Title
Small sample statistics for classification error rates I: error rate measurements

Authors
Martin, J. Kent
Hirschberg, D. S.

Publication Date
1995-11-11

Peer reviewed

Small Sample Statistics for Classification Error Rates I: Error Rate Measurements

J. Kent Martin and D. S. Hirschberg
(jmartin@ics.uci.edu) (dan@ics.uci.edu)
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717
Technical Report No. 95-42
November 11, 1995

Abstract

Several methods (independent subsamples, leave-one-out, cross-validation, and bootstrapping) have been proposed for estimating the error rates of classifiers. The rationale behind the various estimators and the causes of the sometimes conflicting claims regarding their bias and precision are explored in this paper. The biases and variances of each of the estimators are examined empirically. Cross-validation, 10-fold or greater, seems to be the best approach, the other methods are biased, have poorer precision, or are inconsistent. (Though unbiased for linear discriminant classifiers, the 632b bootstrap estimator is biased for nearest neighbors classifiers, more so for single nearest neighbor than for three nearest neighbors. The 632b estimator is also biased for CART-style decision trees. Weiss LOO* estimator is unbiased and has better precision than cross-validation for discriminant and nearest neighbors classifiers, but its lack of bias and improved precision for those classifiers do not carry over to decision trees for nominal attributes.)

1 Introduction

The classification problem is: Given a finite set of classified examples from a population, described by their values for some set of attributes, infer a mechanism for predicting the class of any member of the population given only the member's attributes' values. Many methods have been proposed, most falling into one of the following four families: nearest-neighbors, discriminant analyses, decision trees or symbolic concept learners, and neural networks. Regardless of the inference method, there are three immediate questions: (1) given a classifier, how accurate is it? (usually, this can only be estimated), (2) given an estimate of accuracy, how accurate and how precise is the estimate (what are its bias, variance, and confidence interval)?, and (3) how much confidence can be placed in an assertion that one classifier is more accurate than another?

In this paper we deal with the first question and a portion of the second, with methods for estimating a classifier's accuracy and the bias and variance of the estimates obtained from various methods. A second paper [11] deals with the remainder of the second question, confidence intervals, and with the third, significance tests. The thesis of both papers is that *"...the traditional machinery of statistical processes is wholly unsuited to the needs of practical research ... the elaborate mechanism built on the theory of infinitely large samples is not accurate enough for simple laboratory data. Only by systematically tackling small sample problems on their merits does it seem possible to apply accurate tests to practical data." — R. A. Fisher [8] (1925)*

Given this thesis, it behooves us to provide guidelines as to when a sample is considered small, and when traditional methods will suffice. There is no hard rule here. The 150 instances in the Iris data, for instance, seem adequate for inferring an accurate classifier and for estimating its accuracy and confidence limits by traditional methods. In a more difficult problem (say, one having 16 classes and 100 attributes, contrasted to 3 classes and 4 attributes for the Iris data), a sample of 150 would be very scanty. Schaffer's [16] notion of the sparseness of the data relative to the concept to be learned helps to put this in perspective. Sample size is one component of the equation, complexity of the learned classifier another, and its error rate yet another. The interactions of these factors are discussed in conjunction with experiments in which they arise, and more quantitative guidelines are given in conjunction with specific methods.

There is a substantial body of literature on estimating expected error rates, and a clear consensus that some type of resampling technique is necessary to obtain unbiased estimates. These resampling methods fall into four main families: *independent subsamples* for classifier inference and error rate estimation [3, pp. 11-12], *leave-one-out and k-fold cross-validation* (subsampling without replacement) methods [3, pp. 12-13], *bootstrap* (subsampling with replacement) methods [7], and *hybrid methods,* such as Efron's [7] 632b bootstrap and Weiss' LOO* [20] method. The cross-validation methods are probably the most widely used, especially when the available samples are small, with the independent subsamples methods being preferred by some when very large samples are available. The bootstrap and hybrid methods are computationally expensive and poorly understood and, hence, not widely used.

There are conflicting claims in the literature as to the bias and precision of the various estimators, as well as to their power for testing differences between classifiers. In addition to a tutorial review of the various methods, this paper and the companion paper also present new and more extensive empirical studies and a framework for resolving the seemingly contradictory results.

In Section 2 of the paper, we give a short tutorial on issues relating to error rate measurements, introduce the various methods, and define the terminology used in the remainder of the paper. In Section 3 we present the results of extensive simulation studies on linear discriminant classifiers for very simple data, which reveal fundamental differences in the behavior (bias and precision) of the various methods.

Section 4 presents a brief review of pertinent literature which suggests that the behavior found for linear discriminant classifiers may not generalize to other classifier learning methods for some of the error rate estimators, especially when the classifiers are overfitted (*e.g.,* nearest neighbors and decision tree pruning methods). The results of extensive simulation studies on nearest neighbors and decision tree classifiers for simple, continuous attribute data are presented in Section 5, and Section 6 extends these studies to discrete attribute decision trees.

Significant findings from the various experiments are summarized in Section 7. Only the leave-one-out and cross-validation (10-fold or greater) methods exhibit consistent behavior across all of the learning methods.

2 Error Rate Terminology and Methods

In this section we provide a tutorial on issues relevant to measuring error rates, and define the terminology used in the remainder of the paper.

For practical purposes, a *population* is defined by a set of members, a set of classes, a set of attributes, and the procedures for measuring or assigning the classification and attribute values. Thus, any measurement errors, naming errors, inconsistencies, or omissions are characteristics of the population, not of an inference method.

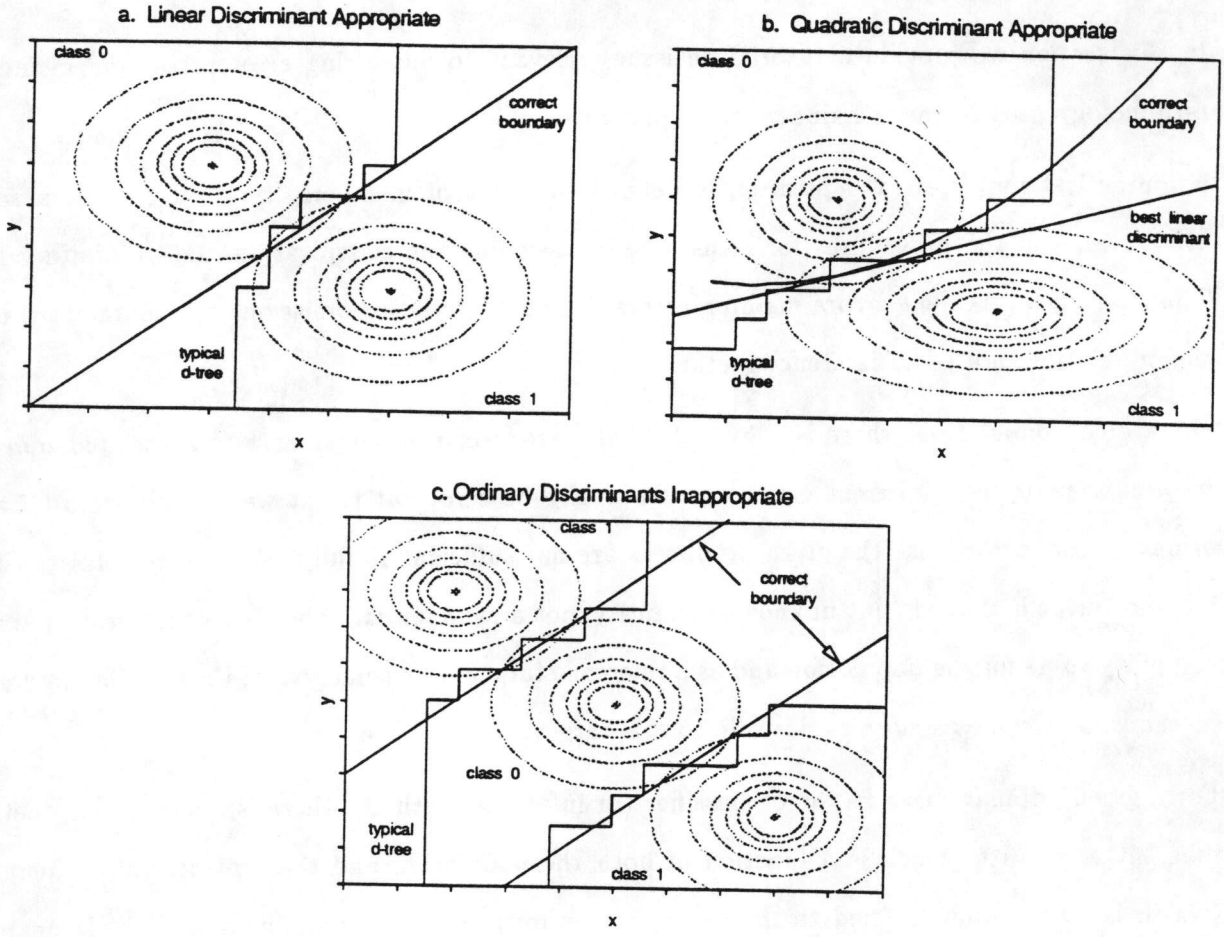
For a given population, there is a hypothetical least-error-rate classifier. Its associated *inherent minimum error rate* (inherent error)[1] would ideally be zero, but might well be non-zero because of data errors or because the given attributes are not sufficient to fully separate the classes. This inherent error is a fixed (but unknown) quantity, not a random variable. It is of interest here as a summary value for the population and as a reference target (the usual goal of the classifier inference methods is to come as close as possible to this target).

For a given population and type of classifier (or inference method), there is another hypothetical inherent error rate, which is a function of both the population and the representation language (which is often implicitly tied to the inference method). Linear discriminant and single nearest-neighbor classifiers, for instance, represent the boundaries between classes by a series of hyperplanes. If the least-error-rate classifier's boundaries are curved surfaces, methods using linear boundaries can only approximate those least-error boundaries. The hypothetical classifier which approximates those boundaries most closely has the *language-intrinsic minimum error rate* (language-intrinsic error), which is greater than the inherent minimum error rate. The language-intrinsic error is a fixed (but unknown) quantity, not a random variable. It is of interest here because it points out one reason that different inference methods can lead to classifiers with very different performance (other causes are the various search strategies and heuristics used).

These distinctions are illustrated in Figure 1. In Figures 1a and 1b, there are two classes labeled 0 and 1, and each class consists of a single multinormal distribution on the real-valued variables

[1]Throughout this paper, the terms error and error rate (meaning misclassification rate) will be used interchangeably. The term bias (rather than error) is used to refer to a systematic difference between an error rate estimate and the true error rate (non-zero average difference), the term precision is used to refer to the rms variability of such differences, and the terms variance or standard deviation to refer to the variability of a particular estimate.

Figure 1: Illustration of Language-Intrinsic Error



x and y, depicted by a set of contour lines of constant probability density. In this situation, the classifier which has least error (*i.e.*, the inherent error) is defined by a curve in the xy plane along which the probability density of class 0 equals that of class 1. When the covariance matrices of x and y for the two classes differ only by a multiplying constant (*i.e.*, when the contours have the same shape and orientation, but not necessarily the same size), this curve is a straight line, as illustrated in Figure 1a, and linear discriminant analysis is appropriate — more generally, the classes differ in the ratio of the x and y variances or in their covariance, and the boundary is a quadratic curve, as illustrated in Figure 1b. In Figure 1b we also show the linear boundary which has the lowest possible error rate for these data (the language-intrinsic error), and for both figures we also show typical CART-style decision tree boundaries. CART-style trees express the boundaries as step functions which can asymptotically approximate the true boundaries here (given a sufficiently large sample and inferring a very complex tree), but cannot exactly express the correct concept with a

finite classifier. Linear discriminant analysis has a fixed complexity, and cannot exactly express the correct concept in Figure 1b (nor even approximate it closely), regardless of the sample size. Of course, there are other data sets (especially those featuring nominal attributes) where CART-style trees are more appropriate, and even a quadratic discriminant cannot express those concepts well.

Figure 1c illustrates another case where ordinary linear and quadratic discriminant analyses fail. Here, there are 3 distinct subpopulations, but only two classes. The correct boundary in this particular case is a pair of parallel lines (the correct concept here is class=0 if $|y-x| \leq c$, else class=1). This case is superficially very similar to that in Figure 1a, but the correct boundary cannot be found or even closely approximated by the usual discriminant analysis because these data violate the fundamental assumption underlying those techniques — namely, that each class is homogeneous, closely approximated by a single multinormal distribution. The best that a linear discriminant classifier can do in this case is to set the boundary as a single line perpendicular to the correct boundary lines, and outside the range of the data, *i.e.,* to default to the simple rule of always guessing the more frequent class. Here, even though the decision tree boundaries are a poor approximation to the correct boundaries, they are a significant improvement over the usual linear discriminant (*i.e.,* a CART-style decision tree actually has a lower language-intrinsic error).

These examples illustrate the fact that, in choosing to use a particular learning algorithm (inference method), we are implicitly making assumptions about the population (the nature and distribution of the attributes and classes) and the language of a correct, minimum error rate classifier. As in all problems of statistical inference, probably the most crucial step is correctly matching these premises or underlying assumptions to the problem at hand.

References are frequently found in the literature (*e.g.,* in the classic CART text [3, pp. 13-17]) to a Bayes' rule or Bayes' or Bayesian classifier or rate. As defined by CART, the *Bayes' optimal error rate* is synonymous with the inherent error, in that any other classifier has at least this error rate. This Bayes' nomenclature is confusing, for two reasons: (1) the term Bayes' rule is sometimes used in the context of a particular kind of classifier (*e.g.,* a CART-style decision tree), of a "no data optimal rule" [3, pp. 178,354], or of finding a Bayes' optimal classifier for the sample — these are references to the language-intrinsic error, not to the inherent error, (*e.g.,* the ideal CART-style decision tree is not necessarily the best possible classifier) and (2) these terms are easy

to confuse with Bayes' Theorem and Bayesian statistical analysis — they might be misconstrued as any classifier inferred using Bayesian techniques (*e.g.*, AutoClass [4]), or as only those classifiers.

Given a population, a sample of N items from the population (the *sample* is here defined to be all data currently available for inference and testing), and a classifier inferred from the sample by some means, that classifier has a *true error rate* — the fraction of items that would be misclassified if the entire population could be tested. For any particular classifier, the true error rate is a fixed (but unknown) quantity, a function of the population and classifier, and not a random variable.

If only a random subset of Q items from the sample is used to infer the classifier (a *training set*, the unused items forming a *test set*), there is usually a very large number[2] of distinct possible training/test splits. Since the training set is random, the inferred classifier is random (if the experiment is repeated, a different result will probably be obtained, even though the population, sample, and inference method remain the same), due simply to the *random subsampling variance.*

Under these (random subsampling) circumstances, although the true error rate of the particular classifier is a fixed quantity, it is more appropriate to speak of the *true error rate of the subsampling inference method* — the expected (mean) value of the true error rates of these individual splits' inferred classifiers, averaged over all possible splits. A particular split's classifier's true error rate, or the average of the true error rates over several splits, is only an estimate of that expected value. Such an estimate is a random function of the population, sample, and inference method.

Since the sample is not the entire population, the true error rate of any particular classifier can only be estimated from this sample data by some method. When random subsampling is used in obtaining the particular classifier(s), this becomes a process of estimating the value of an estimate.

When more than one training/test split is used and estimated error rates averaged, a troublesome question arises: to exactly what classifier does this averaged estimated error rate correspond? (When the classifiers are decision trees, for example, there is no practical notion of what it would mean to average the classifiers.) To answer this question note that (as shown later, see Table 4) for any inference method, the inferred classifier whose true error is closest to the language-intrinsic error is most likely to be obtained by using the entire sample for classifier inference. Then, the solution is fairly clear: infer a classifier using all available data and some inference method, and

[2] $N!\ /\ Q!\ (N-Q)!$ if Q is fixed, otherwise $\mathcal{S}(2,N) = 2^{N-1} - 1$ (a Stirling number of the second kind [1]).

estimate that classifier's true error using one or another *estimator* (estimation method). One set of criteria for evaluating an estimator are the *bias* and *precision* with which it estimates that whole-sample true error, measured by the average and rms values of (EST – TER) over a wide range of populations and sample sizes (where EST is the estimated error and TER the true error).

As noted in the introduction, various estimators have been proposed:

- *Apparent error rate* — The fraction of items misclassified when testing on the same items used to infer the classifier (*i.e.*, on the training set). Sometimes called the resubstitution estimate [3, p. 11]. The apparent error rate is known to be biased (optimistic). In simple nearest-neighbor classifiers, for instance, every item in the training set is its own nearest neighbor, resulting in an apparent error rate of zero (if the data are consistent). This problem is sometimes solved by finding the nearest non-identical neighbor, which can be extended to other classifier types as the leave-one-out method (see below).

- *Independent subsamples* — The sample is randomly split into a training set from which the classifier is inferred and a test set from which the estimated error rate is later determined. Typically either one-half, one-third, or one-fourth of the sample is used for the test set.

- *k-fold cross-validation* — The sample is randomly divided into k approximately equal-size subsets. For each of the subsets, the remaining $k-1$ subsets are combined to form a training set and the resulting classifier's error rate determined on the reserved subset. A weighted average of the k error rate estimates is used (weighted for the test set size). For $k \ll N$, the entire procedure may be iterated many (typically 100) times and those results averaged. When k equals the sample size, N, the *leave-one-out* (LOO) estimate is obtained.

- *Bootstrapping* — A training set of size N is chosen randomly with replacement. Thus, each item in the size N sample may appear 0, 1, or more times in the training set (for large N, an average of $(1-1/e) = 63.2\%$ of the items will be used in the training set). Only those items which do not appear in the training set are used for the test set, and only once each. This procedure is iterated many (typically 200) times and the error rates averaged.

- *Hybrid methods* — Various combinations of the preceeding estimators have been proposed, such as Efron's [7] 632b bootstrap and Weiss' [20] LOO* method. The principal advantage

Figure 2: A Simple Classifier

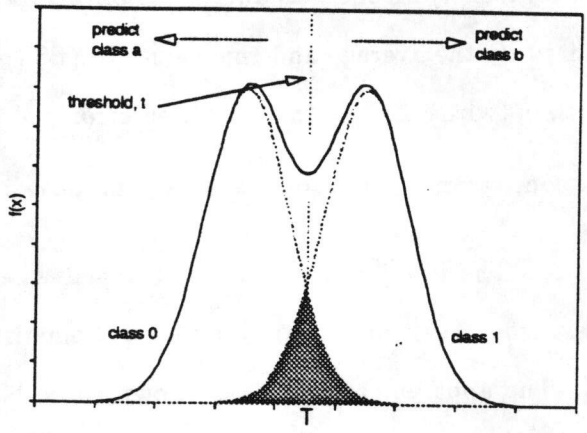
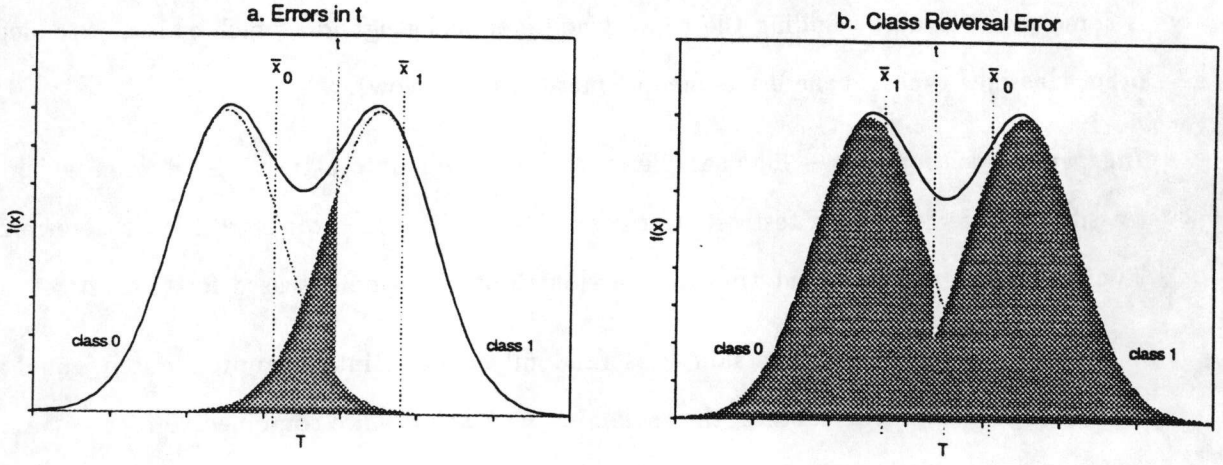


Figure 3: Sampling Errors



claimed for the 632b estimator is that, though biased, it has a lower variance than the other estimators. 632b is a weighted combination of the bootstrapping error rate (BOOT) and the apparent error rate (APP), 632b = 0.632 BOOT + 0.368 APP. Weiss' LOO* estimator is

$$\text{LOO}^* = \begin{cases} \text{632b}, & \text{if LOO} < \text{632b} \\ \text{2-CV}^*, & \text{if 2-CV}^* < \text{LOO and 632b} \leq \text{LOO} \\ \text{LOO}, & \text{otherwise} \end{cases}$$

where 2-CV* is 2-fold cross-validation iterated 100 times.

3 Experimental

Some essential properties of these various estimators can be shown using very simple data and a simple kind of classifier, as illustrated in Figure 2. The data population consists of two equally

likely classes (labeled 0 and 1), each normally distributed on a single real-valued attribute (x), with different means (μ_0 and μ_1, $\mu_0 \leq \mu_1$) but a common variance σ^2 (assume $\sigma^2 = 1.0$, without loss of generality). From a sample of size N, the inferred classifier is:

$$C(t, a, b) \equiv \text{if } (x \leq t) \text{ then class} = a \text{ else class} = b$$

where the threshold, t, and predicted class labels, (a, b), are determined from the training set (of size $Q \leq N$) using a simple linear discriminant procedure [10]. The least-error-rate classifier for these data would be $\mathcal{C} \equiv C(T, 0, 1)$, where $T = (\mu_0 + \mu_1)/2$ is the point where the two classes' density functions cross, as shown in Figure 2. The inherent error is equal to the shaded area in Figure 2, provided that the two distribution curves are normalized so that their combined area is unity. For this model, several things can go wrong:

1. All of the items in the training set could be the same class. For equally frequent classes, this is unlikely (the probability of this happening in a random sample of size Q is 0.5^{Q-1}). If one of the classes is rare, this can be a real problem even for large Q, and special sampling techniques may be needed. In these cases, the classifier always predicts whichever class is observed in the training set, the true error is 50%, and the apparent error is zero.

2. The class means in the training set might be equal (this is rare, but may happen when the x data are rounded values with few significant digits). In these cases, the classifier always predicts whichever class is more frequent in the training set, the true error is 50%, and the apparent error is the proportion of the other class in the training set.

3. The estimated threshold t can differ from T (as shown in Figure 3a), so that the induced classifier $C(t, 0, 1)$ has a greater true error rate than does $\mathcal{C} = C(T, 0, 1)$.

4. The training set's mean values for classes 0 and 1 might be reversed (as shown in Figure 3b), causing the true error rate to be very large. In the figure, this would be pretty rare for large training sets (but still with a finite likelihood). If μ_0 and μ_1 were closer together, however, the likelihood of this reversal would increase (reaching 50% when μ_0 and μ_1 are equal).

Monte Carlo techniques were used to generate 100 samples each of various sizes (10, 20, 30, 50, 100) from populations with different inherent error rates (50, 40, 25, 10, 5, 2, 1, and 0.1%). The $C(t, a, b)$

Table 4: Overall Results

Estimator †		ΔTER ‡	Bias	Precision
ISS	k=2	.012	−.001	.099
ISS	k=3	.007	.000	.113
ISS	k=4	.005	−.001	.135
APP	(k=∞)	0	−.015	.080
2-CV			.013	.100
5-CV			.002	.081
10-CV			.000	.079
LOO	(N-CV)		−.002	.082
2-CV	× 100		.013	.067
5-CV	× 100		.002	.069
10-CV	× 100		.000	.075
BOOT	× 200		.008	.065
632b			−.000	.063
LOO*			.002	.063

† ISS is independent subsamples, $Q = \lfloor (k-1)N/k \rfloor$
APP is apparent error, $(Q = N)$
k-CV is k-fold cross-validation
LOO is leave-one-out, N-CV
BOOT is bootstrapping
‡ ΔTER = average (TER(Training Set) - TER(Whole Sample))
where TER is the true error rate

classifier was calculated using the entire sample, and its true error rate was directly computed from our knowledge of the population's normality and its characteristics (μ_0, μ_1, and σ). The estimated error rates using the various methods were determined for each sample. Table 4 summarizes the mean bias and precision (the average and rms values of (EST − TER) over all 4,000 samples).

The independent subsamples (ISS) estimators appear to be unbiased, but they give unbiased estimates of the error of a less accurate classifier (on the average) than would have been inferred using the whole sample (see the column headed ΔTER). These estimators also have relatively poor precision. ΔTER is approximately proportional to k^{-1}, and the precision (excepting that of the apparent accuracy, APP) is approximately proportional to $k^{1/2}$, where $\lceil N/k \rceil$ is the test set size.

The apparent error rate (APP) is optimistically biased. Two-fold cross-validation (2-CV) is biased to about the same degree as the apparent error, but in the opposite (pessimistic) direction. Its

precision is somewhat poorer than the precision of APP, and about the same as for ISS with $k=2$. Since the driving force behind using the various re-sampling methods is that using the apparent error is unsatisfactory because it is biased [3, pp. 10-13], the 2-CV estimator does not appear to be useful. In common with the 50% ($k=2$) ISS method, it merely replaces the apparent error with an estimator having the opposite bias; further, its cost is greater than either APP or ISS. Iterating 2-CV 100 times significantly improves the precision, but does not remove its bias.

5-CV is less biased than 2-CV and 10-CV is virtually unbiased, and both have approximately the same precision as APP. Iterating these improves precision, but less so for 10-CV and less for 5-CV than for 2-CV. The precision of the bootstrap estimator is comparable to that of either 2-CV or 5-CV iterated 100×, and its bias lies between the biases of those estimators. These bootstrap and iterated cross-validation results are consistent, since the bootstrap uses 63.2% of of the sample items for training, which lies between the 50% used by 2-CV and the 80% used by 5-CV. The leave-one-out (LOO) estimator seems to have no significant bias, and approximately the same precision as APP, 5-CV, and 10-CV. It would be pointless to iterate leave-one-out, since the same result would be obtained on each iteration. As k increases, the improvement in precision resulting from iterating k-CV decreases rapidly, vanishing as $k \rightarrow N$.

The ISS methods (including APP) have the lowest computational costs among these estimators, because only a single classifier need be inferred for ISS. The cost increases with the training set size, so that APP is the most costly and 50% ISS ($k=2$) the least costly of these ISS methods. For small samples, the methods seem to offer a choice between a poorer classifier and a less precise estimate (ISS) or a better classifier but a biased estimate (APP).

For k-CV, we must infer $k+1$ classifiers, and the computational cost of cross-validation ranges from about 1.5× the APP cost for 2-CV to about N× the APP cost for LOO. For the simple classifier problems summarized in Table 4, there is little to choose (so far as bias and precision are concerned) between 5-CV, 10-CV, and LOO. CART reports that 10-CV is approximately unbiased even for more complex problems, and most researchers use 10-CV rather than LOO because of its lower cost (the LOO cost increases more rapidly with sample size than does the 10-CV cost).

The iterated cross-validation and bootstrap methods have relatively high computational cost, since 100 or 200 classifiers must be inferred for these methods. The improvement in precision gained

by iterating is not significant for 10-CV, and it seems necessary to accept an increased risk of bias (5-CV, 2-CV, or bootstrapping) as well as a higher cost to achieve a significantly improved precision. Considering all the factors (lack of bias, precision, and computational cost), uniterated 10-CV seems to be the best of the cross-validation methods for these linear discriminant classifiers.

The 632b hybrid has about the same precision and cost as the unmodified bootstrap without its bias, and 632b is thus a better estimator in these cases than the unmodified bootstrap. The LOO* hybrid has greater cost than, and about the same precision as 632b. Efron's 632b hybrid seems to be the best of the bootstrap and hybrid methods for these simple classifiers. 632b has better precision than uniterated 10-CV, but higher computational cost (by a factor of about 20).

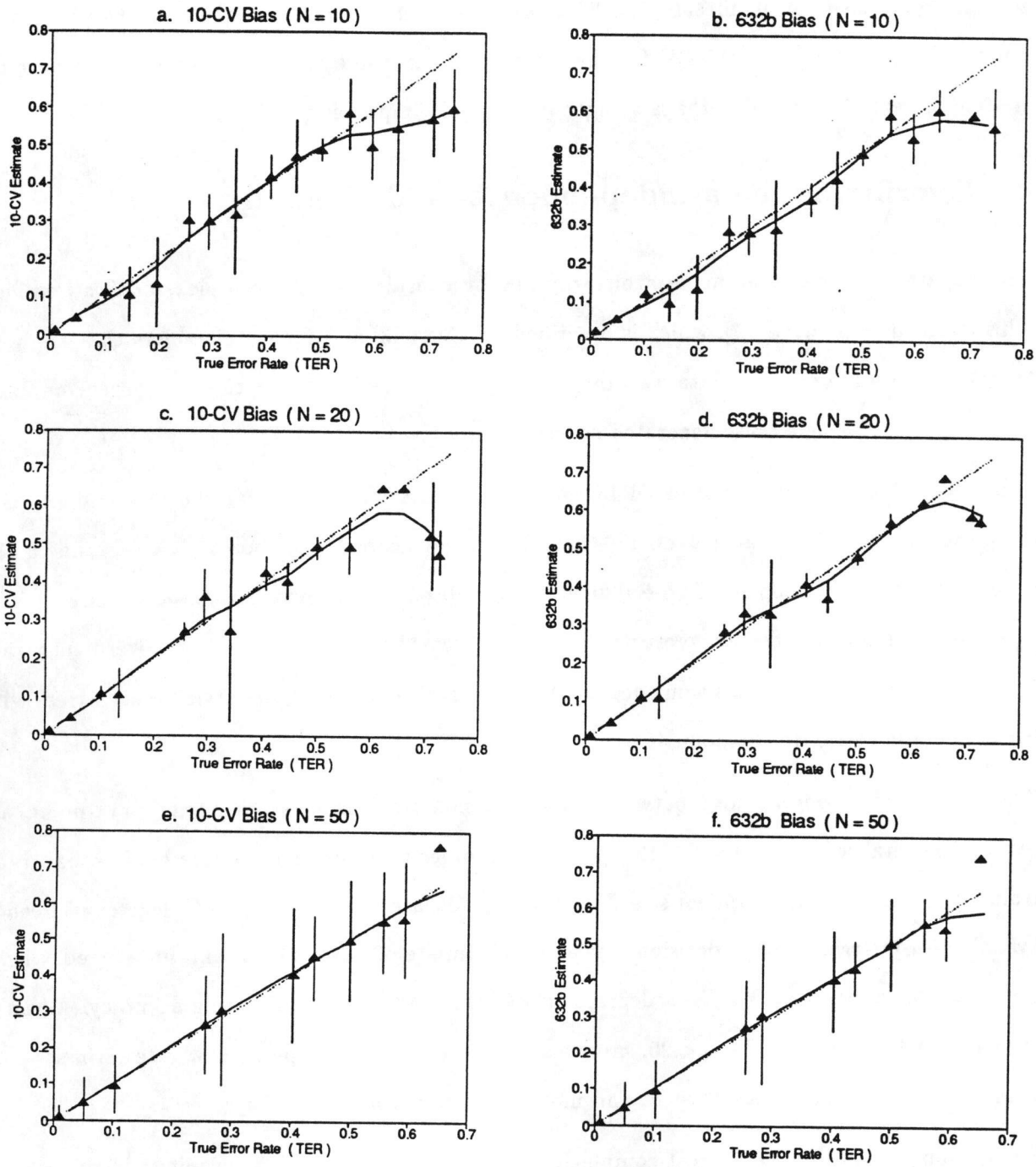
The 632b results are surprising in light of Efron's [7] original results showing that 632b was optimistically biased. Efron's [7] 632b conclusions were based on only 5 experiments, all at very low sample sizes ($N = 14$ or 20), high inherent error (about 31%, only 0.5σ separation of the classes), and multivariate normal distributions. Recent work by Davison and Hall [6] and by Fitzmaurice, *etal* [9] showed that the differences in bias and variability emerge strongly only when the populations are very close or the sample sizes very small.

Figure 5 shows the bias for small samples ($N = 10, 20, 50$), for all populations in the test relative to the true error rate. Each point in Figure 5 represents the average error rate within an interval defined by (true error rate = $.05k \pm .025$, for $k = 0 \ldots 20$), with a vertical bar representing the $\pm 2\sigma$ limits for each of the points. These data were then smoothed,[3] which is shown as a solid curve in Figure 5, contrasted to the dashed line representing zero bias. The curves all show an optimistic bias when the true error rate is very high (above about 50%), that is, when the training set's class means are reversed (see Figure 3) or the class proportions in the training set are very different from those in the population. Both of these conditions arise from a subsample that is not representative of the population, and each is more likely to occur in very small samples.

In the middle range of true error (between 0.1 and 0.5), 632b has a consistent bias for $N = 10$ which is absent for $N > 20$. 10-CV is relatively unbiased in this region for all sample sizes. Both are biased at very high error rates, especially for very small samples. (At such high error rates and small samples, this is likely to be of little consequence because these classifiers are not useful.) The

[3] By applying the filter $z_k = (y_{k-2} + 2y_{k-1} + 3y_k + 2y_{k+1} + y_{k+2}) \,/\, 9$.

Figure 5: Small Sample Bias of 10-CV and 632b



precision of a biased estimator (b) is $\sqrt{\text{bias}^2 + \text{variance}(b)}$, while for an unbiased estimator (u) the precision is $\sqrt{\text{variance}(u)}$. If variance($b$) < variance($u$) and bias^2 < variance($u$) − variance($b$) then the biased estimator b is less likely to stray too far from the truth[4] than is the unbiased estimator u. Though biased for small samples, 632b appears to meet these criteria, and may be a more trustwothy estimator than 10-CV for these problems. In these cases, a tradeoff might be made, trading increased cost and perhaps a slight bias to gain improved precision.

4 Overfitting, Non-independence, and Generality

The apparent error can be made arbitrarily low by considering very complex, *ad hoc* classifiers. This is called *overfitting* [17], which is described by CART [3] as inferring classifiers that are larger than the information in the data warrant, and by ID3 [14] as increasing the classifier's complexity to accomodate a single noise-generated special case.

Weiss' LOO* estimator is motivated by empirical results indicating that the bias and precision relationships shown in Table 1 do not hold for single nearest neighbor (1-NN) classifiers [18], especially for small samples. These difficulties are absent or strongly mitigated in three nearest neighbors (3-NN) classifiers, suggesting that the problems are due to the extreme overfitting which is characteristic of 1-NN. This same degree of overfitting is found in CART-style decision trees when every numeric attribute cut-point is used.

There is a very strong analogy between 1-NN, decision trees using every numeric cut-point, and fitting a generalized linear model [13] where the number of adjustable parameters is equal to the training set size. For a sample of size N containing C classes, there are $N-C$ degrees of freedom available for inferring both a decision tree and an estimate of its accuracy; and for a tree having L leaves, there are at most $N-C-L$ degrees of freedom available for estimating accuracy. As a rule of thumb, whenever $N-C-L < 20$, cross-validation and bootstrapping error rate estimates and their confidence intervals are suspect and more complex methods may be necessary.

Post-pruning strategies (*e.g.*, cost-complexity [3] and reduced-error [15] pruning) begin with an overfitted tree and seek a most accurate and least complex pruned version of that tree. Error

[4] This is a question of the confidence interval, the interval within which, given the value of the estimate, we expect with high confidence to find the true error rate. These issues are addressed more fully in the companion paper [11].

rate estimates for the series of candidate trees generated during post-pruning are subject to all the difficulties of 1-NN classifiers, and to the additional difficulty that the trees and their error rates are not independent. Breiman, *etal* [3], adopted the 1-SE rule (in a series of pruned trees, choose the simplest tree whose 10-CV error rate is no more than 1 standard error greater than that of the tree having the lowest 10-CV rate) to deal with the lack of independence. Weiss & Indurkhya [19] recommend a novel form of iterated (10×) 2-fold cross-validation for cost-complexity pruning.

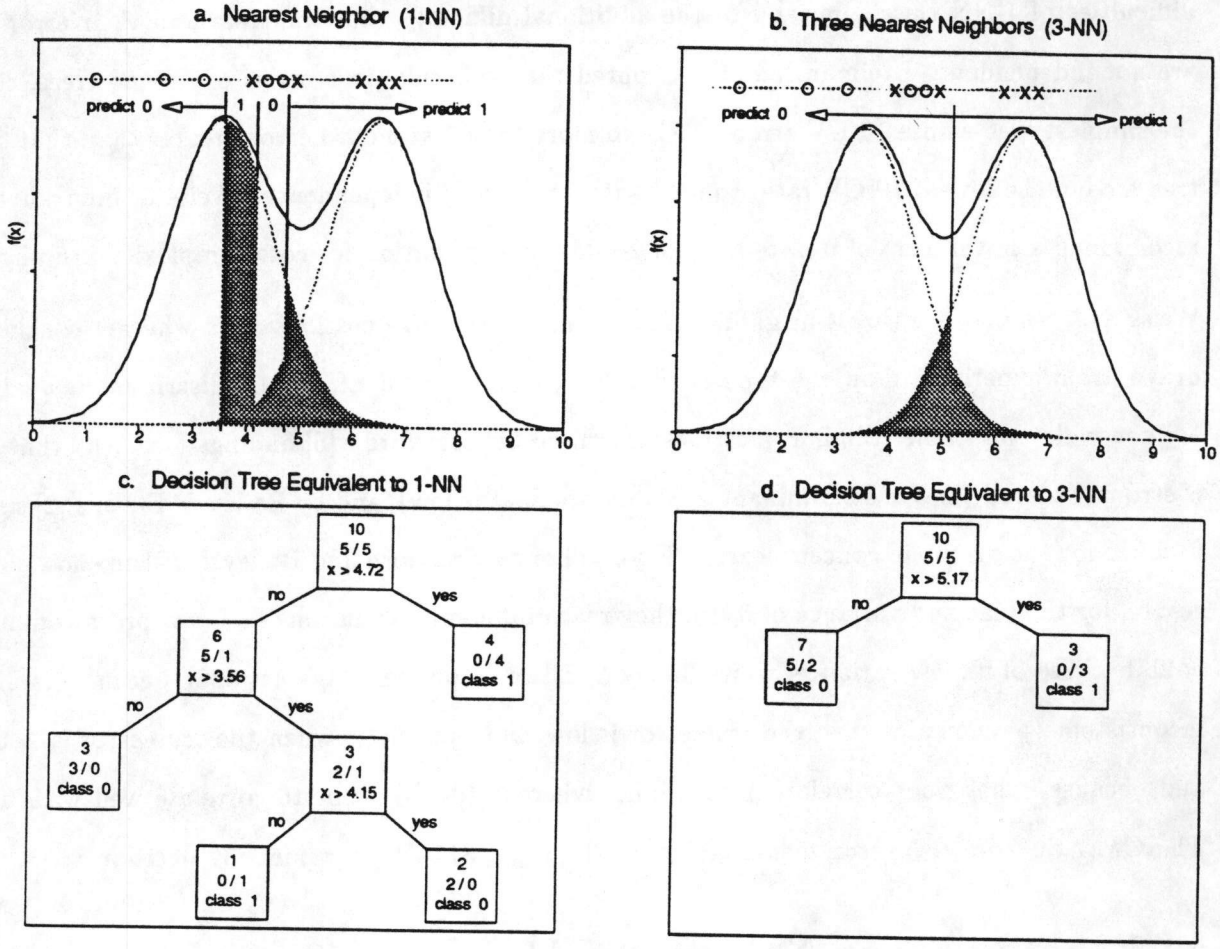
Weiss' [18] results for nearest-neighbors classifiers raise serious questions as to whether conclusions drawn from experiments on one type of classifier (as in our and Efron's [7] discriminant analyses) are generally applicable. Similar questions are raised by Crawford's [5] findings for CART that 632b is strongly biased (in our discriminant analyses, the bias is low), and by Bailey & Elkan's [2] similar findings for the symbolic concept learner FOIL. Though Crawford and Bailey & Elkan show similar results for the bias and variance of 632b, they reach different conclusions — Crawford recommends 632b because of its low variance, while Bailey & Elkan recommend against 632b because its bias is inconsistent (pessimistic when the true error is low, but optimistic when the true error is $> 30\%$) and because it has poor correlation with TER whereas 10-CV seems to correlate well with TER. The CART authors also recommend against 632b [3, pp. 311-313] for inferring decision trees.

5 Experiments on Nearest-Neighbors Classifiers

In Figure 6 we illustrate nearest-neighbors classifiers derived from a small sample from the same sort of population as was used in our discriminant examples. In Figures 6a and 6b we show the population density function and, above that, the location along the x-axis of the elements of a sample of 10 items from that population ('O' denoting a class 0 and 'X' a class 1 item) — the population and sample are the same in both cases. We also show the class predictions and boundaries, and the true error (shaded area) for two nearest-neighbors classifiers, single-nearest-neighbor (1-NN) in Figure 6a and three-nearest-neighbors (3-NN) in Figure 6b.

We note the following features in the figures: (1) 3-NN may make different predictions for adjacent sample items which have the same class (as is also the case in discriminant analysis), while 1-NN cannot do this; (2) 1-NN is sensitive to outliers (isolated instances near the extremes of a class distribution), which can lead to a very high error rate for a small sample, while 3-NN smoothes

Figure 6: Nearest Neighbor Classifiers and Equivalent Trees



these fluctuations in the sample density; and (3) 1-NN always tends to overfit the sample (*i.e.*, to infer an overly complex classifier), while 3-NN does not (it may overfit, underfit or, as in Figure 6b, be about right, depending on the sample and population).

In Figures 6c and 6d we show decision trees corresponding to the nearest-neighbors classifiers of Figures 6a and 6b, respectively. Note that Figure 6d is not merely a pruned version of Figure 6c. Also note that the classifiers depicted in Figures 6a and 6c are entirely equivalent, as are the pair depicted in Figures 6b and 6d — for mutually exclusive classes, any deterministic classifier, in whatever form, can also be expressed as a decision tree, or as a set of rules in disjunctive normal form (DNF) (albeit that the translation may be non-trivial). The tree shown in Figure 6c is equivalent to that which would be inferred by CART or ID3 using the usual method of placing cut-points for continuous variables midway between adjacent items having different classes (a slightly

different tree would be inferred using C4.5's [15] method of placing cut-points only at one of the values occuring in the sample). The tree shown in Figure 6d is not equivalent to that which would be inferred by CART or ID3 using the usual methods, but could be inferred by these algorithms if 3-NN's method for placing cut-points were substituted.

Thus, we must be careful not to over-generalize conclusions from experiments involving different induction algorithms, as in making assertions that X is true for decision trees, but not for nearest-neighbors. As we have illustrated, a decision tree and a nearest-neighbors classifier are not inherently different things[5]. We emphasize that observed differences in behavior are the result of differences in the classifier induction algorithms and error estimation methods and interactions between them, and not due to the format in which the classifier is represented. Differences in induction algorithms may express themselves in either or both of two ways:

1. Through differences in the language (not the format) which the algorithm uses to express concepts. CART, for instance, uses DNF where the elements (individual propositions) are assertions about the value of a single attribute, whereas linear discriminant analysis utilizes a single assertion about the value of a linear function of all of the attributes. There is a very significant qualitative difference between $(p-q) < 0.5$ and $(p < 0.5) \vee [(p > 1) \wedge (q < 0.5)]$, and there are concepts which can be expressed correctly in the language used for the first example, but not the language used for the second, and *vice-versa.*

2. Through differences in the search patterns of algorithms when the language is the same. These differences are illustrated in our single real-valued attribute examples. All of these classifiers consist of a set of n cut-points and class predictions $(t_1, p_1) \cdots (t_n, p_n)$ where the prediction rule is: predict class p_i for $t_{i-1} < x \leq t_i$ where $t_0 = -\infty$ and $t_{n+1} = +\infty$. Though the language is identical, the set of potential values of the t_i for a fixed given sample differs from one algorithm to another, both in the number of cut-points allowed and in the permitted values. (Linear discriminant analysis allows only one cut-point, 1-NN considers all $(x_j + x_{j-1})/2$ in the sample as potential values for t_i with a maximum $n = N - 1$, and 3-NN considers all $(x_j + x_{j-3})/2$ in the sample as potential values for t_i with a maximum $n = N - 3$.)

[5] Although decision tree algorithms may be able to deal with nominal attributes for which there is no meaningful *a priori* concept of distance. We can always translate a nearest-neighbors classifier into an equivalent decision tree, but the converse is not always true.

A series of experiments was conducted to explore the behavior of 1-NN and 3-NN classifier error rates for populations similar to that shown in Figure 6 — 20 random samples each of various sizes ($N = 10, 20, 30, 50, 100$) for populations with different inherent error (0.1, 1, 2, 5, 10, 25, 40, 50%). Both a 1-NN and a 3-NN classifier were calculated from each sample, and TER, APP, LOO, 10-CV, 632b, and LOO* error rates calculated for each classifier.

In Figure 7 we show the mean of each estimator plotted *vs.* the mean TER for each of the 40 experiments. LOO and 10-CV give virtually the same results, and only 10-CV is shown — both are unbiased but have high variability for both 1-NN and 3-NN. The APP and 632b results are less variable than LOO or 10-CV, but they are biased and their biases are different for 1-NN than for 3-NN. LOO* is approximately unbiased for these classifiers, but highly variable. The 632b variances are essentially the same for both 1-NN and 3-NN, and lower (by about 40%) than the LOO variances. LOO* has roughly the same precision as LOO overall, but has a lower variance for small samples and high error rates.

Detailed examination of the data shown in Figure 7 verifies Weiss' [18] findings that the lack of bias and improved precision of 632b applied to linear discriminant classifiers do not carry over to nearest neighbors methods, especially to 1-NN. The LOO and 10-CV results (their lack of bias and relatively high variance), however, apparently do carry over. Weiss' LOO* estimator, developed for nearest neighbors, is approximately unbiased for both discriminant and nearest neighbors classifiers. For discriminant classifiers, LOO* has about the same variance as 632b, but LOO* has a higher variance than does 632b for nearest neighbors methods.

6 Experiments on Decision Tree Induction

The discriminant and nearest neighbors classifier results were all derived from continuous attributes. To test whether those findings generalize to non-numeric attributes, a series of experiments was conducted using the contact lens prescription data set [12]. In this artificial problem, patients are classified into 3 categories (hard, soft, none) based on the values of 4 attributes (1 tertiary and 3 binary). The 24 instances given are complete (cover all cases) and noise free. Figure 8 shows a correct decision tree for this problem (other correct trees, permuting the order of the attribute splits, are possible — the 9 leaves are necessary and sufficient).

Figure 7: Mean Error Rates of 3-NN and 1-NN Classifiers

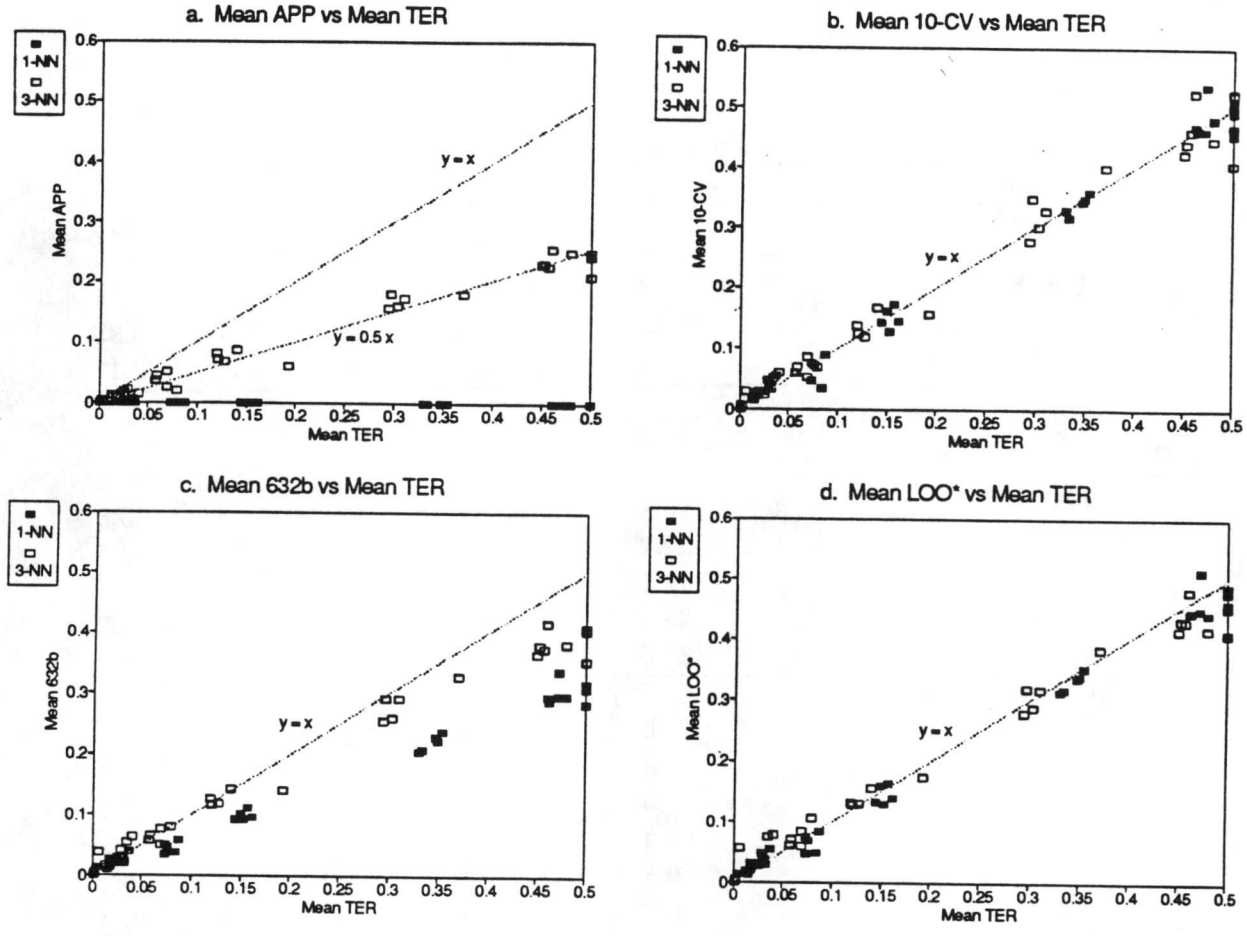


Figure 8: A Correct Decision Tree for Contact Lens

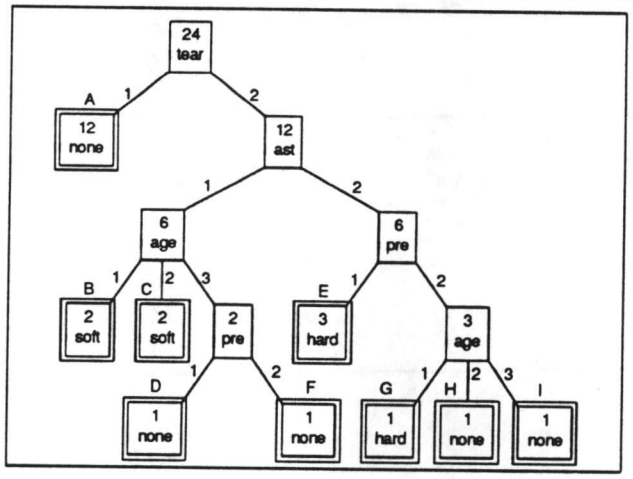


Table 9: Contact Lens Resampling Decision Trees

	Original Data Set	Sampling with Replacement $N=24$	$N=48$	$N=72$	$N=96$
Tree Size:					
No. Nodes	15	9	14.7	15	15
No. Leaves	9	5.3	8.7	9	9
Depth	4	3.33	4	4	4
Avg. Depth	3.33	2.58	3.26	3.33	3.33
Wtd. Avg. Depth	2.21	1.83	2.24	2.29	2.17
Error Rate (%):					
TER	0	12.5	2.8	0	0
APP	0	0	0	0	0
LOO	20.8	5.6	4.2	0.5	0.5
10-CV	22.2	5.6	3.8	0.9	0.5
632b	17.6	10.0	5.3	2.0	1.1
2-CV*	29.5	21.8	11.1	5.6	3.2
LOO*	20.8	10.6	6.1	2.0	1.1
Std. Deviation (%):					
TER	0	8.7	3.4	0	0
APP	0	0	0	0	0
LOO	0	6.8	2.9	0.7	0.6
10-CV	3.4	6.3	2.4	1.7	0.6
632b	0.5	3.9	1.5	0.4	0.4
2-CV*	0.8	6.6	3.6	0.6	0.4
LOO*	0	4.6	1.4	0.4	0.4
RMS (EST - TER):					
APP	0	14.8	4.2	0	0
LOO	20.8	8.3	5.0	0.8	0.7
10-CV	22.4	8.0	4.4	1.8	0.7
632b	17.6	10.6	5.5	2.0	1.1
2-CV*	29.5	22.6	11.6	5.7	3.2
LOO*	20.8	11.4	6.2	2.0	1.1
Correlation with TER:					
APP					
LOO		-.47	.35		
10-CV		-.63	.35		
632b		-.76	-.63		
2-CV*		-.74	-.61		
LOO*		-.64	-.33		

In the second column of Table 9, headed 'Original Data Set', we summarize the results for various error rate estimators applied to these data (the table summarizes the results of 6 repetitions). It is interesting that, in this case, the apparent error rate (APP) is correct, while the various resampling estimates all show a strong pessimistic bias. These techniques are not applicable for this particular data set because two of the key assumptions underlying the methods do not hold, namely, that the data set is a random sample from a large population and that the data contain errors. These results underscore the important point that the estimation of error rates is a statistical inference, working from a set of observations and premises (*a priori* assumptions about the data, many of which are implicit in the methods but not explicitly stated) — the results of applying a method may be nonsensical if its premises are not satisfied by the data.

But, suppose that the first assumption does hold and that, by chance, we happened to draw a minimal complete sample. The remaining columns in Table 9 show the results of averaging 6 random samples each of various sizes from this larger population (which is equivalent to simply sampling the 24 original items with replacement). In each case, the true error rate TER is determined by testing the inferred tree on the original 24 complete cases. For samples of $N = 24$ items, 2-CV* is pessimistically biased, while all of the other estimates are optimistic. 632b and LOO* have the least bias and variance, but LOO and 10-CV are closest to the true error as measured by the RMS. For $N \geq 48$, all of the resampling estimates are pessimistically biased, more so for 2-CV* than for the other estimators, and more so for LOO* and 632b than for LOO or 10-CV. The 632b and 2-CV* methods appear to be more strongly correlated with true error than LOO and 10-CV. However, these correlations are negative, which is undesirable in the sense that the estimators diverge from the true error rate. (Note that the change from optimistic to pessimistic bias for LOO and 10-CV between $N = 24$ and $N = 48$ is accompanied by a change in the sign of the correlation).

One argument for 632b and LOO* is that, though they have a greater bias than 10-CV or LOO, they have a lower variance and may, therefore, be more powerful for distinguishing between competing classifiers. We address this question more fully in the companion paper [11], but the data in Table 9 raise some interesting points:

1. 632b and LOO* are not always more biased than LOO and 10-CV. The RMS precision does appear to be always lower for LOO and 10-CV, but the mean difference $\overline{(\mathrm{EST}-\mathrm{TER})}$ is

smaller for 632b and LOO* at $N = 24$. Also note that LOO and 10-CV are optimistic for $N = 24$, but pessimistic for $N \geq 48$, while 632b and LOO* are consistently pessimistic.

2. Minimal variance alone is not the proper criterion. APP has the least variance of any of the estimators, yet it has no ability to distinguish among the various trees (it predicts a zero error rate for every tree inferred from these data).

3. What we want is an estimator that is correlated with TER, *i.e.*, that a difference in the estimate implies a difference in TER. On this basis, the limited data in Table 9 suggest that 632b or 2-CV* might be better for smaller samples. The negative sign of the correlation is troubling, however, as this implies that, having concluded that the TER's of two classifiers are different, the one with the higher estimated error rate will actually perform better.

However, we caution that conclusions drawn from the data in Table 9 may not generalize well, since the population for these data is free of attribute or class errors (the data are correct, though noisy because of random resampling variation).

An additional set of experiments was conducted, simulating the presence of attribute and class errors in a manner such that the inherent error of the population is controlled — let p be the desired inherent error, then let each of the 24 possible attribute value combinations be equally likely but let the class labels in our infinite population be randomly assigned as follows:

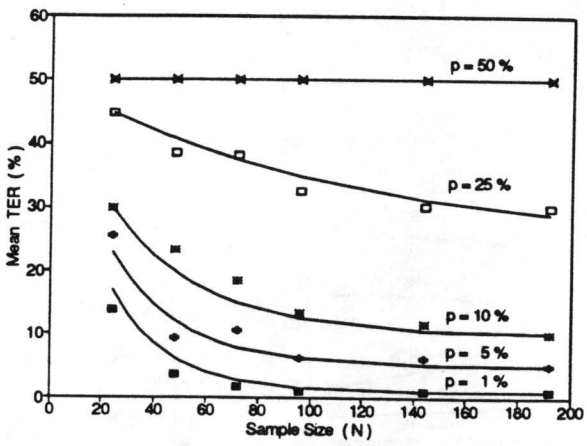
$$\text{class label} = \begin{cases} \text{correct label}, & \text{with probability } 1-p \\ \text{correct label } \textit{modulo } 3 \ + \ 1, & \text{with probability } p \end{cases}$$

(Note that this treatment simulates attribute errors as well as class errors, since there is no way to distinguish whether a tuple such as (11111) is the result of an error in the class label or in the values of one or more attributes).

Our experiments simulated 6 samples each of several sizes from populations with different inherent error rates. In these experiments, TER is calculated using the 24 base cases, as follows (where class_i is the true class of the i^{th} case and prediction_i is the tree's prediction for the case):

$$\text{TER} = \frac{1}{24} \sum_{i=1}^{24} \begin{cases} p, & \text{if class}_i = \text{prediction}_i \\ 1-p, & \text{if class}_i \neq \text{prediction}_i \end{cases}$$

Figure 10: True Error Rate *vs.* Sample Size & Noise Level



In Figure 10 we show the mean TER's of the various populations and sample sizes. TER approaches the inherent error asymptotically from above (in general, TER converges to the language intrinsic error, not the inherent error; here, the two are equal). The smooth curves shown in Figure 10 capture a general behavior which has great practical significance: (1) larger training samples tend to yield more accurate classifiers, (2) the true error rate is bounded below by the language intrinsic error (if the problem is ill-suited for the inference method, we may not be able to infer a good classifier, regardless of the sample size), and (3) the larger the intrinsic error of the population and inference method, the more slowly does TER approach its asymptotic value as the sample size increases (the greater the noise level or the more ill-suited the problem and inference method, the greater the sample size required to achieve a near-asymptotic error). Similar curves for the means of APP and LOO are shown in Figure 11. Note that APP approaches its asymptotic level from below, while TER, LOO, and the other re-sampling estimates approach from above.

The relationship of the various estimators' means to the mean true error for various noise levels and sample sizes is shown in Figure 12. APP is optimistically biased (about 60% of TER), and highly variable. LOO and 10-CV are unbiased and have about the same precision. 632b is pessimistically biased for low ($< 10\%$) error rates, and optimistically biased for higher error rates. The standard deviation (vertical spread) of 632b is much lower than that of LOO or 10-CV, but its rms precision is poorer, due to its bias. 2-CV* has a consistent pessimistic bias, and is more variable than 632b. The bias of LOO* is similar to that of 632b, but it is much more variable. For these decision trees, the precisions of LOO* and 632b are poorer than that of 10-CV or LOO.

Figure 11: Estimators *vs.* Sample Size & Noise Level

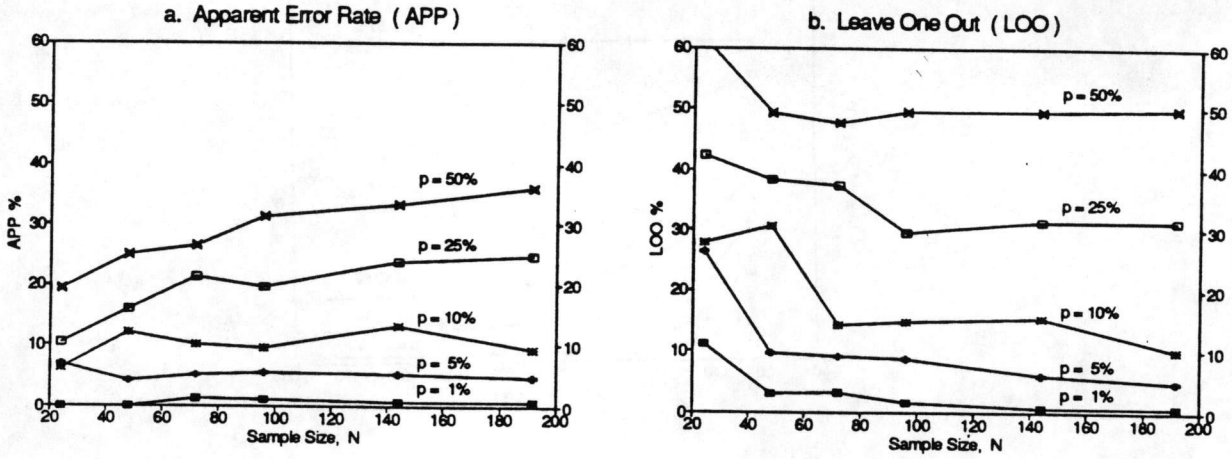


Table 13: Correlation of Repeated Sample Estimates

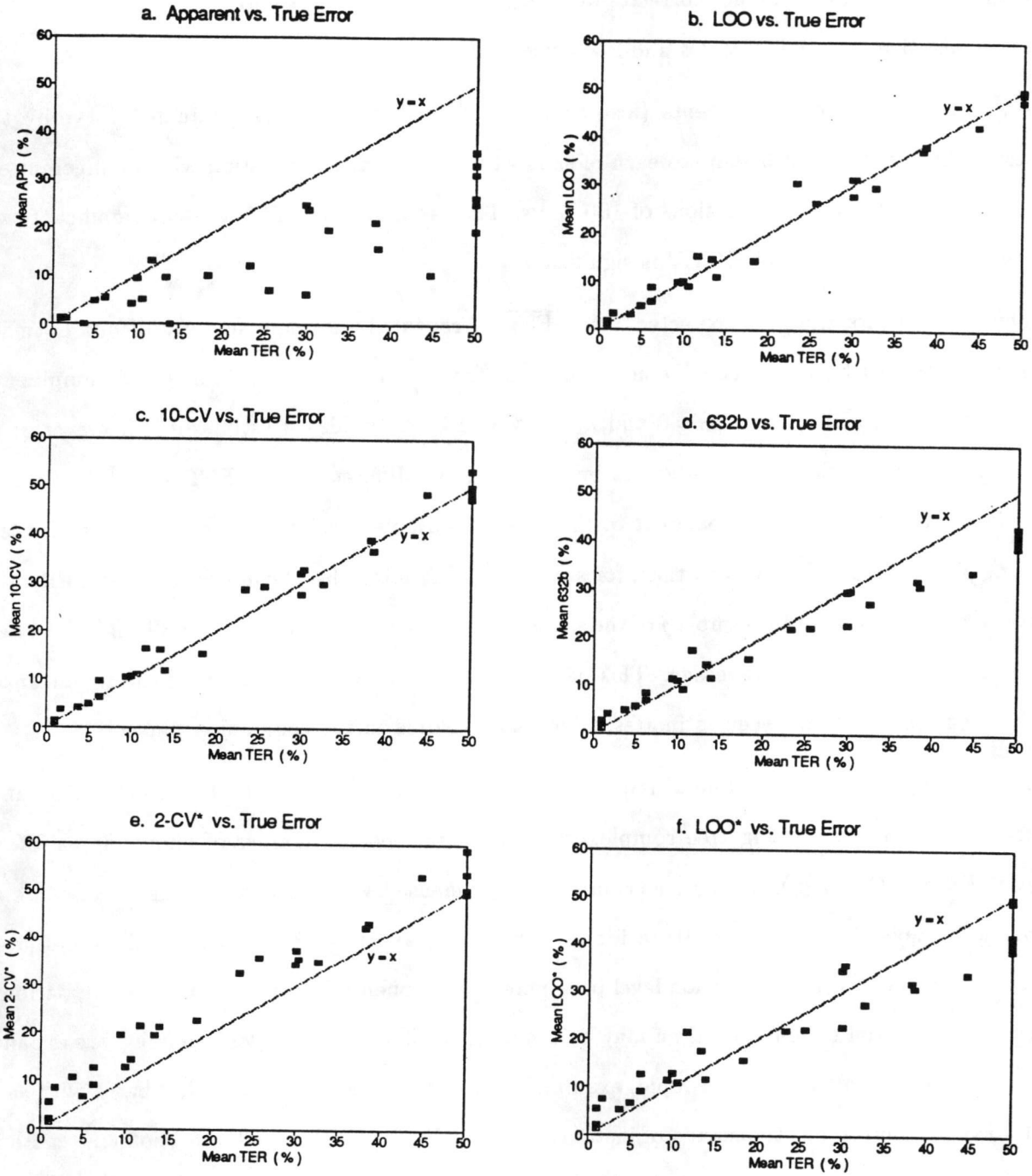
Six Samples of Each Size and Inherent Error

Sample Size	p	0.01	0.05	0.10	0.25	0.01	0.05	0.10	0.25	0.01	0.05	0.10	0.25
		10-CV				632b				LOO*			
		Correlation with TER											
24		−.65	.54	.69	−.06	−.38	.47	.75	.17	−.38	.47	.75	.17
48		−.77	.55	.84	.49	−.68	.45	.89	.45	−.74	.45	.89	.45
72		.88	.35	.83	−.17	.75	.11	.48	.11	.58	.07	.48	.11
96			.44		.21		.55		.40		.58		.37
144			.49	.44	.11		.53	.34	.21		.49	.33	.18
192					.18				.08				.11
		Correlation with 10-CV											
24						.70	.99	.92	.93	.70	.99	.92	.93
48						.72	.90	.98	.92	.68	.90	.98	.92
72						.95	.87	.80	.76	.80	.87	.80	.76
96						.54	.83	.98	.87	.53	.80	.93	.85
144						.99	.93	.85	.98	.96	.94	.81	.98
192						.99	.99	.93	.96	.94	.99	.92	.95

Correlation of 200 Samples' Estimates for p=0.01

$N = 24$		$N = 96$	
10-CV = 15.6 − 0.19(TER)	$r = -.19$	10-CV = 2.0 + 0.06(TER)	$r = +.05$
632b = 14.0 − 0.16(TER)	$r = -.28$	632b = 2.7 + 0.03(TER)	$r = +.03$
632b = 6.6 + 0.42(10-CV)	$r = +.78$	632b = 1.4 + 0.66(10-CV)	$r = +.80$

Figure 12: Mean Estimated Error *vs.* Mean True Error



All of the charts in Figure 12 show a fairly strong positive correlation between the various estimators' means and the mean TER. However, there is no such correlation of the individual estimates and TER's within a replicated experiment, as shown in Table 13 — though the estimators correlate with one another, they do not correlate well with TER (and the weak correlation with TER appears to be negative for small samples and low noise levels).

An additional set of experiments (also summarized in Table 13) was conducted to verify these results by simulating 200 samples each of sizes 24 and 96 from a population with an inherent error of $p = 0.01$. The weak correlations of 10-CV *vs.* TER and 632b *vs.* TER are not significant, while the correlation of 632b *vs.* 10-CV is significant.

Thus, it appears that the expected value $\overline{\text{EST}}$ of repeated sampling for any of our estimators is correlated with the expected true error $\overline{\text{TER}}$ for the trees inferred from these samples, *i.e.*, $\overline{\text{EST}} \approx k_0 + k_1 \overline{\text{TER}}$ (where $k_0 = 0$ and $k_1 = 1$ would be an unbiased estimator). However, it also appears that, for the i^{th} individual estimate EST_i, the difference $\Delta_i = \text{EST}_i - \overline{\text{EST}}$ is a random variable, and that Δ_i is independent of the random variable $\delta_i = \text{TER}_i - \overline{\text{TER}}$ which is of interest (*i.e.*, $E(\Delta_i \delta_i) = 0$). This means that, for sample i, EST_i might be above average and TER_i below average, while for another sample j of the same size from the same population, EST_j might be below average and TER_j above average. This has important consequences regarding the significance of observed differences between estimates, which are explored in the companion paper [11].

In these experiments, the time T required to infer a tree increased with both increasing sample size N and with increasing tree complexity η (measured by the number of nodes in the inferred tree) $T \approx k_0 + (k_1 + k_2 \eta) N$. The tree complexity η increased with both increasing sample size and increasing noise level, nearing saturation ($\eta = 46$, or 24 leaves) for the most noisy data and largest sample. In general, the saturation level of η increases exponentially with the number of attributes, and this potential exponential time may be a matter of concern as problem domains are expanded beyond the current, relatively simple, example data sets to large, real-world databases with scores of noisy attributes and thousands of instances (especially for the iterated and bootstrap methods, which must infer several hundred classifiers for each sample) .

7 Conclusions and Recommendations

1. 10-fold cross-validation (10-CV) appears to be the best method for estimating the whole-sample classifier's error rate. Its lack of bias and its precision are equivalent to those of the leave-one-out method, at lower computational cost.

2. The single independent subsamples (ISS) method results in a classifier with poorer expected accuracy and significantly greater variance than 10-CV.

3. Iterating k-fold cross-validation reduces its variance, but the effect is small for $k \geq 10$. Cross-validation is pessimistically biased for $k < 10$, and iteration does not affect the bias.

4. The 632b bootstrap method has lower variance than 10-CV (its variance averages 80% that of 10-CV), but at a greater computational cost. The 632b method may be optimistically biased for very small samples or high error rates, and the sign and magnitude of its bias are different for different learning algorithms. For that reason, 632b is not suitable for comparing 1-NN and 3-NN classifiers, nor for comparing stopped and unpruned decision trees.

5. LOO* is approximately unbiased for discriminant functions and nearest neighbors, and has lower variance than LOO or 10-CV for these classifiers. This lack of bias and improved precision apparently do not carry over to nominal attribute decision trees, and LOO* is not recommended for those applications.

6. Extreme overfitting, as in the 1-NN classifier and unpruned decision trees, can affect both the bias and precision of cross-validation and bootstrapping. More complex methods may be necessary when classifiers are overfitted.

7. Though the mean estimated error over several samples from the same population is correlated with the mean true error for all of the resampling estimators, the estimated error rates of classifiers inferred from different samples from the same population is not correlated with the individual classifiers' true error rates.

8 Acknowledgement

The authors are indebted to the editor and reviewers of an earlier version of these papers. Their suggestions have been invaluable in correcting many of the weaknesses and oversights.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions.* Dover Publications, Inc., New York, 1972.

[2] T. L. Bailey and C. Elkan. Estimating the accuracy of learned concepts. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, volume 2, pages 895–900, San Mateo, CA, 1993. Morgan Kaufmann.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA, 1984.

[4] P. Cheeseman, J. Kelley, M. Self, J. Stutz, W. Taylor, and D. Freeman. Autoclass: A Bayesian classification system. In *Proceedings Fifth International Conference on Machine Learning*, pages 54–64, San Mateo, CA, 1988. Morgan Kauffman.

[5] S. L. Crawford. Extensions to the CART algorithm. *International Journal of Man-Machine Studies*, 31:197–217, 1989.

[6] A. C. Davison and P. Hall. On the bias and variability of bootstrap and cross-validation estimates of error rate in discrimination problems. *Biometrika*, 79:279–284, 1992.

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

[8] R. A. Fisher. *Statistical Methods for Research Workers.* Oliver & Boyd, Edinburgh, 14th edition, 1970. (the quotation is from the preface to the first (1925) edition).

[9] G. M. Fitzmaurice, W. J. Krzanowski, and D. J. Hand. A Monte Carlo study of the 632 bootstrap estimator of error rate. *Journal of Classification*, 8:239–250, 1991.

[10] M. James. *Classification Algorithms.* W. M. Collins & Sons, London, 1985.

[11] J. K. Martin and D. S. Hirschberg. Small sample statistics for classification error rates, II: confidence intervals and significance tests. Technical Report 95-43, University of California, Irvine, Irvine, CA, 1995.

[12] P. M. Murphy and D. W. Aha. *UCI Repository of Machine Learning Databases.* University of California, Irvine, Department of Information and Computer Science, Irvine, CA. (machine-readable data depository).

[13] J. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society*, A135:370–384, 1972.

[14] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[15] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.

[16] C. Schaffer. Overfitting avoidance as bias. *Machine Learning*, 10:153–178, 1993.

[17] J. W. Shavlik and T. G. Dietterich, editors. *Readings in Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1990.

[18] S. M. Weiss. Small sample error rate estimation for k-nearest neighbor classifers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:285–289, 1991.

[19] S. M. Weiss and N. Indurkhya. Small sample decision tree pruning. In *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 335–342, San Francisco, 1994. Morgan-Kaufman.

[20] S. M. Weiss and C. A. Kulikowski. *Computer Systems that Learn: Classification and Prediction Methods From Statistics, Neural Nets, Machine Learning, and Expert Systems.* Morgan Kaufmann, San Mateo, CA, 1991.