Available online at www.sciencedirect.com
ScienceDirect
Neural Networks 21 (2008) 476–483
Neural Networks
www.elsevier.com/locate/neunet

2008 Special Issue

Classifier performance estimation under the constraint of a finite sample size: Resampling schemes applied to neural network classifiers☆

Berkman Sahiner*, Heang-Ping Chan, Lubomir Hadjiiski

Department of Radiology, University of Michigan, Ann Arbor, United States

Received 10 August 2007; received in revised form 5 December 2007; accepted 11 December 2007

Abstract

In a practical classifier design problem the sample size is limited, and the available finite sample needs to be used both to design a classifier and to predict the classifier's performance for the true population. Since a larger sample is more representative of the population, it is advantageous to design the classifier with all the available cases, and to use a resampling technique for performance prediction. We conducted a Monte Carlo simulation study to compare the ability of different resampling techniques in predicting the performance of a neural network (NN) classifier designed with the available sample. We used the area under the receiver operating characteristic curve as the performance index for the NN classifier. We investigated resampling techniques based on the cross-validation, the leave-one-out method, and three different types of bootstrapping, namely, the ordinary, .632, and .632+ bootstrap. Our results indicated that, under the study conditions, there can be a large difference in the accuracy of the prediction obtained from different resampling methods, especially when the feature space dimensionality is relatively large and the sample size is small. Although this investigation is performed under some specific conditions, it reveals important trends for the problem of classifier performance prediction under the constraint of a limited data set.

Keywords: Performance estimation; Finite sample size; Resampling

1. Introduction

Classification of a set of vectors into two classes is a common application for neural networks (NNs). In practice, the NN classifier is designed using a finite training set, which may not fully represent the true distribution of the two classes in the general population. Finite training sample size presents two problems. First, one has to pay a penalty for the finite sample size, i.e., the classifier is less accurate than one that would be trained using the true distributions. Second, if one is also required to make a prediction of the accuracy of the designed classifier, one has to use the finite sample not only to train the classifier but also to predict its performance for the true population. The goal of this study is to investigate the bias and error of this prediction for several resampling techniques.

We are particularly motivated by applications in computer-aided diagnosis (CAD), which aims at designing automated image analysis systems to aid radiologists in detecting and characterizing lesions in medical images. In most medical applications, one has only a relatively small population of patient samples with ground truth available for training the CAD system. An intuitive approach would be to use all the available cases for classifier design, with the expectation that using as many cases as possible potentially maximizes the accuracy of the designed classifier. However, the CAD developer will also typically be required to predict the performance that can be expected from the designed classifier. Since all cases have been used for classifier design, one has to use a resampling technique to estimate the performance of the designed classifier when it is applied to the true population.

Classifier performance estimation has previously been addressed in many contexts. In the context of CAD, a commonly used performance measure is the area under the receiver operating characteristics (ROC) curve, referred to as AUC below. To our knowledge, only very preliminary studies have been conducted to investigate the problem of AUC

☆ An abbreviated version of some portions of this article appeared in Sahiner, Chan, and Hadjiiski (2007) as part of the IJCNN 2007 Conference Proceedings, published under IEE copyright.

* Corresponding address: Department of Radiology, University of Michigan, 1500 E. Medical Center Drive, CGC B2102 Ann Arbor, MI 48109-0904, United States. Tel.: +1 734 647 7429; fax: +1 734 615 5513.
E-mail address: berki@umich.edu (B. Sahiner).

0893-6080/$ - see front matter
doi:10.1016/j.neunet.2007.12.012

prediction for a classifier under the constraint of a limited data set, and mostly for linear classifiers (Sahiner et al., 2001; Yousef, Wagner, & Loew, 2004). In a previous study, we investigated the performance of different resampling methods for a backpropagation NN using Monte Carlo simulation under a number of conditions for the number of available samples from each class feature space dimensionality, and a limited set of feature space distributions (Sahiner et al., 2007). The current study extends this previous work by including additional conditions for class distributions and class separability.

2. Methods

Five resampling methods were compared, including three variations of the bootstrap method, namely, the ordinary, .632 and .632+ bootstrap, and the Fukunaga–Hayes (F–H) and the leave-one-out (LOO) methods.

2.1. The ordinary bootstrap

Let F represent the true distribution of the data, and let $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ be the available sample of size N, where the boldface letter $\mathbf{x}$ denotes a set of cases (i.e., a sample), and the italic letter x denotes a data vector (i.e., a case). In bootstrap, an empirical discrete distribution $\hat{F}$ is defined such that a probability of $1/N$ is assigned to each case x_i. A bootstrap sample, denoted as $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_N^*)$, of size N is then randomly drawn with replacement from $\mathbf{x}$, which is equivalent to randomly drawing from the empirical distribution $\hat{F}$. In classifier performance evaluation, the bootstrap method generally involves the estimation of the bias of the resubstitution method, and the removal of this bias from the resubstitution performance to obtain an estimate of the true performance (Efron, 1983). Application to the estimation of the test AUC is described below.

Let $\mathrm{AUC}(\mathbf{S_{train}}, \mathbf{S_{test}})$ denote the test AUC value, obtained when the classifier trained on the set $\mathbf{S_{train}}$ is applied to the test set $\mathbf{S_{test}}$. Let w denote the bias of the resubstitution method. Using the bootstrap technique, one generates B bootstrap samples, $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \ldots, \mathbf{x}^{*B}$, where each sample $\mathbf{x}^{*b} = (x_1^{*b}, x_2^{*b}, \ldots, x_N^{*b})$ is obtained by randomly drawing N data vectors (with replacement) from the original data set $\mathbf{x}$. In the ordinary bootstrap method, the bias of the resubstitution method is estimated from the bootstrap sample $\mathbf{x}^{*b}$ as

$$\hat{w}_{\mathrm{ord}}^b = \mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}) - \mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}). \tag{1}$$

This difference, averaged over B bootstrap samples, provides an estimate for the bias of the resubstitution method:

$$\hat{w}_{\mathrm{ord}} = \frac{1}{B}\sum_{b=1}^{B} \hat{w}_{\mathrm{ord}}^b. \tag{2}$$

This estimate is then subtracted from the true resubstitution the AUC value, obtained by training and testing on the set of all available data, to correct for the bias

$$\widehat{\mathrm{AUC}}_{\mathrm{ord}} = \mathrm{AUC}(\mathbf{x}, \mathbf{x}) - \hat{w}_{\mathrm{ord}}. \tag{3}$$

2.2. The .632 bootstrap

Let $\mathbf{x}^{*b}(0)$ denote the subset of data vectors in $\mathbf{x}$ that do not appear in $\mathbf{x}^{*b}$ for the bth bootstrap. A possibility is to consider $\mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))$ as the classifier performance estimate. Using a probabilistic argument, Efron demonstrates that $\mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))$ is pessimistically biased, because $\mathbf{x}^{*b}(0)$ are farther away from $\mathbf{x}$ than a typical test sample randomly drawn from the true population (Efron, 1983). On the average, the ratio of the distances from these two groups to $\mathbf{x}$ is $1/(1 - \mathrm{e}^{-1}) = 1/0.632$. The bias of the resubstitution method can therefore be estimated from the bootstrap sample $\mathbf{x}^{*b}$ as

$$\hat{w}_{.632}^b = 0.632 \cdot \left[\mathrm{AUC}(\mathbf{x}, \mathbf{x}) - \mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))\right]. \tag{4}$$

The estimate for the bias of the .632 method, $\hat{w}_{.632}$, is found by averaging Eq. (4) over B bootstrap samples. The AUC value estimated from the .632 method is then given by

$$\begin{aligned}\widehat{\mathrm{AUC}}_{.632} &= \mathrm{AUC}(\mathbf{x}, \mathbf{x}) - \hat{w}_{.632} \\ &= (1 - 0.632)\mathrm{AUC}(\mathbf{x}, \mathbf{x}) \\ &\quad + \frac{0.632}{B}\sum_{b=1}^{B} \mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0)).\end{aligned} \tag{5}$$

2.3. The .632+ bootstrap

The .632+ estimator was designed by Efron to address the issue of the bias of the .632 estimator (Efron & Tibshirani, 1997). Starting with the example of a classification problem in which the data are useless (AUC = 0.5), Efron shows that for overtrained classifiers, the .632 estimator for the classifier performance can be optimistically biased. The original definition of the .632+ estimator can be found in the literature. In this study, the AUC value estimated from the .632+ method is defined as

$$\begin{aligned}\widehat{\mathrm{AUC}}_{.632+} &= \mathrm{AUC}(\mathbf{x}, \mathbf{x}) + \frac{1}{B}\sum_{b=1}^{B} \alpha(b)[\mathrm{AUC}'(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0)) \\ &\quad - \mathrm{AUC}(\mathbf{x}, \mathbf{x})],\end{aligned} \tag{6}$$

where

$$\mathrm{AUC}'(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0)) = \max\{0.5, \mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))\}, \tag{7}$$

$$\alpha(b) = \frac{0.632}{1 - 0.368 \cdot R(b)}, \tag{8}$$

and

$$R(b) = \begin{cases} 1 & \text{if } \mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0)) < 0.5 \\ \dfrac{\mathrm{AUC}(\mathbf{x}, \mathbf{x}) - \mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))}{\mathrm{AUC}(\mathbf{x}, \mathbf{x}) - 0.5} & \\ \quad \text{if } \mathrm{AUC}(\mathbf{x}, \mathbf{x}) > \mathrm{AUC}(x^{*b}, \mathbf{x}^{*b}(0)) > 0.5 & \\ 0 \quad \text{otherwise.} & \end{cases} \tag{9}$$

Notice that the .632 estimate (Eq. (5)) can be thought of as a special case of the .632+ estimate with $\alpha = 0.632$ and $\mathrm{AUC}'(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0)) = \mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))$. This definition is

slightly different from that used in Efron and Tibshirani (1997) in that the relative overfitting rate R and the weight α are calculated for each bootstrap replication. Also, the definition (9) for the overfitting rate contains an additional condition related to whether $\mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))$ is smaller than the chance (no-information) AUC value of 0.5, which was not included in Efron and Tibshirani (1997).

2.4. The Fukunaga–Hayes method

Another method to estimate the performance of a classifier that can be designed with N cases is to partition them into a training group of N_{train} cases and a test group of $N_{\text{test}} = N - N_{\text{train}}$ cases. One can repeat the partitioning process P times, and use the average test AUC as the performance estimate. A disadvantage of this method is that since $N_{\text{train}} < N$, the designed classifier may have a lower performance than the one that would have been trained with N cases. Fukunaga and Hayes studied the dependence of the classifier performance on the training sample size N_{train}, and showed that under a wide range of conditions, the probability of misclassification (PMC) error varies linearly with $1/N_{\text{train}}$ (Fukunaga & Hayes, 1989). Based on this observation, they suggested that one can vary $N_{\text{train}} < N$ in a range of values, obtain a linear regression to the PMC, and then extrapolate to find the PMC for $N_{\text{train}} \geq N$. In our previous work, we applied this method for performance estimation using the AUC. For various classifiers and Gaussian class distributions, it was observed that the dependence of the AUC value can be closely approximated by a linear relationship in a sample size range where higher-order terms $1/N_{\text{train}}$ can be neglected (Chan, Sahiner, Wagner, & Petrick, 1999). The implementation in the current study uses four values for $N_{\text{train}} < N$ for finding the linear regression, and P training-test partitioning sets at each of these values to obtain the F–H prediction of the classification performance, $\widehat{\mathrm{AUC}}_{FH}$, at $N_{\mathbf{train}} = N$.

2.5. The leave-one-out method

In the leave-one-out (LOO) technique, one designs N classifiers using the sample $\mathbf{x} = (x_1, x_2, \ldots, x_N)$. In the design of the ith classifier, all cases are used except the case x_i, which is reserved as a test case. Since each classifier is designed using $N - 1$ cases, the number of trainers is very close to the number of available cases. In our application, we accumulated all N test results in an array and computed $\widehat{\mathrm{AUC}}_{\mathrm{LOO}}$ for the LOO method.

2.6. Summary measures of prediction accuracy

As discussed in the introduction, our goal is to predict the performance of the classifier trained with the given set of N cases when it is applied to the true population. The true performance is therefore $\mathrm{AUC}(\mathbf{x}, F)$, which is referred to below as the true conditional performance because it is conditional on the available training set $\mathbf{x}$. Several different class distributions F were simulated, as discussed below. We define one experiment j as the selection of a sample $\mathbf{x}$ from F. The true conditional performance for the jth experiment (conditioned on the available training set), $\mathrm{AUC}_j(\mathbf{x}, F)$, is obtained by training the classifier with $\mathbf{x}$, drawing an additional random test sample of 2000 cases from the distribution of each class, and testing the designed classifier with this data set of 4000 test cases. The number of test cases, 4000, is chosen to be large enough so that the distribution of the test data is close to F. The AUC is calculated using the LABROC program (Metz, Herman, & Shen, 1998), which uses a maximum likelihood estimation algorithm to fit a binormal ROC curve to the classifier output after proper binning. Note that the true conditional performance $\mathrm{AUC}_j(\mathbf{x}, F)$ depends on $\mathbf{x}$, and therefore changes in each experiment. The prediction error of a resampling method for the jth experiment is then defined as

$$E_{j,r} = \widehat{\mathrm{AUC}}_{j,r} - \mathrm{AUC}_j(\mathbf{x}, F), \tag{10}$$

where r stands for one of the five different sampling methods, i.e., the ordinary bootstrap, .632 bootstrap, .632+ bootstrap, F–H, or LOO, and $\widehat{\mathrm{AUC}}_{j,r}$ denotes the predicted AUC for experiment j using the resampling method r. For each condition discussed below, we performed $J = 200$ experiments. The bias of the resampling technique was defined as the average of $E_{j,r}$ over J experiments.

To quantify how close the predicted and true conditional performances are, we used the root-mean-squared error (RMSE)

$$\mathrm{RMSE}_r = \sqrt{\frac{1}{J}\sum_{j=1}^{J} E_{j,r}^2}. \tag{11}$$

The mean and the standard deviation of the predicted accuracy were also estimated:

$$\mathrm{Avg}(\widehat{\mathrm{AUC}}_r) = \frac{1}{J}\sum_{j=1}^{J} \widehat{\mathrm{AUC}}_{j,r}, \tag{12}$$

$$\mathrm{SD}(\widehat{\mathrm{AUC}}_r) = \sqrt{\frac{1}{J-1}\sum_{j=1}^{J}\left(\widehat{\mathrm{AUC}}_{j,r} - \mathrm{Avg}(\widehat{\mathrm{AUC}}_r)\right)^2}. \tag{13}$$

2.7. Feature spaces and sample sizes

We investigated two categories of class distributions. Class distributions in the first category were assumed to follow multivariate normal distributions with equal covariance matrices. It has been shown in the literature (Fukunaga, 1990) that the covariance matrices can be simultaneously diagonalized without affecting the analysis. We therefore used an identity matrix for the covariance of both classes. We assumed that the mean difference between the two classes was equal for each feature. The value of the difference was chosen such that the AUC of the classifier designed and tested with infinite sample size would be approximately 0.8. The dimensionality of the feature space, k, was varied from 3 to 15.

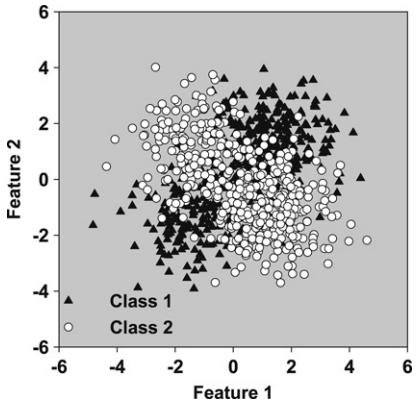

Fig. 1. An example scatter plot of the class distributions from the second category.

Class distributions in the second category were assumed to be a mixture of overlapping Gaussian distributions. The dimensionality of the feature space, k, was varied from 2 to 8. Fig. 1 shows an example scatter plot of the class distributions for $k = 2$. Each class was a mixture of two Gaussians where each Gaussian had identity covariance matrix. The mean vectors of the two Gaussians for class 1 were $[M, \quad M]$ and $[-M, \quad -M]$ respectively, while those for class 2 were $[M, \quad -M]$ and $[-M, \quad M]$ respectively. For $k \geq 3$, the means of the Gaussians in the first two dimensions were $\pm M$, and those in the remaining dimensions were all zero, so that the separability of the two classes was the same for different values of k. The separability of the two classes was varied by changing M. For each k, we performed experiments with three different values of M. The AUCs of the classifier designed and tested with an infinite sample size with these three different values are approximately 0.95, 0.89 and 0.79. These are referred to below as the high, medium-high and medium separability conditions, respectively.

The number of cases from class 1 and class 2, denoted by n_1 and n_2, were assumed to be equal so that $n_1 = n_2 = N/2$. The total number of cases in the sample N was varied between 50 and 120.

2.8. The NN classifier

Many different NN architectures and training methods can be used for a classification task. In this study, we used a three-layered NN trained with a backpropagation method. The NN had k input nodes, h hidden nodes, one output node, and the architecture is denoted as $k - h - 1$. The nodes were fully connected, and the weights were trained using a minimum sum-of-squares error criterion. Since our purpose was not to optimize the NN architecture but to test different resampling schemes to predict the accuracy of a NN that would be trained with the given sample of size N, we did not perform an exhaustive search to determine how many nodes in the hidden layer and what kind of stopping criterion maximized the AUC. However, if the selected NN parameters result in a classifier having a performance that is far from the optimal, the simulation results would be irrelevant in practice. To ensure that this did not happen, we first trained different NNs with a sample size varying between 50 and 120 while the number of hidden nodes and the number of training iterations were varied. The trained NNs were tested with a very large sample size. Using the resulting test A_z values, we chose a k-3-1 architecture for class distributions in the first category (Gaussian) and a k-5-1 architecture for class distributions in the second category (mixture of Gaussians). For the first category, the number of training iterations were 750, 400 and 100 for $k = 3$, 9 and 15, respectively. For the second category, the number of training iterations were 1000, 900 and 800 for $k = 2$, 5 and 8, respectively.

3. Results

Fig. 2 shows the results obtained with the five different resampling methods and five sample sizes ($N = 50$, 60, 80, 100 and 120) for Gaussian class distributions and $k = 15$. The RMSE, and the average performance were plotted in Fig. 2(a) and (b), respectively. Fig. 2(b) shows that the F–H method has a low bias for this condition. However, the .632 and .632+ bootstrap methods perform better than the F–H in terms of the RMSE (Fig. 2(a)).

The simulation conditions in Fig. 3 are similar to those in Fig. 2, except that the feature space dimensionality was reduced to three. The differences in the bias for the different resampling schemes are smaller and the RMSE errors are closer to each other for $k = 3$ compared to $k = 15$, although the .632 and .632+ bootstrap methods still have a small advantage over the others for small N.

Figs. 4 and 5 show the simulation results for a mixture of Gaussians with medium separability, and with $k = 2$ and 8, respectively. The trends in Fig. 4 are similar to those for Fig. 3. At small sample sizes, the average AUC estimated by the F–H and .632+ methods are closest to independent testing. The .632 and .632+ methods have the smallest RMSE. However, Fig. 5 shows that when the dimensionality of the feature space is larger, the average AUC estimated by the .632+ method at small sample sizes is no longer unbiased, whereas the bias of the F–H method is still close to zero. The .632 method has a large bias, and, as a result, has a higher RMSE than the LOO and F–H methods (Fig. 5(b)).

Fig. 6 shows the simulation results for a mixture of Gaussians with $k = 8$ and high separability. In contrast to the other conditions discussed above, the .632+ method has the largest RMSE for sample sizes between 80 and 120 under this condition. This is partly explained by observing that the .632+ method has a relatively large negative bias. On the other hand, the .632 method, which had a positive bias for other simulation conditions, seems almost unbiased under this condition, and has the lowest RMSE.

The average true conditional AUC, and the bias, standard deviation and RMSE obtained over 250 experiments for different resampling methods and different conditions for the simulated data sets are summarized in Table 1.

4. Discussion and conclusion

In many applications, one has only a finite sample size to design a classifier and to assess its performance. Since a larger

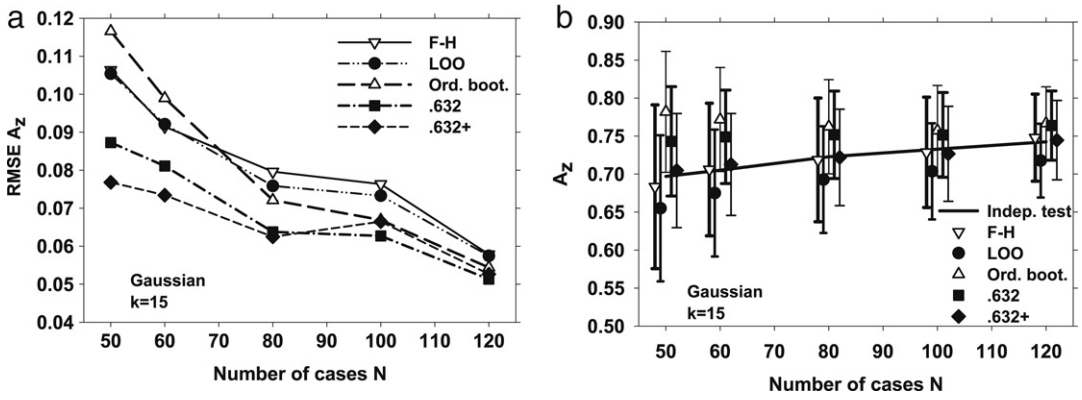

Fig. 2. Simulation results for a 15-dimensional Gaussian feature space. (a) The root-mean-squared-error (RMSE) of the AUC estimated with the five resampling techniques. (b) The mean of the AUC and ±(standard deviation), shown as error bars. The solid line is the average of the true conditional AUC, the standard deviation of which is not shown for the sake of clarity. The data points are plotted slightly offset, centered around a given value of N to prevent the marks and error bars from overlapping with each other.

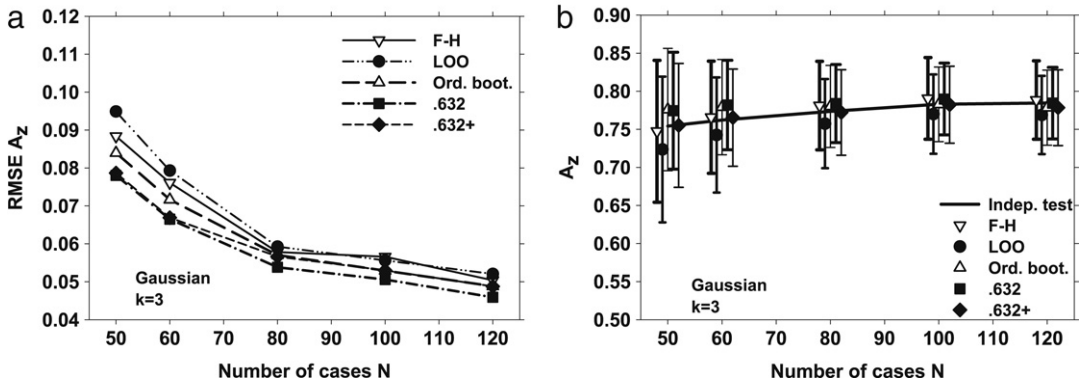

Fig. 3. Simulation results for a 3-dimensional Gaussian feature space: (a) the RMSE of the AUC and (b) the mean of the AUC and ±(standard deviation), shown as error bars.

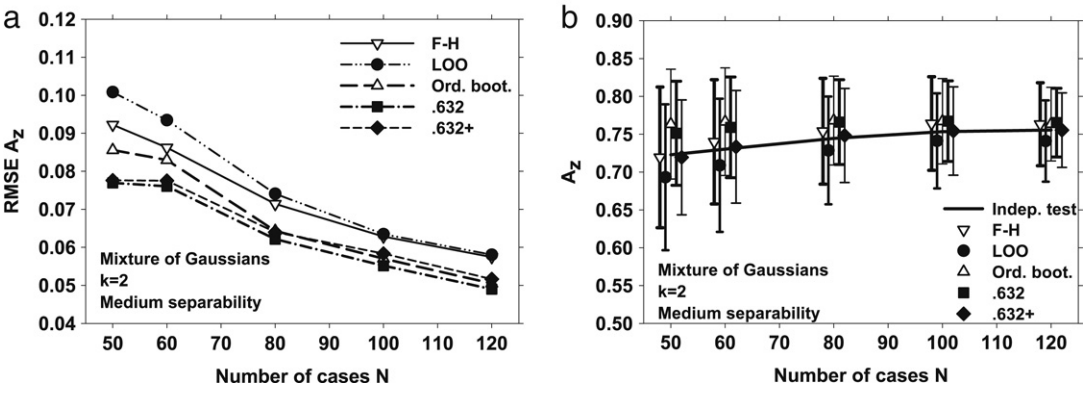

Fig. 4. Simulation results for a 2-dimensional feature space of mixture of Gaussians with medium separability: (a) the RMSE of the AUC and (b) the mean of the AUC and ±(standard deviation), shown as error bars.

sample is more representative of the population, it is advantageous to design the classifier with all the available cases, and then to assess the accuracy of the designed classifier using a resampling technique. In this study, we compared the performances of five such resampling techniques for backpropagation NN classifiers, using the AUC as the performance index.

The bias of the resampling method is important because the use of a method that consistently provides optimistic estimates may be misleading about the potential of a classification technique, especially in CAD. The ordinary bootstrap method had a relatively large optimistic bias for all the conditions in this study, and we believe that, in the form that was used in this investigation, it is not suitable for classifier performance estimation. The F–H method had the smallest overall bias among the methods studied, followed by the .632+ method (Table 1). However, depending on the feature space distribution

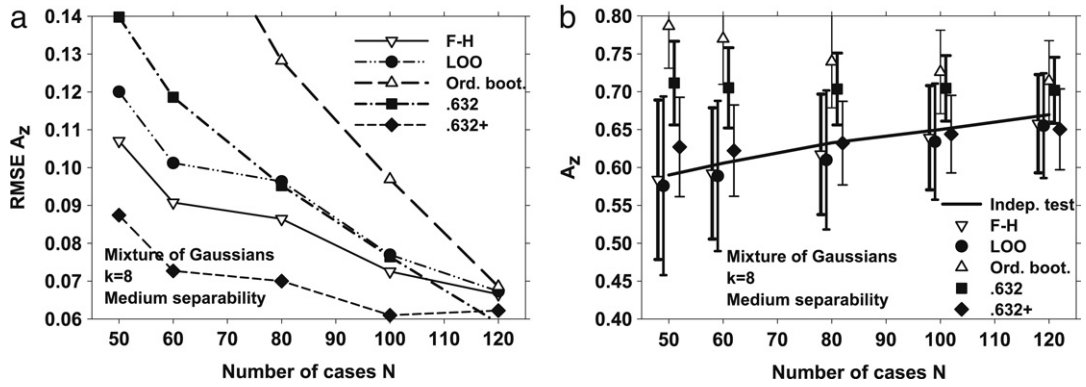

Fig. 5. Simulation results for an 8-dimensional feature space of mixture of Gaussians with medium separability: (a) the RMSE of the AUC and (b) the mean of the AUC and ±(standard deviation), shown as error bars.

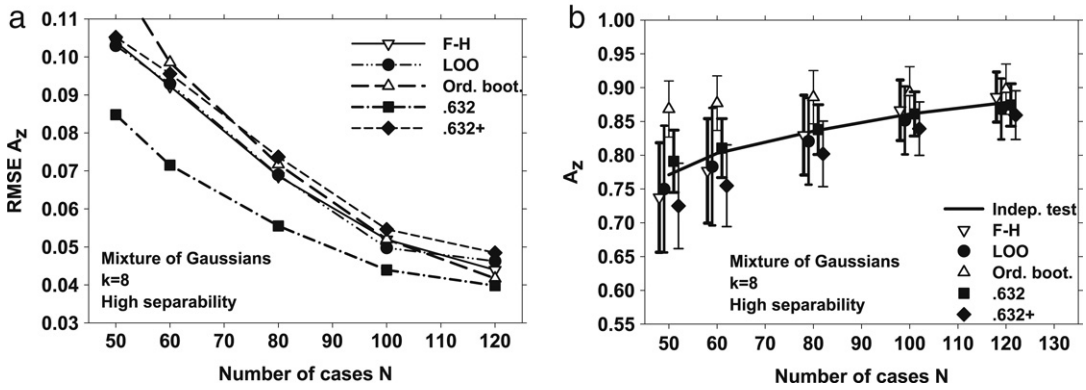

Fig. 6. Simulation results for an 8-dimensional feature space of mixture of Gaussians with high separability: (a) the RMSE of the AUC and (b) the mean of the AUC and ±(standard deviation), shown as error bars.

and dimensionality, some of these trends may be reversed. For example, for the 8-dimensional mixture of Gaussians with high separability, the .632 and LOO methods had lower bias than F–H and .632+ at low training sample size (Fig. 6(b)).

The magnitude of the error in performance estimation, measured by the RMSE in this study, may be more important than the bias in many situations. A smaller error magnitude for a resampling method means that, for a given sample, one has a higher chance to get an estimate that is closer to the truth. For the class distributions considered in this study, the difference in the RMSE obtained using different resampling methods can be large, especially when the feature space dimensionality is large and the number of available samples is small. Under most of the conditions studied, the .632 and .632+ bootstrap methods had either lower or similar RMSE compared to other resampling methods. An exception to this was the 8-dimensional mixture of Gaussians with medium separability, for which the .632 technique had considerably higher RMSE than LOO and F–H (Fig. 5(a)).

Both the bias and standard deviation of the resampling method contribute to the RMSE. The relatively large, optimistic bias of the .632 technique for the 8-dimensional mixture of Gaussians with medium separability (Fig. 5(b)) may be the major reason why this technique had higher RMSE than LOO and F–H in Fig. 5(a). However, under most of the other conditions, the standard deviation of the resampling technique may have been the major contributor to the RMSE. When this is the case, one may expect the.632 and .632+ methods to outperform LOO and F–H because, by design, the bootstrap methods have a smaller standard deviation.

The .632+ method was designed primarily to reduce the optimistic bias of the .632 method. Our simulation study indicates that this goal was met particularly well when the two classes were not well-separated in the feature space. However, when the two classes had high or medium–high separation, the .632+ method may result in an over-correction of the .632 method, and provide a pessimistic bias (Fig. 6(b) and conditions F and H in Table 1).

The method used for finding the .632+ AUC estimate in this study is different from that used in Efron and Tibshirani (1997) for finding the error rate, and the equations are different from those used in Yousef et al. (2004) and Efron and Tibshirani (1997) for AUC, in two aspects. First, our method performs the nonlinear operation given by (9) for each bootstrap replication, whereas Yousef et al. (2004) and Efron and Tibshirani (1997) perform an average over all bootstrap replications before applying the nonlinearity. Our experiments with the latter method (not shown) indicated that the difference in the two methods did not result in a major difference in the .632+ AUC estimate. The second difference in our 632+ AUC estimate was the additional

Table 1
The average true conditional AUC, and the bias, standard deviation (Std) and root-mean-squared error (RMSE) obtained over 250 experiments for different resampling methods for each condition **A**–**H**

	N	True AUC	Fukunaga–Hayes			Leave-one-out			Ordinary bootstrap			.632 bootstrap			.632+ bootstrap		
			Bias	Std	RMSE	Bias	Std	RMSE	Bias	Std	RMSE	Bias	Std	RMSE	Bias	Std	RMSE
A	50	0.754	−0.007	0.093	0.088	−0.031	0.096	0.095	0.022	0.080	0.084	0.020	0.077	0.078	0.001	0.081	0.079
G	60	0.762	0.004	0.074	0.076	−0.020	0.076	0.079	0.017	0.063	0.072	0.020	0.059	0.066	0.003	0.064	0.067
f3	80	0.774	0.008	0.058	0.058	−0.016	0.059	0.059	0.007	0.054	0.057	0.010	0.051	0.054	−0.001	0.056	0.057
	100	0.783	0.008	0.054	0.057	−0.013	0.052	0.056	0.000	0.049	0.053	0.007	0.047	0.051	−0.001	0.051	0.053
	120	0.785	0.004	0.051	0.050	−0.016	0.051	0.052	−0.006	0.049	0.049	0.000	0.047	0.046	−0.006	0.050	0.049
B	50	0.697	−0.014	0.108	0.106	−0.042	0.096	0.105	0.085	0.080	0.117	0.046	0.072	0.087	0.008	0.075	0.077
G	60	0.705	0.001	0.087	0.091	−0.030	0.083	0.092	0.067	0.068	0.099	0.044	0.061	0.081	0.007	0.067	0.073
f15	80	0.723	−0.004	0.081	0.080	−0.030	0.070	0.076	0.040	0.062	0.072	0.029	0.057	0.064	−0.001	0.063	0.062
	100	0.733	−0.004	0.073	0.076	−0.029	0.063	0.073	0.024	0.060	0.067	0.019	0.056	0.063	−0.006	0.063	0.066
	120	0.742	0.005	0.057	0.058	−0.025	0.049	0.057	0.024	0.048	0.054	0.021	0.046	0.051	0.002	0.052	0.053
C	50	0.723	−0.003	0.093	0.092	−0.030	0.096	0.101	0.040	0.073	0.086	0.028	0.069	0.077	−0.004	0.076	0.078
MG	60	0.730	0.010	0.082	0.086	−0.021	0.088	0.093	0.036	0.071	0.083	0.029	0.067	0.076	0.003	0.074	0.078
f2	80	0.745	0.010	0.070	0.071	−0.016	0.071	0.074	0.024	0.059	0.064	0.022	0.056	0.062	0.004	0.062	0.064
M	100	0.753	0.011	0.062	0.063	−0.012	0.063	0.063	0.014	0.056	0.057	0.014	0.053	0.055	0.001	0.058	0.058
	120	0.756	0.008	0.055	0.057	−0.015	0.054	0.058	0.008	0.049	0.050	0.010	0.045	0.049	0.000	0.049	0.052
D	50	0.590	−0.006	0.105	0.107	−0.014	0.118	0.120	0.197	0.056	0.208	0.121	0.055	0.140	0.037	0.066	0.087
MG	60	0.606	−0.014	0.087	0.091	−0.017	0.099	0.101	0.164	0.060	0.178	0.099	0.053	0.119	0.017	0.060	0.073
f8	80	0.633	−0.015	0.080	0.086	−0.023	0.092	0.096	0.107	0.061	0.128	0.071	0.047	0.095	0.000	0.055	0.070
M	100	0.650	−0.011	0.069	0.073	−0.016	0.077	0.077	0.076	0.055	0.097	0.054	0.043	0.076	−0.006	0.051	0.061
	120	0.670	−0.012	0.065	0.067	−0.015	0.069	0.067	0.045	0.053	0.069	0.032	0.044	0.059	−0.019	0.053	0.062
E	50	0.844	−0.009	0.061	0.066	−0.022	0.072	0.079	0.026	0.055	0.066	0.013	0.053	0.060	−0.005	0.061	0.066
MG	60	0.850	0.001	0.058	0.062	−0.014	0.065	0.069	0.024	0.053	0.062	0.013	0.051	0.057	0.000	0.058	0.061
f2	80	0.862	0.005	0.047	0.050	−0.008	0.049	0.052	0.016	0.042	0.048	0.011	0.040	0.046	0.003	0.044	0.048
MH	100	0.869	0.006	0.041	0.042	−0.009	0.041	0.043	0.010	0.038	0.039	0.006	0.036	0.037	0.001	0.038	0.039
	120	0.873	0.001	0.037	0.039	−0.014	0.036	0.041	0.001	0.034	0.036	0.000	0.032	0.035	−0.004	0.034	0.036
F	50	0.685	−0.025	0.095	0.110	−0.023	0.105	0.116	0.148	0.046	0.164	0.068	0.050	0.104	−0.011	0.067	0.093
MG	60	0.709	−0.021	0.086	0.097	−0.020	0.099	0.101	0.118	0.052	0.137	0.048	0.049	0.089	−0.024	0.064	0.089
f8	80	0.746	−0.012	0.076	0.085	−0.013	0.078	0.082	0.073	0.054	0.098	0.027	0.044	0.071	−0.029	0.056	0.081
MH	100	0.777	−0.004	0.060	0.065	−0.013	0.064	0.063	0.046	0.047	0.067	0.014	0.039	0.052	−0.026	0.049	0.064
	120	0.799	0.000	0.053	0.058	−0.014	0.059	0.059	0.025	0.047	0.053	0.003	0.040	0.049	−0.027	0.049	0.064
G	50	0.911	−0.018	0.044	0.055	−0.016	0.051	0.060	0.021	0.036	0.049	0.005	0.036	0.045	−0.005	0.041	0.049
MG	60	0.919	−0.011	0.037	0.042	−0.013	0.048	0.053	0.016	0.035	0.043	0.003	0.035	0.040	−0.004	0.039	0.044
f2	80	0.926	−0.001	0.029	0.032	−0.005	0.032	0.037	0.013	0.027	0.034	0.006	0.026	0.032	0.002	0.028	0.034
H	100	0.933	−0.001	0.026	0.028	−0.008	0.029	0.031	0.006	0.025	0.027	0.002	0.024	0.026	−0.001	0.025	0.027
	120	0.935	−0.003	0.023	0.024	−0.010	0.024	0.028	0.001	0.022	0.024	−0.002	0.021	0.023	−0.004	0.022	0.024
H	50	0.771	−0.034	0.081	0.104	−0.021	0.094	0.103	0.097	0.042	0.121	0.020	0.046	0.085	−0.046	0.063	0.105
MG	60	0.803	−0.026	0.077	0.092	−0.020	0.087	0.093	0.074	0.040	0.099	0.008	0.044	0.072	−0.048	0.060	0.096
f8	80	0.836	−0.006	0.059	0.069	−0.015	0.064	0.069	0.050	0.039	0.072	0.002	0.037	0.056	−0.034	0.048	0.074
H	100	0.861	0.005	0.045	0.052	−0.010	0.050	0.050	0.032	0.038	0.052	0.000	0.033	0.044	−0.022	0.040	0.055
	120	0.878	0.008	0.037	0.044	−0.010	0.045	0.046	0.020	0.037	0.042	−0.004	0.031	0.040	−0.019	0.036	0.048

N denotes the total number of training samples drawn in each experiment. The legend for the first column is: G, Gaussian data; MG, mixture of Gaussian data; f3, 3-dimensional; f15, 15 dimensional; f2, 2-dimensional; f8, 8-dimensional data; M, medium separability; MH, medium–high separability; H, high separability between the two classes.

condition in (9) related to whether $\mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))$ is smaller than the chance (no-information) AUC value of 0.5. Our experiments indicated that without this condition in (9), our .632+ AUC estimate for the medium separability condition may have a larger positive bias. Note that without this condition, the over-fitting rate R will have a large discontinuity when it is plotted as a function of $\mathrm{AUC}(\mathbf{x}^{*b}, \mathbf{x}^{*b}(0))$.

The understanding of bias, variance and RMSE issues in classifier performance estimation will provide us a useful guide for reducing errors in the assessment of classifier performance.

Acknowledgments

This work was supported in part by the USPHS under grants CA095153 and CA118305. The authors are grateful to Charles E. Metz, PhD, for the LABROC program.

References

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

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, *78*, 316–331.

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, *92*, 548–560.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). New York: Academic Press.

Fukunaga, K., & Hayes, R. R. (1989). Effects of sample size on classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*, 873–885.

Metz, C. E., Herman, B. A., & Shen, J. H. (1998). Maximum-likelihood estimation of receiver operating characteristic (ROC) curves from continuously-distributed data. *Statistics in Medicine*, *17*, 1033–1053.

Sahiner, B., Chan, H.P., & Hadjiiski, L. 2007. Classifier performance estimation under the constraint of a finite sample size: Resampling schemes applied to neural network classifiers. In *Proc. 2007 international joint conference on neural networks* (pp. 1762–1766).

Sahiner, B., Chan, H.P., Petrick, N., Hadjiiski, L.M., Paquerault, S., & Gurcan, M.N. 2001. Resampling schemes for estimating the accuracy of a classifier designed with a limited data set. In *Medical image perception conference*, vol. IX, Warrenton, VA. Airlie Conference Center.

Yousef, W.A., Wagner, R.F., & Loew, M.H. 2004. Comparison of non-parametric methods for assessing classifier performance in terms of ROC parameters. In *Applied imagery pattern recognition workshop, 33rd* (pp. 190–195). IEEE Computer Society.