AUC$_\mu$: A Performance Metric for Multi-Class Machine Learning Models

Ross S. Kleiman[1] **David Page**[1 2]

Abstract

The area under the receiver operating characteristic curve (AUC) is arguably the most common metric in machine learning for assessing the quality of a two-class classification model. As the number and complexity of machine learning applications grows, so too does the need for measures that can gracefully extend to classification models trained for more than two classes. Prior work in this area has proven computationally intractable and/or inconsistent with known properties of AUC, and thus there is still a need for an improved multi-class efficacy metric. We provide in this work a multi-class extension of AUC that we call AUC$_\mu$ that is derived from first principles of the binary class AUC. AUC$_\mu$ has similar computational complexity to AUC and maintains the properties of AUC critical to its interpretation and use.

1. Introduction

The area under the Receiver Operating Characteristic (ROC) curve, commonly referred to as the AUC, is ubiquitous in machine learning, yet it is limited to classification tasks with only two classes. There have been a variety of prior attempts to extend AUC to the multi-class setting but there is no consensus on the appropriate way to proceed. Multi-class AUC analogs must deal with new challenges in both computational complexity and decisions of which properties of the binary AUC are most important to preserve. Current approaches largely fall into two camps: those that are theoretically rooted and those that are focused on ease of use. We believe that the community has implicitly stated a preference for practicality, as the most widely used measure, M, introduced by Hand & Till (2001), is an easy to use multi-class AUC analog. However, in our work we show that M can fail to return a score of 1 (perfect performance), even when for every example a model gives the correct label the highest probability.

In our work we first consider those properties of AUC that we believe to be most critical to its use and interpretation. The properties are based on work by (Fawcett, 2006).

Property 1. If a model gives the correct label the highest probability on every example, then AUC = 1

Property 2. Random guessing on examples yields AUC = 0.5

Property 3. AUC is insensitive to class skew

We note that these three properties are all a consequence of the relationship between AUC and the Mann Whitney U-Statistic (Hanley & McNeil, 1982). The U-statistic, and hence the two-class AUC, is the probability the model will correctly rank two instances of difference classes. Therefore, rather than generalizing the ROC curve to handle $K > 2$ classes as others have done before, we instead turn our attention to generalizing the U-statistic for $K > 2$. We call our measure AUC$_\mu$ using the Greek letter mu (μ) as an acronym for "**m**ulti-class **U**-statistic." While this paper derives and presents AUC$_\mu$ in the context of multi-class models with probabilistic outputs, it is also compatible with multi-class models that output scores or ranks for query instances across the K classes.

This paper is organized as follows. In Section 2 we present a survey of prior work performed on extending AUC to the multi-class setting. In Section 3 we present background on the U-statistic form of AUC, multi-class AUC, and partition matrices (a tool we use in computing AUC$_\mu$). In Section 4 we formulate the AUC$_\mu$ statistic. In Section 5 we provide several theoretical results for AUC$_\mu$ and we also demonstrate some special cases for AUC$_\mu$. Finally, in Section 6 we provide concluding remarks on the work and some interesting future directions.

2. Prior Work on Multi-Class AUC

Prior work on extending AUC to the multi-class setting has focused on both the theoretical aspects of the problem

[1]Department of Computer Sciences, University of Wisconsin - Madison, Madison, Wisconsin [2]Department of Biostatistics and Medical Informatics, University of Wisconsin - Madison, Madison, Wisconsin. Correspondence to: Ross S. Kleiman <rkleiman@cs.wisc.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

and producing useable measures for real world problems. Interestingly, one of the earliest works in this area was a theoretical piece by Srinivasan (1999) who proved which classifiers may be optimal in an n-dimensional ROC space. Given a set of possible hard-labeling multi-class classifiers, it was shown that regardless of the choice of misclassification cost matrix, the optimal classifier lies on the convex hull of the n-dimensional ROC "surface". This is an extension of a known property of AUC that was shown by Provost & Fawcett (1997); that is, we may consider only those classifiers on the convex hull of the ROC curve regardless of the misclassification costs. While Srinivasan (1999) did not suggest how one would construct such an n-dimensional ROC space, the contributions were useful for future work.

A reasonable notion for the construction of a multi-class analog of AUC is that if in the two-class case we integrate under the ROC curve, then for the K-class case we should integrate under the ROC surface. This resulted in work on computing the volume under the ROC surface (VUS), though there is a disagreement on exactly how one should construct an ROC surface. In the two-class case, an ROC curve is plotted using the true positive rate and false positive rate values that are derived from the 2×2 confusion matrix. In general, a problem with K classes has a $K \times K$ confusion matrix from which we would construct the ROC surface. Two schools of thought arose on how to construct an ROC surface. Mossman (1999) believed that one needed only K dimensions for construction of the ROC surface, while Ferri et al. (2003) believed that $K(K-1)$-dimensions were necessary.

While a VUS-based approach is a reasonable extension to AUC, it suffers greatly from both computational complexity and interpretability. Both the construction of the ROC surface and computation of its volume are computationally intense problems. Lane (2000) notes that finding the convex hull of N points in d dimensions requires $O(N \log N + N^{\lfloor \frac{d}{2} \rfloor})$ time. This makes finding the ROC surface itself challenging for problems with even a moderate number of classes and instances. Because of this, both Mossman (1999) and Ferri et al. (2003) choose to approximate the points on the ROC surface, which ultimately leads to an inexact and underestimated volume. Further, even with an exact computation of volume, VUS no longer adheres to the same scale that AUC does, namely when AUC is 1 a classifier is perfect and when AUC is 0.5 it is equivalent to random guessing. VUS-based approaches have scales that get increasingly smaller as the number of classes grows and this makes interpreting how good a multi-class model is with VUS a challenge.

Perhaps it is for these reasons that the most widely used multi-class AUC measure, M (Hand & Till, 2001), is not VUS-based but rather an average of pairwise AUCs amongst the k classes. M is an easy to compute and class-skew insensitive performance measure for multi-class problems. However, M loses many of the properties that we believe are crucial for successful use and interpretation. Most importantly M can return values much less than 1 even when all points are correctly labeled. Consider the example in Figure 1 where three predictions, $\hat{\mathbf{p}}^{(1)}$, $\hat{\mathbf{p}}^{(2)}$, and $\hat{\mathbf{p}}^{(3)}$, yield correct classifications by the standard argmax rule (assigning an instance to the label for which its prediction has the highest probability). For each pair of classes, i and j, M considers all points whose true label is i or j, and computes the AUC amongst these instances twice, once with the ith component considered positive, and once with the jth component considered positive. These two calculations can yield different results and thus in cases such as Figure 1, M can return a score as low as 0.67 even though the points are perfectly labeled. Finally, M loses the elegance of a simple probabilistic interpretation as it is no longer equivalent with the U-statistic. That is, M is not the probability that two random instances will be ranked correctly.

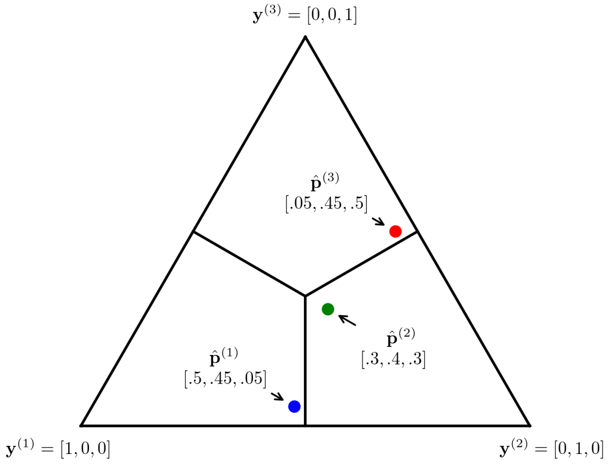

Figure 1. The M measure proposed by Hand & Till (2001) can yield a result much less than 1 even when a model assigns the correct label the highest probability on every example. In this figure, the larger outside equilateral triangle is the space of all possible model outputs and is known as the 2-simplex. Consider three model predictions plotted on the 2-simplex, $\hat{\mathbf{p}}^{(1)}$, $\hat{\mathbf{p}}^{(2)}$, and $\hat{\mathbf{p}}^{(3)}$, belonging to classes 1, 2, and 3 respectively. We divide the simplex using the argmax partitioning, that is, a prediction is assigned to the class for which it has the highest probability. The points are separable and correctly classified, however, M returns a value of 0.67 for this example, suggesting the model quality is much closer to random guessing than it is to perfect performance. This behavior is also true of the metric proposed by Provost & Domingos (2000).

While not nearly as widely used, Provost & Domingos

(2000) proposed another method of extending AUC to the multi-class domain. Their approach performs a weighted average of K one-versus-all calculations of AUC for each of the individual class probabilities. However, because this approach weighs each individual AUC calculation by its class weight, it is inherently sensitive to class skew and thus violates Property 3. Additionally, like M it does not satisfy Property 1 and can return values less than 1 even when all examples would be accurately labeled using the argmax rule. Using the example in Figure 1, the method proposed by Provost & Domingos (2000) would also return a value of 0.67.

3. Background

Here we provide background material necessary for our derivation of AUC$_\mu$. First, in Section 3.1 we discuss the relationship of AUC and the aforementioned Mann-Whitney U-statistic. The U-statistic is a metric based on the ranking of probabilistic model predictions from the two-class case. We then discuss in Section 3.2 how the probabilistic predictions of a model differs when there are more than 2 classes as multi-class predictions are specified as categorical distributions. Finally, in Section 3.3 we discuss partition matrices and decision boundaries, two tools that we use eventually use to rank categorical distributions.

3.1. AUC and the Mann-Whitney U-Statistic

True to its moniker, AUC is most commonly understood as an integration under the ROC curve. The Mann-Whitney U-statistic relationship shows a probabilistic interpretation of AUC. That is, AUC is the probability that a random instance whose label is positive will receive a higher ranking than a random instance whose label is negative. Let D^+ and D^- represent the sets of model predictions for positive and negative instances respectively (e.g. if $\hat{p}^{(i)} \in D^+$, then $\hat{p}^{(i)}$ is some probability in $[0, 1]$, and the true label $y^{(i)}$ for instance $\mathbf{x}^{(i)}$ is positive). Further, let $n_+ = |D+|$ and $n_- = |D^-|$ be the number of positive and negative instances respectively. Then we can calculate AUC as specified in Equation 1,

$$\text{AUC} = \frac{1}{n_+ n_-} \sum_{\hat{p}^{(i)} \in D^+} \sum_{\hat{p}^{(j)} \in D^-} \tilde{I}(\hat{p}^{(i)} - \hat{p}^{(j)}), \quad (1)$$

where $\tilde{I}(\cdot)$ is a modified indicator function that returns 1 if the argument is positive, 0 if the argument is negative, and 0.5 if the argument is 0.

3.2. Multi-Class Classification Models and Predictions

Whereas binary classification problems are concerned with labeling an instance as one of 2 categories, we call a task where an instance can belong to one of K categories a multi-class classification problem.

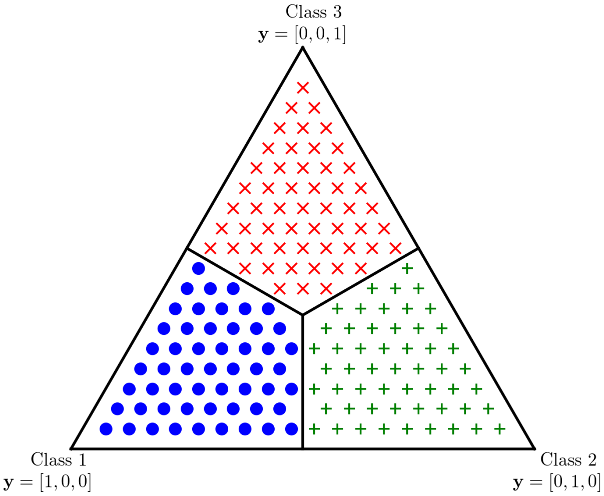

Figure 2. A partitioning of the 2-simplex, Δ_2, for a 3-class classification problem. Here we show the argmax partitioning, $A_{\max}$, and the decision boundaries it induces. The regions with blue circles, green pluses, and red crosses are assigned to classes 1, 2, and 3 respectively.

Definition 3.1. *$\mathcal{M}$ is a **multi-class model** over a domain $\mathcal{X}$ of possible examples that maps each $\mathbf{x} \in \mathcal{X}$ to a categorical distribution $\hat{\mathbf{p}} = [\hat{\mathbf{p}}_1, \ldots, \hat{\mathbf{p}}_K]$, where $\hat{\mathbf{p}}_j$ is the probability $\mathbf{x}$ belongs to category j. The domain of possible model predictions for a task with K classes is described by the $(K-1)-$simplex, Δ_{K-1}.* Figure 1 *shows Δ_2 along with 3 different $\hat{\mathbf{p}}$ model outputs.*

While in two-class tasks a scalar threshold is used to map $\hat{\mathbf{p}}$ to a label, we need more complex tools for a multi-class task. With $K > 2$ classes, we now must define a partitioning of the $(K-1)-$simplex that maps a categorical distribution, $\hat{\mathbf{p}}$, to a hard label. That is, we divide Δ_{K-1} into K regions corresponding to values of $\hat{\mathbf{p}}$ that map to each of the K labels. Consider Figure 2 where we demonstrate what a partitioning of the 2-simplex looks like for a classification task with 3 classes.

3.3. Partition Matrices and Decision Boundaries

Recall that in Equation 1 the U-statistic computation involves ranking of predictions for two instances of different classes. Thus, extending the U-statistic to $K > 2$ classes requires some way of ranking the categorical distributions outputted by a multi-class model. We propose the use of a partition matrix, which divides the probability space of model outputs into distinct labeling regions.

Definition 3.2. ***Partition matrix:*** *Let A be a $K \times K$ matrix and let $A_{i,j}$ be the cost of classifying an instance as class i when its true class is j. Then A defines a partition on the $(K-1)-$simplex and induces decision boundaries between the K classes.*

The partition matrix is analogous to a threshold value in the two-class case and it has been shown that the two-class threshold can be derived from a 2×2 partition matrix (O'Brien et al., 2008). The misclassification cost matrix, that specifies the cost of mislabeling an instance of one label as another, is in fact a partition matrix. In general, a partition matrix is any matrix that divides the probability space into labeling regions for each of the K classes. Here we present background on partition matrices and their relationship with calculating AUC. The work presented by O'Brien et al. (2008) relies heavily on partition matrices, and their study provides many useful properties, proofs, and definitions. We restate some of their results (Definitions 3.2 and 3.3) as they are useful building blocks for our work.

Further, as shown by O'Brien et al. (2008), any partition matrix A can be expressed by some other matrix A' with the properties $A'_{i,i} = 0 \, \forall i$ and $A'_{i,j} \neq 0 \, \forall i \neq j$. From here forward, when referring to a partition matrix we assume it is in this form with all diagonal entries zero.

Definition 3.3. ***Decision boundary:*** *A decision boundary between class i and class j, $i \neq j$, is the hyperplane that separates the two classes in Δ_{K-1}. The decision boundary is calculated using the partition matrix to solve for the hyperplane of solutions that have equal cost-sensitive losses if assigned to class i or class j.*

$$\sum_{k=1}^{K} A_{i,k}\hat{\mathbf{p}}_k = \sum_{k=1}^{K} A_{j,k}\hat{\mathbf{p}}_k \tag{2}$$

An equivalent formulation of Equation 2 can use dot products and may be written as $A_{i,\cdot}\hat{\mathbf{p}} = A_{j,\cdot}\hat{\mathbf{p}}$.

Definition 3.4. ***Argmax partition matrix,*** $A_{\max}$***:*** *When the costs in a partition matrix are 1 everywhere, except the diagonal where they are 0, we call this the argmax partition matrix. It is so named because the label it assigns to any prediction, $\hat{\mathbf{p}}$, is $\arg\max_k \hat{\mathbf{p}}_k$. Because we reference this heavily in this work, we give it a special identifier: $A_{\max}$.*

Figure 3 shows how the choice of partition matrix can change not just the label of a point, but in fact reverse the orientation of two points. By this we mean that two points that are both correctly labeled (correctly oriented with respect to the decision boundary) with one partition matrix, can become incorrectly labeled with another partition matrix.

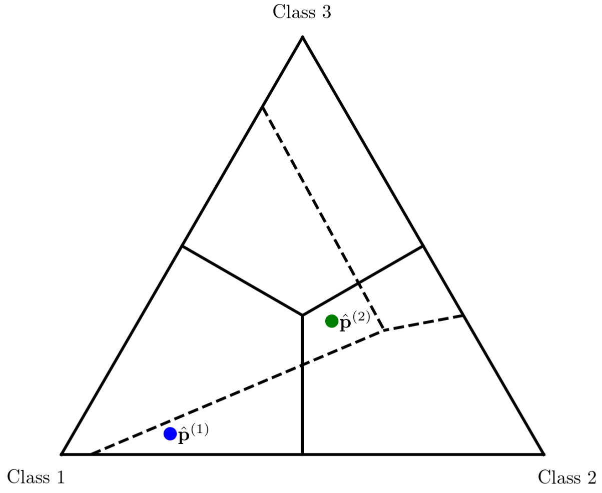

Figure 3. The choice of partition matrix can reverse the labeling of two points. Consider first the argmax partitioning of Δ_2, using $A_{\max}$, shown in solid lines. Then $\hat{\mathbf{p}}^{(1)}$ and $\hat{\mathbf{p}}^{(2)}$ are assigned to the correct classes, 1 and 2 respectively. Now consider an alternative partitioning shown in dashed lines . Now $\hat{\mathbf{p}}^{(1)}$ is assigned to class 2, while $\hat{\mathbf{p}}^{(2)}$ is assigned to class 1. The first choice of partition matrix correctly labels both points, while the second choice of partition matrix incorrectly labels both points. It is also possible to choose a partition matrix that labels one point correctly and one point incorrectly.

Finally, we note the connection between the partition matrix and the decision boundaries learned by a linear-kernel multi-class SVM (Weston & Watkins, 1999). Both the linear-kernel multi-class SVM and the partition matrix divide a space into K regions, each corresponding to a class. However, a linear-kernel multi-class SVM divides the feature space whereas the partition matrix divides a probablity space, Δ_{K-1}. Moreover, a partition matrix has at least one *equal-risk point* (O'Brien et al., 2008), where all classes are equally likely, whereas a multi-class SVM may produce solutions with no equal-risk point.

4. AUC$_\mu$ Derivation

In this section we derive the formula for AUC$_\mu$. We wish for AUC$_\mu$ to be a multi-class extension of the U-statistic presented in Equation 1. Thus, AUC$_\mu$ must compute the probability that two random instances from different classes are ranked correctly by a model. However, recall that in multi-class classification our model output $\hat{\mathbf{p}}$ is a categorical distribution which makes extending the concept of ranking unclear. What does it mean for one categorical distribution to be of a "higher rank" than another? We must provide some means to map a categorical distribution to a scalar value which can be used for ranking. In Section 4.1 we demonstrate how such rankings can be performed using a partition matrix. Then, in Section 4.2 we use this method of ranking to derive the expression for AUC$_\mu$.

4.1. Ranking Categorical Distributions

The intuition behind our approach is most easily described through an analogy to standard linear kernel support vector machines (SVMs). A linear SVM generates a decision hyperplane which divides the feature space into two regions, one where instances are labeled as positive and the other negative. The further an instance is from the decision hyperplane the more confident the SVM is in its label. In this way, model confidence for an SVM is measured by the orthogonal distance of an instance to the decision hyperplane. Similarly, when ranking two instances from different classes, we use the decision hyperplane between those two classes that is derived from the partition matrix.

Recall that Equation 2 describes the decision boundary as a set of categorical distributions for which the expected cost of labeling an instance as class i or class j is equal. Let $\mathbf{v}_{i,j} = A_{i,\cdot} - A_{j,\cdot}$, then $\mathbf{v}_{i,j} \cdot \hat{\mathbf{p}} = 0$ is the equation of the hyperplane and an equivalent formulation of Equation 2. This decision boundary divides our $(K-1)-$simplex into two regions, one where we are more confident to label an instance class i and one where we are more confident to label an instance class j. If $\mathbf{v}_{i,j} \cdot \hat{\mathbf{p}}$ is positive, then we see it is more costly to assign the label of class i than class j. The more positive $\mathbf{v}_{i,j} \cdot \hat{\mathbf{p}}$, the larger the difference in cost, and therefore the more favorable a labeling class j becomes. Therefore, $\mathbf{v}_{i,j}$ provides a way to rank various points in terms of their cost difference between assignments of class i and j. It should be noted that $\mathbf{v}_{i,j}$ is the orthogonal vector to our equal-cost hyperplane and $\mathbf{v}_{i,j} \cdot \hat{\mathbf{p}}$ is proportional to the length of the projection of $\hat{\mathbf{p}}$ onto $\mathbf{v}_{i,j}$. We are in essence calculating an unscaled orthogonal distance of our prediction, $\hat{\mathbf{p}}$, to the equal cost hyperplane. This scalar value provides the ranking that is the critical piece that is needed to extend the indicator function I in Equation 1 and thus derive multi-class AUC.

We now show how to determine if two model outputs are ranked correctly in a multi-class problem. Let $\hat{\mathbf{p}}^{(a)}$ and $\hat{\mathbf{p}}^{(b)}$ be the categorical output of our model for two instances $\mathbf{x}^{(a)}$ and $\mathbf{x}^{(b)}$. Further, without loss of generality, let the true classes of $\mathbf{x}^{(a)}$ and $\mathbf{x}^{(b)}$ be classes 1 and 2 such that $\mathbf{y}^{(a)} = [1, 0, \ldots, 0]$ and $\mathbf{y}^{(b)} = [0, 1, \ldots 0]$ are the true class vertices on the $(K-1)-$simplex for these two instances. Let A be our partition matrix. We first calculate our normal vector to our decision boundary as $\mathbf{v}_{1,2} = A_{1,\cdot} - A_{2,\cdot}$. Note that $\mathbf{v}_{1,2} \cdot \mathbf{y}^{(1)}$ and $\mathbf{v}_{1,2} \cdot \mathbf{y}^{(2)}$ are the unscaled distances of our class vertices from the hyperplane. This provides us the "correct" orientation of two points projected onto $\mathbf{v}_{1,2}$. That is, if $\mathbf{v}_{1,2} \cdot \mathbf{y}^{(a)} > \mathbf{v}_{1,2} \cdot \mathbf{y}^{(b)}$, then if $\mathbf{v}_{1,2} \cdot \hat{\mathbf{p}}_i > \mathbf{v}_{1,2} \cdot \hat{\mathbf{p}}_j$ we know that our model correctly ranked the two points. Figure 4 illustrates an example of this projection and how we can use our partition-matrix-derived decision boundary to induce rankings on multi-class predictions. We can efficiently compute if two points are ranked correctly through the introduction of an orientation function.

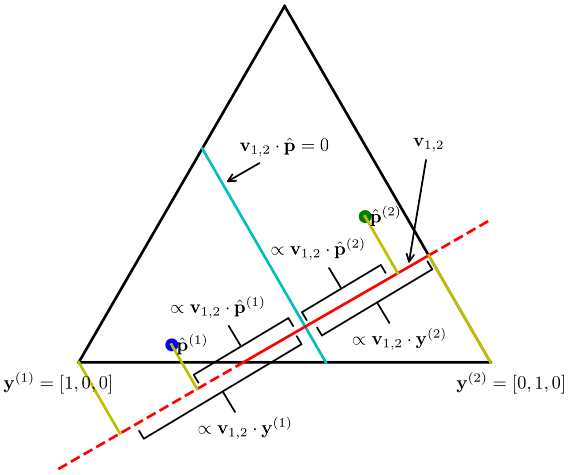
$\mathbf{v}_{1,2} \cdot \hat{\mathbf{p}} = 0$
$\mathbf{v}_{1,2}$
$\hat{\mathbf{p}}^{(2)}$
$\propto \mathbf{v}_{1,2} \cdot \hat{\mathbf{p}}^{(2)}$
$\propto \mathbf{v}_{1,2} \cdot \hat{\mathbf{p}}^{(1)}$
$\hat{\mathbf{p}}^{(1)}$
$\propto \mathbf{v}_{1,2} \cdot \mathbf{y}^{(2)}$
$\mathbf{y}^{(1)} = [1, 0, 0]$
$\mathbf{y}^{(2)} = [0, 1, 0]$
$\propto \mathbf{v}_{1,2} \cdot \mathbf{y}^{(1)}$

Figure 4. A depiction of how the partition-matrix-derived decision boundary, $\mathbf{v}_{1,2} \cdot \hat{\mathbf{p}} = 0$, (in cyan) can be used to induce a ranking of categorical distributions. The normal vector to the decision hyperplane, $\mathbf{v}_{1,2}$ (shown in red), provides a means to rank points in the simplex. The dot product of $\mathbf{v}_{1,2}$ with the true labels and model outputs form an un-normalized projection onto $\mathbf{v}_{1,2}$ and thus a means of ranking categorical distributions. The ranking of $\hat{\mathbf{p}}_1$ and $\hat{\mathbf{p}}_2$ is correct here as their orientation with respect to the decision boundary is the same as the orientation of their labels $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$.

Definition 4.1. *An* ***orientation function****, O, returns a positive value if the predictions are ranked correctly, a negative value if they are ranked incorrectly, and 0 if their rank is tied. Let $\mathbf{x}^{(a)}$ and $\mathbf{x}^{(b)}$ belonging to classes i and j respectively. Further, let their model predictions be $\hat{\mathbf{p}}^{(a)}$ and $\hat{\mathbf{p}}^{(b)}$ respectively, and their true labels be $\mathbf{y}^{(a)}$ and $\mathbf{y}^{(b)}$ respectively.*

$$O(\mathbf{y}^{(a)}, \mathbf{y}^{(b)}, \hat{\mathbf{p}}^{(a)}, \hat{\mathbf{p}}^{(b)}, \mathbf{v}_{i,j}) = (\mathbf{v}_{i,j} \cdot (\mathbf{y}^{(a)} - \mathbf{y}^{(b)}))(\mathbf{v}_{i,j} \cdot (\hat{\mathbf{p}}^{(a)} - \hat{\mathbf{p}}^{(b)})) \quad (3)$$

4.2. AUC$_\mu$

Here we detail our derivation of AUC$_\mu$ as an extension of the U-statistic such that we satisfy Properties 1-3 listed in Section 1. We restate the two-class U-statistic formulation of AUC, Equation 1, for reference.

$$\text{AUC} = \frac{1}{n_+ n_-} \sum_{\hat{p}^{(i)} \in D^+} \sum_{\hat{p}^{(j)} \in D^-} \tilde{I}(\hat{p}^{(i)} - \hat{p}^{(j)}),$$

We begin by modifying the indicator function, $\tilde{I}$, so that it is compatible with multi-class model outputs. Recall that $\tilde{I}$ returns 1 if two instances are ordered correctly, 0 if they are ordered incorrectly, and 0.5 if there is a tie in their rank. We utilize the orientation function described in Equation 3 as the new argument for $\tilde{I}$. Now, for two instances indexed by i and j from different classes, $\tilde{I} \circ O(\mathbf{y}^{(a)}, \mathbf{y}^{(b)}, \hat{\mathbf{p}}^{(a)}, \hat{\mathbf{p}}^{(b)}, \mathbf{v}_{i,j})$, will indicate if the two instances are ordered correctly, incorrectly, or tied. However, we note that O requires the two class decision hyperplane normal vector, $\mathbf{v}_{i,j}$, as an argument. What is the right choice of $\mathbf{v}_{i,j}$? To answer this question we refer to Property 1, that if all instances are labeled according the their highest probability, then AUC$_\mu$ should return a score of 1. We note that this division of the $(K-1)$-simplex is exactly the argmax partitioning, and thus we argue that $A_{\max}$ is the appropriate partition matrix to use in AUC$_\mu$. We later show in the proof in Section 5.2, if there is no *a priori* preference of a particular partition matrix, $A_{\max}$ is the appropriate choice. We further show in Section 5.4.1 how to compute an alternative formulation of AUC$_\mu$ when there is a preference for a particular partition matrix. From here forward, unless otherwise specified we assume that the decision boundaries used when calculating AUC$_\mu$ are derived from $A_{\max}$.

For a problem with K classes, let us first consider, without loss of generality, two classes $i < j \leq K$. Similar to Hand & Till (2001), we aim to construct a separability measure between i and j; we call this measure $S(i,j)$. Let D^i be the set of indices for instances whose true label is class i; we define D^j similarly. Further, let n_i, n_j be the number of instances in each set respectively. Then we define,

$$S(i,j) = \frac{1}{n_i n_j} \sum_{a \in D^i, b \in D^j} \tilde{I} \circ O(\mathbf{y}^{(a)}, \mathbf{y}^{(b)}, \hat{\mathbf{p}}^{(a)}, \hat{\mathbf{p}}^{(b)}, \mathbf{v}_{i,j}).$$

If $K = 2$, then $S(i,j)$ reduces to the U-statistic, and thus AUC. We discuss and prove this equivalence in Section 5.1.

Next we turn our attention to Property 3, that AUC$_\mu$ should be insensitive to class skew. While in the two-class case if two instances from different classes are randomly selected we always get equal representation from both classes (one from each class). However, if instances are randomly selected from different classes when $K > 2$, we are more likely to sample classes with more instances. For this reason, we construct AUC$_\mu$ such that each choice of i and j is weighted equally. This approach is inspired by how Hand & Till (2001) construct their measure M such that it is also class skew insensitive. The final formulation for AUC$_\mu$ is as follows.

$$\text{AUC}_\mu = \frac{2}{K(K-1)} \sum_{i<j} S(i,j) \tag{4}$$

We note that through this formulation, AUC$_\mu$ can also be viewed as an average of pairwise AUCs between the classes.

4.3. Comparison of Algorithms

In Table 1 we present a comparison of AUC$_\mu$ to the four other multi-class classification metrics presented in this work (Mossman, 1999; Ferri et al., 2003; Hand & Till, 2001; Provost & Domingos, 2000). Of these five metrics, AUC$_\mu$ is the only one to preserve the three critical properties of AUC: 1) a perfect classification results in a score of 1, 2) random guessing results in a score of 0.5, and 3) skew insensitivity. Moreover, AUC$_\mu$ has time complexity that is equal or faster than all other algorithms. In Section 5 we present a variety of theoretical analyses and proofs for these claims.

5. Analysis and Extensions of AUC$_\mu$

In this section we provide several properties of AUC$_\mu$, as well as several extensions for special cases. In calculating AUC$_\mu$ we use the argmax partition matrix, $A_{\max}$, yet a partition matrix is notably absent in the two-class case for calculation of AUC. Thus, we provide a proof that the calculation of AUC with only two classes is a special case that does not require a partition matrix. Further, we present a corollary of this theorem showing that AUC$_\mu$ simplifies to the standard two-class AUC when there are only two classes. We then present a proof that when there are $K > 2$ classes a partition matrix is required. In Appendix A.1, we provide proofs that AUC$_\mu$ satisfies Properties 1, 2, and 3. Finally, we present two special cases of AUC$_\mu$ for domains in which there is a strong concern about misclassification costs and/or skew.

5.1. Partition Matrices and AUC

The reader has likely noted that the requirement of a partition matrix seems unnatural, since the standard AUC measure does not require any information about a threshold or partition matrix (the former derivable from the latter). Recall though, as demonstrated in Figure 3, that the choice of partition matrix influences the ranking of points and thus AUC$_\mu$ is sensitive to the choice of partition matrix. In Theorem 5.1 we claim that for two-class classification problems we do not require a partition matrix as the the relative ranking of two points is indifferent to choice of partition matrix. This theorem is proved in Appendix A.2.

Theorem 5.1. *Let $\mathcal{M}$ be a model trained to perform a*

Table 1. Comparison of the various multi-class metrics discussed in this work: VUS-3 (Mossman), VUS (Ferri), H&T (Hand and Till), P&D (Provost and Domingos), and AUC$_\mu$. A $\surd$ indicates a proven property, a $\times$ indicates a proven absence of a property, and a "?" indicates an unknown. Not only is AUC$_\mu$ the only metric to preserve Properties 1, 2, and 3, but also its time complexity is as fast or faster than all other methods.

	VUS-3	VUS	H&T	P&D	AUC$_\mu$
PERFECT = 1	$\surd$	$\times$	$\times$	$\times$	$\surd$
RANDOM = 0.5	$\times$	$\times$	$\surd$	$\surd$	$\surd$
SKEW INSENSITIVE	?	?	$\surd$	$\times$	$\surd$
TIME COMPLEXITY	EXPONENTIAL	EXPONENTIAL	POLYNOMIAL	POLYNOMIAL	POLYNOMIAL

binary classification task. Let A be a 2×2 partition matrix with diagonal zeros and all other entries positive. Then A has no effect on the ranking of predictions from $\mathcal{M}$

A desirable corollary of Theorem 5.1 is that AUC$_\mu$ simplifies to the two-class AUC presented in Equation 1. This corollary is proved in Appendix A.2.

Corollary 5.1.1. *When $K = 2$, AUC$_\mu$ simplifies to the Mann-Whitney U-statistic formulation of AUC.*

When there are more than 2 classes, the choice of partition matrix can impact the ranking of instances and thus it is necessary to specify the partition matrix in calculating AUC$_\mu$. In Figure 3 we showed that the choice of a partition matrix can affect how two instances are ranked in the 3-class case. We claim in Theorem 5.2 that for any $K > 2$ we must provide a partition matrix to rank predictions and we prove this theorem and Appendix A.2.

Theorem 5.2. *Let $\mathcal{M}$ be a model trained for a multi-class classification task with $K > 2$ classes. Then the ranking of predictions from $\mathcal{M}$ is not independent of the choice of $K \times K$ partition matrix, hence calculating AUC$_\mu$ requires a partition matrix.*

5.2. The Argmax Partition Matrix

In Section 4.2 we argue that $A_{\max}$ is a good choice as it satisfies Property 1. Here we claim that if there is no *a priori* preference for choice of partition matrix, then $A_{\max}$ is the appropriate choice as it is the expectation over all possible partition matrices. Theorem 5.3, states that the expectation over all partition matrices, uniformly distributed over $[0, 1]^{K\times K}$, is the argmax partition matrix, $A_{\max}$. We prove our theorem in the space $[0, 1]^{K\times K}$ as any partition matrix with finite values can be expressed by an equivalent partition matrix in $[0, 1]^{K\times K}$ (O'Brien et al., 2008). We prove this theorem in Appendix A.2.

Theorem 5.3. *The expectation over all partition matrices, uniformly distributed over $[0, 1]^{K\times K}$, for a task with K classes is equivalent to the argmax partition matrix, $A_{\max}$, where $(A_{\max})_{i,i} = 0\ \forall i$ and $(A_{\max})_{i,j} = 1\ \forall i \neq j$.*

Unsurprisingly, choosing uniform misclassification costs results in an argmax partitioning of Δ_{K-1}. That is, when we have no knowledge of misclassification costs, we label an instance with the category which contains the highest probability in $\hat{\mathbf{p}}$.

5.3. Time Complexity of AUC$_\mu$

The time complexity of AUC$_\mu$ is $O(Kn \log n)$ when using the argmax partition matrix, $A_{\max}$, where K is the number of classes, and n is the number of instances. This is equivalent to the time complexity of M proposed by Hand & Till (2001). While Fawcett (2006) claims that M has a complexity of $O(K^2 n \log n)$, we show that the bound is in fact tighter and that the complexity is $O(Kn \log n)$. The derivation of both of these results is in Appendix A.3.

5.4. Extensions of AUC$_\mu$

Motivated by the initial important properties we listed for AUC, we use the argmax partition matrix, $A_{\max}$, in the calculation of AUC$_\mu$. While we believe for most cases our initial presentation of AUC$_\mu$ is a suitable measure, there are exceptions. It is not uncommon for domains to have highly skewed class distributions or unequal misclassification costs between classes. AUC$_\mu$ can be easily modified to accommodate both of these scenarios and thus we present two such extensions. In Section 5.4.1 we show that for tasks with unequal misclassification costs one can incorporate an alternative partition matrix when calculating AUC$_\mu$. In Section 5.4.2 we show an alternative formulation of AUC$_\mu$ that can account for class skew in problems where this may be desirable in the performance measure.

5.4.1. Use of an Alternative Partition Matrix

Recall that in the calculation of AUC$_\mu$ we rely on the orientation function presented in Equation 3. This function ranks two instances based on the two-class decision boundary derived from the partition matrix. In the standard calculation of AUC$_\mu$, we use $A_{\max}$ to perform ranking. Thus, we note that it is straightforward to use an alternative partition matrix in this calculation as well. O'Brien et al. (2008) note that if the partition matrix is the misclassification cost matrix for a particular domain, then instances will be labeled in such a manner as to minimize the expected cost for a given instance. Therefore, if the misclassification cost matrix is well established for a particular domain, it may be appropriate then to use that as the partition matrix in place of $A_{\max}$. We show in Appendix A.3 that using an alternative partition matrix has time complexity to $O(Kn(K + \log n))$.

5.4.2. Incorporating Class Skew into AUC$_\mu$

Like Hand & Till (2001), we believe that a multi-class extension of AUC should be insensitive to class skew and thus AUC$_\mu$ is designed to remove the effects of any skew. However, there are tasks with heavy class skew where this property may become problematic. Therefore, we provide here an alternative formulation of AUC$_\mu$ that incorporates a weight for each pair of classes. For a task with K classes, let $i < j \leq K$ be the class labels for two different classes. Let $n = n_1 + \dots n_K$ be the number of total instances and number of instances in each class and let $\tilde{n} = \sum_{i<j} n_i n_j$. Finally, let $w_{i,j} = \dfrac{n_i n_j}{\tilde{n}}$ be the weight assigned for classes i and j. Here, $w_{i,j}$ is there probability that a pair of instances randomly selected from different classes belongs to class i and class j. Then, we formulate the class skew sensitive formulation, AUC$_\mu^S$, as follows.

$$\text{AUC}_\mu^S = \sum_{i<j} w_{i,j} S(i,j) \tag{5}$$

This alternative formulation of incorporates the natural class skew in the dataset as the weighting factor for each separability function, $S(i,j)$. We note that while choosing $w_{i,j} = \dfrac{n_i n_j}{\tilde{n}}$ is a natural option, any weighting scheme may be used so long as $\sum_{i<j} w_{i,j} = 1$ so that AUC$_\mu^S$ is still bounded between 0 and 1.

6. Conclusion

In this paper we introduce AUC$_\mu$, a multi-class classification performance measure that aims to maintain the many desirable properties of AUC. Prior work focused on multi-class analysis of volume under the ROC surface has proven to be computationally intensive and requires stochastic sampling methods for computation (Ferri et al., 2003; Mossman, 1999; Srinivasan, 1999; Lane, 2000). These measures are not well suited to large datasets or tasks such as hyperparameter tuning that require fast calculation of the model quality. The most popular approach as of the time of this writing, that does not utilize an ROC surface, is the measure M introduced by Hand & Till (2001). However, as shown in Figure 1, M can return values less than 1 even when all predictions are separable and would be labeled correctly following the common argmax labeling rule. Thus, we employ an alternative approach to multi-class AUC that is motivated by the relationship of AUC and the Mann Whitney U-statistic and through this relationship we derive AUC$_\mu$, a measure that is easy to compute and interpret.

We provide several theoretical observations of AUC$_\mu$ and some extensions for domains with particular concern regarding misclassification costs and class skew. We prove in Theorems 5.1 and 5.2 that while a partition matrix is not needed for two-class AUC, it is needed for ranking model outputs for more than two classes. As the argmax labeling rule is common in multi-class problems, we suggest that the use of the argmax partition matrix, $A_{\max}$, is the appropriate choice for most tasks and thus we use this in our computation of AUC$_\mu$. We prove in Theorem 5.3 that $A_{\max}$ is the expectation over all partition matrices when uniformly sampled. However, we also note in Section 5.4.1 that an alternative partition matrix can and should be used in domains with known unequal misclassification costs. We additionally present in Section 5.4.2 an alternative of AUC$_\mu$ that can intentionally incorporate class skew where this may provide a more sensible evaluation of the model performance.

There are several exciting avenues for analysis of AUC$_\mu$. While empirical confidence intervals and p-values can be calculated through a bootstrap approach, it would be interesting to see if there exist closed-form solutions for AUC$_\mu$ as they do for the binary AUC. Additionally, we note that the calculation of AUC$_\mu$ involves two dot products and that if either of these dot-products are 0 then the ranking of two instances is tied. This could become troublesome for tasks with very high numbers of classes as the probability of orthogonality between two random vectors increases with dimension. Whether AUC$_\mu$ is susceptible to this or not is a matter for future exploration and could suggest that the U-statistic is not a reliable measure of model performance for tasks with large numbers of classes.

By naturally extending the Mann-Whitney U-statistic, we both introduce a new method for computing multi-class AUC and provide several theoretical observations on how AUC$_\mu$ behaves in multi-class tasks. We believe that a renewed interest in performance metrics for multi-class machine learning models is warranted as many interesting problems in machine learning are not binary class problems. Ultimately, we claim that AUC$_\mu$ is a fast, reliable and easy to interpret method for assessing the performance of a multi-class classification model.

Acknowledgments

This work was supported by the NLM training grant 5T15LM007359 and the Clinical and Translational Science Award program, through the NIH National Center for Advancing Translational Sciences, grant UL1TR002373.

References

Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, jun 2006.

Ferri, C., Hernández-Orallo, J., and Salido, M. A. Volume under the ROC Surface for Multi-class Problems. In *Proceedings of the 14th European Conference on Machine Learning*, pp. 108–120. Springer, Berlin, Heidelberg, 2003.

Hand, D. J. and Till, R. J. A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning*, 45(2):171–186, 2001.

Hanley, J. A. and McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, apr 1982.

Lane, T. Position paper: Extensions of ROC analysis to multi-class domains. *Proceedings of ICML-2000 workshop on cost-sensitive learning, Stanford*, 2000.

Mossman, D. Three-way ROCs. *Medical Decision Making*, 19(1):78–89, jan 1999.

O'Brien, D. B., Gupta, M. R., and Gray, R. M. Cost-sensitive multi-class classification from probability estimates. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 712–719, New York, New York, USA, 2008. ACM Press.

Provost, F. and Domingos, P. Well-Trained PETs: Improving probability estimation trees. *CDER Working Paper #00-04-IS*, 2000.

Provost, F. and Fawcett, T. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. *KDD-97 Proceedings*, 1997.

Srinivasan, A. Note On The Location Of Optimal Classifiers In N-Dimensional ROC Space. Technical report, Oxford University Computing Laboratory, 1999.

Weston, J. and Watkins, C. Support Vector Machines for Multi-Class Pattern Recognition. *Proceedings of the 7th European Symposium on Artificial Neural Networks (ESANN-99)*, 1999.