Journal of Statistical Software
MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

ROC and AUC with a Binary Predictor: a Potentially Misleading Metric

John Muschelli
Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health

Abstract

In analysis of binary outcomes, the receiver operator characteristic (ROC) curve is heavily used to show the performance of a model or algorithm. The ROC curve is informative about the performance over a series of thresholds and can be summarized by the area under the curve (AUC), a single number. When a **predictor** is categorical, the ROC curve has only as many thresholds as the one less than number of categories; when the predictor is binary there is only one threshold. As the AUC may be used in decision-making processes on determining the best model, it important to discuss how it agrees with the intuition from the ROC curve. We discuss how the interpolation of the curve between thresholds with binary predictors can largely change the AUC. Overall, we believe a linear interpolation from the ROC curve with binary predictors, which is most commonly done in software, can lead to misleading results.

Keywords: roc, auc, area under the curve, R.

1. Introduction

In many applications, receiver operator characteristic (ROC) curves are used to show how a predictor compares to the true outcome. One of the large advantages of ROC analysis is that it is threshold-agnostic; performance of a predictor is estimated without a specific threshold and also gives a criteria to choose an optimal threshold based on a certain cost function or objective. Typically, an ROC analysis shows how sensitivity (true positive rate) changes with varying specificity (true negative rate or $1 -$ false positive rate). Analyses also typically weigh false positives and false negatives equally.

Many predictors, especially medical tests, result in a binary decision; a value is higher than a pre-determined threshold or a substance is present. Similarly, some predictors are simply categorical or discrete such as low, normal, or high blood pressure while others are categorical by nature such as having a specific gene or not. These are useful indicators of presence a disease, the primary outcome of interest. The predictive capabilities of the variable is commonly summarized by the area under the curve (AUC). Additionally, partial ROC (pROC) analysis keeps a specificity fixed and can summarize a predictor by the partial AUC (pAUC) or the optimal sensitivity at that fixed false positive rate.

If one assumes the binary predictor is generated from a continuous distribution that has been thresholded, then the sensitivity of this thresholded predictor actually represents one point on the ROC curve for the underlying continuous value. Therefore the ROC curve of a binary predictor is not really appropriate, but should be represented by a single point on the curve. But alas, ROC and AUC analysis is done on binary predictors and used to inform if one variable is more predictive than the other (E, C, K, and et al 2018; TV, GH, JA, S, K, and GY 2017; Glaveckaite, Valeviciene, Palionis, Skorniakov, Celutkiene, Tamosiunas, Uzdavinys, and Laucevicius 2011; Blumberg, De Moraes, Liebmann, Garg, Chen, Theventhiran, and Hood 2016; Budwega, Sprengerb, De Vere-Tyndalld, Hagenkordd, Stippichd, and Bergera 2016; **?**). For example, these cases show that researchers use ROC curves and AUC to evaluate predictors, even when the predictors are categorical or binary. Although there is nothing inherently wrong with this comparison, it can lead to drastically different predictors being selected based on these criteria if ties are treated slightly different ways.

A more appropriate comparison of a continuous predictor and the binary predictor may be to compare the sensitivity and specificity (or overall accuracy) of the continuous predictor given the optimal threshold versus that of the binary predictor.

Fawcett (2006) describes how ties are handled in a predictor. Ties are distinctly relevant for discrete and binary predictors or models that predict a discrete number of values, where many observations can have the same value/risk score. When drawing the ROC curve, one can assume that all the ties do not correctly classify the outcome (Fawcett called the "pessimistic" approach) or that all the ties do correctly classify the outcome (called the "optimistic" approach), see Fig. 6 in (Fawcett 2006). But Fawcett notes (emphasis in original):

> Any mixed ordering of the instances will give a different set of step segments within the rectangle formed by these two extremes. However, the ROC curve should represent the *expected* performance of the classifier, which, lacking any other information, is the average of the pessimistic and optimistic segments.

This "expected" performance directly applies to the assignment of a half probability of success when the data are tied, which is implied by the "trapezoidal rule" from Hanley and McNeil (1982). Fawcett (2006) also states in the calculation of AUC that "trapezoids are used rather than rectangles in order to average the effect between points". This trapezoidal rule applies additional areas to the AUC based on ties of the predictor, giving a half a probability. This addition of half probability is linked to how ties are treated in the Wilcoxon rank sum test. As much of the theory of ROC curve testing, and therefore testing of differences in AUC, is based on the theory of the Wilcoxon rank-sum test, this treatment of ties is relevant to statistical inference.

Others have discussed insights into binary predictors in addition to Fawcett (2006), but they are mentioned in small sections of the paper (Saito and Rehmsmeier 2015; Pepe, Longton, and Janes 2009). Other information regarding ties and binary data are blog posts or working papers such as `http://blog.revolutionanalytics.com/2016/11/calculating-auc.html` or https://www.epeter-stats.de/roc-curves-and-ties/, which was written by the author of the **fbroc** (Peter 2016) package, which we will discuss below. Most notably, Hsu and Lieli (2014) is an extensive discussion of ties, but the paper was not published.

Although many discuss the properties of ROC and AUC analyses, we wish to plainly show the math and calculations of the AUC with a binary predictor explore commonly-used statistical software for ROC curve creation and AUC calculation in a variety of packages and languages. Overall, we believe that AUC calculations alone may be misleading for binary or categorical predictors depending on the definition of the AUC. We propose to be explicit when reporting the AUC in terms of the approach to ties.

2. Mathematical Proof of AUC for Single Binary Predictor

First, we will show how the AUC is defined in terms of probability. This representation is helpful in discussing the disconnect between the stated interpretation of the AUC, the formal definition, and how the treatment of ties is crucial when the data are discrete. Let us assume we have a binary predictor X and a binary outcome Y, such that X and Y only take the values 0 and 1, the number of replicates is not relevant here. Let X_i be the values of $X|Y = i$, where $i \in \{0, 1\}$.

Fawcett (2006) goes on to state:

> AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

In other words, we could discern the definition AUC $= P(X_1 > X_0)$. As there are only two outcomes for X, we can expand this probability using the law of total probability:

$$\begin{aligned} P(X_1 > X_0) &= P(X_1 > X_0|X_1 = 1)P(X_1 = 1) \\ &\quad + P(X_1 > X_0|X_1 = 0)P(X_1 = 0) \qquad (1) \\ &= P(X_1 > X_0|X_1 = 1)P(X_1 = 1) \qquad (2) \end{aligned}$$

as $P(X_1 > X_0|X_1 = 0) = 0$ because $X_0 \in \{0, 1\}$. We see that $P(X_1 = 1)$ in equation (2) is the sensitivity:

$$\begin{aligned} P(X_1 = 1) &= P(X = 1|Y = 1) \\ &= \frac{TP}{TP + FN} \\ &= \text{sensitivity} \end{aligned}$$

and that $P(X_1 > X_0|X_1 = 1)$ in equation (2) is the specificity:

Table 1: A simple 2x2 table of a binary predictor (rows) versus a binary outcome (columns)

	0	1
0	52	35
1	32	50

$$
\begin{aligned}
P(X_1 > X_0 | X_1 = 1) &= P(X_1 > X_0 | X_1 = 1, X_0 = 1) P(X_0 = 1) \\
&+ P(X_1 > X_0 | X_1 = 1, X_0 = 0) P(X_0 = 0) \\
&= P(X_1 > X_0 | X_1 = 1, X_0 = 0) P(X_0 = 0) \\
&= P(X_0 = 0) \\
&= P(X = 0 | Y = 0) \\
&= \frac{TN}{TN + FP} \\
&= \text{specificity}
\end{aligned}
$$

Therefore, we combine these two to show that equation (2) reduces to:

$$P(X_1 > X_0) = \text{specificity} \times \text{sensitivity}$$

Using the definition of AUC as $P(X_1 > X_0)$, it is the sensitivity times the specificity.

If we change the definition of AUC slightly while accounting for ties, which we call $\text{AUC}_{\text{w/ties}}$, to

$$\text{AUC}_{\text{w/ties}} = P(X_1 > X_0) + \frac{1}{2} P(X_1 = X_0)$$

This AUC is the one reported by most software, as we will see below.

2.1. Simple Concrete Example

To give some intuition of this scenario, we will assume X and Y have the following joint distribution, where X is along the rows and Y is along the columns:

Therefore, the AUC should be equal to $\frac{50}{85} \times \frac{52}{84}$, which equals 0.364.

We will define this as the the strict definition of AUC, where ties are not taken into account and we are using strictly greater than in the probability and will call this value $\text{AUC}_{\text{definition}}$.

Note, if we reverse the labels, then the sensitivity and the specificity are estimated by 1 minus that measure, or $\frac{35}{85} \times \frac{32}{84}$, which is equal to 0.157.

Thus, as this AUC is less than the original labeling, we would choose that with the original labeling.

If we used the calculation for $\text{AUC}_{\text{w/ties}}$ we see that we estimate AUC by $\text{AUC}_{\text{definition}} + \frac{1}{2}\left(\frac{50+52}{169}\right)$, which is equal to 0.604.

Monte Carlo Estimation of AUC

We can also show that if we use simple Monte Carlo sampling, we can randomly choose X_0 and X_1. From these samples, we can estimate these AUC based on the definitions above. Here, the function `est.auc` samples 10^6 random samples from X_1 and X_0, then calculates $\widehat{\text{AUC}}_{\text{definition}}$ and $\widehat{\text{AUC}}_{\text{w/ties}}$:

```
R> est.auc = function(x, y, n = 1000000) {
R+   x1 = x[y == 1] # sample x | y = 1
R+   x0 = x[y == 0] # sample x | y = 0
R+   c1 = sample(x1, size = n, replace = TRUE)
R+   c0 = sample(x0, size = n, replace = TRUE)
R+   auc.defn = mean(c1 > c0) # compare
R+   auc.wties = auc.defn + 1/2 * mean(c1 == c0) # compare
R+   return(c(auc.definition = auc.defn,
R+            auc.wties = auc.wties))
R+ }
R> sample.estauc = est.auc(x, y)
R> sample.estauc

auc.definition      auc.wties
      0.364631       0.603990
```

And thus we see these simulations agree with the values estimated above.

Geometric Argument of AUC

We will present a geometric discussion of the ROC as well. In Figure 1, we show the ROC curve for the simple concrete example. In panel A, we show the point of sensitivity/specificity connected by the step function, and the associated AUC is represented in the shaded blue area, representing $\text{AUC}_{\text{definition}}$. In panel B, we show the additional shaded areas that are due to ties in orange and red; all shaded areas represent $\text{AUC}_{\text{w/ties}}$. We can see this by expanding $P(X_1 = X_0)$ such that:

$$\begin{aligned} P(X_1 = X_0) &= P(X_1 = 1, X_0 = 1) + P(X_1 = 0, X_0 = 0) \\ &= P(X_1 = 1)P(X_0 = 1) + P(X_1 = 0)P(X_0 = 0) \\ &= (\text{sensitivity} \times (1 - \text{specificity})) + ((1 - \text{sensitivity}) \times \text{specificity}) \end{aligned}$$

so that we have

$$\begin{aligned} \text{AUC}_{\text{w/ties}} &= \text{specificity} \times \text{sensitivity} \\ &+ \frac{1}{2}\,(\text{sensitivity} \times (1 - \text{specificity})) \\ &+ \frac{1}{2}\,((1 - \text{sensitivity}) \times \text{specificity}) \end{aligned}$$

Thus, we can see that geometrically from Figure 1:

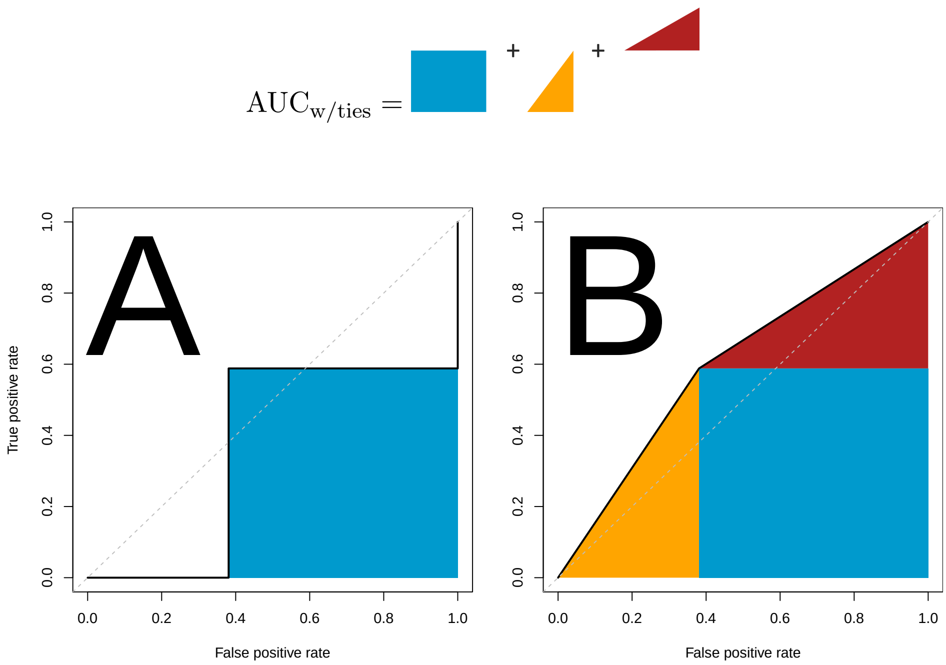


Figure 1: ROC curve of the data in the simple concrete example. Here we present a standard ROC curve, with the false positive rate or $1 -$ specificity on the x-axis and true positive rate or sensitivity on the y-axis. The dotted line represents the identity. The shaded area in panel represents the AUC for the strict definition. The additional shaded areas on panel B represent the AUC when accounting for ties.

2.2. AUC Calculation in Statistical Software

We will explore the estimated ROC curve and AUC from the implementations in the following `R` packages: **ROCR** (Sing, Sander, Beerenwinkel, and Lengauer 2005), **caTools** (Tuszynski 2018), **pROC** (Robin, Turck, Hainard, Tiberti, Lisacek, Sanchez, and Müller 2011), and **fbroc** (Peter 2016). We will also show these agree with the Python implementation in `sklearn.metrics` from **scikit-learn** (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, and Duchesnay 2011), the Stata functions `roctab` and `rocreg` (Bamber 1975; DeLong, DeLong, and Clarke-Pearson 1988), and the SAS software functions `proc logistic` with `roc` and `roccontrast` . We note that these functions all count half the probability of ties, which raises the AUC.

3. AUC Calculation: Current Implementations

This section will present code and results from commonly-used implementations of AUC estimation from R, Python, and Stata. We will note agreement with the definitions of AUC above and any discrepancies. This section is not to be exhaustive, but show that these definitions are consistently used in AUC analysis, primarily $\widehat{\mathrm{AUC}}_{\mathrm{w/ties}}$.

3.1. R

Here we will show the AUC calculation from the common R packages for ROC analysis. We will show that each report the value calculated in $\mathrm{AUC}_{\mathrm{w/ties}}$. The **caTools** package calculates AUC using the `colAUC` function:

```
R> library(caTools)
R> colAUC(x, y)

             [,1]
0 vs. 1 0.6036415
```

In **ROCR**, AUC is calculated from a `performance` object, which takes in a `prediction` object:

```
R> library(ROCR)
R> pred = prediction(x, y)
R> auc.est = performance(pred, "auc")
R> auc.est@y.values[[1]]

[1] 0.6036415
```

The **pROC** package calculates AUC using the `roc` function:

```
R> library(pROC)
R> pROC.roc = pROC::roc(predictor = x, response = y)
R> pROC.roc[["auc"]]

Area under the curve: 0.6036
```

The **fbroc** package is one of the most popular packages for doing ROC analysis (Peter 2016). Using the `fbroc::boot.roc` and `fbroc::perf` functions, we have:

```
R> library(fbroc)
R> fbroc.default = boot.roc(x, as.logical(y),
R+                          n.boot = 1000, tie.strategy = 2)
R> auc.def = perf(fbroc.default, "auc")
R> auc.def[["Observed.Performance"]]

[1] 0.6036415

R> fbroc.alternative = boot.roc(x, as.logical(y),
R+                              n.boot = 1000, tie.strategy = 1)
R> auc.alt = perf(fbroc.alternative, "auc")
R> auc.alt[["Observed.Performance"]]
```

```
[1] 0.6036415
```

which give identical results to above. Although these strategies for ties are different, they are relevant for the plotting for the ROC curve. The standard error calculation uses the second strategy (the "pessimistic" approach), which is described in a blog post (https://www.epeter-stats.de/roc-curves-and-ties/) and can be seen in Figure 2C.

3.2. SAS Software

In SAS software (version 9.4 for Unix) (SAS and Version 2017), let's assume we have a data set named `roc` loaded with the variables/columns of `x` and `y` as above. The following commands will produce the ROC curve in figure

```
R>   proc logistic data=roc;
R+        model y(event='1') = x;
R+        roc; roccontrast;
R+        run;
```

https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_logistic_sect040.htm

3.3. Stata

In Stata (StatCorp, College Station, TX, version 13) (Stata 2013), the function `roctab` is one common way to calculate an AUC:

```
R> roctab x y

 . roctab x y

                      ROC                    -Asymptotic Normal--
           Obs       Area     Std. Err.      [95% Conf. Interval]
         --------------------------------------------------------
           169     0.6037       0.0379        0.52952     0.67793
```

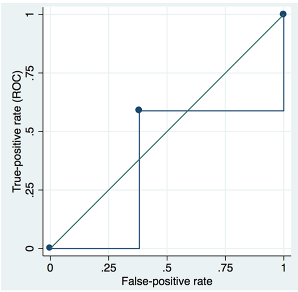
which agrees with the calculation based on $\text{AUC}_{\text{w/ties}}$ and agrees with the estimates from above. One can also calculate the AUC using the `rocreg` function:

```
R> rocreg y x, nodots auc

 . rocreg y x, nodots auc

Bootstrap results                                Number of obs      =        169
                                                 Replications       =       1000

Nonparametric ROC estimation
```

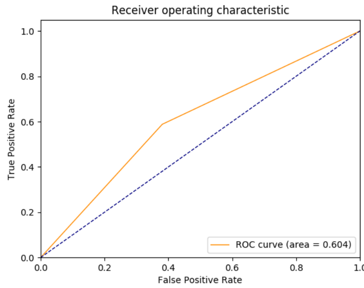
```
Control standardization: empirical
ROC method             : empirical

Area under the ROC curve

   Status    : y
   Classifier: x
------------------------------------------------------------------------------
             |    Observed               Bootstrap
         AUC |       Coef.       Bias    Std. Err.     [95% Conf. Interval]
-------------+----------------------------------------------------------------
             |    .3641457  -.0004513    .0451334     .2756857   .4526056  (N)
             |                                        .2771778    .452824  (P)
             |                                        .2769474   .4507576 (BC)
------------------------------------------------------------------------------
```

which agrees with the definition of $\text{AUC}_{\text{definition}}$ and is different from the output from `roctab`. The variance of the estimate is based on a bootstrap estimate, but the point estimate will remain the same regardless of using the bootstrap or not. This disagreement of estimates is concerning as the reported estimated AUC may be different depending on the command used in the estimation.

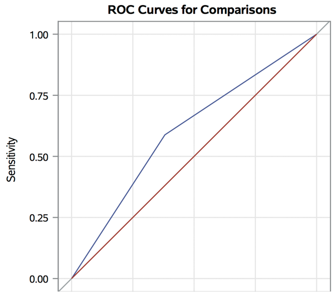
Using `rocregplot` after running this estimation, we see can create an ROC curve, which is shown in Figure 2a. We see that the estimated ROC curve coincides with the estimated AUC from `rocreg` and the blue rectangle in Figure 1.



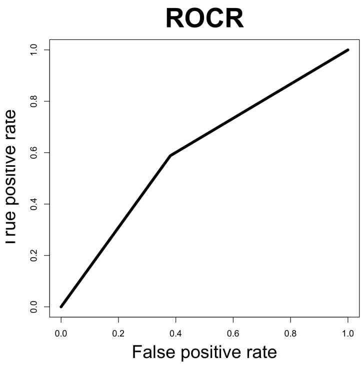
(a) Stata ROC Plot



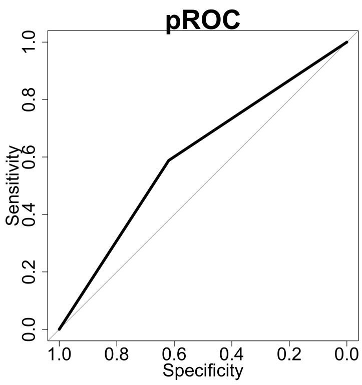
(b) Python ROC Plot from scikit-learn



(c) SAS ROC Plot



(d) ROCR ROC plot

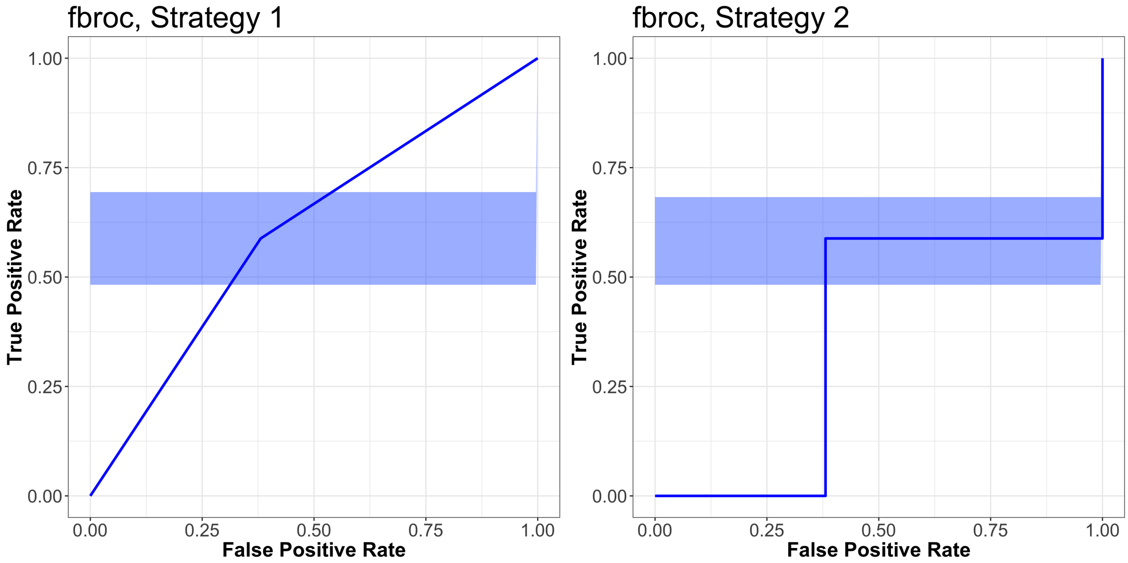


(e) pROC ROC plot

(f) fbroc Strategy 2 (default)

Figure 2: Comparison of different ROC curves for different `R` packages, `scikit-learn` from `Python`, `SAS`, and `Stata`. Each line represents the ROC curve, which corresponds to an according area under the curve (AUC). The blue shading represents the confidence interval for the ROC curve in the `fbroc` package. Also, each software represents the curve as the false positive rate versus the true positive rate, though the `pROC` package calls it sensitivity and specificity (with flipped axes). Some put the identity line where others do not. Overall the difference of note as to whether the ROC curve is represented by a step or a linear function. Using the first tie strategy for ties (non-default) in `fbroc` gives the same confidence interval but an ROC curve using linear interpolation.

We see that all ROC curves are interpolated with a linear interpolation, which coincides with the calculation based on $\text{AUC}_{\text{w/ties}}$, except for the Stata ROC curve, which interpolates using a step function and coincides with $\text{AUC}_{\text{definition}}$. The confidence interval estimate of the ROC curve for `fbroc`, which is shaded in blue in Figures 2f, corresponds to variability based on $\text{AUC}_{\text{definition}}$, though Figure **??** shows the ROC curve based on $\text{AUC}_{\text{w/ties}}$.



(a) fbroc Strategy 1 (b) fbroc Strategy 2 (default)

Figure 3: Comparison of different strategies for ties in the **fbroc** package. The blue shading represents the confidence interval for the ROC curve. Overall the difference of note as to whether the ROC curve is represented by a step or a linear function. Using the first tie strategy for ties (non-default) in **fbroc** gives the same confidence interval as the second strategy but an ROC curve using linear interpolation, which may give an inconsistent combination of estimate and confidence interval.

In Figure 3 we show the differing plots for using the first (panel 3a) and second (panel 3b, default, duplicated). We see that **fbroc** gives a different estimate of AUC based on the tie strategy, but the same estimate of the confidence interval of the ROC curve, regardless of tie strategy. Therefore, there may give an inconsistent combination of estimate and confidence interval.

3.4. Fawcett Example

In Fawcett (2006), the authors introduce a data set where there there should be a completely correct classifier.

```
R> faw = data.frame(y = c(rep(TRUE, 6), rep(FALSE, 4)),
R+                  x = c(0.99999, 0.99999, 0.99993,
R+                        0.99986, 0.99964, 0.99955,
R+                        0.68139, 0.50961, 0.48880, 0.44951))
R> faw = faw %>% mutate(hyp = x > 0.5)
R> pred = prediction(predictions = faw[, "x"], labels = faw[, "y"])
R> auc.estimated = performance(pred, "auc")
R> auc.estimated@y.values[[1]]

[1] 1
```

```
R> est.auc(x = faw[, "x"], y = faw[, "y"])

auc.definition      auc.wties
             1              1
```

References

Bamber D (1975). "The area above the ordinal dominance graph and the area below the receiver operating characteristic graph." *Journal of mathematical psychology*, **12**(4), 387–415.

Blumberg DM, De Moraes CG, Liebmann JM, Garg R, Chen C, Theventhiran A, Hood DC (2016). "Technology and the glaucoma suspect." *Investigative ophthalmology & visual science*, **57**(9), OCT80–OCT85.

Budwega J, Sprengerb T, De Vere-Tyndalld A, Hagenkordd A, Stippichd C, Bergera CT (2016). "Factors associated with significant MRI findings in medical walk-in patients with acute headache." *Swiss Med Wkly*, **146**, w14349.

DeLong ER, DeLong DM, Clarke-Pearson DL (1988). "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach." *Biometrics*, pp. 837–845.

E M, C M, K S, et al (2018). "Diagnostic criteria of ulcerative pyoderma gangrenosum: A delphi consensus of international experts." *JAMA Dermatology*, **154**(4), 461–466. doi:10.1001/jamadermatol.2017.5980. /data/journals/derm/936924/jamadermatology_maverakis_2018_cs_170003.pdf, URL +http://dx.doi.org/10.1001/jamadermatol.2017.5980.

Fawcett T (2006). "An introduction to ROC analysis." *Pattern recognition letters*, **27**(8), 861–874.

Glaveckaite S, Valeviciene N, Palionis D, Skorniakov V, Celutkiene J, Tamosiunas A, Uzdavinys G, Laucevicius A (2011). "Value of scar imaging and inotropic reserve combination for the prediction of segmental and global left ventricular functional recovery after revascularisation." *Journal of Cardiovascular Magnetic Resonance*, **13**(1), 35.

Hanley JA, McNeil BJ (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, **143**(1), 29–36.

Hsu YC, Lieli R (2014). "Inference for ROC curves based on estimated predictive indices: A note on testing AUC = 0.5." *Unpublished manuscript.*

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, **12**, 2825–2830.

Pepe M, Longton G, Janes H (2009). "Estimation and comparison of receiver operating characteristic curves." *The Stata Journal*, **9**(1), 1.

Peter E (2016). *fbroc: Fast Algorithms to Bootstrap Receiver Operating Characteristics Curves*. R package version 0.4.0, URL https://CRAN.R-project.org/package=fbroc.

Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez JC, Müller M (2011). "pROC: an open-source package for R and S+ to analyze and compare ROC curves." *BMC Bioinformatics*, **12**, 77.

Saito T, Rehmsmeier M (2015). "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets." *PloS one*, **10**(3), e0118432.

SAS S, Version S (2017). "9.4 [Computer Program]." *Cary, NC: SAS Institute*.

Sing T, Sander O, Beerenwinkel N, Lengauer T (2005). "ROCR: visualizing classifier performance in R." *Bioinformatics*, **21**(20), 7881. URL http://rocr.bioinf.mpi-sb.mpg.de.

Stata S (2013). "Release 13. Statistical software." *StataCorp LP, College Station, TX*.

Tuszynski J (2018). *caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc.* R package version 1.17.1.1, URL https://CRAN.R-project.org/package=caTools.

TV L, GH B, JA C, S S, K K, GY O (2017). "A revised approach for the detection of sight-threatening diabetic macular edema." *JAMA Ophthalmology*, **135**(1), 62–68. doi:10.1001/jamaophthalmol.2016.4772. /data/journals/ophth/935943/jamaophthalmology_litvin_2016_oi_160098.pdf, URL +http://dx.doi.org/10.1001/jamaophthalmol.2016.4772.

Affiliation:

John Muschelli
Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health
615 N Wolfe St Baltimore, MD 21205
E-mail: jmuschel@jhsph.edu
URL: http://johnmuschelli.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd