Maximization of AUC and Buffered AUC in Binary Classification

Matthew Norton,Stan Uryasev

March 2016

RESEARCH REPORT 2015-2
Risk Management and Financial Engineering Lab
Department of Industrial and Systems Engineering
303 Weil Hall, University of Florida, Gainesville, FL 32611.
E-mail: *mdnorton@ufl.edu, uryasev@ufl.edu*

First Draft: October 2014, This Draft: August 2016

Abstract

In binary classification, performance metrics that are defined as the probability that some error exceeds a threshold are numerically difficult to optimize directly and also hide potentially important information about the magnitude of errors larger than the threshold. Defining similar metrics, instead, using Buffered Probability of Exceedance (bPOE) generates counterpart metrics that provide information about the magnitude of errors exceeding the threshold and, under certain conditions on the error function, can be optimized directly via convex or linear programming. We apply this approach to the case of AUC, the Area Under the ROC curve, and define Buffered AUC (bAUC). We show that bAUC can provide insights into classifier performance not revealed by AUC, while being closely related as a lower bound and representable as the area under a modified ROC curve. Additionally, while AUC is numerically difficult to optimize directly, we show that bAUC optimization often reduces to convex or linear programming. Extending these results, we show that AUC and bAUC are special cases of Generalized bAUC and that popular Support Vector Machine (SVM) formulations for approximately maximizing AUC are equivalent to direct maximization of Generalized bAUC. As a central component to these results, we provide a novel formula for calculating bPOE, the inverse of Conditional Value-at-Risk (CVaR). Using this formula, we show that particular bPOE minimization problems reduce to convex and linear programming.

1 Introduction

In binary classification, some performance metrics can be defined as the probability that some error function exceeds a particular threshold, i.e. by using Probability of Exceedance (POE) and an error function. For example, if one uses misclassification error, Accuracy is one minus the probability that misclassification error exceeds the threshold of zero. The Area Under the Receiver Operating Characteristic Curve (AUC) is a popular performance metric in classification that can also be viewed in this way, as the probability that 'ranking' error exceeds a threshold of zero. With a long history in signal detection theory (Egan (1975), Swets et al. (2000)), diagnostic systems analysis (Swets (1988)), and medical decision making (Zou (2002)), AUC has found much success as a measure of a model's ability to differentiate different classes of events. In machine learning, AUC has gained popularity due to its advantages over Accuracy, particularly when one has no

knowledge of misclassification costs or must deal with imbalanced classes (Bradley (1997), Provost et al. (1998, 1997), Ling et al. (2003), Cortes and Mohri (2004)). In both of these cases, AUC has benefits over Accuracy, with Accuracy implying equal misclassification costs and heightened emphasis on correctly classifying the majority class.

Defining metrics by using POE, although intuitive, produces metrics with undesirable properties. First, these metrics only consider the number of errors larger than the threshold and do not consider the magnitude of these errors which exceed the threshold. This information, which can sometimes be viewed as the classifier's 'confidence,' may be important when gauging classifier performance. Additionally, there is evidence that consideration of this information can lead to improved generalization, guaranteeing a classification margin when optimally considered (Vapnik and Vapnik (1998)). Second, these metrics are difficult to optimize directly. When dealing with empirical observations of data, direct optimization of these metrics yields a non-convex and discontinuous optimization problem. For example, with Accuracy it is common to utilize some convex surrogate to the $0-1$ loss to attempt to optimize Accuracy (e.g., the hinge or exponential loss). With AUC defined with POE, these issues are directly applicable.

Instead of defining metrics with POE, we take the approach of defining metrics with Buffered Probability of Exceedance (bPOE). Focusing our in-depth analysis on the case of AUC, we show that this approach produces a metric that accounts for the magnitude of errors, with direct optimization of the metric reducing to convex, sometimes linear, programming. Recently introduced as a generalization of Buffered Probability of Failure, a concept introduced by Rockafellar (2009) and explored further in Mafusalov and Uryasev (2015) and Davis and Uryasev (2015), bPOE equals one minus the inverse of the superquantile. The superquantile is also commonly known as the Conditional Value-at-Risk (CVaR) from the financial engineering literature. To facilitate our results, we first introduce a novel formula for simultaneously calculating bPOE and the quantile. We then show that this formula reduces many bPOE optimization problems to convex and linear programming.

Furthermore, we apply bPOE to the case of AUC to create a new, AUC-like counterpart metric called Buffered AUC (bAUC). This new metric is indeed a counterpart to AUC. It is a lower bound of AUC. Like AUC, it measures a classifiers ability to discriminate instances belonging to positive and negative classes. It can also be represented as the area under a modified ROC curve, which we call the bROC curve. This new metric is also an informative counterpart that does not simply move linearly with AUC. It can reveal information about classifier performance that may be hidden by AUC, particularly when the magnitude of ranking errors is an important discriminatory characteristic. In addition, the bROC curve can serve as an informative supplement to the information provided by the bAUC summary metric. We show on real data sets that the two measures may disagree on which classifier is superior. We also show that in the case of classifiers yielding similar AUC and ROC curves, important discriminatory information can be revealed by bAUC and the bROC curves.

With AUC defined with POE, it is extremely difficult to optimize directly, yielding a non-convex and discontinuous objective function when faced with discrete observations. We show that bAUC has substantial benefits in this regard, with direct optimization reducing to convex and linear programming. We then introduce Generalized bAUC, a natural extension of bAUC, and show that this produces a family of metrics, in which AUC and bAUC belong, all having interpretations as areas under modified ROC curves. We then provide a formulation for optimizing Generalized bAUC and show that the popular AUC maximizing RankSVM of Herbrich et al. (1999), Brefeld and Scheffer (2005) is a special case of maximizing Generalized bAUC. Thus, we show that bAUC has already found its way into the AUC maximization literature, albeit not explicitly, as an easily optimizable metric alternative to AUC that leads to a classification margin. Additionally, this

allows us to reinterpret the RankSVM, showing that the tradeoff parameter is related to bPOE threshold and that the optimal objective value is, in fact, equal to one minus Generalized bAUC.

As evidence for the viability of a more general scheme, in which one can apply bPOE to create a counterpart for POE defined metrics, we briefly address the case of Accuracy. We show that the result of Norton et al. (2015) can be interpreted in the following way: The classical soft-margin SVM formulation of Cortes and Vapnik (1995) is a special case of direct maximization of a bPOE counterpart to Accuracy called Buffered Accuracy. This serves to show that the idea of applying bPOE to define informative metric counterparts that are easy to optimize has already been applied to Accuracy, once again not explicitly, yielding the soft-margin SVM formulation.

The remainder of this paper is organized in the following manner. Section 2 reviews the AUC performance metric and issues associated with AUC, including difficulties with direct maximization. Section 3 reviews superquantiles and bPOE. We then introduce a calculation formula for bPOE and show that under particular circumstances, minimization of bPOE can be reduced to convex, sometimes linear, programming. Section 4 uses the bPOE concept to introduce bAUC. We discuss its value as a natural counterpart to AUC as a classifier performance metric and show that bAUC is easy to optimize. We then show that it can be presented as the area under a modified ROC curve and demonstrate experimentally its value as an AUC counterpart with an accompanying case study demonstrating available software implementations for efficient calculation and optimization. Section 5 generalizes the bAUC definition, presents it as a family of modified ROC curves with corresponding areas under these curves, and presents a formulation for maximizing this quantity. We then discuss its relation to existing SVM-based AUC maximization formulations. Section 6 discusses application of bPOE to define Buffered Accuracy and discusses its relation to SVM's.

2 The AUC Performance Metric

In this paper, we consider the binary classification task where we have random vectors X^+, X^- in $\mathbb{R}^n$ that belong, respectively, to classes $(Y = +1)$ and $(Y = -1)$. We are given N samples $X_1, ..., X_N$ of the random vector $X = X^+ \cup X^-$, of which m^+ have positive label, m^- have negative label, and we must choose a scoring function $h : \mathbb{R}^n \to \mathbb{R}$ and decision threshold $t \in \mathbb{R}$ to create a classifier with decision rule

$$Y_i = \begin{cases} +1 & \text{if } h(X_i) > t \\ -1 & \text{if } h(X_i) \leq t \ . \end{cases}$$

2.1 Defining AUC: Two Perspectives

AUC is a popular performance metric that measures the ability of a scoring function, h, to differentiate between two randomly selected instances from opposite classes. As opposed to a metric such as Accuracy, which considers the threshold t, AUC does not and is a measure of separation between score distributions $h(X^+)$ and $h(X^-)$. In other words, while accuracy is a direct measure of a classifiers ability to properly classify a single randomly chosen sample, AUC is concerned with a classifiers ability to properly rank two randomly selected samples that are presumed to be in different classes. This is a beneficial measure when classes are imbalanced or misclassification costs are unknown.(Bradley (1997), Provost et al. (1998, 1997), Ling et al. (2003), Cortes and Mohri (2004))

The AUC metric is defined as the Area Under the Receiver Operating Characteristic Curve (the ROC curve). Figure 1 shows an example ROC curve[1], which plots the True Positive Rate,

[1]In this paper, our examples are Empirical ROC curves, where we have a fixed h and samples of the random

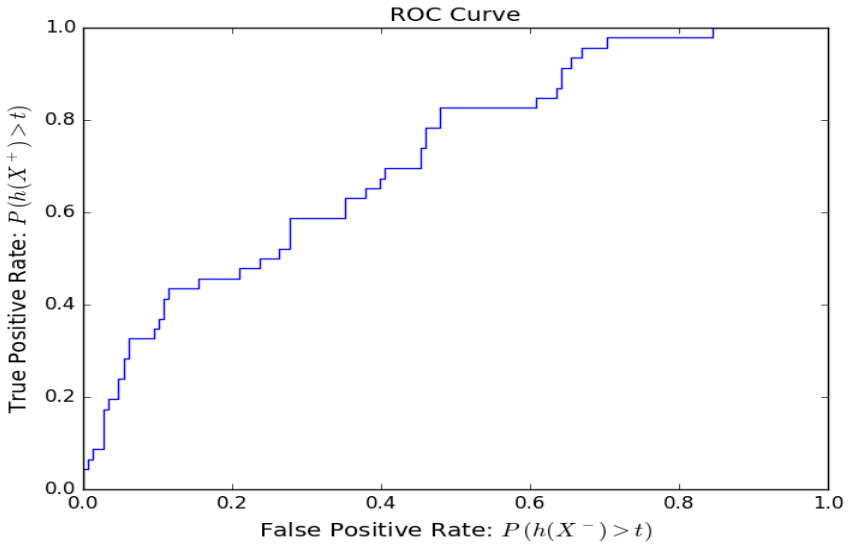

Figure 1: An example of an Empirical ROC curve for fixed h and data set consisting of samples of X. We plot the Empirical True Positive Rate, $P(h(X^+) > t)$, on the vertical axis and the Empirical False Positive Rate, $P(h(X^-) > t)$, on the horizontal axis for all values of decision threshold $t \in \mathbb{R}$.

$P(h(X^+) > t)$, on the vertical axis and the False Positive Rate, $P(h(X^-) > t)$, on the horizontal axis for different values of t. The AUC is the area under the curve formed by plotting pairs $(P(h(X^-) > t), P(h(X^+) > t))$ for all thresholds $t \in \mathbb{R}$. Specifically, we can write this in integral form. If we let $P(h(X) > t) = 1 - F_{h(X)}(t)$ be one minus the cumulative density function of $h(X)$, AUC for a scoring function h can be written as (1).

$$AUC(h) = \int_t P(h(X^+) > t)\, dP(h(X^-) > t) \tag{1}$$

In this paper, we focus more-so on an equivalent probabilistic definition of AUC provided by Hanley and McNeil (1982). Hanley and McNiel showed that the area under the ROC curve is equal to the probability that a randomly selected positive sample will be scored higher than a randomly selected negative sample. Specifically, they show that

$$AUC(h) = P\left(h(X^+) > h(X^-)\right). \tag{2}$$

With this paper focusing on POE and bPOE, we write AUC as one minus the probability of 'ranking error' $\xi(h) = -\left(h(X^+) - h(X^-)\right)$ exceeding zero. Specifically,

$$AUC(h) = 1 - P(\xi(h) \geq 0).$$

Additionally, since the true distribution of X^+ and X^- are rarely known, we often work with samples $X_1^+, ..., X_{m^+}^+, X_1^-, ..., X_{m^-}^-$. In this case, denote our ranking errors as $\xi_{ij}(h) = -\left(h(X_i^+) - h(X_j^-)\right)$ and let I_λ denote an indicator function where,

$$I_\lambda = \begin{cases} 1, & \text{if } \lambda \text{ is True,} \\ 0, & \text{if } \lambda \text{ is False.} \end{cases}$$

variables X^+, X^-.

We then have that AUC is approximated as,

$$AUC(h) = \frac{1}{m^+ m^-} \sum_{i=1}^{m^+} \sum_{j=1}^{m^-} I_{h(X_i^+) > h(X_j^-)}$$
$$= 1 - \frac{1}{m^+ m^-} \sum_{i=1}^{m^+} \sum_{j=1}^{m^-} I_{\xi_{ij}(h) \geq 0}$$

Furthermore, the ROC curve is estimated by plotting the True Positive Rate and False Positive Rate for thresholds $t \in S = \{h(X_1^+), ..., h(X_{m^+}^+), h(X_1^-), ..., h(X_{m^-}^-)\}$ on the ROC plot and connecting these points by some means to make an ROC curve (e.g. throughout the paper, we simply use linear interpolation to connect these points in our ROC plots).

For a more thorough introduction to AUC and the use of the ROC curve, we refer readers to Fawcett (2006). Additionally, for a broader view of AUC and its relation to other performance metrics, we refer readers to Hernández-Orallo et al. (2012).

2.2 Properties of AUC

As a performance metric, AUC provides insight into the ranking quality of a classifier by considering pairwise differences of scores given to samples from opposing classes. With each sample data point receiving a *score*, $h(X_i)$, the ordering of these scores (i.e. the 'ranking' induced by the scoring function) can be an important indicator of classifier performance (see e.g. Caruana et al. (1996), Schapire and Singer (1998), Cortes and Mohri (2004), Herbrich et al. (1999)). Specifically, AUC considers the distribution of ranking errors $\xi_{ij}(h)$, where a pair of samples X_i^+, X_j^- are properly ranked by h if $\xi_{ij}(h) < 0$, and equals the proportion of ranking errors $\xi_{ij} < 0$. AUC, though, does not consider the *magnitude* of ranking errors, i.e. the *confidence* with which the classifier correctly or incorrectly ranks pairs of samples. Therefore, if the magnitude of ranking errors is an important performance indicator, AUC may not be a desirable performance measure. This characteristic parallels that of Value-at-Risk (VaR) in Financial Engineering. It hides potentially important information about tail behavior by failing to consider the magnitude of tail losses.

Maximizing AUC is also a challenging task, as it is akin to probability minimization for discrete distributions, an optimization task which yields a discontinuous and non-convex objective function. Many AUC optimization approaches exist (see e.g. Brefeld and Scheffer (2005), Miura et al. (2010), Krm et al. (2012), Cortes and Mohri (2004), Herschtal and Raskutti (2004), Mozer (2003)). These approaches, though, utilize approximations of the AUC objective and do not optimize AUC directly. For example, Miura et al. (2010) optimizes an AUC approximation by replacing the indicator loss with a continuous sigmoid function. This yields a continuous optimization problem, though still non-convex.

3 bPOE and bPOE Optimization

With AUC defined using POE, we explore the use of a counterpart to POE called bPOE. Specifically, a generalization of Buffered Probability of Failure (Rockafellar and Royset (2010)), bPOE is the inverse of the superquantile (CVaR) defined in Rockafellar and Uryasev (2000). In this section, after reviewing these concepts, we present a novel formula for bPOE that simultaneously calculates POE. We show that this formula allows certain bPOE minimization problems to be reduced to convex, sometimes linear, programming. This result is particularly important when we apply

bPOE to create bAUC in Section 4.

3.1 bPOE and Tail Probabilities

When working with optimization of tail probabilities, one frequently works with constraints or objectives involving *probability of exceedance* (POE), $p_z(X) = P(X > z)$, or its associated quantile $q_\alpha(X) = \min\{z|P(X \leq z) \geq \alpha\}$, where $\alpha \in [0, 1]$ is a probability level. The quantile is a popular measure of tail probabilities in financial engineering, called within this field Value-at-Risk by its interpretation as a measure of tail risk. The quantile, though, when included in optimization problems via constraints or objectives, is quite difficult to treat with continuous (linear or non-linear) optimization techniques.

A significant advancement was made in Rockafellar and Uryasev (2000) in the development of an approach to overcome the difficulties raised by the use of the quantile function in optimization. They explored a replacement for the quantile, called CVaR within the financial literature, and called the superquantile in a general context. The superquantile is a measure of uncertainty similar to the quantile, but with superior mathematical properties. Formally, the superquantile (CVaR) for a continuously distributed X is defined as

$$\bar{q}_\alpha(X) = E\left[X|X > q_\alpha(X)\right].$$

For general distributions, the superquantile can be defined by the following formula,

$$\bar{q}_\alpha(X) = \min_\gamma \gamma + \frac{E[X-\gamma]^+}{1-\alpha}, \tag{3}$$

where $[\cdot]^+ = \max\{\cdot, 0\}$.

Similar to $q_\alpha(X)$, the superquantile can be used to assess the tail of the distribution. The superquantile, though, is far easier to handle in optimization contexts. It also has the important property that it considers the magnitude of events within the tail. Therefore, in situations where a distribution may have a heavy tail, the superquantile accounts for magnitudes of low-probability large-loss tail events while the quantile does not account for this information.

Working to extend this concept, bPOE was developed as the inverse of the superquantile in the same way that POE is the inverse of the quantile. Specifically, bPOE is defined in the following way, where $\sup X$ denotes the essential supremum of random variable X.

Definition 1 (Mafusalov and Uryasev (2015))**.** *bPOE of random variable X at threshold z equals*

$$\bar{p}_z(X) = \begin{cases} \max\{1-\alpha|\bar{q}_\alpha(X) \geq z\}, & \text{if } z \leq \sup X, \\ 0, & \text{otherwise.} \end{cases}$$

In words, bPOE calculates one minus the probability level at which the superquantile equals the threshold. Roughly speaking, bPOE calculates the proportion of worst case outcomes which average to z. We note that there exist two slightly different variants of bPOE, called Upper and Lower bPOE. For this paper, we utilize Upper bPOE. For the interested reader, details regarding the difference between Upper and Lower bPOE are contained in the appendix.

3.2 Calculation of bPOE

Using Definition 1, bPOE would seem troublesome to calculate. In Proposition 1, we introduce a new calculation formula for bPOE. We view this new formula as a critical step in development

of the bPOE concept, as it allows some bPOE minimization problems to be reduced to convex and linear programming. Additionally, calculating bPOE at threshold z with this formula allows simultaneous calculation of the quantile at probability level one minus bPOE.

Proposition 1. *Given a real valued random variable X and a fixed threshold z, bPOE for random variable X at z equals*

$$\bar{p}_z(X) = \inf_{\gamma<z} \frac{E[X-\gamma]^+}{z-\gamma} = \begin{cases} \lim\limits_{\gamma\to-\infty} \frac{E[X-\gamma]^+}{z-\gamma} = 1 \,, & \text{if } z \le E[X] \,, \\ \min\limits_{\gamma<z} \frac{E[X-\gamma]^+}{z-\gamma} \,, & \text{if } E[X] < z < \sup X \,, \\ \lim\limits_{\gamma\to z^-} \frac{E[X-\gamma]^+}{z-\gamma} = P(X = \sup X) \,, & \text{if } z = \sup X \,, \\ \min\limits_{\gamma<z} \frac{E[X-\gamma]^+}{z-\gamma} = 0 \,, & \text{if } \sup X < z. \end{cases} \tag{4}$$

Furthermore, if $z \in (E[X], \sup X)$ then $\gamma^* = q_{1-\bar{p}_z(X)} \in \operatorname*{argmin}_{\gamma<z} \frac{E[X-\gamma]^+}{z-\gamma}$.

Proof. We prove four cases. Note that case 1 and 3 coincide for constant random variable X, when $z = \sup X$.

Case 1: $z \le E[X]$.
Assume $z \le E[X]$. First, note that $\bar{p}_z(X) = \max\{1-\alpha | \bar{q}_\alpha(X) \ge z\} = 1$. This follows from the fact that $\bar{q}_0(X) = E[X]$. Then, notice that

$$\inf_{\gamma<z} \frac{E[X-\gamma]^+}{z-\gamma} = \inf_{0<z-\gamma} E[\frac{X}{z-\gamma} - \frac{\gamma}{z-\gamma}]^+ \,. \tag{5}$$

Letting $a = \frac{1}{z-\gamma}$, we get

$$\inf_{0<z-\gamma} E[\frac{X}{z-\gamma} - \frac{\gamma}{z-\gamma}]^+ = \inf_{a>0} E[aX + a(\frac{1}{a} - z)]^+ = \inf_{a>0} E[a(X-z)+1]^+ \,. \tag{6}$$

Now, $0 \le E[X] - z \implies$ for every $a > 0$, $E[a(X-z)+1]^+ \ge E[a(X-z)+1] \ge a(E[X]-z)+1 \ge 1$. This implies that,

$$0 \in \operatorname*{argmin}_{a\ge0} E[a(X-z)+1]^+ \,.$$

Then, notice that since $0 \in \operatorname*{argmin}_{a\ge0} E[a(X-z)+1]^+$ and that for every $a > 0$, $E[a(X-z)+1]^+ \ge 1$ we have that

$$\inf_{a>0} E[a(X-z)+1]^+ = \min_{a\ge0} E[a(X-z)+1]^+ = E[0(X-z)+1]^+ = 1 \,.$$

Finally, noting that if $a = \frac{1}{z-\gamma}$ then $\lim_{(z-\gamma)\to\infty} \frac{1}{z-\gamma} = 0 = a$ and

$$\begin{aligned} \inf_{0<z-\gamma} \frac{E[X-\gamma]^+}{z-\gamma} &= \min_{a\ge0} E[a(X-z)+1]^+ = E[0(X-z)+1]^+ \\ &= \lim_{(z-\gamma)\to\infty} \frac{E[X-\gamma]^+}{z-\gamma} = 1 \,. \end{aligned}$$

Case 2: $E[X] < z < \sup X$.
Assume that $E[X] < z < \sup X$. This assumption and Definition 2 imply that

$$\bar{p}_z(X) = \max\{1-\alpha|\bar{q}_\alpha(X) \geq z\} = \min\{1-\alpha|\bar{q}_\alpha(X) \leq z\} \; . \tag{7}$$

Recall the formula for the superquantile given in Rockafellar and Uryasev (2000),

$$\bar{q}_\alpha(X) = \min_\gamma \; \left[\gamma + \frac{E[X-\gamma]^+}{1-\alpha}\right] = \min_\gamma \; g(X,\alpha,\gamma) \; . \tag{8}$$

Note also Rockafellar and Uryasev (2000) states that if $\gamma^* = \underset{\gamma}{\operatorname{argmin}}\; g(X,\alpha,\gamma)$, then
$\bar{q}_\alpha(X) = \gamma^* + \frac{E[X-\gamma^*]^+}{1-\alpha}$ and $\gamma^* = q_\alpha(X)$.
Next, using (7) and (8) we get

$$\bar{p}_z(X) = \min\{1-\alpha : \min_\gamma \; g(X,\alpha,\gamma) \leq z\} \; . \tag{9}$$

Then, considering (8) we can write (9) as,

$$\begin{aligned} \bar{p}_z(X) = & \min_{\alpha,\gamma} \quad 1-\alpha \\ & s.t. \quad \gamma + \frac{E[X-\gamma]^+}{1-\alpha} \leq z \; . \end{aligned} \tag{10}$$

Let (γ^*, α^*) denote an optimal solution vector to (10). Since $z < \sup X$, the formula (8) implies that

$$\gamma^* = q_{\alpha^*}(X) < \bar{q}_{\alpha^*}(X) = z \; .$$

This implies that $\gamma^* < z$. Explicitly enforcing the constraint $\gamma < z$ allows us to rearrange (10) without changing the optimal solution or objective value,

$$\begin{aligned} \bar{p}_z(X) = & \min_{\alpha,\gamma<z} \quad 1-\alpha \\ & s.t. \quad 1-\alpha \geq \frac{E[X-\gamma]^+}{z-\gamma} \; . \end{aligned} \tag{11}$$

Simplifying further, this becomes

$$\bar{p}_z(X) = \min_{\gamma<z} \quad \frac{E[X-\gamma]^+}{z-\gamma} \; . \tag{12}$$

Case 3: $z = \sup X$.
Assume $z = \sup X$. First, note that $\bar{p}_z(X) = \max\{1-\alpha|\bar{q}_\alpha(X) \geq z\} = P(X = \sup X)$. This follows from the fact that $\bar{q}_{(1-P(X=\sup X))}(X) = \sup X$. Next, recall that with (5) and (6) for $a = \frac{1}{z-\gamma}$, we get

$$\inf_{\gamma<z} \frac{E[X-\gamma]^+}{z-\gamma} = \inf_{a>0} E[a(X-z)+1]^+ \; .$$

Since $\sup X - z = 0$, we have

$$\inf_{a>0} E[a(X-z)+1]^+ = \lim_{a\to\infty} E[a(X-z)+1]^+ = P(X = \sup X) \; .$$

To see this, notice that for any realization X_0 of X, where $X_0 - z < -\frac{1}{a}$, we get $[a(X_0 - z) + 1]^+ = 0$. Furthermore, for any realization X_1 of X where $X_1 = \sup X = z$ we have that $[a(X_1 - z) + 1]^+ = [0 + 1]^+ = 1$. Thus,

$$\lim_{a\to\infty} E[a(X - z) + 1]^+ = 0 * \left(\lim_{a\to\infty} P(X - z < -\frac{1}{a}) \right) + 1 * P(X = \sup X) = P(X = \sup X) \ .$$

Case 4: $z > \sup X$.
Assume that $z > \sup X$. First, note that $\bar{p}_z(X) = 0$. This follows immediately from Definition 2 (i.e. the 'otherwise' case). Next, recall again that with (5) and (6) for $a = \frac{1}{z-\gamma}$, we get

$$\inf_{\gamma<z} \frac{E[X - \gamma]^+}{z - \gamma} = \inf_{a>0} E[a(X - z) + 1]^+ \ .$$

Since $\sup X - z < 0$, then for any $0 < a \leq z - \sup X$ we have that $P(\frac{X-z}{a} \leq -1) = 1$ implying that $E[\frac{X-z}{a} + 1]^+ = 0$. This gives us that

$$\inf_{a>0} E[a(X - z) + 1]^+ = \min_{a>0} E[a(X - z) + 1]^+ = 0 \ .$$

□

Thus, via Proposition 1 we have provided a surprisingly simple formula for calculating bPOE that is extremely similar to formula (3). In the following section, we show that the true power of formula (4) lies in the fact that it can be utilized to reduce particular bPOE minimization problems to convex, sometimes even linear, programming. For an in-depth study of bPOE and its specific properties, we refer readers to Mafusalov and Uryasev (2015).

3.3 bPOE Optimization

To demonstrate the ease with which bPOE can be integrated into optimization frameworks, particularly when compared to POE, consider the following optimization setup. Assume we have a real valued positive homogenous random function $f(w, X)$ determined by a vector of control variables $w \in \mathbb{R}^n$ and a random vector X. By definition, a function $f(w, X)$ is "positive homogeneous" with respect to w if it satisfies the following condition: $af(w, X) = f(aw, X)$ for any $a \geq 0, a \in \mathbb{R}$. Note that we consider only positive homogeneous functions since they are the type of error function we consider in the case of AUC.

Now, assume that we would like to find the vector of control variables, $w \in \mathbb{R}^n$, that minimize the probability of $f(w, X)$ exceeding a threshold of $z = 0$. We would like to solve the following POE optimization problem.

$$\min_{w\in\mathbb{R}^n} \quad p_0(f(w, X)) \ . \tag{13}$$

Here we have a discontinuous and non-convex objective function (for discretely distributed X) that is numerically difficult to minimize. Consider minimization of bPOE, instead of POE, at the same threshold $z = 0$. This is posed as the optimization problem

$$\min_{w\in\mathbb{R}^n} \quad \bar{p}_0(f(w, X)) \ . \tag{14}$$

Given Proposition 1, (14) can be transformed into the following.

$$\min_{w\in\mathbb{R}^n,\gamma<0} \quad \frac{E[f(w, X) - \gamma]^+}{-\gamma} \ . \tag{15}$$

Notice, though, that the positive homogeneity of $f(w, X)$ allows us to further simplify (15) by getting rid of the γ variable. Thus, we find that bPOE minimization of $f(w, X)$ at threshold $z = 0$ can be reduced to (16).

$$\min_{w \in \mathbb{R}^n} \quad E[f(w, X) + 1]^+ . \tag{16}$$

For convex f, (16) is a convex program. Furthermore, if f is linear, (16) can be reduced to linear programming. This is substantially easier to handle numerically than the non-convex and discontinuous POE minimization (13).

Given the attractiveness of bPOE and the superquantile within the optimization context, we are inclined to apply these concepts to define a bPOE variant of AUC. Not only would this buffered variant give way to more well behaved optimization problems, but it would provide a measure of classifier performance that considers the magnitude of ranking errors $\xi_{ij}(h)$ instead of only a discrete count of the number of ranking errors exceeding zero.

4 Buffered AUC: A New Performance Metric

4.1 Buffered AUC

With AUC defined as $1 - P(\xi(h) \geq 0)$, we can create a natural alternative to AUC called *Buffered AUC* (bAUC) by using bPOE instead of POE. If we assume that we have samples of our random vectors X^+, X^- and are thus working with the empirical distribution of ranking errors $\xi_{ij}(h)$, we have that bAUC equals one minus the proportion of largest ranking errors $\xi_{ij}(h)$ that have average magnitude equal to zero. Specifically, we have the following general definition.

Definition 2 (*Buffered AUC*)**.** *For a scoring function $h : \mathbb{R}^n \to \mathbb{R}$, bAUC of h is defined as*

$$bAUC(h) = 1 - \bar{p}_0(\xi(h)) . \tag{17}$$

To begin, we can look at a graphical example comparing bAUC and AUC in Figure 2. Here, we plot the distribution of ranking errors $\xi_{ij}(h)$ for a fixed scoring function h for some dataset. In the bottom chart, we highlight the errors exceeding zero, i.e. the ranking errors considered by AUC. Thus, in the bottom chart, AUC equals one minus the proportion of errors larger than zero. In the top chart, we highlight the largest errors that have average magnitude equal to zero, i.e. the ranking errors considered by bAUC. Thus, in the top chart, we see that bAUC is smaller than AUC, as it considers not only errors larger than zero but also some negative errors most near to zero.

This metric, utilizing bPOE instead of POE, is similar to AUC. Both are concerned with ranking errors, measuring the tail of the error distribution $\xi(h)$. In fact, as shown below in Proposition 2, bAUC is a lower bound for AUC. Thus, classifiers with large bAUC necessarily have large AUC.

Proposition 2. *For a scoring function $h : \mathbb{R}^n \to \mathbb{R}$,*

$$bAUC(h) \leq AUC(h)$$

Proof. From Mafusalov and Uryasev (2015), we know that for any threshold $z \in \mathbb{R}$ and real valued random variable X that, $P(X > z) \leq \bar{p}_z(X)$. Therefore, $1 - \bar{p}_0(\xi(h)) \leq 1 - P((\xi(h) \geq 0)$.

□

Unlike AUC, though, bAUC is sensitive to the magnitude of ranking errors $\xi(h)$. In addition, bAUC does not only consider ranking *errors*, meaning $\xi_{ij}(h) > 0$. It also takes into account the

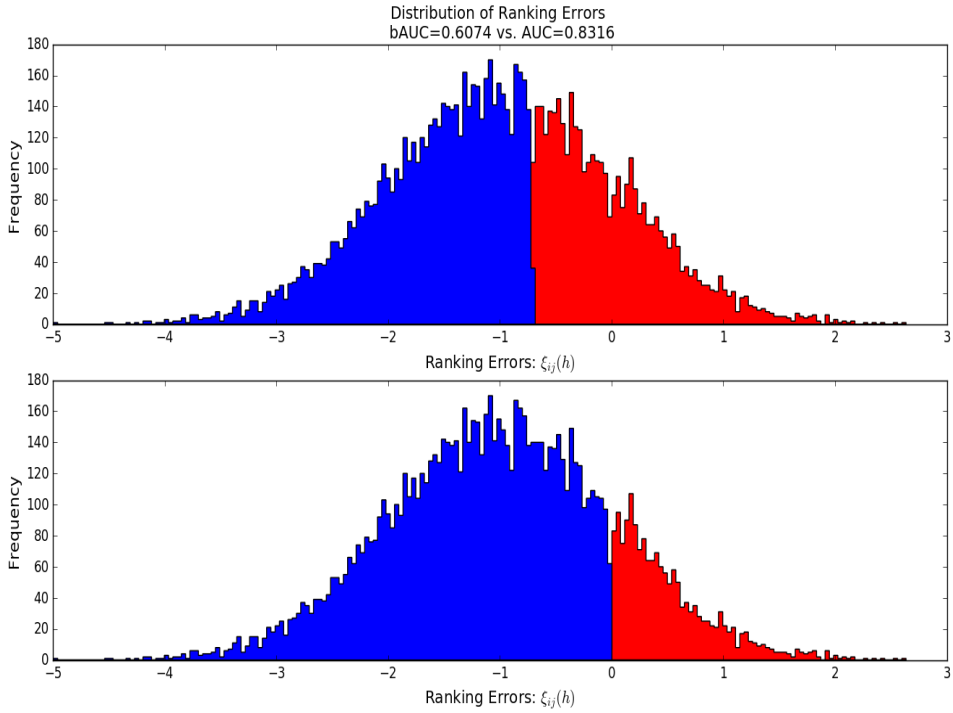

Figure 2: In both charts, we plot the same distribution of ranking errors $\xi_{ij}(h)$ for a fixed h. In the top chart, we highlight the largest errors that have average magnitude equal to zero, i.e. the errors considered by bAUC. In the bottom chart, we highlight the errors that exceed zero, i.e. the errors considered by AUC. We have that bAUC=.6074 and AUC=.8316.

confidence with which the classifier *correctly* ranked some instances, meaning the 'errors' that are less than, but most near to zero. These correctly ranked instances constitute the *buffer*. We discuss this concept and other differences further in the next section.

4.2 The bAUC Buffer and Sensitivity to Classifier Confidence

Focusing on the benefits of bAUC's sensitivity to the magnitude of ranking errors $\xi_{ij}(h)$, we provide two examples illustrating situations where two classifiers give the same AUC, but where one of the classifiers is clearly a better ranker than the other. We show how bAUC reveals this discrepancy. The first example focuses on the importance of the bAUC buffer. The second example simply illustrates a situation where the magnitude of the ranking errors larger than zero, $\xi_{ij}(h) > 0$, would be important when selecting between classifiers.

As already mentioned, bAUC considers the magnitude of the positive errors, $\xi_{ij}(h) > 0$. Importantly, bAUC also considers the magnitude of the 'errors' that are less than, but most near to zero. This buffer may be important as illustrated in the following example. Let I_λ be an indicator function as specified in Section 2.1. Consider the task of comparing the ranking ability (on the

same data set) of two imperfect classifiers[2], h_1 and h_2, that have equal AUC values, meaning that

$$(a) \quad \left(1 - \frac{1}{m^+m^-}\sum_i\sum_j I_{\xi_{ij}(h_1)\geq 0}\right) = \left(1 - \frac{1}{m^+m^-}\sum_i\sum_j I_{\xi_{ij}(h_2)\geq 0}\right) > 0 \; .$$

Assume also that both classifiers produce incorrect rankings with the same magnitude (i.e. confidence), meaning that

$$(b) \;\; \xi_{ij}(h_1) = 1 \;\; \forall \;\; (i,j) \text{ with } \xi_{ij}(h_1) \geq 0 \text{ and } \xi_{ij}(h_2) = 1 \;\; \forall \;\; (i,j) \text{ with } \xi_{ij}(h_2) \geq 0 \; .$$

Finally, assume that h_1 produces correct rankings more confidently than h_2, where

$$(c) \;\; \forall \;\; (i,j) \text{ such that } \xi_{ij}(h_1) < 0 \text{ we have that } \xi_{ij}(h_1) < \min_{s,k} \xi_{sk}(h_2) \;\; .$$

From (a), we see that both classifiers will have equal AUC. Then from (b), we see that the classifiers have identical distributions of errors greater than or equal to zero. But finally, considering (c) reveals that the distribution of negative errors (i.e. the correct rankings) for h_1 is more favorable than that of h_2. Thus, we see that h_1 is superior w.r.t. ranking ability. The AUC metric does not reveal this fact, since both classifiers have equal AUC. The bAUC metric, though, because of the *buffer*, correctly distinguishes between the ranking ability of h_1 and h_2. Specifically, we will find that $bAUC(h_1) > bAUC(h_2)$ with the *buffer* accounting for the magnitude of errors not only in (b), but also in (c).[3]

Illustrating a similar situation, not necessarily involving the buffer but instead involving bAUC's sensitivity to the magnitude of positive ranking errors, consider again two classifiers, h_1 and h_2, with equal AUC (i.e. satisfying (a)). Assume also that both classifiers produce correct rankings with the same magnitude (i.e. confidence), meaning that

$$(d) \;\; \xi_{ij}(h_1) = -1 \;\; \forall \;\; (i,j) \text{ with } \xi_{ij}(h_1) < 0 \text{ and } \xi_{ij}(h_2) = -1 \;\; \forall \;\; (i,j) \text{ with } \xi_{ij}(h_2) < 0 \; .$$

Finally, assume that h_2 produces incorrect rankings more severe than those produced by h_1, where

$$(e) \;\; \forall \;\; (i,j) \text{ such that } \xi_{ij}(h_2) \geq 0 \text{ we have that } \xi_{ij}(h_2) > \max_{s,k} \xi_{sk}(h_1) \;\; .$$

From (a), we see that both classifiers will have equal AUC. Then from (d), we see that the classifiers have identical distributions of errors less than zero. But finally, considering (e) reveals that the distribution of errors larger than or equal to zero (i.e. the incorrect rankings) for h_1 are more favorable than that of h_2. Once again, AUC indicates that these classifiers perform equivalently with respect to ranking ability. The bAUC metric, though, by considering the magnitude of errors, is able to properly distinguish between the two classifiers. Specifically, because of (d) and (e), we will have that $bAUC(h_1) > bAUC(h_2)$.[4]

4.3 Optimizing bAUC

Direct maximization of AUC is rarely done due to the troublesome properties of probabilistic objectives, even for the simplest classifier such as the linear classifier $h(X) - t = w^T X - t$, $w \in \mathbb{R}^n$.

[2]Although we say "classifier", we are omitting the decision thresholds t_1, t_2 since they are not necessary for AUC and bAUC.

[3]We do make the assumption that $bAUC(h_1) \neq 0$, which is to assume that $E[\xi(h_1)] < 0$.

[4]Again, we make the assumption that $bAUC(h_1) \neq 0$, which is to assume that $E[\xi(h_1)] < 0$.

Direct maximization of bAUC, on the other hand, reduces to convex programming and linear programming for the linear classifier. Let $\xi(w) = -w^T(X^+ - X^-)$. Maximization of AUC takes the form,

$$\max_{w \in \mathbb{R}^n} \quad 1 - P\left(\xi(w) \geq 0\right) \ , \tag{18}$$

where the probabilistic objective is discontinuous and non-convex when dealing with empirical observations of X^+ and X^-. Maximization of bAUC takes the form

$$\max_{w \in \mathbb{R}^n} 1 - \bar{p}_0\left(\xi(w)\right) = 1 - \min_{w \in \mathbb{R}^n} \bar{p}_0\left(\xi(w)\right) \ . \tag{19}$$

Applying Proposition 1, (19) becomes

$$1 - \min_{w \in \mathbb{R}^n, \gamma < 0} \frac{E\left[\xi(w) - \gamma\right]^+}{-\gamma} \ . \tag{20}$$

Finally, given the positive homogeneity of $\xi(w)$, we can apply minimization formula (16) and simplify to (21),

$$\min_{w \in \mathbb{R}^n} \quad E[\xi(w) + 1]^+ \ . \tag{21}$$

In financial optimization literature, the function $E[.]^+$ is called *Partial Moment*. It is a very popular function in various applications of stochastic programming.

Here, (21) is a convex optimization problem and, moreover, can be reduced to linear programming with reduction to (22) via auxiliary variables. Thus, in the case of a linear classifier, maximizing bAUC is substantially easier to handle than AUC maximization, a non-convex and discontinuous optimization problem.

$$\begin{array}{ll} \min\limits_{w \in \mathbb{R}^n, \beta_{ij} \in \mathbb{R}} & \dfrac{1}{m^+ m^-} \displaystyle\sum_{i=1}^{m^+} \sum_{j=1}^{m^-} \beta_{ij} \\ s.t. & \beta_{ij} \geq \xi_{ij}(w) + 1, \ \forall \ i = 1, ..., m^+, j = 1, ..., m^- \\ & \beta_{ij} \geq 0 \ . \end{array} \tag{22}$$

4.4 bAUC and the ROC curve

As discussed in Section 2.1, AUC can also be defined as the area under the ROC curve. We show here that bAUC can also be represented as the area under a slightly modified ROC curve, which we call the Buffered ROC (bROC) curve.

Proposition 3. *For a fixed scoring function $h : \mathbb{R}^n \to \mathbb{R}$, assume that $\xi(h)$ is continuously distributed and that*

$$bAUC(h) = 1 - \alpha = 1 - \frac{E[\xi(h) - \gamma^*]^+}{-\gamma^*} \ \textit{where} \ \gamma^* \in \underset{\gamma < 0}{\textit{argmin}} \frac{E[\xi(h) - \gamma]^+}{-\gamma} \ .$$

Then,

$$bAUC(h) = \int_t P\left(h(X^+) \geq t - \gamma^*\right) dP\left(h(X^-) > t\right) .$$

Proof. This follows from Proposition 1. Specifically, if $z \in (E[X], \sup X)$ and $\gamma^* \in \underset{\gamma<z}{\operatorname{argmin}} \frac{E[X-\gamma]^+}{z-\gamma}$ then, since $\xi(h)$ is continuously distributed, we know that $\gamma^* = q_{1-\bar{p}_z(X)}$. This, along with the continuity again, implies that $P(X > \gamma^*) = \bar{p}_z(X)$. Applying this to bAUC, we get that

$$\begin{aligned} bAUC(h) &= 1 - \bar{p}_0\left(-h(X^+) + h(X^-)\right) \\ &= 1 - P(-h(X^+) + h(X^-) > \gamma^*) \\ &= P(-h(X^+) + h(X^-) \leq \gamma^*) \\ &= P(h(X^+) + \gamma^* \geq h(X^-)) \\ &= \int_t P\left(h(X^+) \geq t - \gamma^*\right) dP\left(h(X^-) > t\right) \ . \end{aligned}$$

□

Proposition 3 is shown graphically in Figure 3 with a slightly modified ROC plot. Here, instead of plotting the pairs $(P(h(X^-) > t), P(h(X^+) > t))$ for all thresholds $t \in \mathbb{R}$ to generate an ROC curve, we plot the pairs $(P(h(X^-) > t), P(h(X^+) \geq t - \gamma^*))$ for all thresholds $t \in \mathbb{R}$ to generate the bROC curve. This yields a curve on the modified ROC plot that has area underneath it equal to $bAUC(h)$. Note that the continuity assumption on $\xi(h)$ assures that $P(X > \gamma^*) = \bar{p}_z(X)$. For non-continuous distributions, this relation is approximate, since we can only assure that $\gamma^* = q_{1-\bar{p}_z(X)}$ is an argmin. In this case, though, we follow the methodology of AUC and simply approximate the curve via interpolation.

We can also interpret the bROC curve as the normal ROC curve of a more conservative scoring function $\hat{h}$ with positive class score distribution $\hat{h}(X^+) = h(X^+) + \gamma^*$ and negative class score distribution $\hat{h}(X^-) = h(X^-)$.[5] We have shifted the distribution of positive class scores toward the distribution of negative class scores by the amount γ^*. By Proposition 4, we know that $\gamma^* = \{z \in \mathbb{R} \mid E[\xi(h)|\xi(h) > z] = 0\} < 0$. Therefore, since γ^* is determined by the tail of the error distribution $\xi(h)$, the bROC curve is the ROC curve of a conservative variant of the original scoring function, where the magnitude of conservatism is based upon the ranking errors produced by the scoring function h.

4.5 Numerical Experiments

In this section, we present numerical examples demonstrating that: (1) bAUC is a counterpart of AUC that can be used to differentiate between classifiers with almost identical AUC; (2) bAUC does not always agree with AUC and is indeed a unique performance metric that does not simply move linearly with AUC. In these experiments, we also show that the bROC curve can provide additional discriminatory insights not revealed by the ROC curve.

For all experiments, we use real data sets from the UCI machine learning repository Lichman (2013) and compare an SVM classifier with a logistic regression classifier (both $L2$ regularized[6]).

4.5.1 bAUC as a tiebreaker

In some cases, two classifiers can yield similar AUC values, even if their score distributions are vastly different. Here, much like the theoretical example provided in Section 4.2, we show that bAUC can reveal classifier properties which are not reflected by AUC. We also show that while

[5] The choice of score distribution to shift is arbitrary. One can shift $h(X^-)$ by $-\gamma^*$.

[6] The specific value of the tradeoff parameter is not particularly noteworthy, so we do not list it. The purpose here is to simply compare classifiers as opposed to searching for an optimal one via parameter search.

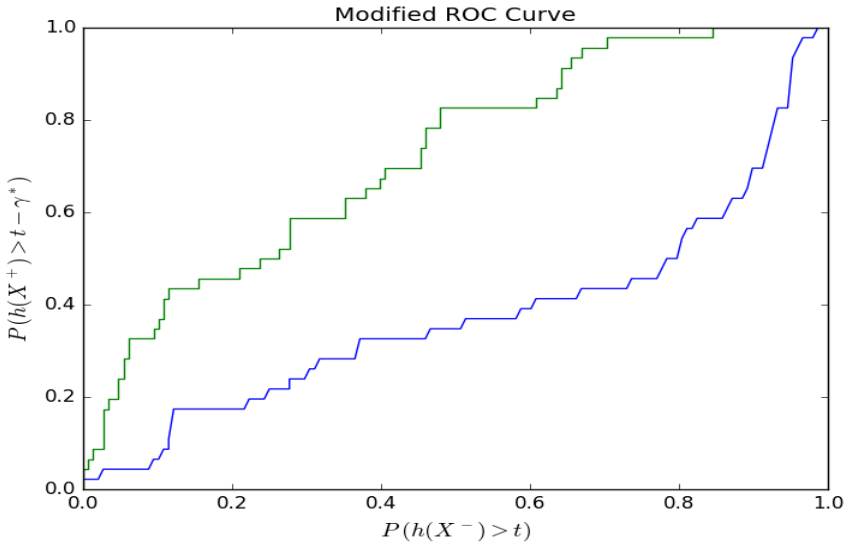

Figure 3: We have a fixed classifier h. The area under the upper curve corresponds to $AUC(h)$, where $\gamma^* = 0$. The area under the lower curve corresponds to $bAUC(h)$, where $\gamma^* < 0$.

the ROC curves may be almost identical, the bROC curves can be different, offering additionally discriminatory insights to supplement the summary insight of the bAUC metric.

For the first experiment, we compared two classifiers trained on the Page-Blocks dataset. Figure 4 shows the ROC and bROC curves for these classifiers as well as their AUC and bAUC values. First, notice that for these two classifiers, AUC values and ROC curves are almost identical, providing little discriminatory insight to compare classifiers. Looking at bAUC values, we see that the logistic regression classifier has larger bAUC. Additionally, we see that the bROC curves are dramatically different. Looking at the slope of the bROC curves, we can see that the SVM classifier is quit unstable, with the large slope revealing that the score distribution is highly concentrated on a small threshold interval. We can demonstrate this fact by looking at the score distributions themselves in Figure 5. Clearly, the logistic regression classifier produces a more stable score distribution with respect to threshold changes, as the SVM has a score distribution that is very sensitive to changes

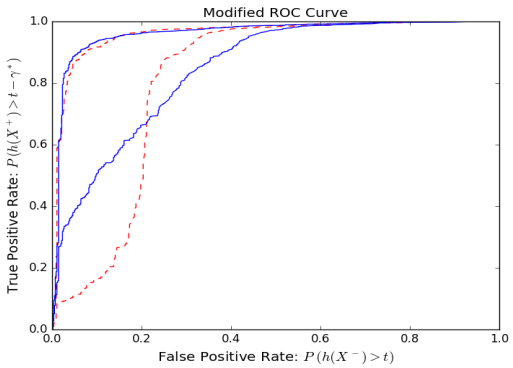

Figure 4: Solid line corresponds to logistic regression with AUC=.962 and bAUC=.844, dashed line is SVM with AUC=.961 and bAUC=.810.

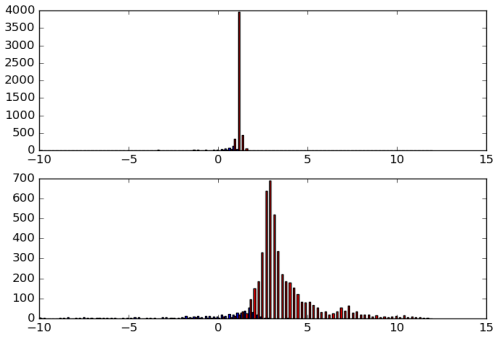

Figure 5: Histogram of scores provided by classifiers. Upper chart is SVM scores. Lower chart is logistic regression scores.

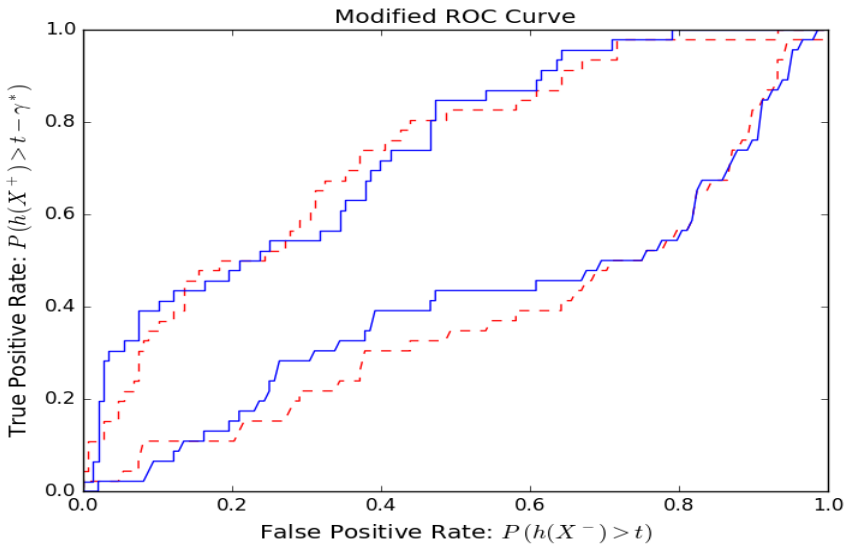

Figure 6: Solid line is logistic regression with AUC=.738 and bAUC=.406, dashed line is SVM with AUC=.731 and bAUC=.376.

in decision threshold because of the concentration of scores in a very small range.

For the second experiment, we compare a logistic regression and SVM classifier trained on the Breast Cancer Wisconsin Prognostic dataset. Figure 6 shows the resulting bAUC and AUC values and the ROC and bROC curves. In addition to the bAUC metric yielding a more definitive discrimination between classifiers, we see that it is easier to gain insight regarding the dominance of a classifier over thresholds by looking at the bROC curve. Looking at the ROC curves, it is difficult to differentiate, with the curves crossing multiple times. Looking at the bROC curve, though, the logistic regression classifier tends to dominate the SVM classifier except for thresholds at the extremes of the spectrum. If having to select between these two classifiers, bAUC and the bROC curve add confidence to the argument that the logistic classifier is superior.

4.5.2 bAUC as a different metric

With bAUC being a lower bound of AUC that is similarly measuring a classifier's ability to properly rank, these values will often be in agreement as to the superior classifier. We demonstrate, though, that this is not always the case and present two examples. For the first experiment, we trained classifiers on the Liver Disorders dataset. Figure 7 shows the resulting AUC and bAUC values as well as the ROC and bROC curves. Not only are the bAUC and AUC metrics in disagreement as to the optimal classifier, but the ROC and bROC curves are almost complete opposites. The ROC curves and bROC curves indicate that the classifiers are superior for completely opposite ranges of threshold.

For the second experiment, we trained again on the Breast Cancer Wisconsin Prognostic dataset yielding different classifiers by altering the regularization parameters from previous experiments. Figure 8 shows that the bAUC and AUC values do not agree. Additionally, one can see that the bROC curve is much easier to read. The ROC curves cross multiple times, making it almost impossible to differentiate via visual inspection. The bROC curves only cross once, making analysis of classifier performance much easier with respect to different ranges of threshold choice.

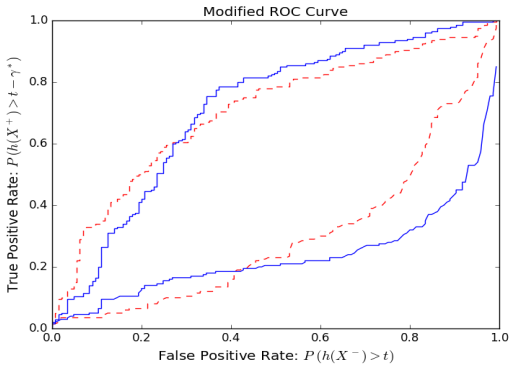

Figure 7: Solid line is logistic regression with AUC=.711 and bAUC=.234, dashed line is SVM with AUC=.704 and bAUC=.298.

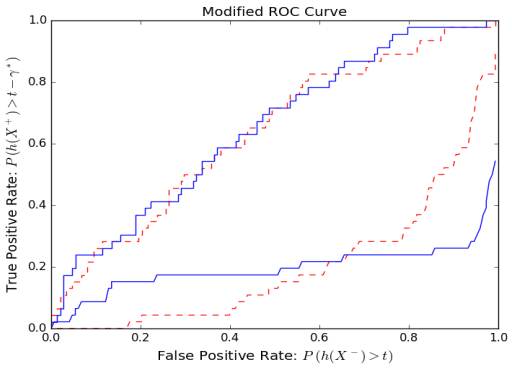

Figure 8: Solid line is logistic regression with AUC=.651 and bAUC=.195, dashed line is SVM with AUC=.638 and bAUC=.200.

4.5.3 Software Implementation for Linear Classifiers

For linear classifier $h(X) - t = w^T X - t$, $w \in \mathbb{R}^n$ the maximization of bAUC reduces to bPOE minimization for linear function $\xi(w) = -w^T(X^+ - X^-) = -w^T X^+ + wX^-$, see (19) and (21). Although this minimization is convex w.r.t. decision variables, the implementation of bAUC calculation and optimization is non-trivial. In particular, calculating bAUC for a sample distribution requires calculating m^+m^- instances of $\xi_{ij}(h)$. Additionally, the LP representation (22) for bAUC optimization has $\mathcal{O}(m^+m^-)$ constraints. Because of this, many commercial optimization packages may struggle to handle bAUC calculation and optimization efficiently.

From a practical point of view, if bAUC is to be effectively utilized in experimentation, it is critical that there exist a highly efficient implementation of bAUC calculation and optimization. Software should take into account the special structure of the optimization problem. We have used Portfolio Safeguard (PSG)[7] which has specialized routines for Partial Moment and bPOE minimization. The Partial Moment and bPOE function, as well as many other stochastic functions, are precoded allowing the user to include them in analytic format in optimization problem statements. Because of this precoding, PSG can efficiently invoke specially developed algorithms for these analytic expressions.

For the interested reader, a PSG case study using the precoded partial moment function can be found online.[8] This case study provides data sets and PSG codes for both MATLAB and Run-File (text) environments. Example PSG code for Partial Moment minimization, which implements the problem statement (21), is as follows:

```
minimize
  pm_pen(-1,L(matrix_0)-L(matrix_1))
```

The function `pm_pen(-1,L(matrix_0)-L(matrix_1))` is the partial moment function

$$E[w^T X^- - w^T X^+ + 1]^+ .$$

This function is applied to the difference of random values $w^T X^- - w^T X^+$ exceeding -1, with matrices of scenarios `matrix_0` and `matrix_1` defining these random differences. The code can handle large sample sizes. For example, Problem 1 in the case study minimizes the partial moment

[7] www.aorda.com

[8] http://www.ise.ufl.edu/uryasev/research/testproblems/advanced-statistics/case-study-bAUC-maximization/

with $m^+ = 3,990$, $m^- = 2,788$ in 0.13 second on a 3.14GHz PC. This problem is equivalent to a bPOE minimization problem with $m^+m^- = 11,124,120$ scenarios.

PSG can also maximize bAUC directly by using the precoded bPOE function, as specified in the problem statement (19). An example of bAUC maximization using the precoded bPOE function can be found online[9].

5 Generalized bAUC: Utilizing Non-zero Thresholds

Previously, we considered the definition of bAUC to be one minus the buffered probability of the error function $\xi(h)$ exceeding the threshold of $z = 0$ (i.e definition (2)). Consider now a more general definition of bAUC with thresholds $z \in \mathbb{R}$.

Definition 3. *Generalized bAUC is defined as follows,*

$$bAUC_z(h) = 1 - \bar{p}_z(\xi(h)) \ .$$

5.1 Generalized bAUC and the ROC Curve

Just as bAUC was shown to correspond to the area under a modified ROC curve, we have that $bAUC_z$ for any $z \in \mathbb{R}$ corresponds to the area under a curve on the same, modified ROC plot. This generates a family of ROC curves, in which AUC and bAUC are members. Specifically, we have the following proposition, the proof of which we omit since it is essentially identical to that of Proposition 3.

Proposition 4. *For a fixed scoring function $h : \mathbb{R}^n \to \mathbb{R}$, assume that $\xi(h)$ is continuously distributed and that $bAUC_z(h) = 1 - \alpha = 1 - \frac{E[\xi(h)-\gamma^*]^+}{z-\gamma^*}$ where $\gamma^* \in \underset{\gamma<z}{argmin} \frac{E[\xi(h)-\gamma]^+}{z-\gamma}$. Then,*

$$bAUC_z(h) = \int_t P\left(h(X^+) \geq t - \gamma^*\right) dP\left(h(X^-) > t\right) \ .$$

Notice in Proposition 4, that if we choose z_0 such that $\gamma^* = 0$, we will have $bAUC_{z_0}(h) = AUC(h)$. Thus, we see that AUC belongs to the family of curves associated with $bAUC_z$, $z \in \mathbb{R}$. Showing this on the ROC plot, we have Figure 9 which displays a family of $bAUC_z$ curves.

5.2 Maximizing Generalized bAUC

Given Generalized bAUC, it is not immediately clear how to utilize it. Here, we show that Generalized bAUC has already been utilized successfully for AUC maximization, albeit not explicitly. Specifically, we find that the popular AUC maximizing RankSVM from Brefeld and Scheffer (2005), Herbrich et al. (1999) is equivalent to a special case of direct maximization of Generalized bAUC. We first provide a formulation for maximizing $bAUC_z$ and then show that the AUC maximizing RankSVM is a special case of this formulation (specifically, for threshold range $z \leq 0$). In this context, we work with $h(X) = w^T X$ and ranking error $\xi(w) = -w^T(X^+ - X^-)$.

Consider the problem of finding the vector $w \in \mathbb{R}^n$ which maximizes $bAUC_z(w)$. In other words, we would like to solve the following optimization problem.

$$\min_{w\in\mathbb{R}^n, \gamma<z} \quad \frac{E[\xi(w)-\gamma]^+}{z-\gamma} \equiv \quad \min_{w\in\mathbb{R}^n} \bar{p}_z\left(\xi(w)\right) \ . \tag{23}$$

[9] http://www.ise.ufl.edu/uryasev/research/testproblems/financial_engineering/%20classification-in-loan-application-process%20/

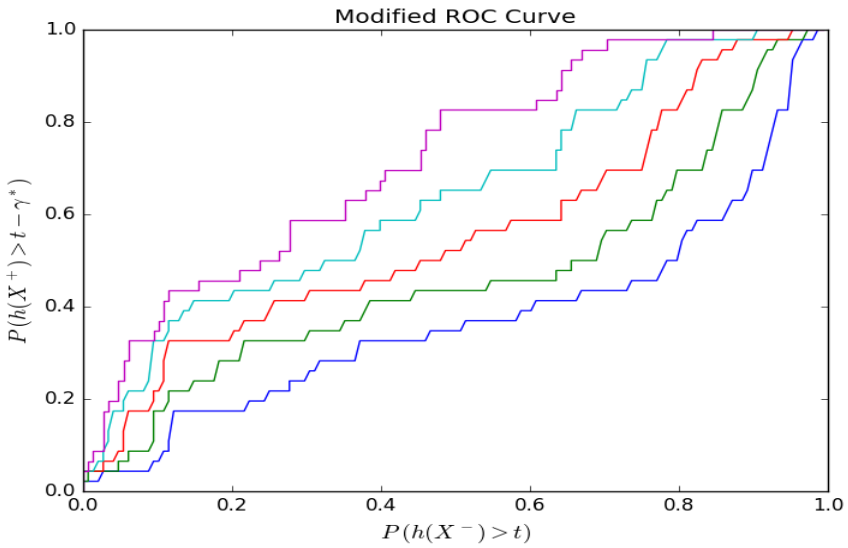

Figure 9: A modified ROC plot for a fixed classifier h. The lower most curve corresponds to $bAUC_0(h)$ while the uppermost curve corresponds to $bAUC_{z_0}(h) = AUC(h)$. The curves in-between correspond to $bAUC_z(h)$ for values of $z \in (0, z_0)$.

However, this problem is ill-posed. As was shown in Norton et al. (2015), this formulation yields trivial solutions for thresholds $z \neq 0$ due to the positive homogeneity of the error function $\xi(w)$ (see appendix of Norton et al. (2015) for details). This issue, though, can be alleviated by fixing the scale of the vector w. This can be accomplished by fixing any general norm on w, effectively minimizing bPOE of the *normalized* error distribution $\frac{\xi(w)}{\|w\|}$. Thus, we can consider the following optimization problem which maximizes $bAUC_z$ for non-zero thresholds, where $\|\cdot\|$ is any general norm,

$$\begin{aligned} \min_{w \in \mathbb{R}^n, \gamma < z} \quad & \frac{E[\xi(w) - \gamma]^+}{z - \gamma} \equiv \min_{w \in \mathbb{R}^n} \bar{p}_z \left(\frac{\xi(w)}{\|w\|} \right) \\ s.t. \quad & \|w\| = 1 \ . \end{aligned} \tag{24}$$

Furthermore, using the result from Norton et al. (2015) we know that to maximize $bAUC_z$, we can alternatively solve the following equivalent problem, which is convex for thresholds $z \leq 0$,

$$\min_{w \in \mathbb{R}^n} \bar{p}_z \left(\frac{\xi(w)}{\|w\|} \right) \equiv \min_{w \in \mathbb{R}^n} E[\xi(w) - z\|w\| + 1]^+ \ . \tag{25}$$

The last formula is easy to interpret. Specifically, adapting a result from Norton et al. (2015), we have the following proposition.

Proposition 5. *For $z \in \mathbb{R}$, assume that*

$$1 - \alpha^* = \min_{w \in \mathbb{R}^n} E[\xi(w) - z\|w\| + 1]^+ = E[\xi(w^*) - z\|w^*\| + 1]^+ \ .$$

Then for the normalized error, $F := \xi\left(\frac{w^*}{\|w^*\|}\right)$, at the optimal point w^*:

$$\bar{p}_z(F) = 1 - \alpha^*, \quad \bar{q}_{\alpha^*}(F) = z, \quad q_{\alpha^*}(F) = z - \frac{1}{\|w^*\|} \ .$$

In the next section, after showing that (25) and the RankSVM are equivalent over the parameter range $z \leq 0$, we find that Proposition 5 provides us with a novel interpretation for the optimal objective value and free parameter of the RankSVM.

5.3 RankSVM Maximizes Generalized bAUC

In Brefeld and Scheffer (2005), Herbrich et al. (1999), the AUC maximizing RankSVM is derived and shown to maximize AUC better than the traditional max-margin SVM's proposed by Cortes and Vapnik (1995). Utilizing a result from Norton et al. (2015), we can show that RankSVM is equivalent to direct maximization of Generalized bAUC for thresholds $z \leq 0$. This serves to show in a more exact manner that the AUC maximizing SVM is, in fact, maximizing a lower bound on AUC, specifically Generalized bAUC. This equivalence also suggests a novel interpretation for the optimal objective value of the RankSVM and the free parameter.

The RankSVM is formulated as follows, where $z \leq 0$ is typically introduced as a parameter specifying the tradeoff between ranking error and regularization. Traditionally, the squared L_2 norm is used, but we use any general norm.

$$\min_{w} \quad -z\|w\| + \frac{1}{m^+m^-}\sum_{i=1}^{m^+}\sum_{j=1}^{m^-}[\xi_{ij}(w)+1]^+ \ . \tag{26}$$

This is a reformulation of the well known C-SVM of Cortes and Vapnik (1995), reformulated for AUC maximization. Let $Y_i \in \{-1,+1\}, i = 1,..,N$ indicate the class of samples $X_1,...,X_N$, let $z \leq 0$, and let $(w,b) \in \mathbb{R}^{n+1}$. The C-SVM is formulated as follows,

$$\min_{w,b} \quad -z\|w\| + \frac{1}{N}\sum_{i=1}^{N}[-Y_i(w^TX_i+b)+1]^+ \ . \tag{27}$$

Relating the C-SVM to bPOE minimization, Norton et al. (2015) introduced the EC-SVM formulation, which is identical to bPOE minimization problem (25) but with error function $\xi(w,b) = -Y(w^TX+b)$. The EC-SVM is formulated as follows, where $z \in \mathbb{R}$.

$$\min_{w,b} \quad E[-Y(w^TX+b) - z\|w\| + 1]^+ \ . \tag{28}$$

Specifically, the EC-SVM and C-SVM were related through the following proposition which shows that the traditional soft margin SVM of Cortes and Vapnik (1995) is equivalent to minimizing bPOE.

Proposition 6. *Consider (27) and (28) formulated with the same norm and assume that we have N equally probable samples (X_i, Y_i), $i = 1,...,N$. Then, over the parameter range $z \leq 0$, (27) and (28) achieve the same set of optimal solutions.*

Using Proposition 6, we can prove that RankSVM is simply maximizing Generalized bAUC.

Proposition 7. *Consider (26) and (25) formulated with the same norm and assume that we have m^+m^- equally probable realizations of the random error $\xi_{ij}(w)$, $i = 1,...,m^+$, $j = 1,...,m^-$. Then, over the parameter range $z \leq 0$, (26) and (25) achieve the same set of optimal solutions.*

Proof. Note that (26) is exactly formulation (27) with m^+m^- samples $(X_i^+ - X_j^-)$, $i = 1,...,m^+$, $j = 1,...,m^-$ all having class $Y_{ij} = +1$ and with the classifier intercept $b = 0$. Thus, applying Proposition 6, we have that (26) and (25) produce the same set of optimal solutions over the parameter range $z \leq 0$. □

With this equivalence, we can draw a novel interpretation of the free parameter of the RankSVM and its optimal objective from Property 5. Specifically, we can now interpret the trade off parameter $z \leq 0$ as Generalized bAUC threshold. Additionally, we can conclude that one minus the optimal objective of the RankSVM equals a probability level, specifically Generalized bAUC for some z. We do not present this formally here, as it follows directly from Norton et al. (2015) and the analysis of the dual formulations of (27) and (28).

6 Buffered Accuracy and SVM's

As already mentioned in the introduction, Accuracy can also be viewed as the probability that misclassification error exceeds the threshold of zero. In binary classification, misclassification error is often characterized as the *margin error*,

$$\xi(w,b) = \frac{-Y(w^T X + b)}{\|w\|} .$$

With this, Accuracy (Acc) is defined as the following.

Definition 4. *Accuracy*

$$Acc(w,b) = 1 - P(\xi(w,b) \geq 0)$$

Just as we did with AUC, we can apply bPOE to create Buffered Accuracy ($bAcc$).

Definition 5. *Buffered Accuracy*

$$bAcc(w,b) = 1 - \bar{p}_0(\xi(w,b))$$

We can also define this in a more general manner, creating Generalized $bAcc$.

Definition 6. *Generalized Buffered Accuracy*

$$bAcc_z(w,b) = 1 - \bar{p}_z(\xi(w,b))$$

In this paper, we do not fully explore the properties and benefits of Buffered Accuracy as an alternative metric. To motivate the general theme of this paper, though, we emphasize the result of Norton et al. (2015) showing that the classical soft margin SVM from Cortes and Vapnik (1995) is simply maximizing Generalized Buffered Accuracy directly. This is exactly what is shown in Proposition 6 which can be seen more clearly by noting that optimization problem (28) is equivalent to the following,

$$\max_{w,b} \quad bAcc_z(w,b) . \tag{29}$$

Therefore, we see that $bAcc$ already plays a major role in classification as an easily optimizable metric alternative to Acc. This lends credibility to the idea of defining bPOE counterparts for metrics defined with POE.

7 Conclusion

AUC is a useful and popular metric for measuring the ranking quality of scores given by a classifier. As a metric defined with POE, though, it does not consider the magnitude of ranking errors and is numerically difficult to optimize. We utilize bPOE to create an informative counterpart

metric called bAUC. We show that bAUC is indeed a counterpart. It is a readily optimizable lower bound of AUC that can also be viewed as the area under a modified ROC curve. We also show that bAUC is an important counterpart to AUC when the magnitude of ranking errors yields important discriminatory information. Additionally, the bROC curve can provide additional discriminatory insights when comparing classifiers, particularly when ROC curves are very similar and cross multiple times.

To facilitate the creation of bAUC, we focused our attention on deriving a novel formula for calculating bPOE, the inverse of the superquantile (CVaR). We show that this formula is significant, allowing certain bPOE minimization problems to be reduced to convex and linear programming. Applying this to bAUC, we show that this reduction applies to bAUC allowing for efficient bAUC maximization.

By considering non-zero bPOE thresholds in the definition of bAUC, we also introduce Generalized bAUC. We show that Generalized bAUC generates a family of metrics, in which AUC and bAUC belong. Furthermore, we show that Generalized bAUC has already found its way into the AUC maximization literature. Specifically, we show that the popular AUC maximizing RankSVM is equivalent to maximization of Generalized bAUC. Thus, bAUC has already, in some sense, been used as a metric counterpart to AUC that is much simpler to optimize.

In the broader scheme, we find evidence that utilizing bPOE to create informative, efficiently optimizable metrics is a fruitful approach. Although we focus our in-depth analysis on creating a bPOE variant of AUC, we show that the bPOE variant of Accuracy already has deep roots in the SVM literature by showing that the famous soft margin C-SVM is equivalent to maximization of Buffered Accuracy. Therefore, this suggests that POE and bPOE can be used in tandem to create counterpart metrics like AUC and bAUC. Defining metrics with POE is highly intuitive, but produces metrics that are numerically difficult to optimize. For example, Accuracy, while being difficult to optimize directly, is an intuitive concept relating to the probability that some error exceeds zero. Utilizing bPOE, one can create a complimentary counterpart to the intuitive POE metric that reveals information about the magnitude of errors while proving to be efficiently optimizable with convex or linear programming.

Acknowledgements

Authors would like to thank Prof. R.T. Rockafellar and Mr. Alexander Mafusalov for their valuable comments and suggestions. This work was partially supported by the following USA Air Force Office of Scientific Research grants: "Design and Redesign of Engineering Systems", FA9550-12-1-0427, and "New Developments in Uncertainty: Linking Risk Management, Reliability, Statistics and Stochastic Optimization", FA9550-11-1-0258.

8 Appendix

Here, we discuss the slight differences between Upper and Lower bPOE. First, Lower bPOE is defined as follows.

Definition 7. *Let X denote a real valued random variable and $z \in \mathbb{R}$ a fixed threshold parameter. bPOE of random variable X at threshold z equals*

$$\bar{p}_z^L(X) = \begin{cases} 0, & \text{if } z \geq \sup X \ , \\ \{1-\alpha | \bar{q}_\alpha(X) = z\}, & \text{if } E[X] < z < \sup X \ , \\ 1, & \text{otherwise.} \end{cases}$$

Upper bPOE is defined as follows.

Definition 8. *Upper bPOE of random variable X at threshold z equals*

$$\bar{p}_z^U(X) = \begin{cases} \max\{1-\alpha|\bar{q}_\alpha(X) \geq z\}, & \text{if } z \leq \sup X \ , \\ 0, & \text{otherwise.} \end{cases}$$

Upper and Lower bPOE do not differ dramatically. This is shown by the following proposition.

Proposition 8.

$$\bar{p}_z^U(X) = \begin{cases} \bar{p}_z^L(X), & \text{if } z \neq \sup X \ , \\ P(X = \sup X), & \text{if } z = \sup X. \end{cases}$$

Proof. We prove four cases.
Case 1: Assume $z > \sup X$. By Definition 1, $\bar{p}_z^L(X) = 0$. By Definition 2, $\bar{p}_z^U(X) = 0$.
Case 2: Assume $E[X] < z < \sup X$. By Definition 1, $\bar{p}_z^L(X) = \{1-\alpha|\bar{q}_\alpha(X) = z\}$. By Definition 2, $\bar{p}_z^U(X) = \max\{1-\alpha|\bar{q}_\alpha(X) \geq z\}$. Since $\bar{q}_\alpha(X)$ is a strictly increasing function of α on $\alpha \in [0, 1-P(X = \sup X)]$, $\bar{q}_\alpha(X) = z$ has a unique solution. Therefore, we have that $\bar{p}_z^U(X) = \max\{1-\alpha|\bar{q}_\alpha(X) \geq z\} = \{1-\alpha|\bar{q}_\alpha(X) = z\} = \bar{p}_z^L(X)$.
Case 3: Assume $z \leq E[X]$, $z \neq \sup X$. By Definition 1, $\bar{p}_z^L(X) = 1$. Since $\bar{q}_0(X) = E[X]$, $\max\{1-\alpha|\bar{q}_\alpha(X) \geq z\} = 1$ implying that $\bar{p}_z^U(X) = 1$.
Case 4: Assume $z = \sup X$. Following from the fact that $\bar{q}_{(1-P(X=\sup X))}(X) = \sup X$, we have that $\bar{p}_x^U(X) = \max\{1-\alpha|\bar{q}_\alpha(X) \geq z\} = P(X = \sup X)$. □

Thus, one will notice that Upper and Lower bPOE are equivalent when $z \neq \sup X$. The difference between the two definitions arises when threshold $z = \sup X$. In this case, we have that $\bar{p}_z^L(X) = 0$ while $\bar{p}_z^U(X) = P(X = \sup X)$. Thus, for a threshold $z \in (E[X], \sup X)$, both Upper and Lower bPOE of X at z can be interpreted as one minus the probability level at which the superquantile equals z. Roughly speaking, Upper bPOE can be compared with $P(X \geq z)$ while Lower bPOE can be compared with $P(X > z)$.

The importance of using Upper bPOE instead of Lower bPOE in the definition of bAUC should be noted here. To illustrate, consider a trivial classifier with $w = 0$. Clearly this is not a very good classifier. Using Upper bPOE, we find that $1 - \bar{p}_0^U(\xi(w)) = 1 - P(\xi(w) = \sup \xi(w)) = 1 - 1 = 0$. Using this number as our ranking ability performance metric intuitively makes sense, i.e. assigning the trivial classifier the lowest possible bAUC, reflecting its poor ranking ability. What if we use Lower bPOE instead? Using Lower bPOE, we find that $1 - \bar{p}_0^L(\xi(w)) = 1 - 0 = 1$. Using this as our measure of ranking ability does not make much sense. Thus, we find that Upper bPOE treats losses at the supremum in a manner more fitting to our application, i.e. measuring the ranking ability of a classifier.

References

Bradley, Andrew P. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition* **30**(7) 1145–1159.

Brefeld, U., T. Scheffer. 2005. AUC maximizing support vector learning. *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning* .

Caruana, Rich, Shumeet Baluja, Tom Mitchell, et al. 1996. Using the future to" sort out" the present: Rankprop and multitask learning for medical risk evaluation. *Advances in neural information processing systems* 959–965.

Cortes, C., M. Mohri. 2004. AUC optimization vs. error rate minimization. *Advances in Neural Information Processing Systems, 16(16), 313-320* .

Cortes, C., V. Vapnik. 1995. Support-vector networks. *Machine learning* **20**(3) 273–297.

Davis, Justin R, Stan Uryasev. 2015. Analysis of tropical storm damage using buffered probability of exceedance. *Natural Hazards* 1–19.

Egan, James P. 1975. Signal detection theory and {ROC} analysis .

Fawcett, Tom. 2006. An introduction to roc analysis. *Pattern recognition letters* **27**(8) 861–874.

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

Herbrich, Ralf, Thore Graepel, Klaus Obermayer. 1999. Large margin rank boundaries for ordinal regression. *Advances in neural information processing systems* 115–132.

Hernández-Orallo, José, Peter Flach, Cesar Ferri. 2012. A unified view of performance metrics: Translating threshold choice into expected classification loss. *Journal of Machine Learning Research* **13**(Oct) 2813–2869.

Herschtal, Alan, Bhavani Raskutti. 2004. Optimising area under the roc curve using gradient descent. *Proceedings of the twenty-first international conference on Machine learning*. ACM, 49.

Krm, E., K. Yildirak, G.W. Weber. 2012. A classification problem of credit risk rating investigated and solved by optimization of the ROC curve. *CEJOR 20, 3 (2012) 529-557; in the special issue at the occasion of EURO XXIV 2010 in Lisbon* .

Lichman, M. 2013. UCI machine learning repository. URL `http://archive.ics.uci.edu/ml`.

Ling, Charles X, Jin Huang, Harry Zhang. 2003. Auc: a statistically consistent and more discriminating measure than accuracy. *IJCAI*, vol. 3. 519–524.

Mafusalov, A., S. Uryasev. 2015. Buffered probability of exceedance: Mathematical properties and optimization algorithms. *Research Report 2014-1, ISE Dept., University of Florida* .

Miura, K., S. Yamashita, S. Eguchi. 2010. Area under the curve maximization method in credit scoring. *The Journal of Risk Model Validation, 4(2), 3–25* .

Mozer, Michael C. 2003. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic .

Norton, M., A. Mafusalov, S. Uryasev. 2015. Soft margin support vector classification as buffered probability minimization. *Research Report 2015-2, ISE Dept., University of Florida* .

Provost, Foster J, Tom Fawcett, Ron Kohavi. 1998. The case against accuracy estimation for comparing induction algorithms. *ICML*, vol. 98. 445–453.

Provost, Foster J, Tom Fawcett, et al. 1997. Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. *KDD*, vol. 97. 43–48.

Rockafellar, R.T. 2009. Safeguarding strategies in risky optimization. *Presentation at the International Workshop on Engineering Risk Control and Optimization, Gainesville, FL, February, 2009* .

Rockafellar, R.T., J.O. Royset. 2010. On buffered failure probability in design and optimization of structures. *Reliability Engineering & System Safety, Vol. 95, 499-510* .

Rockafellar, R.T., S. Uryasev. 2000. Optimization of conditional value-at-risk. *The Journal of Risk, Vol. 2, No. 3, 2000, 21-41* .

Schapire, William W Cohen Robert E, Yoram Singer. 1998. Learning to order things. *Advances in Neural Information Processing Systems* **10** 451.

Swets, John A. 1988. Measuring the accuracy of diagnostic systems. *Science* **240**(4857) 1285–1293.

Swets, John A, Robyn M Dawes, John Monahan. 2000. Better decisions through. *Scientific American* **283** 82–87.

Vapnik, Vladimir Naumovich, Vlamimir Vapnik. 1998. *Statistical learning theory*, vol. 1. Wiley New York.

Zou, Kelly H. 2002. Receiver operating characteristic (roc) literature research. *On-line bibliography available from:¡ http://splweb. bwh. harvard. edu* **8000**.