





# Aggregated Performance Measures for Multi-class Classification

Damian Pȩszor<sup>1,2</sup>(✉)  and Konrad Wojciechowski<sup>2</sup> 

<sup>1</sup> Silesian University of Technology, Akademicka 2A, 44-100 Gliwice, Poland  
damian.peszor@polsl.pl

<sup>2</sup> Polish-Japanese Academy of Information Technology, ul. Koszykowa 86,  
02-008 Warszawa, Poland

{dpezor,kwojciechowski}@pjwstk.edu.pl

<https://www.polsl.pl>, <http://bytom.pja.edu.pl>, <https://www.pja.edu.pl/en/>

**Abstract.** This paper aims to present an approach to generalisation of performance measures commonly used in binary classification to the field of multinomial classification to use them in hyperparameter estimation for various machine learning methods and similar techniques. The classical approach is to use a binary classification wherein each representative of any incorrect class is considered as a representative of an umbrella class being a union of all incorrect classes. Such an approach leads to the removal of important information from the classification process and therefore to the lower value of each experiment for the determination of the gradient when trying to optimise hyperparameters. We propose aggregated performance measures that can be thought of as an analogue of classical ones. The proposed measures better represent the multinomial nature of such algorithms and obtain more valuable information that allows selecting the correct direction while analysing the gradient of the resulting measures.

**Keywords:** Multi-class classification · Multiclass classification · Multinomial classification · Performance · Sensitivity · Specificity · Accuracy

## 1 Introduction

Classification is one of the most popular problems in terms of the application of machine learning [1, 2] or generally, statistics-based techniques. It is widely used in many different fields, from the classification of obstacles of flying UAVs [3], to the classification of patients potentially suffering from neurodegenerative diseases [4], to facial recognition [5]. Many classification algorithms are dependent on hyperparameters, which have to be adjusted in order for the algorithm to work properly in a given domain. This becomes computationally expensive if multiple classifiers are used to lower the error probability [6]. Such adjustment is rarely based on domain knowledge, often done in an automatic way by searching the hyperparameter space for the optimum performance of the classification obtained, which can also be a stop criterion for neural network training [7–9].

Few measures of performance are as prevalent in terms of usage as a tool in the evaluation of a specific parameterisation of the algorithm as the Accuracy itself, defined in binary classification as a ratio of correctly evaluated data points to all of them (Eq. 1). Its popularity is mainly caused by an easy interpretation as a fraction which has been correctly classified. The main issue with using Accuracy is that it is not very informative. Parameterisations that allow for many possibly harmless false positives result in the same value as in the case of equally many possibly disastrous false negatives and the other way around.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

To distinguish between such situations, True Positive Rate (TPR) and True Negative Rate (TNR) are mostly used since [10]. TPR, often called Sensitivity or Recall, is the rate of correct recognition of data points as members of a class they belong with, to this number increased by the number of incorrect classifications as another class (Eq. 2). TPR measures the ability of the parameterisation of an algorithm to detect the important class while ignoring the incorrect detections. This is important when a data point of a given class has to be detected to avoid potential high cost, while incorrect detection can be dealt with.

$$TPR = \frac{TP}{TP + FN} \quad (2)$$

True Negative Rate (Specificity), is defined as the rate of correctly rejected data points with this value increased by the number of data points incorrectly accepted as a given class (Eq. 3). This measure is useful when the cost of incorrect detection of a given class is very high, while incorrectly omitting such detection is not. Note that TNR ignores whether an algorithm can actually detect a given class for the data points that belong to it, so the use of both TPR and TNR together is important in most scenarios, with various ratio of importance.

$$TNR = \frac{TN}{TN + FP} \quad (3)$$

The nature of those measures of performance is based on the problem of binary classification, wherein only two classes exist; one representing a feature and one that does not. They are however used as well in multinomial classification, wherein each data point is considered correctly recognised if and only if the predicted class is the actual one. In any other case, the data point is labelled as incorrectly recognised, no matter whether the correct class will be the second-best or the worst fit. Such a one-vs-all approach produces as many results as cardinality of the set of classes  $|C|$ . While some research has been done to aggregate measures for each class [11], the result in multi-value form is not a clear indicator for the direction of change of hyperparameters in the case of machine learning and therefore does not fulfil the necessary role. To use those values as an indicator of whether a given hyperparameter value improved or worsened classification, one has to aggregate those measures  $M_i$  in some way, wherein the most obvious solution is to average them out, as in Eq. 4.

$$M = \frac{1}{|K|} \sum_{k=1}^{|K|} M_k \quad (4)$$

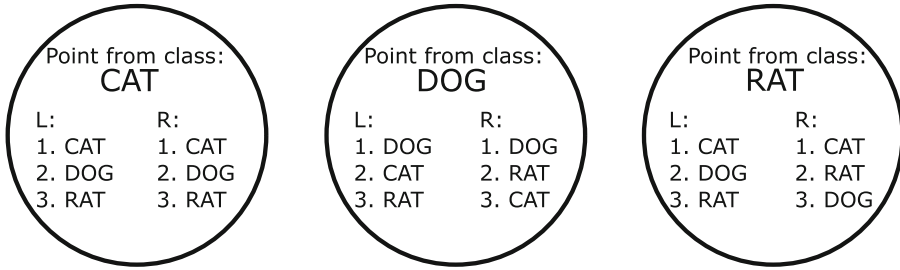
Such an approach is called macro-averaging. [12] and [13] present macro-averaged multinomial classification measures for Accuracy, Error Rate, Recall, Precision, and Fscore. [12] also presents an alternative to macro-averaging in form of the so-called micro-averaging, which is defined as in Eq. 5.

$$\mu = \frac{\sum_{k=1}^{|K|} N(M_k)}{\sum_{k=1}^{|K|} D(M_k)} \quad (5)$$

wherein by  $N(M)$  we mean the numerator of a given measure  $M$  and by  $D(M)$  we mean its denominator. The difference between those kinds of averaging corresponds to resolving the problem of the equal contribution classes with varying number of examples. In micro-averaging, more numerous classes for which there is a stronger belief that the value of measure represents the characteristics of the problem, have a stronger influence over the aggregated measure. While such an approach seems reasonable, the reason for classification might be based on the assumption that the less-represented classes, e.g. uncommon diseases, are interesting ones, so such approach might suppress the important objectives of the classification. Further measures of multi-class performance are presented in [14], mostly aggregations of simple measures presented in this paper.

Neither of those approaches to the aggregation of multi-class performance measures solves the inherent problem of being an extension of a binary approach. Therefore in many cases, the value of  $M_i$  (or  $\mu_i$ ) and  $M$  (or  $\mu$ ) does not change. Consider a simple example. Let there be 3 classes, namely,  $K = \{K_1 = \text{“Cat”}, K_2 = \text{“Dog”}, K_3 = \text{“Rat”}\}$  and two sets of parameters,  $L$  and  $R$ . For simplicity, consider a single data point for each class, as in Fig. 1.

For a binary classification problem,  $L$  and  $R$  give the same results, the predicted class is either correct (cat and dog) or incorrect (rat) so the measures of performance are of equal value. Therefore, a change between  $L$  and  $R$  does not provide any information about which is better for the given problem as if both would produce the same result. That is, however, not the case. Typically, classification produces a measure of similarity between the model of the class and a given sample. The predicted class is the one that corresponds to the highest measure of similarity. Each data point can be described by the measures of similarity for each class. As such, for each data point, there exists a list of classes ordered in descending order by the similarity measure. While for the cat sample the hyperparameters of the classification algorithm do produce the same results, it is not true for all cases. The dog sample contains a different permutation of classes, however, no additional information about how well the classification performs is known. One might deduce that  $L$  parameterisation tends to represent dogs and cats similarly, while  $R$  does that to a lower degree, but such information is useful only in the case of human design. The third sample, wherein the data point corresponding to a rat was labelled as a cat with both  $L$  and



**Fig. 1.** Example of classification results for the multi-class problem with 3 classes and one data point for each of them, represented by a circle.  $L$  and  $R$  are sets of parameters, for each the order of labels from most probable to the least probable is presented.

$R$  parameterisations, is, however, important. Note that in both cases the rat sample was recognised as a cat, and therefore the binary approach would yield the same result. The gradient of the performance measure would not indicate which set of hyperparameters is better, whether  $L$  or  $R$ . It is however clear that  $R$  performs better than  $L$  since  $K_3$  is the second rather than the third-best fit. This additional information allows to select the correct direction in the hyperparameter space and to avoid the pitfalls associated with fluctuations related to the change between the correct and incorrect estimation of a few data points as a result of a slight adjustment to parameters.

In this paper, we propose a method of aggregating performance measures that is dependant on the position of classes on the list of potential classes ordered by the similarity measure. This allows for a better selection of the parameterisation of a classification algorithm due to a higher sensitivity to the overall behaviour of the method rather than only to the binarized result of classification being either correct or incorrect in terms of the given sample. This approach is especially useful whenever a human user is to determine the final result of classification based on options selected by the algorithm, such as in the case of facial re-identification or counselling systems. The presented approach can be placed in the ontology of performance measures presented in [15]. The proposed solution fits the second group in the first dimension, that is to say, it uses the confusion matrix in conjunction with additional information - which can be thought of as an extension of the confusion matrix in the third dimension by the threshold value. In the second dimension, the best fit for our proposal is the extension of the deterministic classifiers branch. The third dimension of ontology is defined by the measure that is the basis for aggregated variant, that is, aggregated Accuracy corresponds to single-value measures, but the pair of Sensitivity-Specificity corresponds to double-valued measures and so on. In the ontology tree, it is clear, that the proposed method belongs to the branch defined by Multiclass Focus.

## 2 Method

### 2.1 Classification of a Single Data Point

Let us consider a single data point  $o_i$  which belongs to the set of all data points  $\forall i : o_i \in O$ . Moreover, the data point  $o_i$  belongs to the set of all data points that share the same class  $O_k$ , which belongs to the set of all represented classes  $\forall k : k \in K$ , such that  $o_i \in O_k$  and  $\forall k : O_k \subset O$ .

Let us, therefore, denote all correctly recognised data points of class  $k$  as a set of true positives, i.e.  $TP_k$ . Similarly, we will define data points belonging to the class  $k$  but recognized as a different class as a set  $FP_k$ .

In the classical approach, we would define the correct recognition as a result of classification, wherein the list of similarity measures between  $o_i \in O_k$  and models of classes  $K$  contains the true class  $k$  on position  $p_k(o_i)$  such that  $p_k(o_i) = 1$ . The set  $TP_k$  would therefore consist of all such  $o_i$ , which would be reflected in its cardinality as in Eq. 6.

$$|TP_k| = \sum_{o_i \in O_k} \begin{cases} 1, & \text{if } p_k(o_i) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

We can, however, use a broader definition of correct recognition. One could, for example, assume that the correct recognition is defined by  $p_k(o_i) \leq 2$  so that either of the first two positions is acceptable. That would be acceptable in many scenarios, such as in the case where a human operator uses classification to narrow the number of possibilities (e.g., in the case of facial recognition). In such a case, several most probable classes are important, and the need for  $p_k(o_i) = 1 | k : o_i \in O_k$  is limited. One could therefore use a threshold value  $t$  and rewrite Eq. 6 as follows in Eq. 7.

$$|TP_{k,t}| = \sum_{o_i \in O_k} \begin{cases} 1, & \text{if } p_k(o_i) \leq t \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Since every data point  $o_i \in O_k$  belongs to either  $TP_{k,t}$  or  $FN_{k,t}$  for any given  $t$ , one can easily define cardinality of  $FN_{k,t}$  as follows in Eq. 8.

$$|FN_{k,t}| = |O_k| - |TP_{k,t}| \quad (8)$$

Every point outside of  $O_k$  is either a true negative or a false positive. The cardinality of the set of all true negatives for a given class and threshold  $|TN_{k,t}|$  can therefore be defined as in Eq. 9.

$$|TN_{k,t}| = \sum_{o_i \in O - O_k} \begin{cases} 1, & \text{if } p_k(o_i) > t \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

This allows us to define the cardinality of the set of all false positives  $|FP_{k,t}|$  for a given class as in Eq. 10, knowing that  $O_k$  is composed of all true positives and false negatives.

$$|FP_{k,t}| = |O| - |O_k| - |TN_{k,t}| \quad (10)$$

## 2.2 Aggregation Over Classes and Thresholds

Let us now consider one of the performance measures, namely, the True Positive Rate as defined in Eq. 2. It can be redefined for a given class  $k$  at some threshold  $t$  as in Eq. 11.

$$TPR_{k,t} = \frac{|TP_{k,t}|}{|TP_{k,t}| + |FN_{k,t}|} \quad (11)$$

Which allows us to redefine the intuitive aggregation of TPR over classes based on Eq. 4 as in Eq. 12.

$$\overline{TPR}_t = \frac{1}{|K|} \sum_{k=1}^{|K|} TPR_{k,t} = \frac{1}{|K|} \sum_{k=1}^{|K|} \frac{|TP_{k,t}|}{|TP_{k,t}| + |FN_{k,t}|} \quad (12)$$

While a domain might indicate that the value of  $t$  should be higher than 1, using any given single number does not solve the problem of an uninformative gradient. Actually, higher  $t$  tends to increase the problem due to the broader range of parameterisation resulting in the same value. Instead of focusing on a single value of  $t$ , we propose to aggregate the measure over all possible values, so that the change of  $p_k(o_i)$  will result in the change of the value of the given measure. Note that the effect on TPR of lower  $t$  values should never be lower than the effect of higher  $t$  as in Eq. 13.

$$\frac{f(t)}{f(t+1)} > 1 \quad (13)$$

If this is fulfilled, one can aggregate the measures over different  $t$  using a simple sum as in Eq. 14.

$$M = \sum_{t=1}^{|K|-1} f(t) \overline{M}_t \quad (14)$$

Let us define  $f(t) = t^{-m}$  for any  $m \geq 0$ . The  $m$  parameter corresponds to the influence of further positions on the measure. In the general case, we assume that the value of  $m = 1$  is quite universal, although specific domains might benefit from adjusting it accordingly to domain demands. For the measure of Sensitivity, Eq. 14 becomes therefore Eq. 15. Note that  $t = |K|$  means that every data point will be considered as True Positive, no matter the  $p_k(o_i)$ , so there is no point in including it as it does not contain any information.

$$TPR = \sum_{t=1}^{|K|-1} \frac{\overline{TPR}_t}{t^m} = \frac{1}{|K|} \sum_{t=1}^{|K|-1} \sum_k \frac{|TP_{k,t}|}{|O_k| t^m} \quad (15)$$

Substituting Eq. 7 in Eq. 15 one can change the order of summations over classes and over the thresholds getting Eq. 16.

$$TPR = \frac{1}{|K|} \sum_k \left( \frac{1}{|O_k|} \sum_{t=1}^{|K|-1} \left( \frac{1}{t^m} \sum_{o_i \in O_k} \begin{cases} 1, & \text{if } p_k(o_i) \leq t \\ 0, & \text{otherwise} \end{cases} \right) \right) \quad (16)$$

Similarly, the order of summations over thresholds and over data points of a given class can be changed leading to the modification of the weight of a single data point for the calculated measure as in Eq. 17.

$$TPR = \frac{1}{|K|} \sum_k \left( \frac{1}{|O_k|} \sum_{o_i \in O_k} \sum_{t=1}^{|K|-1} \begin{cases} \frac{1}{t^m}, & \text{if } p_k(o_i) \leq t \\ 0, & \text{otherwise} \end{cases} \right) \quad (17)$$

wherein the conditional case can be represented by a different initialisation of the summation as in Eq. 18.

$$TPR = \frac{1}{|K|} \sum_k \left( \frac{1}{|O_k|} \sum_{o_i \in O_k} \sum_{t=p_k(o_i)}^{|K|-1} \frac{1}{t^m} \right) \quad (18)$$

One can notice that the last summation in Eq. 17 is actually a difference between two harmonic numbers as presented in Eq. 19, wherein vacuous summation defines  $H_0^m$ . For generality, we use  $H^m$  to denote generalised harmonic number of order  $m$ .

$$TPR = \frac{1}{|K|} \sum_k \left( \frac{1}{|O_k|} \sum_{o_i \in O_k} \left( H_{|K|-1}^m - H_{p_k(o_i)-1}^m \right) \right) \quad (19)$$

Which can then be simplified to the form of Eq. 20.

$$TPR = H_{|K|-1}^m - \frac{1}{|K|} \sum_k \left( \frac{1}{|O_k|} \sum_{o_i \in O_k} H_{p_k(o_i)-1}^m \right) \quad (20)$$

Notice that the summations do not consider the same data point twice, that is to say, the calculation is actually linear ( $\mathcal{O}(|O|)$ ) in terms of complexity. The values of  $H^m$  can be precalculated in an array, which allows for easy access. The entirety of the value is therefore quite efficient to calculate, despite the added complexity behind the aggregation of multiple thresholds.

### 2.3 Normalisation

While the inner sum of Eq. 20 is enough to control the direction of change in the hyperparameter space, it might be beneficial to use a normalised version of the aggregated TPR, especially when one wants to combine it with other measures, as in the case of F1-score. The minimal value of the aggregated TPR is achieved in the pessimistic case, whenever  $\forall i \wedge k : o_i \in O_k, p_k(o_i) = |K|$ . It is easy to

notice that in this particular case, the aggregated TPR is zero. The maximum value is achieved whenever the perfect classification occurs - that is to say, when  $\forall i \wedge k : o_i \in O_k, p_k(o_i) = 1$  in which case the summation equals zero. Since the first summand depends on the class count, the normalised aggregated TPR is defined as in Eq. 21.

$$T\hat{P}R = 1 - \frac{1}{|K|H_{|K|-1}^m} \sum_k \left( \frac{1}{|O_k|} \sum_{o_i \in O_k} H_{p_k(o_i)-1}^m \right) \quad (21)$$

## 2.4 The Case of Specificity

We take a similar approach in the case of True Negative Rate defined in Eq. 3. Redefined for a given class  $k$  at a given threshold  $t$  it becomes Eq. 22

$$TNR_{k,t} = \frac{|TN_{k,t}|}{|TN_{k,t}| + |FP_{k,t}|} \quad (22)$$

Which we aggregate over classes as in Eq. 23.

$$\overline{TNR}_t = \frac{1}{|K|} \sum_{k=1}^{|K|} TNR_{k,t} = \frac{1}{|K|} \sum_{k=1}^{|K|} \frac{|TN_{k,t}|}{|TN_{k,t}| + |FP_{k,t}|} \quad (23)$$

This allows us to aggregate over different values of the threshold in Eq. 24 as in the case of Sensitivity.

$$TNR = \sum_{t=1}^{|K|-1} \frac{\overline{TNR}_t}{t^m} = \frac{1}{|K|} \sum_{t=1}^{|K|-1} \sum_k \frac{|TN_{k,t}|}{(|O| - |O_k|)t^m} \quad (24)$$

Again, we substitute Eq. 9 in Eq. 24 and change the order of summations getting Eq. 25.

$$TNR = \frac{1}{|K|} \sum_k \left( \frac{1}{|O| - |O_k|} \sum_{t=1}^{|K|-1} \left( \frac{1}{t^m} \sum_{o_i \in O - O_k} \begin{cases} 1, & \text{if } p_k(o_i) > t \\ 0, & \text{otherwise} \end{cases} \right) \right) \quad (25)$$

Changing the order of summations over thresholds and over data points outside of a given class leads again to the weight factor in condition as in Eq. 26.

$$TNR = \frac{1}{|K|} \sum_k \left( \frac{1}{|O| - |O_k|} \sum_{o_i \in O - O_k} \sum_{t=1}^{|K|-1} \begin{cases} \frac{1}{t^m}, & \text{if } p_k(o_i) > t \\ 0, & \text{otherwise} \end{cases} \right) \quad (26)$$

Which can be represented as changing the range of summation as in Eq. 27.

$$TNR = \frac{1}{|K|} \sum_k \left( \frac{1}{|O| - |O_k|} \sum_{o_i \in O - O_k} \sum_{t=1}^{p_k(o_i)-1} \frac{1}{t^m} \right) \quad (27)$$



The last summation in Eq. 27 is a harmonic number as presented in Eq. 28, wherein vacuous summation defines  $H_0$ .

$$TNR = \frac{1}{|K|} \sum_k \left( \frac{1}{|O| - |O_k|} \sum_{o_i \in O - O_k} H_{p_k(o_i) - 1}^m \right) \quad (28)$$

Notice again that the values of  $H^m$  can be precalculated in an array, similarly to the per-class multiplier. The computational complexity can therefore be defined as  $\mathcal{O}((|K| - 1)|O|)$ . It is also useful to note that the calculation of both aggregated Sensitivity and aggregated Specificity can be easily combined.

The normalisation of aggregated TNR is quite more involved than in the case of TPR. For most cases, it is enough to assume that the minimal value is achieved when  $\forall i \wedge \forall k : o_i \notin O_k, p_k(o_i) = 1$ . In this case, the minimum value would be zero. Similarly, the maximum value would be achieved when  $\forall i \wedge \forall k : o_i \notin O_k, p_k(o_i) = |K|$ , which is to say that all incorrect classes would be classified as least probable. That would lead to the normalised aggregated TNR as in Eq. 29.

$$T\hat{N}R = \frac{1}{|K|H_{|K|-1}^m} \sum_k \left( \frac{1}{|O| - |O_k|} \sum_{o_i \in O - O_k} H_{p_k(o_i) - 1}^m \right) \quad (29)$$

A careful reader will, however, notice that neither all classes can be classified as best fit nor as the worst one. Perfect normalisation is, therefore, dependant on the sizes of classes. The proposed normalisation will generally not achieve neither 0 nor 1 in the general case.

## 2.5 The Compound Measure of Accuracy

Accuracy is often used as a single-value measure of the quality of classification. This is especially true in the case of optimisation, where increasing specificity and sensitivity at the same time might not be possible. Accuracy is therefore an easy way to present the overall score. Let us, therefore, use the same approach for Accuracy, which is a bit more complicated in its form. We start with a redefinition of Eq. 1 for given class  $k$  and threshold  $t$  as presented in Eq. 30.

$$ACC_{k,t} = \frac{|TP_{k,t}| + |TN_{k,t}|}{|TP_{k,t}| + |TN_{k,t}| + |FP_{k,t}| + |FN_{k,t}|} \quad (30)$$

We aggregate over  $|K|$  classes and  $|K| - 1$  threshold values as in Eq. 31.

$$ACC = \sum_{t=1}^{|K|-1} \frac{ACC_t}{t^m} = \frac{1}{|K|} \sum_{t=1}^{|K|-1} \sum_k \frac{|TP_{k,t}| + |TN_{k,t}|}{|O|t^m} \quad (31)$$

In this case, we change the order of summations and then use both Eq. 7 and Eq. 9 in Eq. 31 getting Eq. 32.

$$ACC = \frac{1}{|K|} \sum_k \frac{1}{|O|} \sum_{t=1}^{|K|-1} \frac{1}{t^m} \left( \sum_{o_i \in O_k} \begin{cases} 1, & \text{if } p_k(o_i) \leq t \\ 0, & \text{otherwise} \end{cases} + \sum_{o_i \in O-O_k} \begin{cases} 1, & \text{if } p_k(o_i) > t \\ 0, & \text{otherwise} \end{cases} \right) \quad (32)$$

Then we include the threshold factor in the conditional summations and remove the conditions by changing the range of summations as in Eq. 33.

$$ACC = \frac{1}{|K||O|} \sum_k \left( \sum_{o_i \in O_k} \sum_{t=p_k(o_i)}^{|K|-1} \frac{1}{t^m} + \sum_{o_i \in O-O_k} \sum_{t=1}^{p_k(o_i)-1} \frac{1}{t^m} \right) \quad (33)$$

Which leads us to harmonic numbers in Eq. 34, wherein vacuous summation defines  $H_0$ .

$$ACC = \frac{1}{|K||O|} \sum_k \left( \sum_{o_i \in O_k} \left( H_{|K|-1}^m - H_{p_k(o_i)-1}^m \right) + \sum_{o_i \in O-O_k} H_{p_k(o_i)-1}^m \right) \quad (34)$$

As in the case of Sensitivity, we can remove the constant factor from the summation as in Eq. 35.

$$ACC = \frac{H_{|K|-1}^m}{|K|} + \frac{1}{|K||O|} \sum_k \left( \sum_{o_i \in O-O_k} H_{p_k(o_i)-1}^m - \sum_{o_i \in O_k} H_{p_k(o_i)-1}^m \right) \quad (35)$$

Reversing the order of summations allows us to represent the result as in Eq. 36.

$$ACC = \frac{H_{|K|-1}^m}{|K|} + \frac{1}{|K||O|} \sum_{o_i \in O} \left( \sum_{k: o_i \notin O_k} H_{p_k(o_i)-1}^m - \sum_{k: o_i \in O_k} H_{p_k(o_i)-1}^m \right) \quad (36)$$

Note that the first inner sum is actually the partial sum of generalised harmonic numbers with the second inner sum removed. It can therefore be precalculated. It is clear if we rewrite it for  $m \in \mathbb{N}$ , where we can represent the sum of generalised harmonic numbers as  $\sum_{i=1}^n H_i^m = (n+1)H_n^{m+1} - H_n^{(m+1)}$ , obtaining Eq. 37.

$$ACC = \frac{H_{|K|-1}^m}{|K|} + \frac{1}{|K||O|} \sum_{o_i \in O} \left( (|K|+1)H_{|K|}^m - H_{|K|}^{(m+1)} - 2H_{p_k: o_i \in O_k(o_i)-1}^m \right) \quad (37)$$

Which allows us to represent it as in Eq. 38.

$$ACC = \frac{H_{|K|-1}^m + (|K| + 1)H_{|K|}^m - H_{|K|}^{(m-1)}}{|K|} - \frac{2}{|K||O|} \sum_{o_i \in O} H_{p_k: o_i \in O_k}^m (o_i) - 1 \quad (38)$$

Which, despite quite involved general case formulae, can be evaluated with linear ( $\mathcal{O}(|O|)$ ) complexity. It is important to note, that for hyperparameter optimisation, the constant part of Accuracy or other measures does not have to be calculated at all, as the difference between compared classifiers will be fully enclosed in the inner summation.

The minimal value of Accuracy occurs when each data point actual class has been assigned the last position, that is  $\forall i, k : o_i \in O_k, p_k(o_i) = |K|$ . In such a case, we get Eq. 39.

$$ACC = \frac{|K|H_{|K|}^m + \frac{1}{|K|^m} - H_{|K|}^{(m-1)}}{|K|} \quad (39)$$

To put the minimal value at 0, we remove this factor and get Eq. 40.

$$A\hat{C}C = \frac{2H_{|K|-1}^m}{|K|} - \frac{2}{|K||O|} \sum_{o_i \in O} H_{p_k: o_i \in O_k}^m (o_i) - 1 \quad (40)$$

Since the maximum value is dependant on the count of classes, we can further assume that the best result is the one in which  $\forall i, k : o_i \in O_k, p_k(o_i) = 1$ , therefore the appropriately scaled, normalised aggregated Accuracy is in the form as presented in Eq. 41.

$$A\hat{C}C = 1 - \frac{1}{|O|H_{|K|-1}^m} \sum_{o_i \in O} H_{p_k: o_i \in O_k}^m (o_i) - 1 \quad (41)$$

### 3 Discussion

In this paper, we present a modification to commonly used performance measures that leverages the ordering of classes in multinomial classification. Instead of using a one-vs-all approach, we aggregate the performance measure over multiple thresholds of how we understand the correct classification, that is - on which position a proper class should be to recognise the classification as the correct one, which leads to two different effects.

The first effect is especially important in many real-life scenarios, wherein the final decision is done by humans and the machine learning approach is used to narrow the range of possibilities to consider. Such scenarios include facial recognition, neurodegenerative disease diagnosis, and many others. In such a case, the human operator receives a few best fitting classes and can examine the case further. For example, recognising the few most probable identities based on facial recognition among thousands of classes allows investigators to use additional

information (such as clothing) to pinpoint the suspect. In the case of disease diagnosis, this allows considering extra tests that will differentiate between a few potential diseases or between the best fitting “healthy” class and the slightly less probable disease that would remain ignored using binary classification. An approach that considers not only the most probable result but also further ones leads to an algorithm that positions the correct result as high as possible, even if for some reason it might not be the best fit, while otherwise, the correct result might be in the position that does not allow for recognition by a human operator.

The second effect is crucial for the process of machine learning. In the one-vs-all approach to classification, adjustments to hyperparameters that result in the change of the position of the correct label in the resulting ordering are ignored. The only shift that actually matters is either improving the position of the correct label to the first one or decreasing it from the first to any other. Therefore, the difference between two values of the hyperparameter might either be non-existent, even when the prediction of the correct class has improved from, e.g., 100th to the 2nd class, or be driven by fluctuations of a few edge cases. The true change of performance of multinomial classification is therefore hidden and only a limited amount of information is used to control the adjustment of hyperparameters, leading to the algorithm that manages to find a way to differentiate a specific data set rather than capture the nature of the problem. Leveraging information about further positions might result in faster convergence and solutions that are less dependant on the training data.

It should be noted, that the presented approach does not stand in opposition to the trend of aggregating multiple measures into one, such as in the case of F-score or even to the extension of ROC curves to the multi-class case.

**Acknowledgements.** The research described in the paper was supported by grant no. WND-RPSL.01.02.00-24-00AC/19-011 “An innovative system for the identification and re-identification of people based on a facial image recorded in a short video sequence in order to increase the security of mass events.” funded under the Regional Operational Programme of the Silesia Voivodeship in the years 2014–2020.

The work of Damian Peşzor was supported in part by Silesian University of Technology (SUT) through a grant number BKM-647/RAU6/2021 “Detection of a plane in stereovision images without explicit estimation of disparity with the use of correlation space”.

## References

1. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In: Sattar, A., Kang, B. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1015–1021. Springer, Heidelberg (2006). [https://doi.org/10.1007/11941439\\_114](https://doi.org/10.1007/11941439_114)
2. Singh, A., Singh, M.: Evaluation measure selection for performance estimation of classifiers in real time image processing applications. *Res. Cell: Int. J. Eng. Sci.* **17**(1), 168–174 (2016)

3. Pęszor, D., Paszkuta, M., Wojciechowska, M., Wojciechowski, K.: Optical flow for collision avoidance in autonomous cars. In: Nguyen, N.T., Hoang, D.H., Hong, T.-P., Pham, H., Trawiński, B. (eds.) ACIIDS 2018. LNCS (LNAI), vol. 10752, pp. 482–491. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-75420-8\\_46](https://doi.org/10.1007/978-3-319-75420-8_46)
4. Dudek, A., et al.: Analysis of facial expressions in patients with schizophrenia, in comparison with a healthy control - case study. *Psychiatr. Danub.* **29**(3), 584–589 (2017)
5. Pęszor, D., Staniszewski, M., Wojciechowska, M.: Facial reconstruction on the basis of video surveillance system for the purpose of suspect identification. In: Nguyen, N.T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) ACIIDS 2016. LNCS (LNAI), vol. 9622, pp. 467–476. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-49390-8\\_46](https://doi.org/10.1007/978-3-662-49390-8_46)
6. Huk, M., Szczepanik, M.: Multiple classifier error probability for multi-class problems. *Eksploatacja i Niezawodność-Maint. Reliab.* **51**(3), 12–16 (2011)
7. Huk, M.: Notes on the generalized backpropagation algorithm for contextual neural networks with conditional aggregation functions. *J. Intell. Fuzzy Syst.* **32**, 1365–1376 (2017)
8. Huk, M.: Training contextual neural networks with rectifier activation functions: role and adoption of sorting methods. *J. Intell. Fuzzy Syst.* **37**(6), 7493–7502 (2019)
9. Huk, M.: Stochastic optimization of contextual neural networks with RMSprop. In: Nguyen, N.T., Jearanaitanakij, K., Selamat, A., Trawiński, B., Chittayasothorn, S. (eds.) ACIIDS 2020. LNCS (LNAI), vol. 12034, pp. 343–352. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-42058-1\\_29](https://doi.org/10.1007/978-3-030-42058-1_29)
10. Yerushalmy, J.: Statistical problems in assessing methods of medical diagnosis, with special reference to X-ray techniques. *Public Health Rep.* **62**(40), 1432–1449 (1947)
11. Lachiche, N., Flach, P.: Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. In: Proceedings of the Twentieth International Conference on Machine Learning, Washington, DC, USA, pp. 416–423. AAAI Press (2003)
12. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **45**(4), 427–37 (2009)
13. Hossin, M., Sulaiman, M.N.: A review on evaluation metrics for data classification evaluations. *Int. J. Data Mining Knowl. Manag. Process* 1–11 (2015)
14. Ferri, C., Hernández-Orallo, J., Modroi, R.: An experimental comparison of performance measures for classification. *Pattern Recognit. Lett.* **30**, 27–38 (2009)
15. Japkowicz, N., Shah, M.: *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, Cambridge (2011)