Maximization of AUC and Buffered AUC in Classification

Matthew Norton, Stan Uryasev

January 2015

RESEARCH REPORT 2014-2
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*

Abstract

This paper utilizes a new concept, called Buffered Probability of Exceedance (bPOE), to introduce an alternative to the Area Under the Receiver Operating Characteristic Curve (AUC) performance metric called *Buffered AUC* (bAUC). Central to the creation of bAUC is a new technique for calculation and optimization of bPOE. We show this formula to be easily integrable into optimization frameworks, often reducing bPOE minimization to convex, sometimes even linear, programming. Then, we utilize bPOE to create the bAUC performance metric, showing it to be an intuitive counterpart to AUC. In addition, we show that bAUC is much easier to handle in optimization frameworks than AUC, specifically reducing to convex and linear programming. We use these friendly optimization properties to introduce the bAUC Efficiency Frontier, a concept that serves to partially resolve the "incoherency" that arises when misclassification costs need be considered. We conclude that bAUC avoids many of the numerically troublesome issues encountered by AUC and integrates much more smoothly into the general framework of model selection and evaluation.

1 Introduction

The Area Under the Receiver Operating Characteristic Curve (AUC) is a popular performance metric in classification. It measures a classification models ability to differentiate between two randomly selected instances from opposite classes. To put another way, AUC has become a standard measure of model performance when the modeling task is classification and the performance characteristic of importance is the classifiers ability to properly 'rank' a given positive instance correctly with respect to a given negative instance.

Given the popularity of AUC as a model selection metric, it is natural to consider choosing a classifier via direct maximization of the AUC metric via optimization with AUC as the objective function. This, though, is rarely done. Direct AUC maximization is akin to probability optimization , which for discrete distributions is discontinuous and non-convex. Thus, AUC maximization raises substantial numerical difficulties when integrated into optimization frameworks. In Section 2.3, we briefly discuss existing literature which encounter said difficulties.

In addition to these difficulties, AUC as a standalone metric has been criticized as 'incoherent' in Hand 2009, [6]. With AUC considering pairwise 'rankings' of positive and negative instances, it does not consider the decision threshold with which one determines the class of a single instance. This introduces ambiguity when consideration of misclassification costs is of some importance. Assume, theoretically, that one is able to find an AUC maximizing classifier. Then, as a second stage, one would need to determine the optimal decision threshold. Intuitively, one would think it a good policy to minimize misclassification costs via choice in threshold. This, though, has been shown to be a potential trap. Misclassification costs may be far suboptimal because of the first stage AUC maximization, which did not consider threshold, and thus did not consider misclassification costs. In other words, AUC does not imply anything about potential misclassification costs.

Recently, Rockafellar in [12] introduced the concept of *buffered Probability of Failure* (bPOF), studying it further with Royset in [13]. Expanding on this concept, a generalization called *buffered Probability of Exceedance* (bPOE) was studied by Mafusalov and Uryasev in [9]. In addition, Davis and Uryasev in [5] explore its application to hurricane damage assessment. These concepts have shown a great deal of promise as numerically tractable methods for probability optimization. In this paper, we contribute a new technique for calculation and optimization of bPOE, significantly adding to the promise of numerical tractability. Specifically, we show this formula to be easily integrable into optimization frameworks, often reducing bPOE minimization to convex, sometimes even linear, programming. As it pertains to AUC, with AUC interpretable as a probability, bPOE proves highly applicable to the ideas characterized by AUC.

This paper shows bPOE applicability by creating a new AUC-like metric called *buffered AUC* (bAUC). We show that bAUC, as a metric, is a natural counterpart of AUC. In addition, we argue that bAUC can be viewed as a richer measure of classifier ranking ability. Furthermore, we show that direct maximization of bAUC avoids the numerical difficulties associated with AUC maximization. Specifically, we show that maximizing bAUC for discrete distributions can be reduced to convex, sometimes even linear, programming. With bAUC lending itself to efficient optimization, we introduce the bAUC Efficiency Frontier as a framework for avoiding the issues of "incoherency." This framework allows disambiguation of the tradeoff between bAUC/AUC maximization and misclassification cost.

The remainder of this paper is organized in the following manner. In Section 2, we review the AUC performance metric. We first review the definition of AUC and its inter-

pretation as a performance metric in classification. We then discuss issues associated with AUC, including issues related to its optimization and relationship with misclassification costs.

Section 3 begins with a review of *buffered probability of exceedance* (bPOE), a generalization of *buffered probability of failure* (bPOF). In Section 3.1, we expand upon the definition of bPOE by introducing a calculation formula for bPOE. In Section 3.2, we show that under particular circumstances, minimization of bPOE can be reduced to convex programming.

In Section 4, we introduce an alternative to AUC called *buffered AUC* (bAUC). In Section 4.1, we use the bPOE concept to define bAUC and discuss its value as a natural counterpart to AUC as a classifier performance metric. In Section 4.2, we discuss bAUC optimization, specifically showing that direct maximization of bAUC reduces to convex and linear programming. Section 4.3 builds on this result by introducing the bAUC efficiency frontier. Then, after introducing "incoherency" as defined in [6], we show that the bAUC efficiency frontier can be used to effectively avoid the issue of "incoherency."

2 AUC as a Performance Metric

2.1 Probabilistic Definition of AUC

AUC is a popular performance metric for model comparison that measures a classification models ability to differentiate between two randomly selected instances from opposite classes. As opposed to a metric such as error rate, AUC does not give direct measure of a classifiers ability to properly classify a single randomly chosen sample, but instead is concerned with a classifiers ability to properly rank two samples that are presumed to be in different classes.

Let X be a random vector taking values from $\mathbb{R}^n$ and Y be a random variable taking values from $\{-1, +1\}$. Let us consider that for pair (X, Y) we have a sample of outcomes $(x_1, y_1), \ldots, (x_m, y_m)$. For probabilisitic evaluations, we can consider that pair (X, Y) take values $(x_1, y_1), \ldots, (x_m, y_m)$ with equal probabilities $\frac{1}{m}$.

Denote the number of $+1$ outcomes by $m^+ = |\{y_i | y_i = +1, i = 1, \ldots, m\}|$ and the number of -1 outcomes by $m^- = |\{y_i | y_i = -1, i = 1, \ldots, m\}|$. Let random vector X^+ take values $\{x_i | y_i = +1, i = 1, \ldots, m\}$ with equal probabilities $\frac{1}{m^+}$. Let random vector X^- take values $\{x_i | y_i = -1, i = 1, \ldots, m\}$ with equal probabilities $\frac{1}{m^-}$. Also, assume X^+, X^- are independent and let $x_i^+ \in S^+$, $x_j^- \in S^-$ denote their realizations (i.e. S^+, S^- are samples of outcomes of X^+, X^-). We assume that the model is a linear classifier $w^T X + b$ with vector parameter $w \in \mathbb{R}^n$ and intercept denoted by $b \in \mathbb{R}$.

AUC was originally defined using the Receiver Operating Characteristic Curve (the ROC curve). In this paper, we use a probabilistic definition of AUC provided by Hanley and McNiel in [7]. Hanley and McNiel showed that the Area Under the ROC curve is

equivalent to the Wilcoxon Statistic. This allows AUC to be defined probabilistically as

$$P\left(w^T x_i^+ - w^T x_j^- > 0\right) = \frac{1}{m^+ m^-} \sum_{x_i^+ \in S^+} \sum_{x_j^- \in S^-} I_{ij} \ , \tag{1}$$

where I_{ij} is the indicator function

$$I_{ij} = \begin{cases} 1, & \text{if } w^T x_i^+ > w^T x_j^-; \\ 0, & otherwise. \end{cases}$$

Here the right hand side of (1) is the Wilcoxon Statistic.

For the remainder of this paper, in order to avoid complications when we introduce and incorporate bPOE, we make slight modification of the strictness of inequalities and *define* AUC as

$$AUC(w) = P\left((w^T X^+ + b) - (w^T X^- + b) \geq 0\right) = 1 - P\left(-w^T (X^+ - X^-) > 0\right).$$

To simplify notation, letting $L(w) = -w^T(X^+ - X^-)$ denote a random loss variable and $L(w)_{ij} = -w^T(x_i^+ - x_j^-)$ denote its realizations, we write AUC as

$$AUC(w) = 1 - P(L(w) > 0).$$

2.2 Interpreting AUC

As a performance metric, $AUC(w)$ provides insight into the ranking quality of a classifier. With each sample data point receiving a *score*, $w^T X + b$, the ordering of these scores can be an important indicator of classifier performance for particular applications. $AUC(w)$ measures a classifiers ranking quality by considering pairwise differences of scores given to samples from opposing classes. Specifically, it considers instances of the random difference $(w^T X^+ + b) - (w^T X^- + b)$, where a pair of samples $x^+ \in S^+$, $x^- \in S^-$ are properly ranked by classifier w if $(w^T x^+ + b) > (w^T x^- + b)$.

$AUC(w)$, though, can be criticized as a shallow measure of ranking quality because it does not consider the magnitude of this pairwise difference. It only considers a count of the number of sample pairs who have incorrectly ordered scores. In other words, it only considers the number of instances giving $(w^T X^+ + b) - (w^T X^- + b) \geq 0$ and does not consider the magnitude of the difference $(w^T X^+ + b) - (w^T X^- + b)$. Being derived from tail loss probability, i.e. $1 - P(L(w) > 0) = P(w^T(X^+ - X^-) > 0)$, $AUC(w)$ can be criticized in the same way that Value-at-Risk, VaR, has been criticized in Financial Engineering. It does not provide an assessment of tail behavior because it fails to consider the magnitude of instances in the tail of the distribution of losses, $L(w) = -w^T(X^+ - X^-)$.

2.3 Maximizing AUC: Numerical Complications

It is natural to consider the task of finding an $AUC(w)$ maximizing classifier via direct optimization of $AUC(w)$. This, though, is rarely done. Optimization of $AUC(w)$ with discretely distributed data is discontinuous and non-convex, giving rise to substantial numerical difficulties. Many AUC optimization approaches exist, see e.g. [2], [10], [8]. Most of these approaches, though, utilize approximations of the $AUC(w)$ objective and do not optimize AUC directly. For example, [10] optimizes an AUC approximation by replacing the indicator function of (1) with a continuous sigmoid function. This yields a continuous optimization problem, though still non-convex.

2.4 AUC and Incoherence

The "coherency" of AUC as a comparison metric for model selection has been criticized by Hand [6]. Hand pointed out that model comparison via AUC does not correspond analogously to comparison metrics that consider misclassification costs. In other words, given two classifiers w_1 and w_2 with $AUC(w_1) > AUC(w_2)$, we cannot say that this implies anything about the misclassification costs of the classifiers. When one finds an AUC maximizing classifier and secondarily chooses decision threshold to minimize expected misclassification cost, it may turn out that the AUC maximizing classifier found via maximization is suboptimal w.r.t. misclassification costs. This gives rise to issues when deciding how to utilize AUC in a model selection process when misclassification costs are known, either exactly or approximately.

3 bPOE and bPOE Optimization

Paper [9] defines and studies bPOE as an alternative to the regular *probability of exceedance* (POE). As discussed by Rockafellar and Royset in the context of structural reliability in [13], chance constraints and probabilistic objectives prove to be difficult to optimize, mirroring the difficulties associated with optimization problems involving $AUC(w)$. bPOE provides an alternative measure of tail probability, alleviating many optimization difficulties. In this section, we first review the definition of bPOE and its connection to superquantiles. We then introduce a calculation formula for bPOE and show that bPOE has optimization friendly properties. This serves as a lead-in to Section 4, where we use this concept to define Buffered AUC (bAUC) and show that maximization of bAUC can be reduced to convex and linear programming.

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), $p_x(X) = P(X > x)$, or its associ-

ated quantile $q_\alpha(X) = \min\{x|P(X \leq x) \geq \alpha\}$, 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 by Rockafellar and Uryasev [11] 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,

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

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. Though there exists two slightly different variants of bPOE[1], we use the following as the formal definition of bPOE, where $\sup X$ denotes the essential supremum of random variable X.

Definition 1 [9]: Let X denote a real valued random variable and $x \in \mathbb{R}$ a fixed threshold parameter. bPOE of random variable X at threshold x equals

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

Thus, bPOE of X at x can be interpreted as one minus the probability level at which the superquantile equals x.

[1]Notice that $\bar{q}_\alpha(X) = \sup X$ for $\alpha \in [1 - P(X = \sup X), 1]$. Thus, it is possible to define bPOE in two ways. One could define $\bar{p}_x(X) = 0$ at threshold $x = \sup X$. This paper, though, defines $\bar{p}_x(X) = P(X = \sup X)$ at threshold $x = \sup X$. See [9] for details.

Although bPOE seems troublesome to calculate, in Proposition 1 we introduce a readily calculable formula for bPOE.

Proposition 1: Given a real valued random variable X and a fixed threshold x, bPOE for random variable X at x equals

$$\bar{p}_x(X) = \inf_{\gamma<x} \frac{E[X-\gamma]^+}{x-\gamma} = \begin{cases} \lim_{\gamma\to-\infty} \frac{E[X-\gamma]^+}{x-\gamma} = 1 \,, & \text{if } x \le E[X] \,, \\ \min_{\gamma<x} \frac{E[X-\gamma]^+}{x-\gamma} \,, & \text{if } E[X] < x < \sup X \,, \\ \lim_{\gamma\to x^-} \frac{E[X-\gamma]^+}{x-\gamma} = P(X = \sup X) \,, & \text{if } x = \sup X \,, \\ \min_{\gamma<x} \frac{E[X-\gamma]^+}{x-\gamma} = 0 \,, & \text{if } \sup X < x. \end{cases} \tag{2}$$

Proof: We prove four cases.

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

$$\inf_{\gamma<x} \frac{E[X-\gamma]^+}{x-\gamma} = \inf_{0<x-\gamma} E[\frac{X}{x-\gamma} - \frac{\gamma}{x-\gamma}]^+ \,. \tag{3}$$

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

$$\inf_{0<x-\gamma} E[\frac{X}{x-\gamma} - \frac{\gamma}{x-\gamma}]^+ = \inf_{a>0} E[aX + a(\frac{1}{a} - x)]^+ = \inf_{a>0} E[a(X-x)+1]^+ \,. \tag{4}$$

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

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

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

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

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

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

Case 2: $E[X] < x < \sup X$.
Assume that $E[X] < x < \sup X$. This assumption and Definition 1 imply that

$$\bar{p}_x(X) = \max\{1-\alpha|\bar{q}_\alpha(X) \geq x\} = \min\{1-\alpha|\bar{q}_\alpha(X) \leq x\} \ . \tag{5}$$

Recall the formula for the superquantile given by Rockafellar and Uryasev [11],

$$\bar{q}_\alpha(X) = \min_\gamma \ \left[\gamma + \frac{E[X-\gamma]^+}{1-\alpha}\right] = \min_\gamma \ g(X,\alpha,\gamma) \ . \tag{6}$$

Note also [11] 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 (5) and (6) we get

$$\bar{p}_x(X) = \min\{1-\alpha : \min_\gamma \ g(X,\alpha,\gamma) \leq x\} \ . \tag{7}$$

Then, considering (6) we can write (7) as,

$$\begin{aligned} \bar{p}_x(X) = & \min_{\alpha,\gamma} \quad 1-\alpha \\ & s.t. \quad \gamma + \frac{E[X-\gamma]^+}{1-\alpha} \leq x \ . \end{aligned} \tag{8}$$

Let (γ^*, α^*) denote an optimal solution vector to (8). Since $x < \sup X$, the formula (6) implies that

$$\gamma^* = q_{\alpha^*}(f(w^*,X)) < \bar{q}_{\alpha^*}(f(w^*,X)) = x \ .$$

This implies that $\gamma^* < x$. Explicitly enforcing the constraint $\gamma < x$ allows us to rearrange (8) without changing the optimal solution or objective value,

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

Simplifying further, this becomes

$$\bar{p}_x(X) = \min_{\gamma<x} \quad \frac{E[X-\gamma]^+}{x-\gamma} \ . \tag{10}$$

Case 3: $x = \sup X$.
Assume $x = \sup X$. First, note that $\bar{p}_x(X) = \max\{1-\alpha|\bar{q}_\alpha(X) \geq x\} = P(X = \sup X)$. This follows from the fact that $\bar{q}_{(1-P(X=\sup X))}(X) = \sup X$. Next, recall that with (3) and (4) for $a = \frac{1}{x-\gamma}$, we get

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

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

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

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

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

Case 4: $x > \sup X$.
Assume that $x > \sup X$. First, note that $\bar{p}_x(X) = 0$. This follows immediately from Definition 1 (i.e. the 'otherwise' case). Next, recall again that with (3) and (4) for $a = \frac{1}{x-\gamma}$, we get

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

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

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

□

Thus, via Proposition 1 we have shown that bPOE can be efficiently calculated. In the following section, we show that the true power of formula (2) lies in the fact that it can be utilized to reduce bPOE minimization problems to convex, sometimes even linear, programming. This is explored further in [9]. In [9], Mafusalov and Uryasev study the general properties of bPOE, showing, among other things, that this formula has generalized extensions.

3.2 bPOE Optimization

In the same way that the superquantile reflects tail event magnitudes and the quantile does not, bPOE assesses the tail of a distribution in a more informative way than POE. Also, much like the superquantile/quantile relationship within optimization, bPOE avoids many of the numerical difficulties associated with POE when used as either an objective or constraint in optimization problems.

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" w.r.t. w if it satisfies the following condition: $af(w, X) = f(aw, X)$ for any $a \geq 0, a \in \mathbb{R}$.

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 $x = 0$. We would like to solve the following POE optimization problem.

$$\min_{w \in \mathbb{R}^n} \quad p_0(f(w, X)) \, . \tag{11}$$

Here we have a discontinuous and non-convex objective function that is numerically difficult to minimize. Consider alternatively minimization of bPOE instead of POE at the same threshold $x = 0$. This is posed as the optimization problem

$$\min_{w \in \mathbb{R}^n} \quad \bar{p}_0(f(w, X)) \, . \tag{12}$$

Given Proposition 1, (12) can be transformed into the following.

$$\min_{w \in \mathbb{R}^n, \gamma < 0} \quad \frac{E[f(w, X) - \gamma]^+}{-\gamma} \, . \tag{13}$$

Notice, though, that the positive homogeneity of $f(w, X)$ allows us to further simplify (13) by getting rid of the γ variable. Thus, we find that $bPOE$ minimization of $f(w, X)$ at threshold $x = 0$ can be reduced to (14).

$$\min_{w \in \mathbb{R}^n} \quad E[f(w, X) + 1]^+ \, . \tag{14}$$

Thus, we see that instead of non-convex and discontinuous POE minimization, we can minimize bPOE for discrete and continuous distributions with convex programming if $f(w, X)$ is a convex function. Furthermore, if $f(w, X)$ is linear, then (14) can be reduced to *linear programming*. This is substantially easier to handle numerically.

Given the attractiveness of bPOE and the superquantile within the optimization context, we are inclined to apply these concepts and definitions to AUC. Since AUC is simply a measure of tail probability (i.e. a measure of POE), it may be beneficial to redefine AUC with bPOE. Not only would this buffered alternative give way to more well behaved optimization problems, but it would provide a richer measure of classifier performance by considering the magnitude of "ranking" error instead of only a discrete count of the number of ranking errors.

4 Buffered AUC: A New Performance Metric

4.1 Defining Buffered AUC

With AUC defined using POE, specifically as $1 - P(L(w) > 0) = 1 - p_0(L(w))$, we can create a natural alternative to AUC called *buffered AUC* (bAUC). Using bPOE instead of POE as our inspiration, we come to the following definition.

Definition 2 (bAUC) For a fixed classifier $w \in \mathbb{R}^n$, the *bAUC* of w is defined as

$$bAUC(w) = 1 - \bar{p}_0(L(w)). \tag{15}$$

This metric, utilizing bPOE instead of POE in its derivation, is extremely similar to AUC. This metric, though, can be somewhat more informative than AUC with respect to the ranking quality of the scores produced by the classifier. This is due to the fact that it considers the magnitude of ranking errors as opposed to only a discrete count of ranking errors.

One can view bAUC as a stronger version of AUC that considers the confidence with which a classifier ranks instances, i.e., the magnitude of instances of $L(w)$. bAUC penalizes a classifier for confidently ranking instances incorrectly, while rewarding a classifier for confidently ranking instances correctly. It is important to note that bAUC becomes equivalent to AUC when considering classifiers that rank perfectly. When a classifier ranks perfectly, $\bar{q}_\alpha(L(w)) = 0$ at $\alpha = 1$ meaning that $bAUC(w) = 1 - P(L(w) > 0)$.

Therefore, bAUC is a justifiable metric to use in place of AUC. When faced with classifiers that are imperfect rankers, bAUC provides an informative measure of the classifier's ranking quality. When faced with classifier that are perfect rankers, bAUC is effectively identical to AUC.

4.2 Optimizing bAUC

As already discussed, direct maximization of AUC is rarely done due to the troublesome properties of probabilistic objectives. Direct maximization of bAUC, on the other hand, is quite simple to do. In fact, it can be reduced to convex and linear programming in the case of linear classification.

Maximization of AUC takes the form,

$$\max_w \quad 1 - p_0(L(w)) \ . \tag{16}$$

The probabilistic objective, being discontinuous and non-convex for discrete distributions, is difficult to handle. Maximization of bAUC is far simpler to handle, reducing to convex and linear programming.

The problem of maximizing bAUC can be posed as,

$$\max_{w \in \mathbb{R}^n} 1 - \bar{p}_0(L(w)) \tag{17}$$

In light of Proposition 1, (17) can be posed as,

$$1 - \min_{w \in \mathbb{R}^n, \gamma < 0} \frac{E[L(w) - \gamma]^+}{-\gamma} \tag{18}$$

Finally, given the positive homogeneity of $L(w)$, we can apply minimization formula (14). Thus, we see that maximizing bAUC simplifies to (19).

$$\min_{w \in \mathbb{R}^n} \quad E[L(w) + 1]^+ \,. \tag{19}$$

Here, since $L(w)$ is linear, (19) is a convex optimization problem and, moreover, can be reduced to linear programming. Thus, one can solve the linear reduction of (19) to obtain the classifier w that maximizes bAUC as opposed to solving (16), a numerically troublesome optimization problem. Specifically, the linear reduction of (19) is the following optimization problem.

$$\begin{aligned} \min_{w \in \mathbb{R}^n, z_{ij} \in \mathbb{R}} \quad & \sum_{x_i^+ \in S^+} \sum_{x_j^- \in S^-} z_{ij} \\ s.t. \quad & z_{ij} \geq L(w)_{ij} + 1 \qquad \forall\, x_i^+ \in S^+, x_j^- \in S^- \\ & z_{ij} \geq 0 \end{aligned} \tag{20}$$

4.3 The bAUC Efficiency Frontier

As discussed in Section 2.4, AUC becomes "incoherent" as a comparison metric when one begins to consider classification costs. When one finds an AUC maximizing classifier and chooses threshold to minimize expected misclassification cost, it may turn out that the AUC maximizing classifier is suboptimal w.r.t. misclassification costs. These arguments also apply to bAUC.

The ease with which bAUC can be handled in optimization frameworks allows us to tackle this issue. We propose a framework for model selection called the bAUC Efficiency Frontier to effectively alleviate the "incoherency" of bAUC. Specifically, we aim to eliminate the ambiguous nature of tradeoff between misclassification cost and ranking ability.

4.3.1 The AUC Efficiency Frontier

It is common for misclassification costs to be of some importance for a classification task. One might like to find a classifier with large AUC/bAUC, but would also like to minimize misclassification costs. Due to the "incoherency" of AUC/bAUC, though, when an

AUC/bAUC maximizing classifier is produced, it is unknown if (or by how much) misclassification costs are suboptimal. With this in mind, we propose that it is beneficial to consider classifiers generated by optimization problems (23) and (24), which are buffered versions of (21) and (22).

Let $f_c(w) :=$(misclassification cost induced by classifier w)[2] where $C_0 \in \mathbb{R}$ is a constant upper bound on misclassification cost and α_0 is a lower bound on $AUC(w)$.

$$\max_w \quad 1 - P(L(w) > 0) \quad \text{s.t.} \quad f_c(w) \leq C_0 \tag{21}$$

$$\min_w \quad f_c(w) \quad \text{s.t.} \quad 1 - P(L(w) > 0) \geq \alpha_0 \tag{22}$$

Here, (21) is maximizing $AUC(w)$ with an explicit cost constraint while (22) is minimizing cost with an explicit lower bound on $AUC(w)$. These problems are difficult to solve because of the use of $AUC(w)$. By using $bAUC$ instead of AUC, though, we can reformulate these into optimization problems (23) and (24).

$$\min_w \quad E[L(w) + 1]^+ \quad \text{s.t.} \quad f_c(w) \leq C_0 \tag{23}$$

$$\min_w \quad f_c(w) \quad \text{s.t.} \quad 1 - E[L(w) + 1]^+ \geq \alpha_0 \tag{24}$$

For (23), we have already shown that its objective function is convex. Therefore, we can see that (23) is a convex program iff $f_c(w) \leq C_0$ is a convex constraint. A slightly more complicated variant of the constraint in (24) was considered by Rockafellar and Royset [13]. Notice that for any w, we have that $\bar{q}_{\alpha_0}(L(w)) \leq 0 \implies 1 - \bar{p}_0(L(w)) \geq \alpha_0$. This allows us to write the constraint on bAUC as a superquantile constraint, which can be turned into a set of linear constraints via auxiliary variables. We can see, similar to (23), that (24) is convex iff $f_c(w)$ is a convex function of w.

It is also important to notice that $\bar{q}_{\alpha_0}(L(w)) \leq 0 \implies 1 - \bar{p}_0(L(w)) \geq \alpha_0 \implies 1 - P(L(w) > 0) \geq \alpha_0 \implies AUC(w) \geq \alpha_0$. This means that any feasible solution to (24), i.e. a classifier satisfying the bAUC constraint, is also a feasible solution to (22), i.e. is a classifier that satisfies an AUC constraint using the same probability level α_0.

4.3.2 Generating frontier

To produce the bAUC Efficiency Frontier one can solve (23) for many values of C_0 or one can solve (24) for many values of α_0. In doing so, one will generate a set of solution triples (w^*, C^*, α^*). This allows one to explicitly evaluate the tradeoff taking place between cost

[2]Though it is not indicated in this notation, $f_c(w)$ could well be a function of w and additional variables. For example, in classification one typically needs to choose a decision threshold $b \in \mathbb{R}$. This threshold will affect misclassification costs, making $f_c(w, b)$ dependent on w and b.

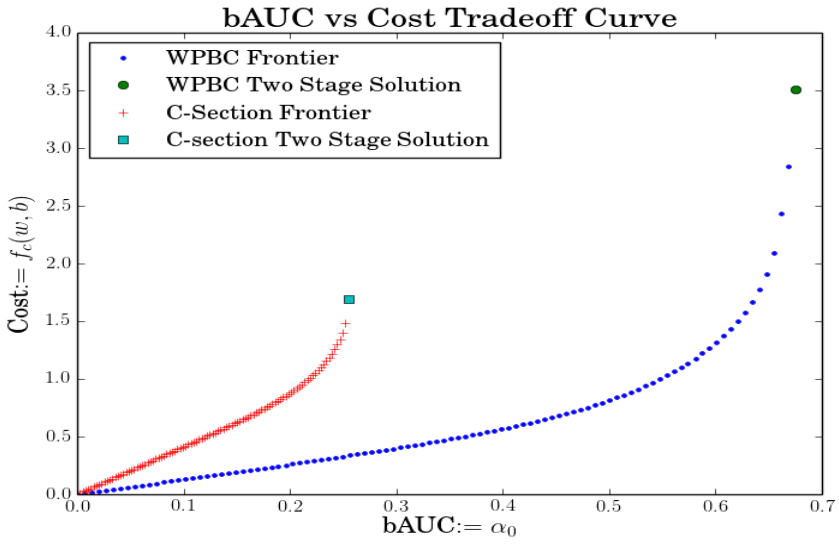

Figure 1: bAUC Efficiency Frontier: This chart shows two bAUC frontier curves. bAUC frontier for C-section data set is left-most curve. bAUC frontier for WPBC data set is right-most curve. For each curve, the solution given via the two-stage problem, i.e. (26) followed by (27), is represented by the "two stage solution" point on the chart. Solving the frontier optimization problem for 99 different α_0 values then generates the rest of the curve by yielding 99 solution pairs, (C^*, α_0).

and bAUC. The issues associated with AUC/bAUC "incoherency" are now alleviated, as we are now able to make these tradeoffs unambiguous.

For a linear cost function, $f_c(w)$, calculation of the frontier can be performed extremely efficiently. Specifically, one can utilize parametric dual simplex to solve for a wide range of C_0 or α_0 values in one algorithmic pass, (see [9]).

This works to resolve the "incoherency" associated with measures of classifier ranking ability. When one chooses a classifier by maximizing AUC (or bAUC) and then independently chooses the threshold that minimizes cost, one falls into the trap of "incoherency." In other words, by choosing the parameters $w \in \mathbb{R}^n$ via AUC or bAUC maximization first, then secondarily choosing the threshold $b \in \mathbb{R}$ via cost minimization, it is possible that one has chosen w such that (w, b) is greatly suboptimal w.r.t. misclassification cost. The bAUC efficiency frontier avoids this trap, allowing for *simultaneous* consideration of cost and classifier ranking ability, $f_c(w)$ vs. $bAUC(w)$.

4.3.3 Frontier Example:

Data: To illustrate the usefulness of this technique, we generate the bAUC Efficiency Frontier for two medical diagnostic tasks. Medical diagnostic classification serves to illustrate an application of classification where AUC is a popular performance metric, but where misclassification (misdiagnosis) costs are important to consider. Here we are using two data sets, both binary classification tasks. The first data set is the Wisconsin Prognostic Breast Cancer (WPBC) from [1], which is a popular classification data set where the task is to predict recurrence of breast cancer. The second data set comes from [3]. Here, the task is to predict whether a woman in labor will require a Cesarean Section (C-section), a surgical procedure used to deliver a child. Details regarding size and solving times for both data sets can be found in Table 1.

Methodology: First, we consider generating a bAUC frontier for each data set by solving the following optimization problem for multiple values of $\alpha_0 \in [0, 1]$.

$$\begin{aligned} \min_{w,b} \quad & f_c(w, b) \\ s.t. \quad & 1 - E[L(w) + 1]^+ \geq \alpha_0 \end{aligned} \tag{25}$$

For each value of α_0, we generate an optimal solution triple $(w^*, C^* = f_c(w^*, b^*), \alpha_0)$. Thus, for each data set, after solving for many values of α_0 we can form a set of solution pairs $(C^* = f_c(w^*, b^*), \alpha_0)$. In Figure 1, we plot these solution pairs to generate the bAUC vs. Cost tradeoff curve for each individual data set.

For comparison with the above method, we also perform a two stage optimization task for each data set. In the first stage, we solve for the classifier that simply maximizes bAUC.

$$\max_w \, bAUC(w) = 1 - \min_w \, E[L(w) + 1]^+ \ . \tag{26}$$

After solving for the bAUC maximizing classifier, which we denote as w^*, we perform a second stage optimization tasks to find the misclassification cost minimizing decision threshold, $b \in \mathbb{R}$. Letting $f_c(w, b)$ represent a misclassification cost function, we solve for

$$b^* = \operatorname*{argmin}_b \, f_c(w^*, b) \ . \tag{27}$$

Note that we have fixed $w = w^*$ within the objective cost function and are only concerned with the optimal threshold, b.

This two stage optimization task serves two purposes. Assume that the two stage optimization problem, for some data set, yields the optimal solution triple $(w^*, C^* = f_c(w^*, b^*), \alpha^* = bAUC(w^*))$. First, this provides us with a baseline optimal pair (C^*, α^*) representative of a common approach that, as we have discussed, can be considered "incoherent" w.r.t. misclassification costs. Although we have maximized bAUC, we do not

know if (or by how much) our misclassification costs are suboptimal. Secondly, solving the above first stage problem before generating the bAUC frontier is useful in that it gives us the feasible range of α_0 for the bAUC frontier optimization problem. Assume that w^* is an optimal solution to the first stage bAUC maximization problem with $bAUC(w^*) = \alpha^*$. When attempting to choose the values of α_0 for which we want to generate bAUC frontier points, we know that any $\alpha_0 > \alpha^*$ yields an infeasible bAUC frontier optimization problem.

Software: For numerical experiments, we utilized Portfolio Safeguard (PSG) of AOrDA.com. When concerned with a difference of two linear losses, as we are with $L(w)$, optimization packages can struggle to handle the large number of pairwise losses that must be evaluated, which is (m^+m^-) in our notation. PSG, on the other hand, can efficiently handle such situations via a precoded Partial Moment function tailored to handle such linear loss differences.

Specifying the Cost Function: For the data set we selected, we assume that a False Negative (FN) is far more costly than a False Positive (FP). This could apply to any predictive medical diagnostic where the cost of preventative measures given a FP are much lower than the cost of a FN which may cause loss of life or costly medical procedures when critically acute symptoms arise after failed detection. Let $C^+ = 10$ denote the cost of a FN, $C^- = 1$ denote the cost of a FP. For simplicity, we consider the misclassification cost function

$$f_c(w, b) = \frac{C^+}{m^+} \sum_{i \in S^+} [-(w^T x_i^+ + b)]^+ + \frac{C^-}{m^-} \sum_{j \in S^-} [w^T x_j^- + b]^+ .$$

Results: Here, we give a brief overview of the results. For the interested reader, full data sets, results, and PSG code for this case study can be found online.[3]

For each of the two data sets, we solved for 100 different values of α_0 to generate each data sets' bAUC frontier. Looking at Figure 1, we see a bAUC frontier curve for each data set generated by plotting 100 different solution pairs $(C^* = f_c(w^*, b^*), \alpha_0)$. The left-most curve in Figure 1 is the bAUC frontier generated for the C-section data set while the right-most curve in Figure 1 is the bAUC frontier generated for the WPBC data set. As noted before, we determined the upper bound on the feasible range of α_0 via the two stage optimization problem. Looking at Figure 1, we have plotted (for each data set) the optimal (C^*, α^*) pair given by the two stage optimization problem. This corresponds to the largest α_0 value on each respective curve, which makes sense because it represents the bAUC maximizing classifier.

Consider now the case of a decision maker relying only on the two stage optimization solution with no knowledge of the bAUC frontier. First, regardless of the shape of the

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

Data Set	Positive Samples	Negative Samples	Solve Time for 100 α_0 values
WPBC	46	148	< 5 seconds
C-section	2788	9902	< 30 seconds

Table 1: Size of data sets and approximate amount of time needed to generate 100 points on the bAUC efficiency frontier curve. Optimization was implemented in Portfolio Safeguard by AOrDa.com

curves given in Figure 1, we see that he is missing out on a wealth of information regarding the bAUC vs. cost tradeoff. Because he first maximized bAUC, he does not know if (or by how much) his costs (given in the second stage of the two stage optimization) are suboptimal. Secondly, considering the shape of the curves in Figure 1, we see that in the particular case of our two data sets and cost function, the bAUC curve reveals that we are paying a substantial cost premium by choosing the bAUC maximizing classifier. The bAUC frontier curve reveals that if we were to accept a slightly lower bAUC, we could considerably lower the resulting cost.

5 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, AUC can be criticized as being a shallow measure similar to Value-at-Risk. As a potential objective or constraint in optimization, AUC is numerically troublesome for discrete distributions. 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, we have found bAUC to be potentially more informative than AUC with regard to measuring the ranking quality of different classifiers. We have also found that maximizing bAUC reduces to convex programming, and even linear programming in the case of a linear decision function.

AUC and bAUC both suffer when misclassification costs need to be considered. The ease with which bAUC can be handled as an objective function or constraint in optimization, though, allows us to develop a simple framework to avoid this issue. Generation of the bAUC efficiency frontier allows the tradeoffs between misclassification cost and ranking ability to be made explicit, providing much more information to the decision maker.

References

[1] Bache, K., Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Brefeld, U., Scheffer, T. (2005). AUC Maximizing Support Vector Learning Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning.

[3] Chen, G., S. Uryasev, Young, T.K. (2004). On Prediction of the Cesarean Delivery Risk in a Large Private Practice. American Journal of Obstetrics and Gynecology 191, 617-25

[4] Cortes, C., Mohri, M. (2004). AUC Optimization vs. Error Rate Minimization. Advances in Neural Information Processing Systems, 16(16), 313-320.

[5] Davis, J.R., Uryasev S. (2014). Analysis of Hurricane Damage using Buffered Probability of Exceedance. Research Report 2014-4, ISE Dept., University of Florida.

[6] Hand, D. (2009). Measuring Classifier Performance: A Coherent Alternative To The Area Under The ROC Curve. Machine learning 77(1), 103-123

[7] Hanley J. A., McNeil B. J. (1982). The Meaning and Use Of The Area Under a Receiver Operating Characteristic (ROC) Curve. Radiology, 143(1), 29-36.

[8] Krm E., Yildirak K., Weber G.W. (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.

[9] Mafusalov A., Uryasev S. (2014). Buffered Probability of Exceedance: Mathematical Properties and Optimization Algorithms. Research Report 2014-1, ISE Dept., University of Florida.

[10] Miura K., Yamashita S., Eguchi S.(2010). Area Under the Curve Maximization Method in Credit Scoring. The Journal of Risk Model Validation, 4(2), 3–25.

[11] Rockafellar R.T. and S. Uryasev (2000) Optimization of Conditional Value-At-Risk. The Journal of Risk, Vol. 2, No. 3, 2000, 21-41.

[12] Rockafellar, R.T. (2009). Safeguarding Strategies in Risky Optimization. Presentation at the International Workshop on Engineering Risk Control and Optimization, Gainesville, FL, February, 2009.

[13] Rockafellar R.T., Royset J.O. (2010). On Buffered Failure Probability in Design and Optimization of Structures. Reliability Engineering & System Safety, Vol. 95, 499-510.