Exact Bootstrap Distributions of Cost Curves

Keywords: cost curves, roc curves, bootstrap, coverage probabilities, model selection

Abstract

Cost curves have recently been introduced as an alternative or complement to ROC curves in order to visualize binary classifiers performance. Of importance to both cost and ROC curves is the computation of confidence intervals along with the curves themselves so that the reliability of a classifier's performance can be assessed. Computing confidence intervals for the difference in performance between two classifiers allows to determine whether one classifier performs *significantly* better than another. A simple procedure to obtain confidence intervals for costs or the difference between two costs, under various *operating conditions*, is to perform bootstrap resampling of the testset. In this paper, we derive *exact* bootstrap distributions, i.e. distributions obtained when an infinite number of bootstrap samples are drawn, for the cost of using a classifier as well as the difference between the costs of two classifiers. These distributions are then used to obtain confidence intervals, under various operating conditions. Performances of these confidence intervals are measured in terms of coverage accuracies. Simulations show excellent results.

1. Introduction

Cost curves (Drummond & Holte, 2000; Drummond & Holte, 2006) have recently been introduced as an alternative or complement to receiver operating characteristics (ROC) curves in order to visualize binary classifiers performance. A cost curve is a plot of a classifier's expected cost as a function of operating conditions, i.e. misclassification costs and class probabilities. Performance assessment in terms of expected cost is paramount but can hardly be visualized through ROC analysis although knowledge of the distribution of a classifier's total misclassification error cost is often among the enduser's interests.

Cost curve analysis can be enhanced if dispersion measures of the curve are provided along with the curve itself, thereby allowing the enduser to assess the reliability of the estimated performance of the classifier considered for implementation. In order to obtain confidence intervals from a single testset, resampling methods such as the bootstrap (Efron & Tibshirani, 1993) technique can be used: from the testset, a certain number of samples are drawn, with remplacement, and from these samples, a distribution of the cost can be obtained. In certain cases, the bootstrap technique lends itself to analytic derivations for the limit case where the number of samples tends to infinity. Distributions thus obtained are referred to as *exact bootstrap* distributions. The purpose of this paper is to derive exact bootstrap distributions for a classifier's total cost of misclassification errors as well as the difference between two classifiers' total costs, for varying operating conditions.

Except for (Drummond & Holte, 2006), little attention has been given to developing and evaluating the performance of confidence intervals for cost curves. In that regard, ROC curves have received much more attention. Arguably, the recency of cost curves explains in part this situation. Recent literature on the derivations of confidence intervals for ROC curves can be segmented in three categories: parametric, semi-parametric or empirical. Semi-parametric methods mainly refer to kernel-based methods (Hall & Hyndman, 2003; Hall et al., 2004; Lloyds, 1998; Lloyds & Wong, 1999). Bootstrap resampling has been used for ROC curves as an empirical method but to date, exact bootstrap distributions for the ROC curve have not been presented.

A technical difficulty arises from the fact that, when sampling from the entire testset, a procedure we shall refer to as *full* sampling, relative proportions of classes will vary from one sample to another. Mathematical derivations of exact bootstrap distributions, in

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

the context of full sampling, are thus more cumbersome. In this paper, we first circumvent this difficulty through the use of a procedure referred to as *stratified sampling* according to which proportions of positive and negative instances of each bootstrap sample are fixed as equal to those of the original testset. Here, an *instance* is an element of the testset. Instances of the class for which the event has (not) taken place are called *positive* (*negative*). For example, for a credit card fraud detection application, fraudulous transactions would be labelled as positive whereas legitimate transactions would be labelled as negative. Within the stratified sampling framework, each sample is obtained from the combination of two independent bootstrap samples: one drawn from the set of positive instances and the other drawn from the set of negative instances. This procedure has previously been used in the context of ROC (Bandos, 2005) as well as cost (Drummond & Holte, 2006) curves. After obtaining results under this simplified stratified sampling approach, we build on these results and derive exact bootstrap distributions for the full sampling approach.

Cost curves are obtained assuming the enduser selects the threshold that minimizes expected cost, given operating conditions, *based on the testset*. One approach to obtain cost distributions is to draw bootstrap samples from the testset, obtain a cost curve for each of the samples and derive a distribution for the cost from these cost curves. Now consider a specific set of values for the operating conditions. Each of the samples will lead a possibly different optimal threshold for this set of operating conditions. Thus, averaging cost curves (fixed operating conditions but varying thresholds) in order to obtain an estimate of the expected cost would correspond to the enduser being able to select the optimal thresholds, depending on the actually observed sample of instances. In other words, the enduser would be required to have knowledge of the testset *before* deciding on a threshold value, something that can't be done. Obviously, thresholds must somehow be selected *prior* to testset cost measurements. This can be done through the standard machine learning process of splitting the data in three sets: training, validation and test. In our simulations, we first assume the user chooses the optimal *theoretical* thresholds for all operating conditions, thus implicitely assuming an infinite sized validation set. Our approach can therefore be considered as a form of threshold averaging of the costs. But since both operating conditions (abscissa values) and thresholds are fixed for each computed distribution, then the approach could be considered as vertical averaging as well.

The rest of the paper is as follows: in section 2, we briefly review the main aspects of ROC and cost curves. Then, mathematical derivations are presented in section 3 for stratified sampling and in section 4 for full sampling. In section 5, we perform simulations and measure coverage accuracies of the confidence intervals. Finally, we conclude in section 6.

2. ROC and cost curves

There are various alternatives for assessing and comparing the predictive performance of binary classifiers. ROC and cost curves are among these. ROC curves have been used in the context of numerous real-world applications (Swets & Pickett, 1982; Swets et al., 2000). An ROC curve is a plot of the probability of correctly identifying a positive instance (a true positive) against the probability of mistakenly identifying a negative instance as positive (a false positive), for various threshold values. When viewed as a test, it corresponds to a plot of sensitivity against $(1 -$ specificity) or power against size. See (Fawcett, 2006a) for an excellent introduction to ROC curves along with descriptions of the essential elements of ROC graph analysis. When obtained through the use of a classifier for scoring instances of a testset, the curve is often referred to as the *empirical ROC*.

An advantage of ROC graph analysis lies in the fact that ROC curves are independent of the relative proportions of positive and negative instances in the population as well as the relative values of error costs (Fawcett, 2006a). Since both axes are scaled as proportions to the total number of positive (y-axis) and negative (x-axis) instances, a change in these numbers should not affect the ROC curve (although this argument has recently been discussed (Webb & Ting, 2005; Fawcett & Flach, 2005)). Since all computations of ROC curves are made independently of cost values, these have no effect on the curve. On the other hand, changes in costs or proportions will cause changes in the value of the optimal (expected total error cost minimizing) threshold, corresponding to a different point on the otherwise unaffected ROC curve. One drawback of ROC graph analysis, is its inability to handle instance-varying benefits (or costs) but an extension has recently been proposed for that purpose (Fawcett, 2006b). Classifier performance assessment in terms of expected total error cost can hardly be done using ROC curves and for this reason (and others (Drummond & Holte, 2006)), cost curves have been introduced as an alternative (or a complement) to ROC curves.

The main objective of cost curves is to visualize classifier performance in terms of expected cost rather than through a tradeoff between misclassification error probabilities. Expected cost is plotted against operating conditions where, as mentionned above, operating conditions include two factors: class probabilities and misclassification costs. Once these values are fixed, all possibly attainable true and false positive rates pairs are considered. Given class probabilities, misclassification costs, and true and false positive rates, a cost is obtained. The pair that minimizes the cost is selected. It is assumed that given certain operating conditions, the enduser would select the cost minimizing pair and set the classifier's threshold accordingly. This process is done for all possible values for operating conditions in order to obtain a cost curve. As shown below, a set of operating conditions can be summarized through a single normalized scalar value ranging between 0 and 1. We now turn to more formal derivations of the cost curves and associated exact bootstrap distributions.

3. Stratified sampling

Consider a testset consisting of n instances from which stratified bootstrap samples are drawn. In this paper, we shall assume bootstrap samples are of the same size as the testset itself, a common procedure. Let n^+ and n^- be the numbers of positive and negative instances in the testset. According to the stratified bootstrap procedure and since we assume sample size equals testset size, the numbers of sampled positive and negative instances are fixed for all samples and also equal to n^+ and n^-, respectively. Let n_t^+ denote the number of instances, among the n^+ positive instances of the testset, with score greater or equal to the threshold $t = t(w)$ associated to operating conditions w, where w will be defined shortly. The corresponding value for a set of sampled positive instances is noted N_t^+ and follows binomial distribution with parameters n_t^+/n^+ and n^+ which we note as $N_t^+ \sim \mathrm{Bin}(n_t^+/n^+, n^+)$. The random variable for the true positive rate, at threshold t, is denoted $TP_t^+ = N_t^+/n^+$. Similarly for negative instances, n_t^- refers to the number of instances with score greater or equal to t among the n^- negative instances of the testset, N_t^- is the random variable for the corresponding number of sampled instances and $FP_t^- = N_t^-/n^-$ is the false positive rate, at threshold t, with $N_t^- \sim \mathrm{Bin}(n_t^-/n^-, n^-)$. Note that, according to the stratified sampling procedure, samples from positive and negative instances are drawn independently so that TP_t^+ and FP_t^- are independent as well.

Let us now formalize the above mentionned operating conditions and define w. Let p_+ and p_- represent class probabilities for positive and negative instances, respectively. Misclassification costs are noted $c_{+/-}$ and $c_{-/+}$ for false positive and false negative errors, respectively. Total cost is therefore given by the following:

$$C_t^T = p_+ c_{-/+}(1 - TP_t^+) + p_- c_{+/-} FP_t^-. \quad (1)$$

In (Drummond & Holte, 2006), total cost is divided by the maximum possible value, in order to obtain a normalized cost with maximum value of one. This maximum total cost value is reached when $1 - TP_t^+ = FP_t^- = 1$ and the total cost is then equal to $p_+ c_{-/+} + p_- c_{+/-}$. Defining w as

$$w = \frac{p_+ \cdot c_{-/+}}{p_+ \cdot c_{-/+} + p_- \cdot c_{+/-}}, \quad (2)$$

the normalized cost is given by

$$C_t^N = w(1 - TP_t^+) + (1 - w) FP_t^- \quad (3)$$

with $w \in [0, 1]$. Considering the distribution of all possible stratified bootstrap samples, the expected value and variance of C_t^N are

$$\begin{aligned} E[C_t^N] &= w(1 - n_t^+/n^+) + (1 - w) n_t^-/n^-. && (4) \\ V[C_t^N] &= w^2 n_t^+/n^+(1 - n_t^+/n^+) \\ &\quad + (1 - w)^2 n_t^-/n^-(1 - n_t^-/n^-). && (5) \end{aligned}$$

We use these expectation and variance of the distribution of C_t^N to fit a gaussian distribution from which confidence intervals are easily obtained. Now, in order to assess the statistical significance of the difference in performance of two classifiers, we need to obtain the distribution of the difference in their normalized costs:

$$\begin{aligned} \Delta C_{t_1,t_2}^N &= C_{t_2}^N - C_{t_1}^N \\ &= w(TP_{t_1}^+ - TP_{t_2}^+) \\ &\quad + (1 - w)(FP_{t_2}^- - FP_{t_1}^-) && (6) \end{aligned}$$

where we use subscripts 1 and 2 to differentiate values obtained for the two classifiers. The values of $C_{t_2}^N$ and $C_{t_1}^N$ can hardly be assumed independent since it is possible that the scores assigned by two different classifiers are correlated: for example, obvious fraudulous transactions will likely obtain high scores on all classifiers. Also note that only instances that are falsely labelled by one and only one of the two classifiers will affect the difference in costs. Errors made by both classifiers will offset each other when computing cost differences. Let $n_{t_1}^+$ represent the number of positive testset instances labelled as positive by the first classifier and negative by the second classifier, given operating conditions w. Note that thresholds $t_1 = t_1(w)$ and $t_2 = t_2(w)$ associated to operating conditions w may differ from one classifier to the other since score distributions and scales may vary from one classifier to the

Algorithm 1 Computing $n^+_{t_1}$ and $n^+_{t_2}$ in $O(n)$

The following are obtained through sorting:
- $s_1(j)$: j^{th} largest score, according to classifier 1. The set of values $s_1(j)$ forms table 1.
- $r_1(j)$: rank, in table 2, for the j^{th} instance of table 1.
- $f_1(j)$: flag associated to j^{th} instance of table 1. Initially set to zero.

$s_2(k)$, $r_2(k)$, and $f_2(k)$ are defined similarly.

```
n_1 ← 0, n_2 ← 0
j ← 1, k ← 1
for all values of w (sorted in increasing order) do
  while j ≤ n^+ and s_1(j) ≥ t_1(w) do
    f_1(j) ← 1
    if f_2(r_1(j)) == 0 then
      n_1 ← n_1 + 1
    else
      n_2 ← n_2 − 1
    end if
    j ← j + 1
  end while
  while k ≤ n^+ and s_2(k) ≥ t_2(w) do
    f_2(k) ← 1
    if f_1(r_2(k)) == 0 then
      n_2 ← n_2 + 1
    else
      n_1 ← n_1 − 1
    end if
    k ← k + 1
  end while
  n^+_{t_1} ← n_1
  n^+_{t_2} ← n_2
end for
```

other. Similarly, let $n^+_{t_2}$ represent the number of positive testset instances labelled as positive by classifier 2 and negative by classifier 1. Values $n^-_{t_1}$ and $n^-_{t_2}$ are defined similarly for negative instances. Let $N^+_{t_1}$, $N^+_{t_2}$, $N^-_{t_1}$, and $N^-_{t_2}$ be the associated random variables for the number of instances in a bootstrap sample. Values $N^+_{t_1}$ and $N^+_{t_2}$ jointly follow a multinomial distribution. This also applies to $N^-_{t_1}$ and $N^-_{t_2}$. Accordingly, moments of $\Delta C^N_{t_1,t_2}$ are easily obtained:

$$
\begin{aligned}
E[\Delta C^N_{t_1,t_2}] &= w\left(\frac{n^+_{t_1}-n^+_{t_2}}{n^+}\right) \\
&+ (1-w)\left(\frac{n^-_{t_2}-n^-_{t_1}}{n^-}\right) \qquad (7)\\
V[\Delta C^N_{t_1,t_2}] &= w^2\left(\frac{n^+_{t_1}+n^+_{t_2}-\frac{(n^+_{t_1}-n^+_{t_2})^2}{n^+}}{(n^+)^2}\right) \\
&+(1-w)^2\left(\frac{n^-_{t_1}+n^-_{t_2}-\frac{(n^-_{t_1}-n^-_{t_2})^2}{n^-}}{(n^-)^2}\right). \qquad (8)
\end{aligned}
$$

Let us now evaluate the computational time required to obtain confidence intervals for the performance of a single classifier and for the difference between the performances of two classifiers. Here, we assume the number of different operating conditions considered, i.e. the number of different values for w is proportional to n. Also, as explained above, we assume the thresholds associated to each of these operating conditions have previously been determined through a validation process. For the case of a single classifier performance, we first need to sort instances with respect to their score, which requires time $O(n \ln n)$. Then, values of n^+_t and n^-_t are easily obtained in linear time. There remains to compute expectations and variances, using equations (4) and (5), and derive confidence intervals using these values. This is realized in constant time for each value of w, thus overall linear time. Globally, the entire process is therefore dominated by the sorting phase and total computational time is $O(n \ln n)$. Confidence intervals for the difference in performance between two classifiers can be obtained in $O(n \ln n)$ computational time as well, although less trivially. Naive solutions lead to quadratic time but, given careful sorting preprocessing, values $n^+_{t_1}, n^+_{t_2}, n^-_{t_1}$, and $n^-_{t_2}$ are computed in linear time. Algorithm 1 describes how $n^+_{t_1}$ and $n^+_{t_2}$ can be computed. Values $n^-_{t_1}$ and $n^-_{t_2}$ are computed similarly. Then, moments and confidence intervals for $\Delta C^N_{t_1,t_2}$ are obtained in linear time (for all values of w) using equations (7) and (8).

4. Full sampling

Within the framework of full sampling, the proportions of positive and negative instances vary from one sample to another. Whereas with stratified sampling, the number of positive and negative instances in each sample, n^+ and n^-, were set as equal to those of the testset, we now consider these numbers as random variables, and accordingly use capital notation N^+ and N^-. Here again, these values follow binomial distributions: $N^+ \sim Bin(n^+/n, n)$. Thus, full sampling implicitly assumes a binomial distribution for the observed class proportions $P_+ = N^+/n$ and $P_- = N^-/n$. If the enduser has some reason to believe that, once the system is deployed, the distribution of the proportions of positive and negative instances should differ from the binomial distribution derived from the testset, then the distribution for N^+ above could be replaced with a more appropriate one. Such a situation could occur because of known trends in the evolution of the proportions or because the testset was taken from a subset of the entire population with one of the two classes being relatively overrepresented in the testset.

Equation (1) still holds in the case of full sampling, but with the difference that P_+ and P_- are now treated as random variables. In the previous section the normalized version of the total cost was obtained by dividing the total cost by the largest possible cost: $p_+c_{-/+}+p_-c_{+/-}$, a weighted average between misclassification costs $c_{-/+}$ and $c_{+/-}$. Since P_+ and P_- are no longer fixed, we must consider the largest possible weighted average which simply is the maximum of the two misclassification costs, $c_{\max} = \max[c_{-/+}, c_{+/-}]$, and is obtained when either $(P_+, P_-) = (1, 0)$ or $(P_+, P_-) = (0, 1)$. Thus, for full sampling, the normalized cost can be written as

$$C_t^N = \frac{N^+ \cdot c_{-/+} \cdot (1 - TP_t^+) + N^- \cdot c_{+/-} \cdot FP_t^-}{n \cdot c_{\max}}.$$

Then, expected normalized cost and normalized cost variance are obtained through iterated expectations:

$$\begin{aligned} E[C_t^N] &= E_{N^+}\{E[C_t^N|N^+]\} \\ &= \frac{c_{-/+}(n^+ - n_t^+) + c_{+/-} \cdot n_t^-}{n \cdot c_{\max}} \end{aligned} \tag{9}$$

$$\begin{aligned} V[C_t^N] &= V_{N^+}\{E[C_t^N|N^+]\} + E_{N^+}\{V[C_t^N|N^+]\} \\ &= \frac{c_{-/+}^2\alpha_t^+ + c_{+/-}^2\alpha_t^- + \delta_t^2}{(n \cdot c_{\max})^2} \end{aligned} \tag{10}$$

where

$$\begin{aligned} \alpha_t^+ &= n_t^+ - \frac{(n_t^+)^2}{n^+} \\ \alpha_t^- &= n_t^- - \frac{(n_t^-)^2}{n^-} \\ \delta_t^2 &= (c_{-/+}(1 - n_t^+/n^+) - c_{+/-} \cdot n_t^-/n^-)^2 \frac{n^+ \cdot n^-}{n} \end{aligned}$$

Here again, equations (9) and (10) can be used to obtain a fitted gaussian distribution for which confidence intervals are easily derived.

Let us now turn to the difference in performance between two classifiers. In the case of full sampling, this difference is

$$\Delta C_{t_1,t_2}^N = \frac{c_{-/+}(N_{t_1}^+ - N_{t_2}^+) + c_{+/-}(N_{t_2}^- - N_{t_1}^-)}{n \cdot c_{\max}} \tag{11}$$

Again, expected normalized cost and normalized cost variance are obtained through iterated expectations:

$$\begin{aligned} E[\Delta C_{t_1,t_2}^N] &= E_{N^+}\{E[\Delta C_{t_1,t_2}^N|N^+]\} \\ &= \frac{c_{-/+}(n_{t_1}^+ - n_{t_2}^+) + c_{+/-} \cdot (n_{t_2}^- - n_{t_1}^-)}{n \cdot c_{\max}} \end{aligned} \tag{12}$$

$$\begin{aligned} V[\Delta C_{t_1,t_2}^N] &= V_{N^+}\{E[\Delta C_{t_1,t_2}^N|N^+]\} \\ &\quad + E_{N^+}\{V[\Delta C_{t_1,t_2}^N|N^+]\} \\ &= \frac{c_{-/+}^2\alpha_{t_1,t_2}^+ + c_{+/-}^2\alpha_{t_1,t_2}^- + \delta_{t_1,t_2}^2}{(n \cdot c_{\max})^2} \end{aligned} \tag{13}$$

where

$$\begin{aligned} \alpha_{t_1,t_2}^+ &= n_{t_1}^+ + n_{t_2}^+ - \frac{(n_{t_1}^+ - n_{t_2}^+)^2}{n^+} \\ \alpha_{t_1,t_2}^- &= n_{t_1}^- + n_{t_2}^- - \frac{(n_{t_1}^- - n_{t_2}^-)^2}{n^-} \\ \delta_{t_1,t_2}^2 &= \left(c_{-/+}\frac{n_{t_1}^+ - n_{t_2}^+}{n^+} - c_{+/-}\frac{n_{t_2}^- - n_{t_1}^-}{n^-}\right)^2 \frac{n^+ \cdot n^-}{n} \end{aligned}$$

This completes mathematical derivations. A total of four distributions have been obtained. For all four distributions, computation of confidence intervals is dominated by the need to sort instances so that computational time is $O(n \ln n)$ in all cases. Note that such time efficiency is obtained because we rely on the gaussian fitting of the variables' distributions. Computing true exact bootstrap distributions would lead to higher computational time orders. But as we show in the next section, results obtained with gaussian fitting are already excellent.

5. Numerical results

In this section, we conduct a series of experiments in order to assess the performance of the confidence intervals derived in sections 3 and 4. Performance is measured in terms of coverage accuracy of confidence intervals.

The first experiment is based on the framework used in (Macskassy et al., 2005) in which four methods for obtaining pointwise confidence intervals for ROC curves are compared: threshold averaging, vertical averaging, kernel smoothing (Hall et al., 2004) and Working-Hotelling bounds. In (Macskassy et al., 2005), positive and negative instance scores follow normal distributions but with various parameter values. Such a pair of normal distributions is often referred to as a binormal distribution. We set the scale parameter to 3.00 for both positive and negative instances scores. The location parameter θ for positive instances varies within the set $\{0.75, 1.5, 3.0, 5.0\}$ and the location parameter

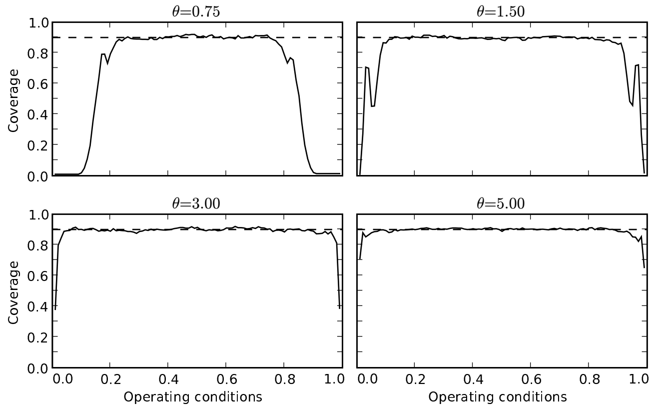

Figure 1. Effect of spread between distributions on coverage. Stratified sampling is used. Confidence intervals are derived for a classifier's cost. Location parameter for positive instances is set equal to 0.75 (up and left), 1.50 (up and right), 3.00 (down and left), and 5.00 (down and right). Sample size is 1,000. Confidence intervals are built with significance level $\alpha = 10\%$. Coverage proportion (solid) for 1,000 simulations and target coverage of 90% (dashed) are plotted against operating conditions.

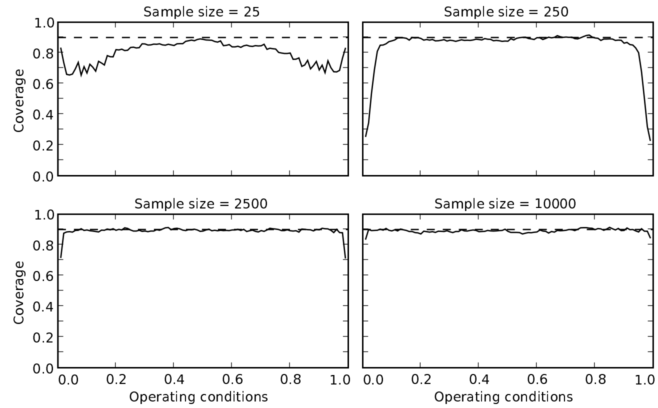

Figure 2. Effect of sample size on coverage. Stratified sampling is used. Confidence intervals are derived for a classifier's cost. Sample sizes of 25 (up and left), 250 (up and right), 2,500 (down and left), and 1,000, (down and right) are considered. Confidence intervals are built for significance level $\alpha = 10\%$. Location parameter for positive instances is set to $\theta = 3.0$. Coverage proportion (solid) for 1,000 simulations and target coverage of 90% (dashed) are plotted against operating conditions.

for negative instances is set equal to $-\theta$. Sample size is set to 1,000, i.e. a set of 1,000 instances is drawn from the positive instances distribution and another set of 1,000 negative instances is drawn from the negative instances distribution. The sampling procedure is repeated 1,000 times, i.e. 1,000 simulations are performed for each value of θ. We shall refer to this experiment as the *spread* experiment. Confidence intervals are obtained for a significance level of 10%.

Figure 1 provides simulation results which clearly show that better results are obtained when score distributions of positive and negative instances have few overlap, i.e. for high values of θ. For values of w close to zero or one variance estimates are unreliable since some of the values used to compute them are close to zero.

As a second experiment, we consider the effect of sample size on coverage accuracy. This experiment is everywhere similar to the previous one except for two modifications: (1) the location parameter not longer varies: it is set to $\theta = 3.0$ and (2) the sample size takes values in $\{25; 250; 2,500; 10,000\}$ instead of being fixed at 1,000. We shall refer to this experiment as the *size* experiment. Simulation results appear in Figure 2. As the sample size increases, the range of operational condition values with good coverage accuracy widens. For sample sizes of 25, only a very narrow range of operational condition values lead to a coverage rate that is on target.

Our third experiment addresses the modeling of the difference in performance between two classifiers. The experiment design is similar to the ones used for the previous two experiments, i.e. the spread and size experiments. Scores are distributed according to a binormal distribution with scale of 3.00 Confidence intervals are obtained for a significance level of $\alpha = 10\%$. The location parameters are set as follows: for positive instances of the first classifier, we consider two values: $\theta \in \{1.0, 3.0\}$. For negative instances of both classifiers the parameter is set equal to $-\theta$. Finally, for positive instances of the second classifier we consider three values: θ, $\theta + 2.0$ and $\theta + 4.0$. The difference between the location parameters of the two classifiers' positive instances distributions, either 0.0, 2.0 or 4.0, is referred to as the *shift* parameter. In order to include some form of dependency between the scores of the two classifiers, three values of a correlation factor are considered: $\rho \in \{0.3, 0.6, 0.9\}$. We shall refer to this experiment as the *difference* experiment.

Results appear in Figure 3. As in the previous two experiments, coverage accuracy breaks for very low or high total positive rates. Comparing curves on the left of Figure 3 with those on the right, we see the spread parameter θ has some impact: higher values of θ cause the range of total positive rate values with good coverage accuracy to widen. With $\theta = 1.0$, higher

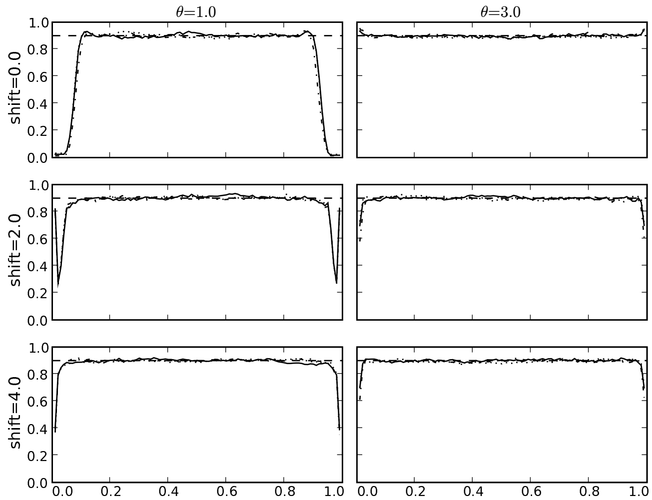

Figure 3. Coverage accuracy of confidence intervals for the difference in performance between two classifiers. Stratified sampling is used. Sample size is 1000, and significance level is $\alpha = 10\%$. Location parameter for positive instances of first classifier is set to $\theta = 1.0$ (left) and $\theta = 3.0$ (right). Location parameter for the score of positive instances according to the second classifier is θ (top), $\theta + 2.00$ (middle) and $\theta + 4.00$ (bottom). Within each plot, correlation factor is equal to 0.3 (dotted), 0.6 (dash-dotted) and 0.9 (solid). Coverage proportions for 1,000 simulations and target coverage of 90% (dashed) are plotted against operating conditions.

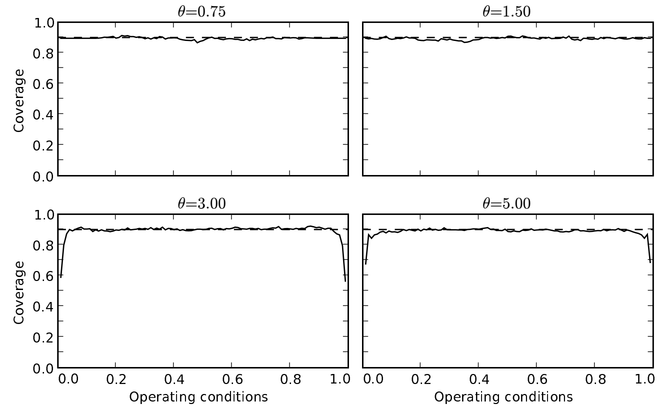

Figure 4. Effect of spread between distributions on coverage. Full sampling is used. Confidence intervals are derived for a classifier's cost. Location parameter for positive instances is set equal to 0.75 (up and left), 1.50 (up and right), 3.00 (down and left), and 5.00 (down and right). Sample size is 1,000. Confidence intervals are built with significance level $\alpha = 10\%$. Coverage proportion (solid) for 1,000 simulations and target coverage of 90% (dashed) are plotted against operating conditions.

shift parameter values lead to better coverage accuracy whereas with $\theta = 3.0$, the shift parameter has the opposite, but less pronounced, effect. The correlation coefficient seems to have very little effect on coverage accuracy which is a welcome property: the performances of the confidence intervals seem independent of the level of correlation between the scores of two models.

Figure 4 and 5 repeat the spread (first) and diffrence (third) experiments described above, but with the use of full sampling. Looking at Figure 4, it it clear that full sampling leads to better coverage accuracy than stratified sampling for low values of the spread parameter ($\theta = 0.75$). In fact, the effect of the spread parameter seems to have reversed although performance at $\theta = 5.00$ is better than with $\theta = 3.0$. Finally, Figure 5 indicates that both stratified and full sampling perform equally well for modeling the difference between two classifiers' performances.

6. Conclusion

In this paper, we have derived exact bootstrap distributions for the (normalized) cost of the misclassification errors of a classifier's decisions. We have also derived exact bootstrap distributions for the difference between the costs of two classifiers. The first and second moments of these distributions have been used to fit gaussian distributions and thus approximate the true exact bootstrap distributions. From these approximated distributions, we were able to obtain confidence intervals for the variables of interest. Table 1 summarizes these results. All confidence intervals can be derived in $O(n \ln n)$ time.

Results obtained in this paper are excellent but limited to a few simulations. In a few cases, severe breaks in coverage accuracy appear when operating conditions take extreme values (0 or 1). Future work includes investigating when these breaking patterns appear so that other techniques could be used in place of the suggested exact bootstrap approach if conditions indicate that a particular testset lies within a region of poor performance. Another possibility would be to combine approaches for such cases.

References

Bandos, A. (2005). *Nonparametric methods in comparing two correlated roc curves*. Doctoral disserta-

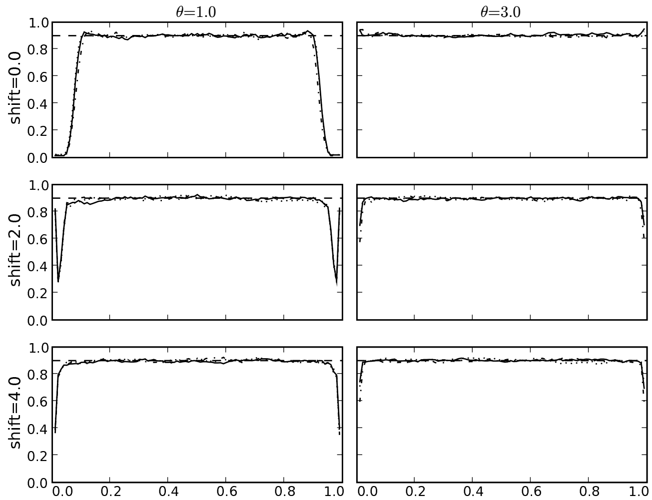

Figure 5. Coverage accuracy of confidence intervals for the difference in performance between two classifiers. Full sampling is used. Sample size is 1000, and significance level is $\alpha = 10\%$. Location parameter for positive instances of first classifier is set to $\theta = 1.0$ (left) and $\theta = 3.0$ (right). Location parameter for the score of positive instances according to the second classifier is θ (top), $\theta + 2.00$ (middle) and $\theta + 4.00$ (bottom). Within each plot, correlation factor is equal to 0.3 (dotted), 0.6 (dash-dotted) and 0.9 (solid). Coverage proportions for 1,000 simulations and target coverage of 90% (dashed) are plotted against operating conditions.

Sampling	Variable	Equations	Figures
Stratified	C_t^N	(4), (5)	1,2
	$\Delta C_{t_1,t_2}^N$	(7), (8)	3
Full	C_t^N	(9), (10)	4
	$\Delta C_{t_1,t_2}^N$	(12), (13)	5

Table 1. Summary of the paper's main results.

tion, Graduate School of Public Health, University of Pittsburgh.

Drummond, C., & Holte, R. (2000). Explicitly representing expected cost: an alternative to roc representation. *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 198–207). ACM.

Drummond, C., & Holte, R. (2006). Cost curves: an improved method for visualizing classifier performance. *Machine Learning*, *65*, 95–130.

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. No. 57 in Monographs on Statistics and Probability. Chapman & Hall.

Fawcett, T. (2006a). An introduction to roc analysis. *Pattern Recognition Letters*, *27*, 861–874.

Fawcett, T. (2006b). Roc graphs with instance varying costs. *Pattern Recognition Letters*, *27*, 882–891.

Fawcett, T., & Flach, A. (2005). A response to webb and ting's on the application of roc analysis to predict classification performance under varying class distributions. *Machine Learning*, *58*, 33–38.

Hall, P., & Hyndman, R. (2003). Improved methods for bandwidth selection when estimating roc curves. *Statistics & Probability Letters*, *64*, 181–189.

Hall, P., Hyndman, R., & Fan, Y. (2004). Nonparametric confidence intervals for receiver operating characteristic curves. *Biometrika*, *91*, 743–750.

Lloyds, C. (1998). The use of smoothed roc curves to summarise and compare diagnostic systems. *Journal of the American Statistical Association*, *93*, 1356–1364.

Lloyds, C., & Wong, Z. (1999). Kernel estimators of the roc curve are better than empirical. *Statistics & Probability Letters*, *44*, 221–228.

Macskassy, S., Provost, F., & Rosset, S. (2005). Pointwise roc confidence bounds: An empirical evaluation. *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning*. Bonn, Germany.

Swets, J., Dawes, R., & Monahan, J. (2000). Better decisions through science. *Scientific American*, *283*, 82–87.

Swets, J., & Pickett, R. (1982). *Evaluation of diagnostic systems: Methods from signal detection theory*. Academic Press.

Webb, G., & Ting, K. (2005). On the application of roc analysis to predict classification performance under varying class distributions. *Machine Learning*, *58*, 25–32.