Comparison of Non-Parametric Methods for Assessing Classifier Performance in Terms of ROC Parameters

Waleed A. Yousef

Department of Electrical and Computer Engineering,
George Washington University
wyousef@aucegypt.edu

Robert F. Wagner

Office of Science and Engineering Laboratories,
Center for Devices & Radiological Health, FDA
rfw@cdrh.fda.gov

Murray H. Loew

Department of Electrical and Computer Engineering,
George Washington University
loew@gwu.edu

Abstract

The most common metric to assess a classifier's performance is the classification error rate, or the probability of misclassification (PMC). Receiver Operating Characteristic (ROC) analysis is a more general way to measure the performance. Some metrics that summarize the ROC curve are the two normal-deviate-axes parameters, i.e., a and b, and the Area Under the Curve (AUC). The parameters "a" and "b" represent the intercept and slope, respectively, for the ROC curve if plotted on normal-deviate-axes scale. AUC represents the average of the classifier TPF over FPF resulting from considering different threshold values. In the present work, we used Monte-Carlo simulations to compare different bootstrap-based estimators, e.g., leave-one-out, .632, and .632+ bootstraps, to estimate the AUC. The results show the comparable performance of the different estimators in terms of RMS, while the .632+ is the least biased.

1. INTRODUCTION

In this article we consider the binary classification problem, where a sample case $t_i = (x_i, y_i)$ has the p-dimensional feature vector x_i, the predictor, and belongs to the class y_i where $y_i = \omega_1, \omega_2$. Given a sample $\mathbf{t} = \{t_i : t_i = (x_i, y_i),\ i = 1,2,\cdots,n\}$ consisting of n cases, statistical learning may be performed on this training data set to design the prediction function $\eta_{\mathbf{t}}(x_0)$ to predict, i.e., estimate the class, y_0 of any future case $t_0 = (x_0, y_0)$ from its predictor x_0.

One of the most important criteria in designing the prediction function $\eta_{\mathbf{t}}(x_0)$ is the expected loss (risk) defined by:

$$R(\eta) = E_{0F}\left[L(\eta_{\mathbf{t}}(x_0), y_0)\right], \tag{1}$$

where F represents the probability distribution of the data, L is the loss function for misclassification, and E_{0F} is the expectation under the distribution F taken over the testers (x_0, y_0). The designed classifier assigns, for each value in the feature space, a likelihood ratio of the posterior probabilities conditional on the given feature vector. The classification $\eta_{\mathbf{t}}(x_0)$ is decided by comparing the log of this likelihood ratio $h_{\mathbf{t}}(x_0)$ to the threshold value th, which is a function in the *a priori* probabilities, P_1 and P_2, of the two classes and the costs. The decision $\eta_{\mathbf{t}}(x_0)$ minimizes the risk if

$$\begin{aligned} h_{\mathbf{t}}(x_0) &\underset{\omega_2}{\overset{\omega_1}{\gtrless}} th, \\ th &= \log \frac{\mathrm{P}_1\, c_{21}}{\mathrm{P}_2\, c_{12}} \end{aligned} \tag{2}$$

where c_{ij} is the cost of predicting class ω_i while the truth is class ω_j (see [1]). This minimum risk is given by:

$$R_{\min} = c_{12} P_1 e_1 + c_{21} P_2 e_2, \tag{3}$$

where:

$$\begin{aligned} e_1 &= \int_{-\infty}^{th} f_h(h(x) \mid \omega_1)\, dh(x), \\ e_2 &= \int_{th}^{\infty} f_h(h(x) \mid \omega_2)\, dh(x) \end{aligned}, \tag{4}$$

are the two probability areas, for the two types of misclassification, where f_h is the pmf of the likelihood ratio. See Figure 1. If the two costs and the *a priori* probabilities are equal, i.e., the case of zero threshold, the value $e_1 + e_2$ is simply called the true error rate or the probability of misclassification (PMC).

In other environments, there will be different *a priori* probabilities. Since the PMC depends on a single fixed threshold, it is not a sufficient metric for the more general problem. A more general way to assess a classifier is provided by the Receiver Operating Characteristic (ROC) curve. This is a plot for the two components of error, e_1 and e_2, under different threshold values. It is conventional in medical imaging to refer to e_1 as the False Negative Fraction (FNF), and e_2 as the False Positive Fraction (FPF). This is because diseased patients typically have a higher output value for a test than non-diseased patients. For example, a patient belonging to class 1 whose test output value is less than the threshold setting for the test will be called "test negative" while he or she is in fact in the diseased class. This is a false negative decision; hence the name FNF. The situation is vice versa for the other error component.

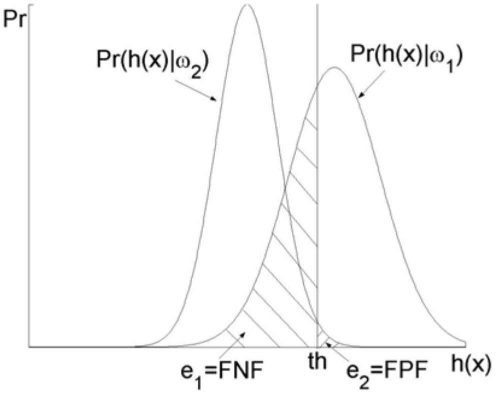

Figure 1. The probability density of log-likelihood ratio conditional under each class. The two components of error are indicated as the FPF and FNF, the conventional terminology in medical imaging.

Under the special case of normal distribution for the log-likelihood ratio $h_t(\cdot)$ the ROC curve can be expressed, using the inverse error function transformation, as:

$$\phi^{-1}(TPF) = \frac{(\mu_1 - \mu_2)}{\sigma_1} + (\frac{\sigma_2}{\sigma_1})\phi^{-1}(FPF) \qquad (5)$$

This means that the whole ROC curve can be summarized in just two parameters: the intercept a, and the slope b; this is shown in Figure 2. We frequently see the Central Limit Theorem at work in higher dimensions driving the ROC curve toward this condition.

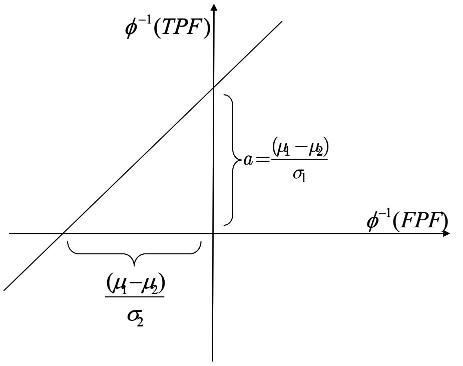
$\phi^{-1}(TPF)$
$a = \frac{(\mu_1 - \mu_2)}{\sigma_1}$
$\phi^{-1}(FPF)$
$\frac{(\mu_1 - \mu_2)}{\sigma_2}$

Figure 2. The double-normal-deviate plot for the ROC under the normal assumption for the log-likelihood ratio is a straight line.

Another important summary metric for the whole ROC curve, which does not require any assumption for the log-likelihood distribution, is the Area Under the ROC curve (AUC) that expresses, on average, how the decision function is able to separate the two classes from each other. The AUC is given formally by:

$$AUC = \int_0^1 TPF \; d(FPF) \qquad (6)$$

The two components of error in (4), or the summary metric AUC in (6), are the parametric forms of these metrics. That is, these metrics can be calculated by these equations if the posterior probabilities are known parametrically, e.g., in the case of the Bayes classifier or by parametric regression techniques. If the posterior probabilities are not known in a parametric form, the error rates can only be estimated numerically from a given data set, called the testing data set. This is done by assigning equal probability mass for each sample case, since this is the Maximum Likelihood Estimation (MLE) for the probability mass function under the nonparametric distribution, i.e.,

$$\hat{F} : mass \; \frac{1}{N} \; on \; t_i, \qquad i = 1, 2, \ldots, N \qquad (7)$$

If n_1, n_2 are the data sizes of the two classes, i.e., $N = n_1 + n_2$, the true risk (3) will be estimated by:

$$\begin{aligned} R(\eta) &= \frac{1}{N} \sum_{i=1}^{N} \Big(c_{12} \, I_{\hat{h}(x_i | \omega_1) < th} + c_{21} \, I_{\hat{h}(x_i | \omega_2) > th} \Big) \\ &= \frac{1}{N} \Big(c_{21} \, \widehat{e_1} \, n_1 + c_{21} \, \widehat{e_2} \, n_2 \Big) \\ &= c_{21} \, \widehat{FNF} \, \widehat{P_1} + c_{21} \, \widehat{FPF} \, \widehat{P_2} \end{aligned} \qquad , (8)$$

I is the indicator function defined by:

$$I_{cond} = \begin{cases} 1 & cond \; is \; True \\ 0 & cond \; is \; False \end{cases} \qquad (9)$$

The AUC (6) can be estimated by the empirical area under the ROC curve, which can be shown to be equal to the Mann-Whitney statistic (a scaled version of the Wilcoxon rank sum test), which is given by:

$$\begin{aligned} \widehat{AUC} &= \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \psi\Big(\hat{h}(x_i \mid \omega_1), \hat{h}(x_j \mid \omega_2) \Big), \\ \psi(a, b) &= \begin{cases} 1 & a > b \\ 1/2 & a = b \\ 0 & a < b \end{cases} \end{aligned} \qquad (10)$$

2. MEAN AUC VS. DATA SET SIZE

To exhibit the basic structure of the problem under the practical limitation of a finite-training set, we carried out simulations inspired by Chan et al. [2] and the work of Fukunaga [3, 4]. In our simulation, we assume that the feature vector has the multinormal distribution with the following parameters: $\mu_1 = \underline{0}, \mu_2 = c\underline{1}$, and $\Sigma_1 = \Sigma_2 = \mathbf{I}$ where $\underline{0}$ is the vector all of whose components are zeros, $\underline{1}$ is the vector all of whose components are ones, $\mathbf{I}$ is the identity matrix, and c is a constant. A fundamental metric is the Mahalanobis distance between the mean vectors of the two classes: it is defined as:

$$\Delta = \left[(\mu_1 - \mu_2)'\Sigma^{-1}(\mu_1 - \mu_2)\right]^{1/2} \tag{11}$$

It expresses how these two vectors are separated from each other with respect to the spread Σ. In the simulation of the present example, the Mahalanobis distance is $c^2 p$. In this simulation, illustrated in Figure 3, the value c is adjusted for every dimensionality to obtain the same asymptotic AUC. This allows us to isolate the effect of the variation in training set sizes. Typically, the simulations described in this context used a value of 0.8 for Δ. For the time being, it is assumed that $n_1 = n_2 = n$, which is referred to as the training set size per class. For a particular dimensionality, and for particular data set size n, two training data sets are generated using the above parameters and distributions. When the classifier is trained, it will be tested on a pseudo-infinite test set, here 1000 cases per class, to obtain a very good approximation to the true AUC for the classifier trained on this very training data set; this is called a single realization or a Monte-Carlo (MC) trial.

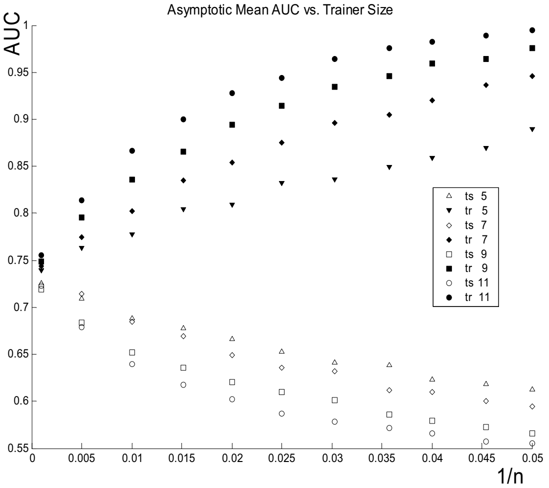

Figure 3. Mean AUC of the Bayes classifier. For every training sample size n, the classifier is tested on pseudo-infinite testers (represented as "ts") and tested as well on the same training sample (represented as "tr"). Each curve shows the average performance over 100 MC trials. The numbers in the legend are the dimensionalities of the feature vectors.

Many realizations of the training data sets with same n are generated over MC simulation to study the mean and variance of the AUC for the Bayes classifier under this training set size. The number of MC trials used is 100.

Several important observations can be made from these results. As was expected, for training size n the mean apparent AUC, i.e., coming from testing on the same training data set, is upwardly biased from the true AUC. It should be cautioned that this is on the average, i.e., over the population of all training sets; it is possible that for a single data set (single realization) the apparent performance can be better or worse than the true one. In addition, the classifier had the same asymptotic performance, approximately 0.74, for all dimensionalities in the simulation (by design as above).

3. NONPARAMETRIC INFERENCE FOR THE TRUE ERROR RATE

In the previous sections, it was assumed that there is a separate data set, i.e., the testing data set, to estimate the performance metric, either Err or AUC, using (8) or (10) respectively. In real life problems, there is scarcity of data, i.e., only a small training set size is available. Moreover, the data distribution is unknown, so that neither the formulas (4) or (6) can be used directly nor a testing data set can be simulated to assess the classification rule.

The conventional method for estimating the true error rate (1) is the so-called method of cross validation. The method relies on dividing the available data set into k-fold sub-sets with equal sizes. Training is carried out on all of them but one, on which testing is performed. This is carried out k times, each time one of the k subsets is used for testing, then the results from the k tests are averaged. A thorough discussion of the method can be found in [5]. An improvement on cross validation was produced by Efron [6] by proposing the .632 bootstrap estimator $\widehat{Err}^{(.632)}$ given by:

$$\widehat{Err}_{\mathbf{t}}^{.632} = .368\,\overline{Err}_{\mathbf{t}} + .632\,\widehat{Err}_{\mathbf{t}}^{(1)}, \tag{12}$$

where the estimator $\overline{Err}_{\mathbf{t}}$ is the apparent error rate obtained by testing the classifier on the same training data set. Formally:

$$\begin{aligned}\overline{Err}_{\mathbf{t}} &= E_{\hat{F}} L(y, \eta_{\mathbf{t}}(x)), \qquad (x,y) \in \mathbf{t} \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(I_{\hat{h}_{\mathbf{t}}(x_i|\omega_1)<th} + I_{\hat{h}_{\mathbf{t}}(x_i|\omega_2)>th}\right)\end{aligned} \tag{13}$$

The estimator $\widehat{Err}_{\mathbf{t}}^{(1)}$ is the leave-one-out bootstrap estimator calculated by:

$$\widehat{Err}_{\mathbf{t}}^{(1)} = \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{b=1}^{B} I_i^b L(y_i, \eta_{\mathbf{t}^{*b}}(x_i)) \Big/ \sum_{b'=1}^{B} I_i^{b'}\right] \tag{14}$$

where B bootstraps are replicated from the available data set and the classifier is trained on each. After each training, the classifier is tested only on those cases in the original data set that did not appear in the bootstrap replicate used for training. The notation I_i^b is used as an indicator function that equals one if the case sample t_i is not included in the bootstrap replicate b, and zero otherwise. The value .632 in (12) is the effective number of cases contained within the bootstrap replications. That means the .632 bootstrap trains on $.632n$ where n is the original data set size; the matter that makes it more biased than the cross validation.

Later in 1997, Efron and Tibshirani [7] proposed the .632+ bootstrap estimator that was considered to be a further improvement on the .632 estimator. It was designed to decrease the bias of the .632 bootstrap, which apparently exists in the case of an overtrained classifier, e.g., the 1-nearest-neighbor classifier. The .632+ estimator is given by

$$\widehat{Err}_{\mathbf{t}}^{(.632+)} = \widehat{Err}_{\mathbf{t}}^{(.632)} + (\widehat{Err}_{\mathbf{t}}^{(1)} - \overline{Err}_{\mathbf{t}})\frac{.368 \cdot .632 \cdot \hat{R}'}{1 - .368\hat{R}'} \quad (15)$$

where $\hat{R}'$ is the corrected relative over fitting, i.e., a modified measure of the relative over-fitting that is used to renormalize the factor 0.632 for that case (see [7] for details).

Efron and Tibshirani [7] carried out different experiments to compare different estimators of the error rate. These estimators include, among others, cross validation, $\widehat{Err}^{(1)}$, $\widehat{Err}^{(.632)}$, and $\widehat{Err}^{(.632+)}$. It was apparent that the .632+ is the winner in terms of the bias. In terms of the RMS error, the estimators were comparable with a little superiority for the .632+.

4. NONPARAMETRIC INFERENCE FOR THE AUC

In the present article, we extend the study carried out in [7] to include the AUC as the performance metric. Similar work has been done by considering the .632 bootstrap and the leave-one-out cross validation [8]

4.1. Mathematical Definitions

Analogously to $\widehat{Err}^{(1)}$, we can test on those cases that were not included in each bootstrap replicate and produce a single estimate of the AUC, then average over the whole set of bootstrap replicates. This gives the estimator

$$\widehat{AUC}_{\mathbf{t}}^{(*)} = \frac{1}{B}\sum_{b=1}^{B}\left[AUC_{\mathbf{t}^{*b}}(\hat{F}^{(*)})\right] = \frac{1}{B}\sum_{b=1}^{B}\frac{\sum_{j=1}^{n_2}\sum_{i=1}^{n_1} I_i^b I_j^b \psi(\hat{h}_{\mathbf{t}^*}(x_i), \hat{h}_{\mathbf{t}^*}(x_j))}{\sum_{i'=1}^{n_1} I_{i'}^b \sum_{j'=1}^{n_2} I_{j'}^b} \quad (16)$$

The AUC .632 estimator is defined analogously to $\widehat{Err}_{\mathbf{t}}^{(.632)}$ as:

$$\widehat{AUC}_{\mathbf{t}}^{.632} = .368\,\overline{AUC}_{\mathbf{t}} + .632\,\widehat{AUC}_{\mathbf{t}}^{(*)} \quad (17)$$

and the .632+ is defined by:

$$\widehat{AUC}_{\mathbf{t}}^{(.632+)} = \widehat{AUC}_{\mathbf{t}}^{(.632)} + (\widehat{AUC}_{\mathbf{t}}^{(*)\prime} - \overline{AUC}_{\mathbf{t}})\frac{.368 \cdot .632 \cdot \hat{R}'}{1 - .368\hat{R}'} \quad (18)$$

where

$$\widehat{AUC}_{\mathbf{t}}^{(*)\prime} = \max(\widehat{AUC}_{\mathbf{t}}^{(*)}, \gamma_{AUC}),$$
$$\hat{R}' = \begin{cases} (\widehat{AUC}_{\mathbf{t}}^{(*)} - \overline{AUC}_{\mathbf{t}})/(\gamma_{AUC} - \overline{AUC}_{\mathbf{t}}), & \\ \quad\quad if\, \overline{AUC}_{\mathbf{t}} > \widehat{AUC}_{\mathbf{t}}^{(*)} > \gamma_{AUC} & \\ 0 \quad\quad otherwise & \end{cases} \quad (19)$$

The no-information AUC γ_{AUC} can be shown to be equal to 0.5.

4.2. Experimental Results

We carried out different experiments to compare these three bootstrap-based estimators, considering different dimensionalities, different parameter values, and training set sizes, all based on the multinormal assumption for the feature vector. The experiments are described in Section 2. Here in this section we illustrate the results when the dimensionality was five. The number of trainer groups per point (the number of MC trials) is 1000 and the number of bootstraps is 100.

It is apparent from Figure 4 that the $\widehat{AUC}_{\mathbf{t}}^{(*)}$ is downward biased. This is a natural opposite of the upward bias observed in [7] when the metric was the true error rate as a measure of incorrectness, by contrast with the true AUC as a measure of correctness. The $\widehat{AUC}_{\mathbf{t}}^{(.632)}$ is designed as a correction for $\widehat{AUC}_{\mathbf{t}}^{(*)}$; it appears in the figure to correct for that but with an over-shoot. The correct adjustment for the remaining bias is almost achieved by the estimator $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$. The $\widehat{AUC}_{\mathbf{t}}^{(.632)}$ estimator can be seen as an attempt to balance between the two extreme biased estimators, $\widehat{AUC}_{\mathbf{t}}^{(*)}$ and $\overline{AUC}_{\mathbf{t}}$.

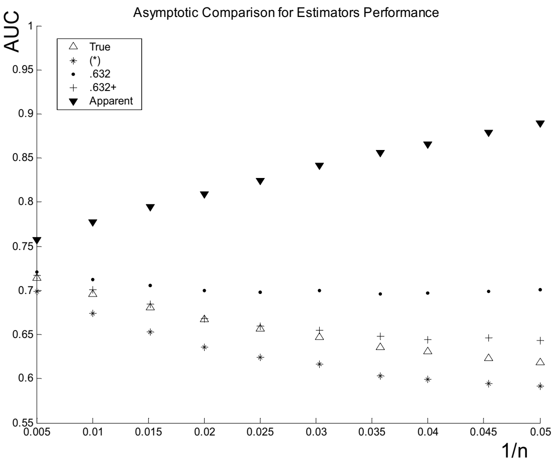

Figure 4. Comparison of the three bootstrap estimators, $\widehat{AUC}_{\mathrm{t}}^{(*)}$, $\widehat{AUC}_{\mathrm{t}}^{(.632)}$, and $\widehat{AUC}_{\mathrm{t}}^{(.632+)}$ for 5-feature predictor. The $\widehat{AUC}_{\mathrm{t}}^{(*)}$ is downward biased, while the $\widehat{AUC}_{\mathrm{t}}^{(.632)}$ is an over correction for that bias. $\widehat{AUC}_{\mathrm{t}}^{(.632+)}$ is almost the unbiased version of the $\widehat{AUC}_{\mathrm{t}}^{(.632)}$.

Estimator	Mean	SD	RMS	Size
AUC_{t}	.6181	.0434	.0434	20
$\widehat{AUC}_{\mathrm{t}}^{(*)}$	.5914	.0947	.0984	
$\widehat{AUC}_{\mathrm{t}}^{(.632)}$	.7012	.0749	.1119	
$\widehat{AUC}_{\mathrm{t}}^{(.632+)}$	.6431	.0858	.0894	
$\overline{AUC}_{\mathrm{t}}$	.8897	.0475	.2757	
AUC_{t}	.6571	.0308	.0308	40
$\widehat{AUC}_{\mathrm{t}}^{(*)}$	.6244	.0711	.0783	
$\widehat{AUC}_{\mathrm{t}}^{(.632)}$	.6981	.0598	.0725	
$\widehat{AUC}_{\mathrm{t}}^{(.632+)}$	.6595	.0739	.0739	
$\overline{AUC}_{\mathrm{t}}$	.8246	.0431	.1730	
AUC_{t}	.6965	.0158	.0158	100
$\widehat{AUC}_{\mathrm{t}}^{(*)}$	.6738	.0454	.0507	
$\widehat{AUC}_{\mathrm{t}}^{(.632)}$	.7119	.0399	.0428	
$\widehat{AUC}_{\mathrm{t}}^{(.632+)}$	.7004	.0452	.0453	
$\overline{AUC}_{\mathrm{t}}$	.7772	.0312	.0866	
AUC_{t}	.7141	.0090	.0090	200
$\widehat{AUC}_{\mathrm{t}}^{(*)}$	.6991	.0298	.0334	
$\widehat{AUC}_{\mathrm{t}}^{(.632)}$	.7205	.0272	.0279	
$\widehat{AUC}_{\mathrm{t}}^{(.632+)}$	.7170	.0285	.0286	
$\overline{AUC}_{\mathrm{t}}$	.7573	.0228	.0489	

Table 1. Comparison of the bias and variance for different bootstrap-based estimators of the AUC. The effect of the training set size is obvious in the variability of the true AUC.

Table 1 gives a comparison for the different estimators in terms of the RMS values. The RMS is defined in the present context as the root of the mean squared difference between an estimate and the population mean, i.e., the mean over all possible training sets.

4.3. Remarks

As shown by Efron and Tibshirani [7], the $\widehat{Err}_{\mathrm{t}}^{(1)}$ estimator is a smoothed version of the leave-one-out cross validation, since for every test sample case the classifier is trained on many bootstrap replicates. This reduces the variability of the cross-validation based estimator. On the other hand, the effective number of cases included in the bootstrap replicates is .632 of the total sample size n. This accounts for training on a less effective data set size; this makes the leave-one-out bootstrap estimator ($\widehat{Err}^{(1)}$) more biased than the leave-one-out cross-validation. This bias issue is observed in [8], as well, when the performance metric was the AUC. This fact is illustrated in Figure 5 for $\widehat{AUC}_{\mathrm{t}}^{(*)}$. At every sample size n the true value of the AUC is plotted. The estimated value $\widehat{AUC}_{\mathrm{t}}^{(*)}$ at data sizes of $n/.632$ and $n/.5$ are plotted as well. It is obvious that these values are lower and higher than the true value respectively, which supports the discussion of whether the leave-one-out bootstrap is supported on 0.632 of the samples or 0.5 of the samples (as mentioned in [7]) or, as here, something in-between.

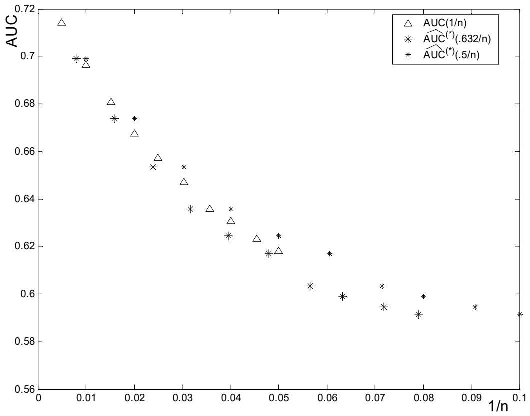

Figure 5.The true AUC and rescaled versions of the bootstrap estimator $\widehat{AUC}_{\mathrm{t}}^{(*)}$. At every sample size n the true AUC is shown along with the value of the estimator $\widehat{AUC}_{\mathrm{t}}^{(*)}$ at $n/.632$ and $n/.5$.

The estimators studied here are used to estimate the mean performance (AUC) of the classifier. However, the basic motivation for the $\widehat{AUC}_{\mathrm{t}}^{(.632)}$ and $\widehat{AUC}_{\mathrm{t}}^{(.632+)}$ is to estimate the AUC conditional on the given data set $\mathbf{t}$. This is the analogue of $\widehat{Err}_{\mathrm{t}}^{(.632)}$ and $\widehat{Err}_{\mathrm{t}}^{(.632+)}$.

Nevertheless, as mentioned in [7] and detailed in [9] the cross-validation, the basic ingredient of the bootstrap based estimators, is weakly correlated with the true performance on a sample by sample basis. This means that no estimator has a preference in estimating the conditional performance.

Work in Progress includes analysis of alternative expressions for the bootstrap estimators; an analysis of their smoothness properties; analysis of the correlation (or lack of it) referred to above; finite-sample estimates of the uncertainties of these estimators; and comparison of these results with the components-of-variance model of [10]

5. CONCLUSION

The bootstrap based estimators proposed in the literature are extended to estimate the AUC of a classifier. They have comparable performance in terms of the RMS of AUC, while the $\widehat{AUC}_{t}^{(.632+)}$ is the least biased.

6. ACKNOWLEDGMENT

This work was supported in part by a CDRH Medical Device Fellowship awarded to Waleed A. Yousef and administered by the Oak Ridge Institute for Science and Education (ORISE).

7. REFERENCES

[1] K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd ed. Boston: Academic Press, 1990.

[2] H. P. Chan, B. Sahiner, R. F. Wagner, and N. Petrick, "Classifier design for computer-aided diagnosis: Effects of finite sample size on the mean performance of classical and neural network classifiers," *Medical Physics*, vol. 26, pp. 2654-2668, 1999.

[3] K. Fukunaga and R. R. Hayes, "Effects of sample size in classifier design," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. - 11, pp. - 885, 1989.

[4] K. Fukunaga and R. R. Hayes, "Estimation of classifier performance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. - 11, pp. - 1101, 1989.

[5] M. Stone, "Cross-Validatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, pp. 111-147, 1974.

[6] B. Efron, "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation," *Journal of the American Statistical Association*, vol. 78, pp. 316-331, 1983.

[7] B. Efron and R. Tibshirani, "Improvements on Cross-Validation: The.632+ Bootstrap Method," *Journal of the American Statistical Association*, vol. 92, pp. 548-560, 1997.

[8] B. Sahiner, H. P. Chan, N. Petrick, L. Hadjiiski, S. Paquerault, and M. Gurcan, (University of Michigan, Department of Radiology Medical Imaging Research Group), "Resampling Schemes for Estimating the Accuracy of a Classifier Designed with a Limited Data Set," *Medical Image Perception Conference IX, Airlie Conference Center, Warrenton VA, 20-23*, September 2001.

[9] P. Zhang, "Assessing Prediction Error In Nonparametric Regression," *Scandinavian Journal Of Statistics*, vol. 22, pp. 83-94, 1995.

[10] S. V. Beiden, M. A. Maloof, and R. F. Wagner, "A general model for finite-sample effects in training and testing of competing classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. - 25, pp. - 1569, 2003.