Bootstrap Techniques for Error Estimation

ANIL K. JAIN, SENIOR MEMBER, IEEE, RICHARD C. DUBES, AND CHAUR-CHIN CHEN, STUDENT MEMBER, IEEE

Abstract—**The design of a pattern recognition system requires careful attention to error estimation. The error rate is the most important descriptor of a classifier's performance. The commonly used estimates of error rate are based on the holdout method, the resubstitution method, and the leave-one-out method. All suffer either from large bias or large variance and their sample distributions are not known. Bootstrapping refers to a class of procedures that resample given data by computer. It permits determining the statistical properties of an estimator when very little is known about the underlying distribution and no additional samples are available. Since its publication in the last decade, the bootstrap technique has been successfully applied to many statistical estimations and inference problems. However, it has not been exploited in the design of pattern recognition systems. We report results on the application of several bootstrap techniques in estimating the error rate of 1-NN and quadratic classifiers. Our experiments show that, in most cases, the confidence interval of a bootstrap estimator of classification error is smaller than that of the leave-one-out estimator. The error of 1-NN, quadratic, and Fisher classifiers are estimated for several real data sets.**

Index Terms—**Bootstrap, confidence interval, error rate estimator, Fisher's classifier, pattern, quadratic classifier, 1-NN classifier.**

I. Introduction

IT is common to use the estimated error rate to evaluate the performance of a classifier. In the nonparametric framework the leave-one-out method (also referred to as cross-validation or the U method) proposed by Lachenbruch and Mickey [13] has been shown to have a much smaller bias than the resubstitution method [2], and has become a popular nonparametric error estimator in small sample size situations. However, Efron [8] has shown that the leave-one-out method can have a much larger variance than competing estimators. In some cases, this variance is sufficiently large that competitors with slightly larger bias but smaller variance will outperform the leave-one-out estimator. In this paper, we establish confidence intervals on various error rate estimators, compare them to those obtained for the leave-one-out method, and show that some estimators based on bootstrapping techniques do provide shorter confidence intervals than the leave-one-out method.

The organization of the paper is as follows. Section II defines the error rate estimators. Section III reports the performance of various estimators for the error rate of the nearest neighbor (1-NN) decision rule. Section IV contains experimental results for estimating the error rate of a quadratic classifier. Section V examines the performance of various estimators for the error rate of 1-NN, quadratic and Fisher classifiers on three real data sets. Section VI gives the conclusions of our study.

Manuscript received April 18, 1986; revised March 16, 1987. Recommended for acceptance by J. Kittler. This work was supported by the National Science Foundation under Grant ECS-8603541.

The authors are with the Department of Computer Science, Michigan State University, East Lansing, MI 48824.

IEEE Log Number 8715813.

II. Definitions of Error Rate Estimators

We follow Efron's [8] notations. Let $\{v_1, v_2, \cdots, v_n\}$ be a set of d-dimensional training vectors with corresponding categories $\{y_1, y_2, \cdots, y_n\}$ taken from classes $\{C_1, C_2, \cdots, C_K\}$. For convenience, denote training pattern $x_i = (v_i, y_i)$ and $X = \{x_i\}_{i=1}^n$.

Let $\eta(v, X)$ be a decision rule based on the training set X and let $Q[y, \eta(v, X)]$ be 0 if the classification of vector v by η is correct.

$$Q[y, \eta(v, X)] = \begin{cases} 0 & \text{if } \eta(v, X) = y, \\ 1 & \text{otherwise.} \end{cases}$$

Several error rate estimators are now defined. The expected error rate (Err) is the probability of misclassifying a randomly selected pattern $x_0 = (v_0, y_0)$ independent of X.

$$\text{Err} = E\{Q[y_0, \eta(v_0, X)]\} \tag{2.1}$$

In general, this expectation cannot be evaluated explicitly and is not known. We estimate it by classifying 1000 test patterns (equal number of patterns from each class) independently generated from the distributions of the training patterns. We report Err to assess the bias of each estimator.

The apparent error rate (App), or the resubstitution estimate, is obtained by reclassifying the training patterns.

$$\text{App} = \frac{1}{n} \sum_{i=1}^{n} Q[y_i, \eta(v_i, X)] \tag{2.2}$$

The leave-one-out estimate saves each training pattern for testing and uses the remaining $(n - 1)$ patterns as new training patterns.

$$\text{Ecv} = \frac{1}{n} \sum_{i=1}^{n} Q[y_i, \eta(v_i, X_{(i)})] \quad \text{where } X_{(i)} = X - \{x_i\}. \tag{2.3}$$

It is well known in pattern recognition literature that the apparent error rate is optimistically biased; i.e., App usually underestimates Err. The leave-one-out estimate, on the other hand, is nearly unbiased but has large variance

[2]. Bootstrapping allows us to define alternative estimators.

Bootstrapping techniques sample the training patterns with replacement to establish nonparametric estimators of bias, variance, and other statistics. This resampling generates "fake" data sets from the original data to assess the variability of a statistic or parameter from its variability over all the sets of fake data [3]. Bootstrapping is similar to other resampling schemes such as cross-validation and jackknifing [7]. The difference lies in the manner in which fake data sets are generated.

Let $\hat{F}$ be the empirical probability distribution of X, i.e.,

$$\hat{F}\text{: mass of } \frac{1}{n} \text{ on } x_i, \qquad i = 1, 2, \cdots, n.$$

A bootstrap sample, X^*, is a random sample of size n from $\hat{F}$. In other words, X^* is a set $\{x_1^*, x_2^*, \cdots, x_n^*\}$ randomly selected from the training set $\{x_1, x_2, \cdots, x_n\}$ with replacement. For a given set of training patterns, let op (X, F) be the positive or optimistic bias defined as

$$\text{op}\,(X, F) = \text{Err} - \text{App}, \tag{2.4}$$

where F is the underlying unknown mixture distribution. The expectation of this bias, denoted as $w(F)$, can be written as

$$w(F) = E_F[\text{op}\,(X, F)]. \tag{2.5}$$

If w were known, then Err could be estimated as

$$\hat{\text{Err}} = \text{App} + \text{w}. \tag{2.6}$$

The bootstrap procedure for estimating the bias w is defined below:

1) Select a bootstrap sample according to $\hat{F}$, say $X^{*b} = \{x_i^{*b}\}_{i=1}^n$.

2) Compute $w_b = \Sigma_{i=1}^n ((1/n) - P_i^{*b})\, Q[y_i, \eta(v_i, X^{*b})]$ with P_i^{*b} indicating the proportion of the bootstrap sample on x_i, i.e.,

$$P_i^{*b} = \text{Cardinality of } \{j \mid x_j^{*b} = x_i\}/n.$$

3) Repeat steps 1) and 2) B times to get $\{w_1, w_2, \cdots, w_B\}$. The bias of the bootstrap error rate is estimated by

$$\text{wboot} = \frac{1}{B} \sum_{b=1}^{B} w_b.$$

The bootstrap error estimate, Boot, is:

$$\text{Boot} = \text{App} + \text{wboot}. \tag{2.7}$$

Bootstrapping mimics the process of selecting many samples of size n. The choice of B is not critical as long as it exceeds 100. Efron [8] suggests that B need not be greater than 200.

Another bootstrap estimator, called the E0 estimator, counts the number of those training patterns misclassified which did not appear in the bootstrap sample. The E0 estimate is obtained by summing the misclassified samples over all bootstrap samples and dividing the sum by the total number of training patterns not appearing in the bootstrap sample. Thus, E0 uses a subset of a training set as a test set. Let $A_b = \{i \mid P_i^{*b} = 0\}$ denote the index set of training patterns which do not appear in the bth bootstrap sample, then

$$\text{E0} = \frac{\sum_{b=1}^{B} \sum_{A_b} Q[y_i, \eta(v_i, X^{*b})]}{\sum_{b=1}^{B} |A_b|}, \tag{2.8}$$

where $|A_b|$ denotes the cardinality of the set A_b.

Finally, we define the "0.632" error estimator, denoted E632. The rationale for the 0.632 estimator is given by Efron [8]. Note that App is the error rate for patterns which are "zero" distance from the training set, whereas patterns contributing to E0 are "too far out" from the training set. Since the (asymptotic) probability that a pattern will not be included in a bootstrap sample is approximately 0.368, the weighted average of App and E0 involves patterns at the "right" distance from the training set in estimating the error rate.

$$\text{E632} = 0.368 * \text{App} + 0.632 * \text{E0} \tag{2.9}$$

Our primary interest in studying the behavior of error rate estimators is in small sample size situations. When the sample size is large, most of the estimators give identical results because of the consistency property. To highlight the difference in the performance of various estimators, we are mainly interested in situations where the true error rates are moderate but not greater than 0.35.

III. Nearest Neighbor Decision Rule

Consider n_i training vectors $\{v_1^{(i)}, v_2^{(i)}, \cdots, v_{n_i}^{(i)}\}$ from class C_i, where $i = 1, 2, \cdots, K$. The nearest neighbor decision rule classifies a d-dimensional test vector v_0 as follows:

> Assign v_0 to class C_j if $\min_i \{\| v_0 - v_i^{(j)} \|\} \leq \min_i \{\| v_0 - v_i^{(m)} \|\}$, for all $m \neq j$, where $\| \cdot \|$ denotes the Euclidean norm.

Ties are resolved randomly. We will now use the estimators defined in Section II to estimate the error rate of the 1-NN decision rule. Since App uses the same data for training and testing, it is almost zero for the 1-NN decision rule (ties in certain interpoint distances may result in App greater than zero). Therefore, Boot and E632, which require App in their computations [see equations (2.7) and (2.9)], were not used in these experiments. Each experiment involves 100 trials, so 100 independent sets of training samples are generated with 200 bootstrap samples for each trial ($B = 200$) with dimensions $d = 2, 4, 8$, and two classes ($K = 2$). The 68 percent nonparametric confidence interval of an error estimator based on these 100 values is defined to be $[a, b]$, where a is the 17th smallest estimate, and b is the 84th smallest estimate among the 100 trials.

Experiment 1 follows Efron's [8] paradigm. The training patterns are generated from d-dimensional Gaussian distributions $N(u_i, \Sigma_i)$, $i = 1, 2$. The parameters of these distributions are as follows:

$$u_1 = (0, 0, \cdots, 0),$$

$$u_2 = (2.5634, 0, 0, \cdots, 0),$$

and

$$\Sigma_1 = \Sigma_2 = I_d,$$

where I_d is the d-dimensional identity matrix. The mean vectors u_1 and u_2 fix the Bayes error at 0.10 when the class prior probabilities are equal.

Class 1: 10 training patterns per class.

Class 2: 20 training patterns per class.

Table I summarizes the results in the form of the mean, standard deviation (s.d.), 68 percent confidence interval (C.I.) and width of the interval for each estimator.

All of the parameters in Experiments 2 and 3 are the same as those in Experiment 1 except for u_2. In Experiment 2, $u_2 = (1.6836, 0, 0, \cdots, 0)$ and in Experiment 3, $u_2 = (1.0488, 0, 0, \cdots, 0)$. This fixes the Bayes error at 0.20 and 0.30, respectively, for equal class prior probabilities. Tables II and III summarize the results of Experiments 2 and 3.

In all of the results reported in Tables I–III, the E0 estimator provides shorter 68 percent confidence intervals and smaller standard deviations than the leave-one-out estimator, although the E0 estimator has a slightly larger bias in some cases. These results are independent of the dimensionality, size of the training samples, and the true error rate. This suggests that E0 is a better estimator of the error rate than the commonly used leave-one-out method (Ecv). Of course, E0 requires more computation than Ecv. Note that the usual definition of confidence interval as (mean − s.d., mean + s.d.) assumes a normal distribution of the error rate estimator. This motivates our use of the nonparametric 68 percent confidence interval. As expected, the mean value of Err always lies in the confidence intervals of both E0 and Ecv.

TABLE I
ERROR RATES FOR 1-NNR, BAYES ERROR = 0.10

Case 1 : $n_1 = n_2 = 10$, K=2

d = 2	Err	E0	Ecv
mean :	0.158	0.188	0.189
s.d. :	0.036	0.103	0.115
C.I. :	[0.127,0.186]	[0.071,0.297]	[0.050,0.300]
width:	0.059	0.225	0.250
d = 4	**Err**	**E0**	**Ecv**
mean :	0.188	0.197	0.181
s.d. :	0.041	0.095	0.112
C.I. :	[0.148,0.230]	[0.102,0.275]	[0.050,0.250]
width:	0.082	0.172	0.200
d = 8	**Err**	**E0**	**Ecv**
mean :	0.216	0.261	0.244
s.d. :	0.039	0.095	0.115
C.I. :	[0.174,0.252]	[0.161,0.355]	[0.150,0.350]
width:	0.078	0.194	0.200

Case 2 : $n_1 = n_2 = 20$, K=2

d = 2	Err	E0	Ecv
mean :	0.156	0.154	0.146
s.d. :	0.028	0.066	0.080
C.I. :	[0.128,0.184]	[0.090,0.210]	[0.075,0.225]
width:	0.056	0.120	0.150
d = 4	**Err**	**E0**	**Ecv**
mean :	0.174	0.188	0.182
s.d. :	0.029	0.069	0.082
C.I. :	[0.146,0.194]	[0.129,0.260]	[0.100,0.275]
width:	0.048	0.131	0.175
d = 8	**Err**	**E0**	**Ecv**
mean :	0.208	0.209	0.196
s.d. :	0.029	0.072	0.087
C.I. :	[0.180,0.233]	[0.141,0.280]	[0.100,0.300]
width:	0.053	0.139	0.200

TABLE II
ERROR RATES FOR 1-NNR, BAYES ERROR = 0.20

Case 1 : $n_1 = n_2 = 10$, K=2

d = 2	Err	E0	Ecv
mean :	0.295	0.324	0.315
s.d. :	0.048	0.126	0.148
C.I. :	[0.247,0.348]	[0.190,0.452]	[0.150,0.450]
width:	0.101	0.261	0.300
d = 4	**Err**	**E0**	**Ecv**
mean :	0.321	0.335	0.327
s.d. :	0.043	0.106	0.130
C.I. :	[0.277,0.366]	[0.240,0.422]	[0.200,0.450]
width:	0.089	0.182	0.250
d = 8	**Err**	**E0**	**Ecv**
mean :	0.347	0.398	0.391
s.d. :	0.043	0.113	0.147
C.I. :	[0.304,0.385]	[0.297,0.505]	[0.250,0.550]
width:	0.081	0.209	0.300

Case 2 : $n_1 = n_2 = 20$, K=2

d = 2	Err	E0	Ecv
mean :	0.295	0.293	0.284
s.d. :	0.036	0.084	0.099
C.I. :	[0.261,0.330]	[0.207,0.381]	[0.175,0.400]
width:	0.069	0.174	0.225
d = 4	**Err**	**E0**	**Ecv**
mean :	0.308	0.326	0.320
s.d. :	0.037	0.083	0.100
C.I. :	[0.274,0.338]	[0.246,0.414]	[0.225,0.425]
width:	0.064	0.168	0.200
d = 8	**Err**	**E0**	**Ecv**
mean :	0.343	0.343	0.333
s.d. :	0.032	0.084	0.104
C.I. :	[0.310,0.373]	[0.252,0.426]	[0.200,0.425]
width:	0.063	0.174	0.225

IV. Quadratic Classifier

Consider n_i training vectors $\{v_1^{(i)}, v_2^{(i)}, \cdots, v_{n_i}^{(i)}\}$ from class C_i for $i = 1, 2, \cdots, K$. The maximum-likelihood estimators of mean vectors and covariance matrices for the pattern classes are defined as:

$$\hat{u}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} v_j^{(i)},$$

$$\hat{\Sigma}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} [v_j^{(i)} - \hat{u}_i][v_j^{(i)} - \hat{u}_i]^T,$$

$$i = 1, 2, \cdots, K.$$

A quadratic classifier based on the multivariate Gaussian model is defined as [5]:

Assign v to class C_i if

$$(v - \hat{u}_j)^T \hat{\Sigma}_j^{-1}(v - \hat{u}_j) - (v - \hat{u}_i)^T \hat{\Sigma}_i^{-1}(v - \hat{u}_i) > 2 \log [\hat{P}(C_j) \mid \hat{\Sigma}_i \mid^{1/2} / \hat{P}(C_i) \mid \hat{\Sigma}_j \mid^{1/2}],$$

for all $j \neq i$, where $|\hat{\Sigma}_i|$ is the determinant of $\hat{\Sigma}_i$, and $\hat{P}(C_i) = n_i/(n_1 + n_2 + \cdots + n_K)$ is the estimated prior probability for class C_i.

TABLE III
ERROR RATES FOR 1-NNR, BAYES ERROR = 0.30

Case 1 : $n_1 = n_2 = 10$, K=2

d = 2	Err	E0	Ecv
mean :	0.402	0.437	0.435
s.d. :	0.045	0.128	0.165
C.I. :	[0.356,0.447]	[0.297,0.559]	[0.250,0.600]
width:	0.091	0.263	0.350
d = 4	Err	E0	Ecv
mean :	0.419	0.439	0.439
s.d. :	0.038	0.104	0.138
C.I. :	[0.378,0.459]	[0.345,0.546]	[0.300,0.550]
width:	0.081	0.201	0.250
d = 8	Err	E0	Ecv
mean :	0.432	0.473	0.469
s.d. :	0.036	0.108	0.146
C.I. :	[0.402,0.464]	[0.382,0.596]	[0.350,0.600]
width:	0.062	0.214	0.250

Case 2 : $n_1 = n_2 = 20$, K=2

d = 2	Err	E0	Ecv
mean :	0.407	0.400	0.388
s.d. :	0.033	0.083	0.098
C.I. :	[0.377,0.438]	[0.315,0.488]	[0.275,0.475]
width:	0.061	0.173	0.200
d = 4	Err	E0	Ecv
mean :	0.411	0.421	0.417
s.d. :	0.034	0.077	0.098
C.I. :	[0.381,0.439]	[0.340,0.491]	[0.325,0.525]
width:	0.058	0.151	0.200
d = 8	Err	E0	Ecv
mean :	0.434	0.437	0.434
s.d. :	0.027	0.081	0.106
C.I. :	[0.406,0.459]	[0.345,0.517]	[0.325,0.525]
width:	0.053	0.172	0.200

TABLE IV
ERROR RATES FOR QUADRATIC CLASSIFIER, $d = 2$, $K = 2$

	mean	s.d.	C.I.	width
Err	0.319	0.006	[0.314,0.324]	0.010
App	0.232	0.069	[0.175,0.275]	0.100
Boot	0.273	0.080	[0.198,0.341]	0.143
E0	0.301	0.077	[0.223,0.372]	0.149
E632	0.276	0.073	[0.202,0.341]	0.139
Ecv	0.289	0.083	[0.200,0.375]	0.175

TABLE V
ERROR RATES FOR QUADRATIC CLASSIFIER, $d = 4$, $K = 2$

	mean	s.d.	C.I.	width
Err	0.318	0.005	[0.314,0.324]	0.010
App	0.160	0.063	[0.100,0.225]	0.125
Boot	0.256	0.078	[0.171,0.336]	0.165
E0	0.339	0.078	[0.261,0.404]	0.143
E632	0.273	0.070	[0.210,0.347]	0.138
Ecv	0.294	0.090	[0.200,0.375]	0.175

TABLE VI
ERROR RATES FOR QUADRATIC CLASSIFIER, $d = 2$, $K = 4$

	mean	s.d.	C.I.	width
Err	0.325	0.011	[0.314,0.334]	0.020
App	0.285	0.046	[0.237,0.337]	0.100
Boot	0.331	0.050	[0.282,0.387]	0.105
E0	0.360	0.045	[0.312,0.403]	0.091
E632	0.332	0.044	[0.288,0.379]	0.091
Ecv	0.335	0.049	[0.276,0.375]	0.099

If the underlying class-conditional densities are multivariate Gaussian with known parameters, then the form of the above decision rule is Bayes optimal. We are using estimated parameters in place of the true parameters, so the above rule is called the "plug-in" rule.

Experiment 4 generates the training patterns from two-dimensional Gaussian distributions $N(u_i, \Sigma_i)$ with the following parameters.

$u_1 = (0, 0)$, $u_2 = (1, 1)$, and $\Sigma_1 = I_2$, $\Sigma_2 = 1.44I_2$. In Experiment 5, the training patterns are generated from four-dimensional Gaussian distributions with the following parameters.

$u_1 = (0, 0, 0, 0)$, $u_2 = (1, 1, 1, 1)$, and $\Sigma_1 = 1.69I_4$, $\Sigma_2 = 2.25I_4$. These parameters provide moderate error rates. Tables IV and V summarize the results for $n_1 = n_2 = 20$.

The purpose of Experiment 6 is to evaluate the performance of bootstrap error estimators for multiclass problems. The training patterns are generated from two-dimensional Gaussian distributions with 4 classes having the following parameters

$$u_1 = (1.5, 0), \quad u_2 = (0, 1.5), \quad u_3 = (-1.5, 0),$$

$$u_4 = (0, -1.5), \quad \text{and} \quad \Sigma_i = 1.1I_2, \quad i = 1, 2, 3, 4.$$

Table VI summarizes the performance of various error estimates for $n_1 = n_2 = n_3 = n_4 = 20$. Tables IV, V, and VI show that almost all of the estimators based on the bootstrapping techniques (Boot, E0, E632) have shorter 68 percent confidence intervals and smaller standard deviations than the leave-one-out estimator Ecv. The only exception is Boot in experiment 6 which has a slightly larger confidence interval than Ecv in Table VI. Among the various error estimators, E632 appears to have the smallest standard deviation and shortest confidence interval. This result agrees with Efron [8] and Chernick *et al.* [1] for Fisher's linear classifier. None of the estimates exhibits consistently lowest error bias.

V. CLASSIFICATION OF REAL DATA SETS

This section evaluates the performance of bootstrap estimators on several real data sets with three classifiers (1-NN, Fisher, quadratic classifiers). We first define Fisher's classifier. Consider n_i training vectors $\{v_1^{(i)}, v_2^{(i)}, \cdots, v_{n_i}^{(i)}\}$ from class C_i, $i = 1, 2, \cdots, K$. Estimators for the mean vector $\hat{u}_i$ and covariance matrix $\hat{\Sigma}_i$ of the ith pattern class are defined in Section IV. The pooled mean vector and within-class scatter matrix are estimated by

$$\hat{u} = \frac{1}{n} \sum_{i=1}^{K} n_i \hat{u}_i, \qquad \text{where } n = \sum_{i=1}^{K} n_i,$$

$$\hat{S} = \frac{1}{n} \sum_{i=1}^{K} n_i \hat{\Sigma}_i.$$

The Mahalanobis distance between pattern v and the estimated mean vector of class C_j is denoted by $g_j(v)$, where

$$g_j(v) = (v - \hat{u}_j)^T \hat{S}^{-1} (v - \hat{u}_j),$$

$$\text{for } j = 1, 2, \cdots, K.$$

The Fisher's classifier can be defined as [14]:

$$\text{Assign } v \text{ to class } C_i \text{ if } g_i(v) = \min_j \{g_j(v)\}.$$

We now estimate various error rates of these three classifiers on three sets of real data. The 80X data set is derived from the Munson's hand printed Fortran character set. Included are 15 patterns from each of the characters "8," "0," "X". Each pattern consists of 8 feature measurements [4]. The IRIS data set contains measurements of three species of IRIS (setosa, versicolor, virginica). It consists of 50 patterns from each species on each of 4 features (sepal length, sepal width, petal length, petal width) [11]. The IMOX data set contains 8 feature measurements on each character of "I," "M," "O," "X". It consists of 192 patterns, 48 in each character. This data set is also derived from the Munson's database [4].

The results are shown in Table VII. We do not know the true error rates of these data sets nor do we know their distributions so we cannot compare the performance of various estimators. The purpose of reporting these results is to provide some feeling about differences between various error estimates for a fixed classifier. Generally, E0 is more conservative than Ecv, that is, E0 provides higher error estimate than Ecv. However, E632 is comparable to Ecv.

Note that when the quadratic classifier is applied to the 80X data set, estimated covariance matrices for each bootstrap sample are frequently singular so we use the first two principal components [14] of the original data as new features and apply the quadratic classifier to the projected data. This suggests that the size of the training sample should exceed the dimensionality by a factor of at least five [12]. Note that, as expected, App (resubstitution estimate) provides an optimistic estimate of the error rate for all data sets. The estimates of confidence intervals for error rates in Tables I–VI were derived from a Monte Carlo analysis in which fresh training patterns were obtained on each trial. This procedure cannot be applied to real data since real data provide only one set of training patterns. Efron [9] shows how confidence intervals on parametric estimators can be obtained with bootstrapping. Each bootstrap sample leads to one estimate and the B bootstrap samples generate a distribution for the estimator from which various statistics, such as confidence intervals, can be estimated. This bootstrap method cannot repeatedly be applied to real data because all B bootstrap samples are used to compute one value of Boot, E0, and E632. Thus one would need to create several sets of bootstrap samples of size B to generate a distribution for these estimators. The smaller confidence intervals of bootstrap estimators compared to Ecv established in Tables I–VI further support our contention that bootstrap estimates should be considered in the design of pattern recognition systems.

TABLE VII
ERROR ESTIMATION FOR REAL DATA: 80X, IRIS, IMOX

A. 80X

	Classifier	App	Boot	E0	E632	Ecv
	1-NN	*	*	0.103	*	0.067
**	QUADRATIC	0.178	0.237	0.303	0.257	0.244
	FISHER	0.022	0.067	0.127	0.088	0.089

B. IRIS

Classifier	App	Boot	E0	E632	Ecv
1-NN	*	*	0.045	*	0.040
QUADRATIC	0.020	0.025	0.026	0.024	0.020
FISHER	0.020	0.022	0.023	0.022	0.033

C. IMOX

Classifier	App	Boot	E0	E632	Ecv
1-NN	*	*	0.055	*	0.052
QUADRATIC	0.026	0.042	0.056	0.045	0.047
FISHER	0.073	0.085	0.095	0.087	0.078

* : Not defined for 1-NN classifier
** : Use two principal components only

VI. Summary and Conclusions

The bootstrap procedure [3], [6]–[10] has been described as a nonparametric maximum likelihood estimation technique. The simulations of Efron [8] and Chernick *et al.* [1] show that estimators based on bootstrapping performed somewhat better than the traditional leave-one-out estimator in estimating the error rate of Fisher's linear classifier. In this paper, we have extended the simulation results to 1-NN classifiers and quadratic classifiers. The apparent error rate is almost zero for the 1-NN classifier, which makes some bootstrap estimators, such as Boot and E632, not appropriate for nearest neighbor classifiers. Therefore, for 1-NN classifier we only used the E0 estimator and compared it with the leave-one-out estimator. In all our experiments, the conditional expected error rate, Err (based on testing 1000 independent test samples) falls in the 68 percent confidence interval of all the error estimators. This suggests that these error estimators are reliable. In almost all of our limited experiments, the bootstrap estimators have smaller variances and shorter 68 percent confidence intervals than the leave-one-out estimator. For quadratic classifiers, the E632 estimator outperforms the other estimators, which is consistent with earlier reported results for Fisher's classifier. This suggests that boostrapping is a powerful nonparametric technique for evaluating a classifier's performance.

References

[1] M. C. Chernick, V. K. Murthy, and C. D. Nealy, "Application of bootstrap and other resampling techniques: Evaluation of classifier performance," *Pattern Recognition Lett.*, vol. 3, pp. 167–178, 1985.
[2] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Englewood Cliffs, NJ: Prentice-Hall International, 1982.
[3] P. Diaconis and B. Efron, "Computer-intensive methods in statistics," *Sci. Amer.*, pp. 116–127, 1983.
[4] R. Dubes and A. K. Jain, "Clustering techniques—The user's dilemma," *Pattern Recognition*, vol. 8, pp. 247–260, 1976.
[5] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.
[6] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, pp. 1–26, 1979.

[7] ——, "The jackknife, the bootstrap, and other resampling plans," in *CBMS–NSF Regional Conf. Series in Applied Mathematics*, no. 38, SIAM, 1982.
[8] ——, "Estimating the error rate of a prediction rule: Improvement on cross-validation," *J. Amer. Statist. Ass.*, vol. 78, pp. 316–331, 1983.
[9] ——, "Nonparametric standard errors and confidence intervals," *Canadian J. Statist.*, vol. 9, pp. 139–172, 1981.
[10] B. Efron and G. Gong, "A leisurely look at the bootstrap, the jackknife and the cross-validation," *Amer. Statistician*, vol. 37, pp. 36–48, 1983.
[11] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, part II, pp. 179–188, 1936.
[12] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, vol. 2, P. R. Krishnaiah and L. N. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, 1982, pp. 835–855.
[13] P. Lachenbruch and M. Mickey, "Estimation of error rates in discriminant analysis," *Technometrics*, vol. 10, pp. 167–178, 1968.
[14] D. F. Morrison, *Multivariate Statistical Methods*. New York: McGraw-Hill, 1976.

Anil K. Jain (S'70–M'72–SM'86), for a photograph and biography, see this issue, p. 620.

Richard C. Dubes was born in Chicago, IL. He received the B.S. degree from the University of Illinois in 1956 and the M.S. and Ph.D. degrees from Michigan State University, East Lansing, in 1959 and 1962, respectively, all in electrical engineering.

He is currently a Professor in the Department of Computer Science at Michigan State University.

Dr. Dubes is a member of the Pattern Recognition Society, the Classification Society, Sigma Xi, and is an Associate Editor of *Pattern Recognition*.

Chaur-Chin Chen (S'85) received the B.S. degree in mathematics from National Taiwan University, Taiwan, in 1977, and the M.S. degree from Michigan State University, East Lansing, in applied mathematics and computer science, in 1982 and 1984, respectively.

He is currently working toward the Ph.D. degree in computer science at Michigan State University. He has been a research assistant since April 1985. His research interests are in the areas of spatial point process, pattern recognition, and image processing.

Mr. Chen is a student member of the Pattern Recognition Society, the Association for Computing Machinery, and is a member of Phi Kappa Phi and Sigma Xi.