to find structure in the classes. Examples can be found in [20] and [14], [21]. In [20], the authors build a chain of classifiers, while in [14], [21], the authors build a hierarchy of classes. The approaches that we have discussed in this paper fall into this category.

In our work, we have based our decomposition strategies on the knowledge about the heterogeneity of decision boundaries. In the literature, these approaches have been based on the similarity of classes. There are two main ways of measuring the similarity of classes: geometrical distance of centroids [21], and classifier based distance [14]. The latter consists of first training a classifier on the data, and estimate which classes are similar or dissimilar based on the classification performance. None of these similarities are directly linked to the heterogeneity of decision boundaries, although we speculate that the classifier based similarity might be correlated.

In the field of regression, heterogeneity has already been treated. Reference [22] presents MARS, a regression method that uses splines to create different fits in different parts of the feature space.

VII. CONCLUSION AND DISCUSSIONS

The main contribution of this paper is the identification and description of a previously untreated phenomenon: how heterogeneity of decision boundaries can affect the performance of multi-class classification performance.

The concept of heterogeneity in decision boundaries speaks about the relationship of these decision boundaries through the lens of a learning algorithm. Two decision boundaries can be heterogeneous under one learning algorithm and homogeneous under another. Heterogeneity of decision boundaries is not a direct result of the geometrical properties of the decision boundaries, but how those properties relate to each other through the learning algorithm. If two decision

boundaries are heterogeneous, a learning algorithm will solve them better individually than simultaneously.

We have identified the inductive bias introduced by the hyperparameter tuning as a source of heterogeneity. Hyperparameter tuning is generally done with the help of validation schemes like cross-validation. This is a strong example of inductive bias: among all possible hypotheses, we are only going to select the ones that can be found by the particular learning algorithm and hyperparameter configuration chosen. If the hyperparameter configurations that achieve maximum performance for two or more decision boundaries do not overlap, it will result in a performance loss.

We have shown how one part of the classification problem can negatively affect another part of the problem in terms of classification performance. To the best of our knowledge, this is the first comprehensive description of pure algorithmic bias mechanism in classification. Some decision boundaries are incorrectly estimated by the presence of other decision boundaries; this is especially true when we compare the same learning algorithms acting individually on the same decision boundaries. Research of fairness in machine learning and AI generally focuses on bias rooting from the quality of the data [23]. Heterogeneity of decision boundaries looks like a promising direction to study fairness in AI.

We have studied how the heterogeneity of decision boundaries affects different approaches to solve multi-class classification problems. The main lesson we have learned is that whenever a learning algorithm has to solve two or more heterogeneous decision boundaries, the classification performance decreases.

We have also shown that the information about the heterogeneity of decision boundaries can be used to devise approaches that obtain better classification performances. The information about the heterogeneity of decision boundaries can also be used to devise approaches that allow us to get similar classification performance at a lower computational cost as compared to approaches that do not use information about heterogeneity. In the field of multi-class classification, this is important, since generally computation time increases significantly with the number of classes.

In our work, we have designed the synthetic dataset with only two types of decision boundaries so that we could easily identify and group them in homogeneous groups. Finding the same structures in real-life problems might not be as simple. Some of the proposed decompositions, like the minimal or 2-Fold multi-output decompositions, are very restrictive; more complex structures in the heterogeneity might not be as easily represented with these schemes. We believe that other approaches, such as ECOC or hierarchical decompositions, are potentially more practical since they are flexible enough to represent complex hidden structures among decision boundaries.

We have applied our findings about the heterogeneity of decision boundaries on a real dataset using a competitive learning algorithm like SVM. We have proposed a way to measure the degree of heterogeneity of decision boundaries and have corroborated its effects on classification performance. Finally, we have used the measurement of the heterogeneity of decision boundaries to extract a hierarchy of classes that outperforms other hierarchies that do not take this information into account.

We have used multi-class classification as a natural field to showcase the heterogeneity of decision boundaries. However, indirectly, we have also shown the effects on binary classification by comparing OVA and OVO. As a hypothesis, one could further decompose a binary problem into a collection of smaller binary problems defining new decision boundaries that might be heterogeneous. How to do it is not obvious. However, positive results in ensemble methods for binary classification problems indicates the potential of this idea, especially if we consider the computational time to output a new prediction for ensemble methods.

An interesting final remark is how we have used learning algorithms to describe decision boundaries. Generally, the complexity of data is too big for humans to understand and visualize its characteristics easily. In quantum mechanics, nature is described by interacting with it through the process of measure. Here, we have taken a similar approach, describing the nature of the data by how it interacts with our measuring instruments (performance of classifiers learned by a particular algorithm). In addition, we have described how decision boundaries can interact with another through the effects of learning.

REFERENCES

[1] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[3] R. Díaz-Uriarte and S. A. de Andrés, "Gene selection and classification of microarray data using random forest," *BMC Bioinf.*, vol. 7, no. 1, pp. 1–13, Dec. 2006.

[4] 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, Aug. 2002.

[5] J. Fürnkranz, "Round Robin classification," *J. Mach. Learn. Res.*, vol. 2, pp. 721–747, Mar. 2002.

[6] V. Melnikov and E. Hüllermeier, "On the effectiveness of heuristics for learning nested dichotomies: An empirical analysis," *Mach. Learn.*, vol. 107, nos. 8–10, pp. 1537–1560, Sep. 2018.

[7] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognit.*, vol. 44, no. 8, pp. 1761–1776, Aug. 2011.

[8] P. D. Moral, S. Nowaczyk, A. Sant'Anna, and S. Pashami, "Pitfalls of assessing extracted hierarchies for multi-class classification," 2021, *arXiv:2101.11095*.

[9] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, Dec. 2004.

[10] R. Babbar, I. Partalas, E. Gaussier, and M. R. Amini, "On flat versus hierarchical classification in large-scale taxonomies," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1–9.

[11] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1st ed. Routledge, 1984, doi: 10.1201/9781315139470.

[12] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Intell. Res.*, vol. 2, pp. 263–286, Jan. 1995.

[13] E. Kong and T. G. Dietterich, "Error-correcting output coding corrects bias and variance," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 313–321.

[14] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multi-class tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, vol. 23, no. 1, pp. 163–171.
[15] P. W. Frey and D. J. Slate, "Letter recognition using Holland-style adaptive classifiers," *Mach. Learn.*, vol. 6, no. 2, pp. 161–182, 1991.
[16] A. Hoffmann, R. Kwok, and P. Compton, "Using subclasses to improve classification learning," in *Machine Learning: ECML*, L. De Raedt and P. Flach, Eds. Berlin, Germany: Springer, 2001, pp. 203–213.
[17] C. Lee and S. Woo, "Linear classifier design in the weight space," *Pattern Recognit.*, vol. 88, pp. 210–222, Apr. 2019.
[18] M. N. Adnan and M. Z. Islam, "One-vs-all binarization technique in the context of random forest," in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, 2015, pp. 385–390.
[19] E. Okafor, M. Groefsema, S. He, L. R. B. Schomaker, and M. A. Wiering, "One-vs-one classification for deep neural networks," *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107528.
[20] W. Liu, I. W. Tsang, and K.-R. Müller, "An easy-to-hard learning paradigm for multiple classes and multiple labels," *J. Mach. Learn. Res.*, vol. 18, no. 94, pp. 1–38, 2017.
[21] V. Vural and J. G. Dy, "A hierarchical method for multi-class support vector machines," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, pp. 831–838.
[22] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Statist.*, vol. 19, no. 1, pp. 1–141, Mar. 1991.
[23] D. Danks and A. J. London, "Algorithmic bias in autonomous systems," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4691–4697.

PABLO DEL MORAL received the master's degree in data science from the University of Granada and the master's degree in nuclear, particle and astrophysics from the Technical University of Munich. He is currently pursuing the Ph.D. degree in data mining with the Center for Applied Intelligent Systems Research, Halmstad University, Sweden.

SŁAWOMIR NOWACZYK received the M.Sc. degree from the Poznan University of Technology, in 2002, and the Ph.D. degree from the Lund University of Technology, in 2008. He is currently a Professor in machine learning at the Center for Applied Intelligent Systems Research, Halmstad University, Sweden. He is the Research Leader of the School of Information Technology, Halmstad University. During the last decade, his research focused on knowledge representation, data mining, and self-organizing systems, especially in large and distributed data streams, including unsupervised modeling. He has led multiple research projects related to applying artificial intelligence and machine learning in many different domains, such as transport and automotive, energy, smart cities, as well as healthcare. In most cases, this research was done in collaboration with industry and public administration organizations—inspired by practical challenges and leading to tangible results and deployed solutions. He is a Board Member for the Swedish AI Society.

SEPIDEH PASHAMI received the Ph.D. degree from the AASS Research Centre, Örebro University, Sweden, in 2016. She is currently a Senior Researcher at RISE and a Lecturer at the Center for Applied Intelligent Systems Research, Halmstad University. She has been involved as a Researcher and the Research Leader in several projects, such as EVE, In4Uptime, ARISE, and HEALTH together with Volvo Group AB, applying machine learning techniques for predictive maintenance of heavy-duty vehicles. Her research interests include predictive maintenance, interactive machine learning, causal inference, and representation learning.