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: March 2016

Abstract

In binary classification, performance metrics can often be defined as the probability that some error exceeds a threshold, i.e. by using Probability of Exceedance (POE) and an error function. This leads to performance metrics that are numerically difficult to optimize directly, which also hide potentially important information about the magnitude of errors larger than the threshold. For example, Accuracy can be viewed as one minus the probability that misclassification error exceeds zero. This probability is rarely minimized directly and does not account for the magnitude of misclassification error. Defining these metrics, instead, with Buffered Probability of Exceedance (bPOE) generates a counterpart metric that provides 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 idea to the case of AUC, the Area Under the ROC curve, and derive Buffered AUC (bAUC). We justify its merit as an informative counterpart to AUC that is also a lower bound and directly optimizable via convex and linear programming. We then derive Generalized bAUC, which generates a family of metrics in which AUC and bAUC belong, and show that popular SVM formulations for 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. Using this formula, we show that particular bPOE minimization problems reduce to convex and linear programming. We briefly mention application of bPOE to Accuracy and show that a buffered variant of Accuracy already has deep roots in the SVM literature.

1 Introduction

In binary classification, performance metrics can often be defined as the probability that some error function exceeds a particular threshold, i.e. by defining a metric 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 some error exceeds a threshold of zero. It measures a classification model's ability to differentiate between two randomly selected instances from opposite classes and is a useful metric when one has no knowledge of misclassification costs or must deal with imbalanced classes. 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. Another viewpoint of AUC is as a measure of a classifiers ability to properly 'rank' a given positive instance correctly with respect to a given negative instance. From this viewpoint, AUC is a natural performance metric for binary classification.

Defining metrics by using POE, though, produces metrics with undesirable properties. First, these metrics only consider the number of errors larger than the threshold and does not consider the magnitude of these errors. This information, which may be viewed as the classifier's 'confidence' may be important when gauging classifier performance. Second, these metrics are difficult to optimize directly. When dealing with empirical observations of your 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. Not only are these convex surrogates simpler to optimize, reducing to convex programming, but it is interesting to note that they inadvertently consider the magnitude of errors. In the AUC optimization literature, we see similar approaches, although some are still non-convex.

Instead of defining metrics with POE, we take the approach of defining metrics with Buffered Probability of Exceedance (bPOE). We show that this produces metrics that account for the magnitude of errors exceeding the threshold, with optimization of these metrics 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), Davis and Uryasev (2014), bPOE is the inverse of the superquantile, also called the Conditional Value-at-Risk (CVaR) in the financial engineering literature.

Specifically, contributions of this paper are as follows.

- We present a novel calculation formula for bPOE and show that this formula simultaneously calculates bPOE and POE. Applying this formula, we show that some bPOE minimization problems can be reduced to convex, sometimes linear, programming.
- We define bAUC and show that it is a natural counterpart to AUC. We show that

bAUC measures the ranking ability of a classifier, is a lower bound for AUC, and can be represented as the area under a slightly modified ROC curve.

- We show that bAUC has advantages over AUC. Specifically, we show that bAUC reveals information about the confidence of a classifiers prediction that is hidden by AUC. Additionally, we show that direct maximization of bAUC reduces to convex and linear programming.

- Generalizing the bAUC definition, we find that this generates a family of modified ROC curves and corresponding areas under these curves, with the traditional ROC curve and AUC belonging to this family. We then present a formulation for maximizing Generalized bAUC. We show that popular AUC maximizing SVM's are a special case of maximizing Generalized bAUC.

- To further the theme of the paper, we briefly discuss applying bPOE to 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) directly maximizes the bPOE variant of 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, albeit not explicitly, yielding the highly successful 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. Section 5 generalizes the bAUC definition, presents it as a family of modified ROC curves with corresponding area 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 then 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 and 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 $Y_i = \text{sign}(h(X_i) - t)$.

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.

Originally, AUC was defined as the Area Under the Receiver Operating Characteristic Curve (the ROC curve). Figure 1 shows an example ROC curve, which plots the True Positive Rate, $P(h(X^+) > t)$, 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, AUC for a scoring function h is equal to the area under this curve, i.e.

$$AUC(h) = \int_t P(h(X^+) > t) \; dP(h(X^-) > t) \; .$$

In this paper, we focus 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

$$AUC(h) = P\left(h(X^+) > h(X^-)\right) \tag{1}$$

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,

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

2.2 Undesirable 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 can be an important indicator of classifier performance for particular applications. 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, can be criticized as a shallow measure of ranking quality because it does not consider the *magnitude* of ranking errors, i.e. the *confidence* with which the classifier

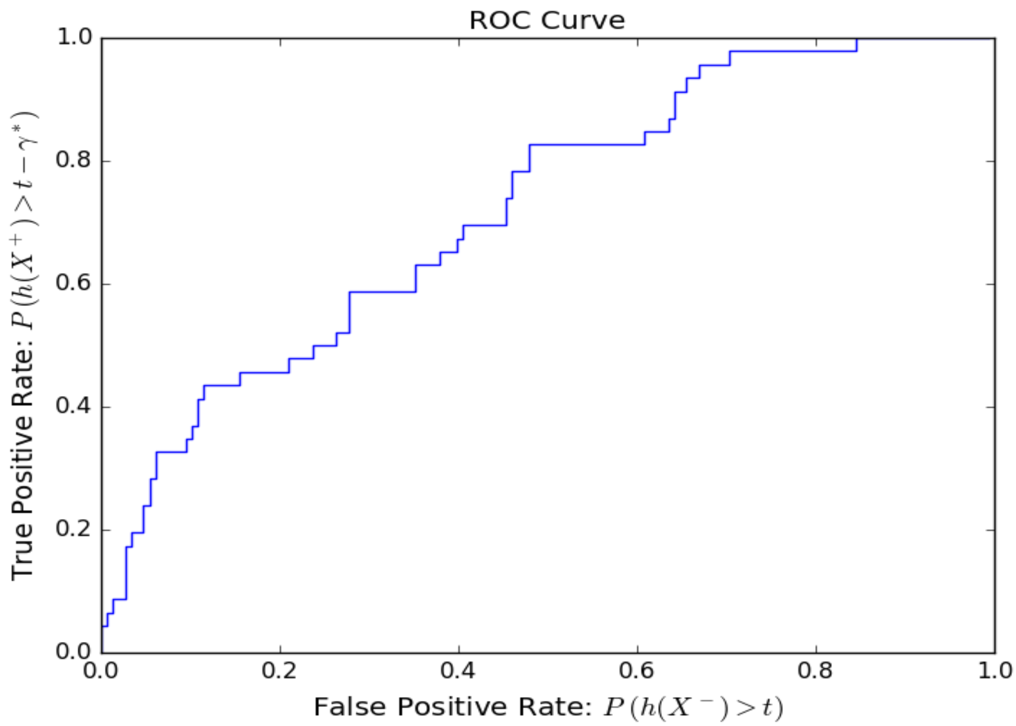

Figure 1: An example of an ROC curve for fixed h. We plot the True Positive Rate, $P(h(X^+) > t)$, on the vertical axis and the False Positive Rate, $P(h(X^-) > t)$, on the horizontal axis for all values of decision threshold $t \in \mathbb{R}$.

correctly or incorrectly ranks pairs of samples. This criticism 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, Yamashita, and Eguchi (2010), Krm, Yildirak, and Weber (2012). 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 combat 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,

$$\bar{q}_\alpha(X) = \min_\gamma \gamma + \frac{E[X - \gamma]^+}{1 - \alpha}, \tag{2}$$

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*

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

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 threshold γ at which $P(X > \gamma) = \bar{p}_z(X)$, providing information about bPOE and POE at the same probability level.

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

$$\bar{p}_z(X) = \underset{\gamma < z}{inf} \frac{E[X-\gamma]^+}{z-\gamma} = \begin{cases} \underset{\gamma \to -\infty}{lim} \frac{E[X-\gamma]^+}{z-\gamma} = 1 \ , & \textit{if } z \leq E[X] \ , \\ \underset{\gamma < z}{min} \frac{E[X-\gamma]^+}{z-\gamma} \ , & \textit{if } E[X] < z < \sup X \ , \\ \underset{\gamma \to z^-}{lim} \frac{E[X-\gamma]^+}{z-\gamma} = P(X = \sup X) \ , & \textit{if } z = \sup X \ , \\ \underset{\gamma < z}{min} \frac{E[X-\gamma]^+}{z-\gamma} = 0 \ , & \textit{if } \sup X < z. \end{cases} \tag{3}$$

Furthermore, if $z \in (E[X], \sup X)$ and $\gamma^* = \underset{\gamma < z}{argmin} \frac{E[X-\gamma]^+}{z-\gamma}$ then $P(X > \gamma^*) = \bar{p}_z(X)$.

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

Case 1: $z \leq E[X]$.
Assume $z \leq E[X]$. First, note that $\bar{p}_z(X) = \max\{1-\alpha | \bar{q}_\alpha(X) \geq z\} = 1$. This follows from

the fact that $\bar{q}_0(X) = E[X]$. Then, notice that

$$\inf_{\gamma<z} \frac{E[X-\gamma]^+}{z-\gamma} = \inf_{0<z-\gamma} E[\frac{X}{z-\gamma} - \frac{\gamma}{z-\gamma}]^+ \ . \tag{4}$$

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

$$\inf_{0<z-\gamma} E[\frac{X}{z-\gamma} - \frac{\gamma}{z-\gamma}]^+ = \inf_{a>0} E[aX + a(\frac{1}{a} - z)]^+ = \inf_{a>0} E[a(X-z)+1]^+ \ . \tag{5}$$

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

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

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

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

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

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

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

$$\bar{p}_z(X) = \max\{1-\alpha | \bar{q}_\alpha(X) \geq z\} = \min\{1-\alpha | \bar{q}_\alpha(X) \leq z\} \ . \tag{6}$$

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

$$\bar{q}_\alpha(X) = \min_\gamma \left[\gamma + \frac{E[X-\gamma]^+}{1-\alpha}\right] = \min_\gamma g(X,\alpha,\gamma) \ . \tag{7}$$

Note also Rockafellar and Uryasev (2000) states that if $\gamma^* = \operatorname*{argmin}_\gamma g(X,\alpha,\gamma)$, then $\bar{q}_\alpha(X) = \gamma^* + \frac{E[X-\gamma^*]^+}{1-\alpha}$ and $\gamma^* = q_\alpha(X)$.
Next, using (6) and (7) we get

$$\bar{p}_z(X) = \min\{1-\alpha : \min_\gamma g(X,\alpha,\gamma) \leq z\} \ . \tag{8}$$

Then, considering (7) we can write (8) as,

$$\begin{aligned} \bar{p}_z(X) = &\min_{\alpha,\gamma} \quad 1-\alpha \\ & s.t. \quad \gamma + \frac{E[X-\gamma]^+}{1-\alpha} \leq z \ . \end{aligned} \tag{9}$$

Let (γ^*, α^*) denote an optimal solution vector to (9). Since $z < \sup X$, the formula (7) 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 (9) without changing the optimal solution or objective value,

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

Simplifying further, this becomes

$$\bar{p}_z(X) = \min_{\gamma<z} \quad \frac{E[X-\gamma]^+}{z-\gamma} \ . \tag{11}$$

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 (4) and (5) 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 (4) and (5) 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 (2). In the following section, we show that the true power of formula (3) lies in the fact that it can be utilized to reduce particular bPOE minimization problems to convex, sometimes even linear, programming. The impact of this formula is further confirmed in Mafusalov and Uryasev (2015). Applying the formula from Proposition 1, Mafusalov and Uryasev (2015) considers more general convex and linear reformulations of bPOE minimization.

3.3 bPOE Optimization

To demonstrate the way in which bPOE alleviates the difficulties associated with POE optimization, 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.

$$\min_{w \in \mathbb{R}^n} \quad p_0(f(w, X)) \ . \tag{12}$$

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

$$\min_{w \in \mathbb{R}^n} \quad \bar{p}_0(f(w, X)) \ . \tag{13}$$

Given Proposition 1, (13) can be transformed into the following.

$$\min_{w \in \mathbb{R}^n, \gamma<0} \quad \frac{E[f(w, X) - \gamma]^+}{-\gamma} \ . \tag{14}$$

Notice, though, that the positive homogeneity of $f(w, X)$ allows us to further simplify (14) 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 (15).

$$\min_{w \in \mathbb{R}^n} \quad E[f(w, X) + 1]^+ \,. \tag{15}$$

For convex f, (15) is a convex program. Furthermore, if f is linear, (15) can be reduced to linear programming. This is substantially easier to handle numerically than the non-convex and discontinuous POE minimization (12).

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 rich measure of classifier performance by considering the magnitude of ranking errors $\xi_{ij}(h)$ instead of only a discrete count of the number of ranking errors.

4 Buffered AUC: A New Performance Metric

4.1 Buffered AUC

With AUC defined as $1 - P\left(\xi(h) \geq 0\right)$, we can create a natural alternative to AUC called *buffered AUC* (bAUC) by using bPOE instead of POE. In words, 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 definition.

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

$$bAUC(h) = 1 - \bar{p}_0\left(\xi(h)\right). \tag{16}$$

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. In the bottom chart, we highlight in red 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 in red. In the top chart, we highlight red the largest errors that have average magnitude equal to zero, i.e. the ranking errors considered by bAUC. Thus, in the top chart, bAUC is one minus the proportion of errors highlighted red.

This metric, utilizing bPOE instead of POE in its derivation, is extremely 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 $h : \mathbb{R}^n \to \mathbb{R}$,*

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

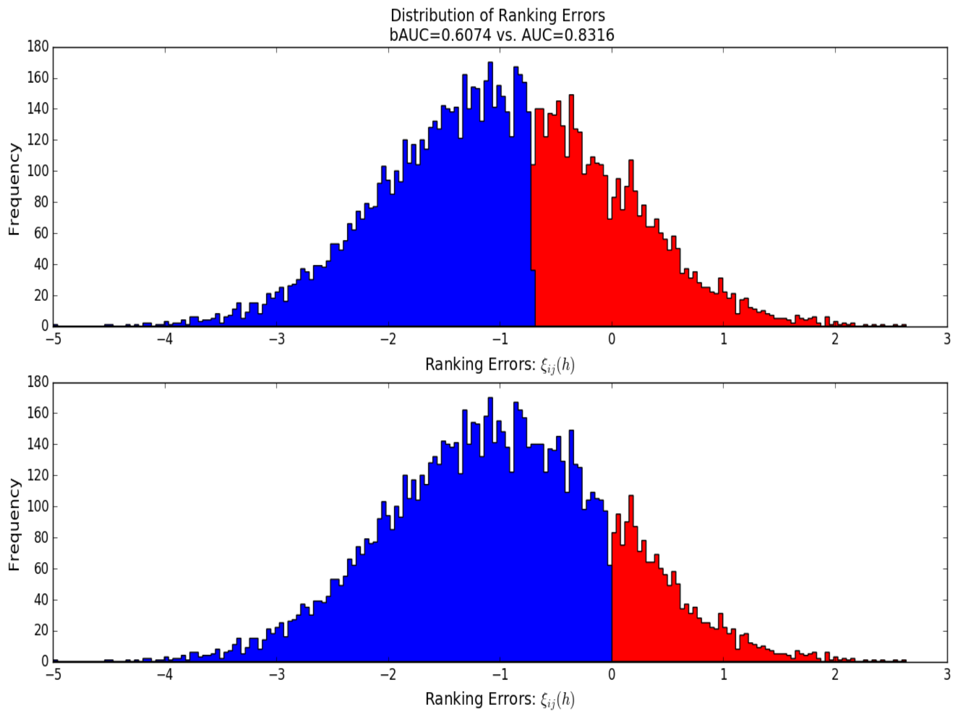

Figure 2: In both charts, we plot the same distribution of ranking errors $\xi_{ij}(h)$ for a fixed h. In the top chart, we highlight the largest errors that have average magnitude equal to zero, i.e. the errors considered by bAUC. In the bottom chart, we highlight the errors that exceed zero, i.e. the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

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, $1 - \bar{p}_0\left(\xi(h)\right) \leq 1 - P((\xi(h) \geq 0)$.

□

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 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 $I_{\lambda>0}$ be an indicator function such that for $\lambda \in \mathbb{R}$,

$$I_{\lambda>0} = \begin{cases} 1, & \text{if } \lambda > 0, \\ 0, & \text{if } \lambda \leq 0. \end{cases}$$

Consider the task of comparing the ranking ability (on the same data set) of two imperfect classifiers[1], h_1 and h_2, that have equal AUC values, meaning that

$$(a) \quad \left(\sum_i \sum_j I_{\xi_{ij}(h_1)>0} \right) = \left(\sum_i \sum_j I_{\xi_{ij}(h_2)>0} \right) > 0$$

Assume also that

$$(b) \quad \left(\sum_i \sum_j \xi_{ij}(h_1) I_{\xi_{ij}(h_1)>0} \right) = \left(\sum_i \sum_j \xi_{ij}(h_2) I_{\xi_{ij}(h_2)>0} \right) \quad ,$$

and that

$$(c) \ \forall \ (i,j) \text{ such that } \xi_{ij}(h_1) < 0 \text{ we have that } \xi_{ij}(h_1) < \xi_{ij}(h_2) \quad .$$

Clearly, h_1 is the better classifier with respect to its ranking ability. We see that h_1 and h_2 have incorrectly ranked the same number of instances, (a), and that they have done so with the same confidence, (b). h_1, though, has correctly ranked instances with much greater confidence, (c). AUC, though, indicates that h_1 and h_2 perform equivalently with respect to ranking ability. bAUC, though, because of the *buffer*, correctly distinguishes between the ranking ability of h_1 and h_2. Specifically, we will find that $bAUC(h_1) > bAUC(h_2)$.

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

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

$$(d) \quad \left(\sum_i \sum_j \xi_{ij}(h_1) I_{\xi_{ij}(h_1)>0}\right) < \left(\sum_i \sum_j \xi_{ij}(h_2) I_{\xi_{ij}(h_2)>0}\right) \quad ,$$

and that

$$(e) \;\; \forall \;\; (i,j) \text{ such that } \xi_{ij}(h_1) < 0 \text{ we have that } \xi_{ij}(h_1) = \xi_{ij}(h_2) \;\; .$$

Clearly, again, h_1 is the better classifier with respect to its ranking ability. We see that h_1 and h_2 have correctly ranked the same number of instances, (a), and that they have done so with the same confidence, (e). h_2, though, has incorrectly ranked instances with much greater confidence, (d), leading to larger magnitude ranking errors. Once again, AUC indicates that these classifiers perform equivalently with respect to ranking ability. bAUC, though, will reveal the discrepancy between the ranking ability of h_1 and h_2. Specifically, because of (d) and (e), we will have that $bAUC(h_1) > bAUC(h_2)$.

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 $h(X)-t = w^T X - t$, $w \in \mathbb{R}^n$. 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,

$$\max_{w \in \mathbb{R}^n} \quad 1 - P\left(\xi(w) \geq 0\right) \; . \tag{17}$$

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

$$\max_{w \in \mathbb{R}^n} 1 - \bar{p}_0\left(\xi(w)\right) \; . \tag{18}$$

Applying Proposition 1, (18) becomes

$$1 - \min_{w \in \mathbb{R}^n, \gamma < 0} \frac{E\left[\xi(w) - \gamma\right]^+}{-\gamma} \; . \tag{19}$$

Finally, given the positive homogeneity of $\xi(w)$, we can apply minimization formula (15) and simplify to (20).

$$\min_{w \in \mathbb{R}^n} \quad E[\xi(w) + 1]^+ \; . \tag{20}$$

Here, (20) is a convex optimization problem and, moreover, can be reduced to linear programming with reduction to (21) 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.

$$\begin{aligned} \min_{w\in\mathbb{R}^n, \beta_{ij}\in\mathbb{R}} \quad & \frac{1}{m^+m^-}\sum_{i=1}^{m^+}\sum_{j=1}^{m^-}\beta_{ij} \\ s.t. \quad & \beta_{ij} \geq \xi_{ij}(w)+1,\ \forall\, i=1,...,m^+, j=1,...,m^- \\ & \beta_{ij} \geq 0\ . \end{aligned} \tag{21}$$

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.

Proposition 3. *For fixed scoring function $h : \mathbb{R}^n \to \mathbb{R}$, assume that $bAUC(h) = 1-\alpha = 1-\frac{E[\xi(h)-\gamma^*]^+}{-\gamma^*}$ where $\gamma^* = \underset{\gamma<0}{argmin}\frac{E[\xi(h)-\gamma]^+}{-\gamma}$. We then have that,*

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

Proof. This follows from Proposition 1, specifically the fact that if $z \in (E[X], \sup X)$ and $\gamma^* = \underset{\gamma<z}{\operatorname{argmin}}\frac{E[X-\gamma]^+}{z-\gamma}$ then $P(X>\gamma^*) = \bar{p}_z(X)$. Applying this to bAUC, we get that

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

where the last equality follows from the result in Section 2 from Hanley and McNeil (1982) that the integral representation of AUC is equivalent to the probabilistic definition. □

Proposition 3 is shown graphically, with respect to a modified ROC plot in Figure 3. Here, instead of plotting the pairs $(P(h(X^-)>t), P(h(X^+)>t))$ for all thresholds $t\in\mathbb{R}$ we plot the pairs $(P(h(X^-)>t), P(h(X^+)>t-\gamma^*))$ for all thresholds $t\in\mathbb{R}$. This yields a curve on the ROC plot that has area underneath it equal to $bAUC(h)$.

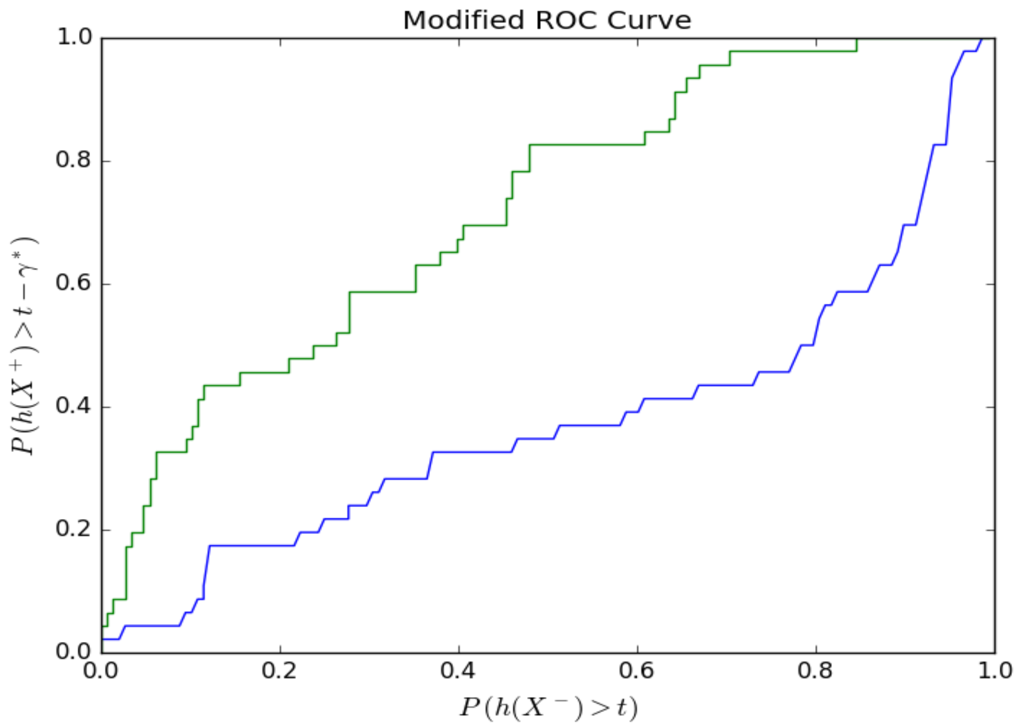

Figure 3: We have a fixed classifier h. The area under the upper curve corresponds to $AUC(h)$, where $\gamma^* = 0$. The area under the lower curve corresponds to $bAUC(h)$, where $\gamma^* < 0$.

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 formula (2)). Consider now a more general definition of bAUC where we consider thresholds $z \in \mathbb{R}$. We can define this as the following.

Definition 3 (Generalized bAUC)**.**

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

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. *Assume that* $bAUC_z(h) = 1-\alpha = 1-\frac{E[\xi(h)-\gamma^*]^+}{z-\gamma^*}$ *where* $\gamma^* = \underset{\gamma<z}{argmin}\frac{E[\xi(h)-\gamma]^+}{z-\gamma}$. *We then have that,*

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

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 4 which displays a family of $bAUC_z$ curves.

5.2 Maximizing Generalized bAUC

Although interesting, it is not immediately clear how to utilize Generalized bAUC. Here, we show, surprisingly, 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 a linear classifier $h(X) = w^T X$ 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.

$$\min_{w\in\mathbb{R}^n,\gamma<z} \frac{E[\xi(w)-\gamma]^+}{z-\gamma} \equiv \min_{w\in\mathbb{R}^n} \bar{p}_z\left(\xi(w)\right) . \tag{22}$$

Unfortunately, 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.

$$\begin{aligned} \min_{w\in\mathbb{R}^n,\gamma<z} \quad & \frac{E[\xi(w)-\gamma]^+}{z-\gamma} \equiv \min_{w\in\mathbb{R}^n} \bar{p}_z\left(\frac{\xi(w)}{\|w\|}\right) \\ s.t. \quad & \|w\| = 1 \end{aligned} \tag{23}$$

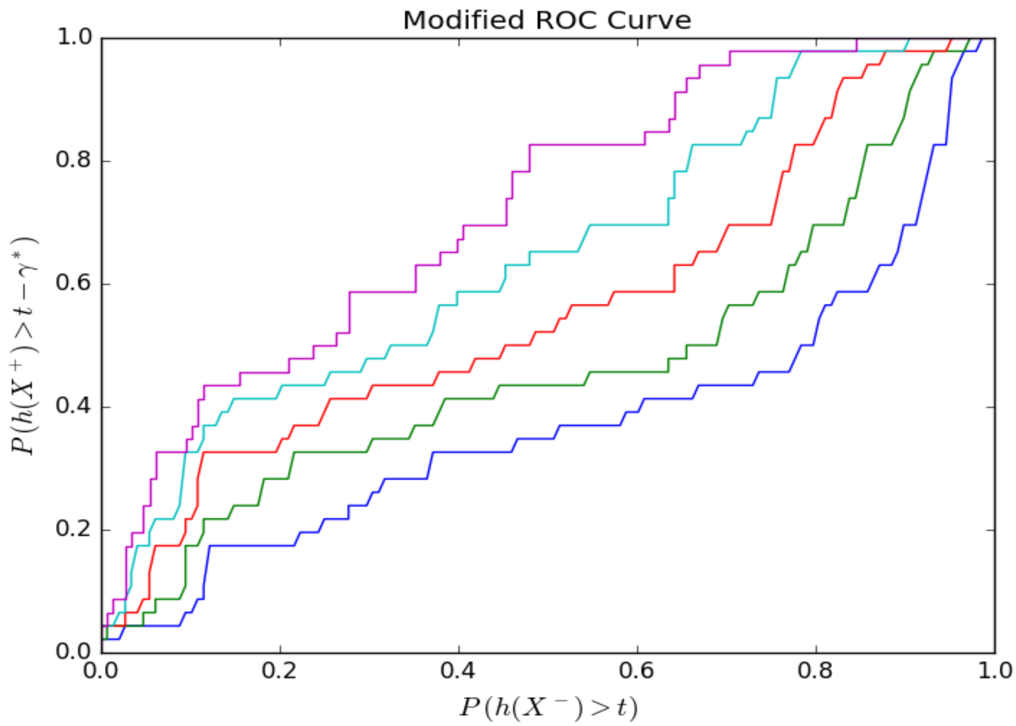

Figure 4: A modified ROC plot for a fixed classifier h. The lower most curve corresponds to $bAUC_0(h)$ while the uppermost curve corresponds to $bAUC_{z_0}(h) = AUC(h)$. The curves in-between correspond to $bAUC_z(h)$ for values of $z \in (0, z_0)$.

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 $z \leq 0$.

$$\min_{w \in \mathbb{R}^n} \bar{p}_z \left(\frac{\xi(w)}{\|w\|} \right) \quad \equiv \quad \min_{w \in \mathbb{R}^n} E[\xi(w) - z\|w\| + 1]^+ \quad . \tag{24}$$

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.

The RankSVM is formulated as follows, where $z \leq 0$ is a fixed tradeoff parameter.

Traditionally, the squared L_2 norm is used, but we use any general norm.

$$\begin{aligned} \min_{w\in\mathbb{R}^n,\beta_{ij}\in\mathbb{R}} \quad & -z\|w\| + \sum_{i=1}^{m^+}\sum_{j=1}^{m^-}\beta_{ij} \\ s.t. \quad & \beta_{ij} \geq \xi_{ij}(w) + 1,\ \forall\, i = 1,...,m^+, j = 1,...,m^- \\ & \beta \geq 0 \end{aligned} \tag{25}$$

Furthermore, in Norton et al. (2015), a variant of the following proposition was proven.

Proposition 5. *Consider formulation (26) and (27), where $N = m^+ + m^-$ and $z \leq 0$. Here, (26) is the traditional C-SVM soft margin support vector machine and (27) is minimization of bPOE for error function $-Y(w^TX + b)$. Over the parameter range $z \leq 0$, (26) and (27) achieve the same set of optimal hyperplanes, up to some scaling factor.*

$$\begin{aligned} \min_{w\in\mathbb{R}^n,b\in\mathbb{R},\beta_i\in\mathbb{R}} \quad & -z\|w\| + \sum_{i=1}^{N}\beta_i \\ s.t. \quad & \beta_i \geq -Y_i(w^TX_i + b) + 1, \quad \forall i \in \{1,...,N\}, \\ & \beta \geq 0. \end{aligned} \tag{26}$$

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

Using Proposition 5, we can prove that RankSVM is simply maximizing Generalized bAUC.

Proposition 6. *Assume (25) and (24) are formulated with the same general norm $\|\cdot\|$. The set of optimal hyperplanes achieved by solving (25) for all $z \leq 0$ is the same, up to some scaling factor, as the set of optimal hypeerplanes achieved by solving (24) for all $z \leq 0$.*

Proof. Note that (25) is exactly formulation (26) with samples $(X_i^+ - X_j^-)$ all having class $Y_{ij} = +1$ and with the classifier intercept $b = 0$. Thus, applying Proposition 3, we have that (25) provides the same set of optimal hyperplanes, up to some scaling factor, as (24) over the parameter range $z \leq 0$. $\square$

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.*

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

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

Definition 4.

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

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

Definition 5.

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

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

Definition 6.

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

In this paper, we do not fully explore the properties and benefits of bAcc 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 bAcc directly. This is exactly what is shown in Proposition 5 which can be seen more clearly by noting that optimization problem (27) is equivalent to the following.

$$\begin{aligned} \max_{w\in\mathbb{R}^n, b\in\mathbb{R}} \quad & \frac{1}{N} bAcc_z(w,b) \\ s.t. \quad & \|w\| = 1 \end{aligned} \tag{28}$$

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 *buffering* metrics that are defined with POE.

7 Conclusion

AUC is a useful and popular metric for measuring the ranking quality of scores given by a classifier. AUC, though, suffers from limitations. As a metric, it can be criticized as being a shallow measure similar to Value-at-Risk that is also numerically difficult to optimize. In order to alleviate some of these issues, we have utilized the new concept of bPOE to define a new AUC-like metric called bAUC. As a metric, bAUC is an informative counterpart to AUC. By developing a new calculation formula for bPOE, we have also found that maximizing bAUC reduces to convex programming, and even linear programming in the case of a linear decision function. We also find that bAUC has already found its way into the AUC maximization literature, with the RankSVM 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 conclusion, we find bAUC to be a promising metric alternative and counterpart to AUC. Therefore, we find bAUC to be a promising counterpart to AUC.

Additionally, we present our methodology as a general approach to generating insightful, easy to optimize metrics that are counterparts to metrics defined with POE. Although we focus our in-depth analysis on creating a buffered variant of AUC, we show that the buffered alternative to Accuracy already has deep roots in the SVM literature. This lends credibility to the idea of utilizing bPOE to create new metrics that are counterparts to metrics defined by POE.

Acknowledgements

Authors would like to thank Prof. R.T. Rockafellar and Mr. Sasha 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*

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

Upper bPOE is defined as follows.

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

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

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

Proposition 7.

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

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

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

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

Davis, J.R., S. Uryasev. 2014. Analysis of hurricane damage using buffered probability of exceedance. *Research Report 2014-4, ISE Dept., University of Florida* .

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.

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* .

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* .

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* .

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* .