OUT-OF-BAG ESTIMATION

Leo Breiman*
Statistics Department
University of California
Berkeley, CA. 94708
leo@stat.berkeley.edu

Abstract

In bagging, predictors are constructed using bootstrap samples from the training set and then aggregated to form a bagged predictor. Each bootstrap sample leaves out about 37% of the examples. These left-out examples can be used to form accurate estimates of important quantities. For instance, they can be used to give much improved estimates of node probabilities and node error rates in decision trees. Using estimated outputs instead of the observed outputs improves accuracy in regression trees. They can also be used to give nearly optimal estimates of generalization errors for bagged predictors.

* Partially supported by NSF Grant 1-444063-21445

Introduction: We assume that there is a training set T= $\{(y_n,\mathbf{x}_n), n=1, \ldots, N\}$ and a method for constructing a predictor $Q(\mathbf{x},T)$ using the given training set. The output variable y can either be a class label (classification) or numerical (regression). In bagging (Breiman[1996a]) a sequence of training sets $T_{B,1}, \ldots, T_{B,K}$ are generated of the same size as T by bootstrap selection from T. Then K predictors are constructed such that the kth predictor $Q(\mathbf{x},T_{k,B})$ is based on the kth bootstrap training set. It was shown that if these predictors are aggregated--averaging in regression or voting in classification, then the resultant predictor can be considerably more accurate than the original predictor.

Accuracy is increased if the prediction method is unstable, i.e. if small changes in the training set or in the parameters used in construction can result in large changes in the resulting predictor. The examples generated in Breiman[1996a] were based on trees and subset selection in regression, but it is known that neural nets are also unstable, as are other well-known prediction methods. Other methods such as nearest neighbors, are stable.

It turns out that bagging, besides its primary purpose of increasing accuracy, has valuable by-products. Roughly 37% of the examples in the training set T do not appear in a particular bootstrap training set T_B. Thus, to the predictor $Q(\mathbf{x},T_B)$ these examples are unused test examples. Thus, if K = 100, each particular example $(y,\mathbf{x})$ in the training set has about 37 predictions among the $Q(\mathbf{x},T_{k,B})$ such that $T_{k,B}$ does not contain $(y,\mathbf{x})$. The predictions for examples "that are out-of-the-bag" can be used to form accurate estimates for important quantities.

For example, in classification, the out-of-bag predictions can be used to estimate the probabilities that the example belongs to any one of the J possible classes. Applied to CART this gives a method for estimating node probabilities more accurately than anything available to date. Applied to regression trees, we get an improved method for estimating the expected error in a node prediction. In regression, using the out-of-bag estimated values for the outputs instead of the actual training set outputs gives more accurate trees. Simple and accurate

out-of-bag estimates can be given for the generalization error of bagged predictors. Unlike cross-validation, these require no additional computing.

In this paper, we first look at estimates of node class probabilities in CART (Section 2), and then at estimates of mean-squared nodes errors in the regression version of CART (Section 3). Section 4 looks at how much accuracy is lost by using estimates that are averaged over terminal nodes. Indications from synthetic data are that the averaging can account for a major component of the error. Section 5 gives some theoretical justification for the accuracy of out-of-bag estimates in terms of a pointwise bias-variance decomposition. Section 6 gives the effect of constructing regression trees using the out-of-bag output estimates. In Section 7 the out-of-bag estimates of generalization error for bagged predictors are defined and studied. The Appendix gives some statistical details.

The present work came from two stimuli. One was the dissatisfaction, over many years, but growing stronger more recently, with the biased node class probability and error estimates in CART. The other consisted of two papers. One, by Tibshirani[1996], proposes an out-of-bag estimate as part of a method for estimating generalization error for any classifier. The second, by Wolpert and Macready [1996], looks at a number of methods for estimating generalization error for bagged regressions--among them, a method using out-of-bag predictions equivalent to the method we give in Section 6.

2. Estimating node class probabilities

Assume that the training set T consists of independent draws from an Y,**X** distribution where Y is a J-class output label and **X** is a multivariate input vector. Define $\mathbf{p}^*(\mathbf{x})$ as that probability vector with components $p^*(j|\mathbf{x}) = P(Y=j|\mathbf{X}=\mathbf{x})$. Most classification algorithms use the training set T to construct a probability predictor $\mathbf{p}^R(\mathbf{x},T)$ that outputs a nonnegative sum-one J-vector $\mathbf{p}^R(\mathbf{x},T) = (p^R(1|\mathbf{x}), \ldots, p^R(J|\mathbf{x}))$ and then classify **x** as that class for which $p^R(j|\mathbf{x})$ is maximum. In many applications, the components of $\mathbf{p}^R$ as well as the classification is important. For instance, in medical survival analysis, estimates of the survival probability is important.

In some construction methods , the resubstitution values $\mathbf{p}^R(\mathbf{x})$ are intrinsically biased estimates of $\mathbf{p}^*(\mathbf{x})$. This is true of methods like trees or neural nets where the optimization over T tries to drive all components of $\mathbf{p}^R$ to zero except for a single component that goes to one. The resulting vectors $\mathbf{p}^R$ are poor estimates of the true class probabilities $\mathbf{p}^*$.

With trees, the $\mathbf{p}^R$ estimates are constant over each terminal node **t** and are given by the proportion of class j examples in the terminal node. In Breiman et. al. [1984] two methods were proposed to improve estimates. In his thesis, Walker[1992] showed that the first of the two methods worked reasonably well on some synthetic data. However, the method only estimates $\max_j p^*(j|\mathbf{t})$, and the results are difficult to compare with those given below.

To define the problem better--here is the target: assume again that the training set T consists of independent draws from an Y,**X** distribution. Assume that a tree $C(\mathbf{x},\mathbf{T})$ with terminal nodes $\{\mathbf{t}\}$ has already been constructed. For a given terminal node **t,** define $p^*(j|\mathbf{t}) = P(Y=j|\mathbf{X} \in \mathbf{t})$. The vector $\mathbf{p}^*(\mathbf{t})$ is what we want to estimate. The resubstitution probability estimate $\mathbf{p}^R(\mathbf{x}, T)$ is constant over each terminal node **t** and consists of the relative class proportions of the training set examples in **t**.

The out-of-bag estimate $\mathbf{p}^B$ is gotten as follows: draw 100 bootstrap replicates of T getting $T_{1,B}, \ldots, T_{100,B}$. For each k, build the tree classifier $C(\mathbf{x},T_{k,B})$. For each (y,**x**) in T, define $\mathbf{p}^B(\mathbf{x})$ as the average of the $\mathbf{p}^R(\mathbf{x}, T_{k,B})$ over all k such that (y,**x**) is not in $T_{k,B}$. Then for any terminal node **t** , let $\mathbf{p}^B(\mathbf{t})$ be the average of $\mathbf{p}^B(\mathbf{x})$ over all **x** in **t**.

2.1 Experimental results

We illustrate, by experiment, the improved accuracy of the out-of-bag estimates compared to the resubstitution method using synthetic and real data. For any two J-probabilities **p** and **p**' denote

$$|\mathbf{p}-\mathbf{p}'| = \sum_j |p(j)-p'(j)|$$

$$\|\mathbf{p}-\mathbf{p}'\|^2 = \sum_j (p(j)-p'(j))^2$$

Let $q^*(\mathbf{t}) = P(\mathbf{X} \in \mathbf{t})$ be the probability that an example falls into the terminal node **t** . For any estimate $\{\mathbf{p}(\mathbf{t})\}$ of the $\{\mathbf{p}^*(\mathbf{t})\}$ define two error measures:

$$E_1 = \sum_{\mathbf{t}} q^*(\mathbf{t})|\mathbf{p}^*(\mathbf{t})-\mathbf{p}(\mathbf{t})|$$

$$E_2 = (\sum_{\mathbf{t}} q^*(\mathbf{t})\|\mathbf{p}^*(\mathbf{t})-\mathbf{p}(\mathbf{t})\|^2)^{1/2}$$

The difference between the two measures is that large differences are weighted more heavily by E_2. To simplify the interpretation we divide E_1 by J and E_2 by $\sqrt{J}$. Then E_1 measures the absolute average error in estimating each component of the probability vector, while E_2 measures the corresponding rms average error.

We use a test set to estimate $q^*(\mathbf{t})$ as the proportion $q'(\mathbf{t})$ of the test set falling into node **t** and estimate $\mathbf{p}^*(\mathbf{t})$ by the proportions of classes $\mathbf{p}'(\mathbf{t})$ in those test set examples in **t**. This lets us estimate the two error measures. With E_2, it is possible to derive a correction that adjusts for the error in using $\mathbf{p}'(\mathbf{t})$ instead of $\mathbf{p}^*(\mathbf{t})$. The correction is generally small if the test set is large and is derived in the Appendix.

Synthetic Data We give results for four sets of synthetic data (see Breiman[1996b]) for specific definitions):

Table 1 Synthetic Data Set Summary

Data Set	Classes	Inputs	Training	Test
waveform	3	21	300	5000
twonorm	2	20	200	5000
threenorm	2	20	200	5000
ringnorm	2	20	200	5000

In all cases, there were 50 iterations with the training and test sets generated anew in each iteration and 100 replications in the bagging. The results given are the averages over the 50 iterations:

Table 2 Node Probability Errors

Data Set	E_{B1}	E_{R1}	E_{B1}/E_{R1}	E_{B2}	E_{R2}	E_{B2}/E_{R2}
waveform	,048	.131	.37	.066	.171	.38
twonorm	.061	.215	.28	.083	.221	.36
threenorm	.069	.263	.26	.085	.278	.30
ringnorm	.085	.202	.42	.123	.230	.54

The ratios of errors in the 3rd and 6th columns shows that the out-of-bag estimates are giving significant error reductions.

Real Data The data sets we used in this experiment are available in the UCI repository. We used some of the larger data sets to insure test sets large enough to give adequate estimates of q* and **p***.

Table 3 Data Set Summary

Data Set	Classes	Inputs	Training	Test
breast-cancer	2	9	200	499
diabetes	2	8	200	568
vehicle	4	18	400	446
satellite	6	36	600	5835
dna	3	60	300	2886

The data sets listed in Table 3 consisted examples whose number was the total of the test and training set numbers. So, for example, the breast-cancer data set had 699 examples. The last two data sets listed came pre-separated into test and training sets. For instance, the satellite data came as a 4435 example training set and a 2000 example test set. These were put together to create a single data set with 6435 examples. Note that the training set sizes are 100 per class.

There were 50 runs on each data set. In each run, the data set was randomly divided into training and test sets with sizes as listed in Table 3, and 100 bootstrap replicates generated with each training set. The results, averaged over the 50 runs, are given in Table 4.

Table 4 Node Probability Errors

Data Set	E_{B1}	E_{R1}	E_{B1}/E_{R1}	E_{B2}	E_{R2}	E_{B2}/E_{R2}
breast cancer	.037	.046	.80	.069	.084	.82
diabetes	.063	.124	.56	.063	.156	.41
vehicle	.054	.091	.60	.058	.151	.39
satellite	.025	.044	.56	.049	.100	.49
dna	.054	.050	1.08	.084	.106	.79

The results generally show a significant decrease in estimation error when the out-of-bag estimates are used.

3. Estimating node error in regression

Assume here that the training set consists of independent draws from the distribution Y,**X** where Y is a numerical output, and **X** a multivariate input. Some methods for constructing a predictor f(**x**,T) of y using the training set also try to construct an estimate of the average error in the prediction--for instance by giving an estimate of the rms error in the prediction. When these error estimates are based on the training set error, they are often biased toward the low side.

In trees, the predicted value f(x,T) is constant over each terminal node **t** and is equal to the average $\bar{y}(\mathbf{t})$ of the training set outputs over the node **t.** The with-in node error estimate $e^{R}(\mathbf{t})$ for **t** is computed as the rms error over all examples in the training set falling into **t**. However, since the recursive splitting in CART is based on trying to minimize this error measure, it is clearly biased low as an estimate of the true error rate e*(**t**) defined as:

$$. e^*(\mathbf{t}) = (E((Y-\bar{y}(\mathbf{t}))^2 | \mathbf{X} \in \mathbf{t}))^{1/2}$$

The out-of-bag estimates are gotten this way: draw 100 bootstrap replicates of T getting $T_{1,B}$, ... ,$T_{100,B}$. For each k, build the tree predictor $f(\mathbf{x},T_{k,B})$. For each (y,**x**) in T, define $s^B(\mathbf{x})$ as the average of $(y - f(\mathbf{x},T_{k,B}))^2$ over all k such that (y,**x**) is not in $T_{k,B}$. Then for any node **t** define $e^B(\mathbf{t})$ as the square root of the average over all **x** in **t** of $s^B(\mathbf{x})$.

3.1 Experimental results

For any estimate e(**t**) of e*(**t**), define two error measures:

$$E_1 = \sum_{\mathbf{t}} q^*(\mathbf{t}) |e^*(\mathbf{t}) - e(\mathbf{t})|$$

$$E_2 = (\sum_{\mathbf{t}} q^*(\mathbf{t}) (e^*(\mathbf{t}) - e(\mathbf{t}))^2)^{1/2}$$

To illustrate the improved accuracy of the out-of-bag estimates of e*(**t**), five data sets are used--the same five that were used in Breiman[1996a]. The first three of the data sets are synthetic data, the last two real.

Table 5 Data Set Summary

Data Set	Inputs	Training	Test
Friedman #1	10	100	2000
Friedman #2	4	100	2000
Friedman #3	4	100	2000
Ozone	8	100	230
Boston	12	100	406

There were 50 runs of the procedure on each data set, and 100 bootstrap baggings in each run. In the synthetic data sets, the training and test set was freshly generated for each run. With the real data sets, for each run a different random split into training and test set was used. The values of q* and e* were estimated using the test set. For E_2 an adjustment was used to correct for this approximation (see Appendix). The results, averaged over the 50 runs, are given in Table 6. For interpretability, the error measures displayed have been divided by the standard deviation of the combined training and test set.

Table 4 Node Estimation Errors

Data Set	E_{B1}	E_{R1}	E_{B1}/E_{R1}	E_{B2}	E_{R2}	E_{B2}/E_{R2}
Friedman#1	.13	.33	.41	.17	.35	.47
Friedman #2	.11	.23	.47	.14	.27	.53
Freidman #3	.14	.31	.47	.20	.42	.48
Ozone	.11	.18	.59	.11	.21	.52
Boston	.17	.31	.53	.21	.39	.53

Again, there are significant reductions in estimation error.

4. Error due to within-node variability

Suppose that we want to estimate a function of the inputs h*(**x**). Only if we want to stay in the structure of a single tree with predictions constant over each terminal node, does it make sense to estimate h*(**x**) by some estimate of h*(**t**). The target function h*(**x**) may have considerable variability over the region defined by a terminal node in a tree. Given a tree with terminal nodes **t** and estimates h(**t**) of h*(**t**), denote the corresponding estimate of h by h(**x**,T). tThe squared error in estimating h* given that the inputs **x** are drawn from the distribution of the random vector **X** can be decomposed as:

$$E_{\mathbf{X}}\left(h(\mathbf{X})-h(\mathbf{X},T)\right)^2=\sum_{\mathbf{t}}E_{\mathbf{X}}(h(\mathbf{X})-h^*(\mathbf{t})^2\,|\,\mathbf{X}\in\mathbf{t})P(\mathbf{X}\in\mathbf{t})+\sum_{\mathbf{t}}(h^*(\mathbf{t})-h(\mathbf{t}))^2P(\mathbf{X}\in\mathbf{t})$$

The first term in this decomposition we call the error due to within-node variability, the second is the node estimation error. The relevant question is how large the within-node variability error is compared to the node estimation error. Obviously, this will be problem dependent , but we give some evidence below that it may be a major portion of the error.

We look at the problem of estimating the conditional probabilities $\mathbf{p}^*(\mathbf{x})$. The expression analogous to the above decomposition is:

$$E_{\mathbf{X}}\left\|\mathbf{p}^*(\mathbf{X})-\mathbf{p}(\mathbf{X},\mathbf{T})\right\|^2=\sum_t q^*(\mathbf{t})E_{\mathbf{X}}(\|\mathbf{p}^*(\mathbf{X})-\mathbf{p}^*(\mathbf{t})\|^2|\mathbf{X}\in\mathbf{t})+\sum_t q^*(\mathbf{t})\|\mathbf{p}^*(\mathbf{t})-\mathbf{p}(\mathbf{t})\|^2 \qquad (4.1)$$

For the four synthetic classification data sets used in Section 3, $\mathbf{p}^*(\mathbf{x})$ can be evaluated exactly. Therefore, replacing the expectations over **X** by averages over a large test set (5000), the error E_V due to within-node variability (first term in (4.1) and the error E_N due to node estimation (second term in (4.1))can be evaluated. There are two methods of node estimation--the standard method leads to error E_{NR} and the out-of-bag method to error E_{NB}.

Another method of estimating $\mathbf{p}^*$ is by utilizing the sequence of bagged predictors. For each (y,**x**) in the test set, define $\mathbf{p}^B(\mathbf{x})$ as the average of the $\mathbf{p}^R(\mathbf{x}, T_{k,B})$ over all k. Then defining E_B as:

$$E_B = E_{\mathbf{X}}\left\|\mathbf{p}^*(\mathbf{X})-\mathbf{p}^B(\mathbf{X})\right\|^2,$$

this quantity is also evaluated for the synthetic data by averaging over the test data.

The experimental procedure consists of 50 iterations for each synthetic data set. In each iteration, a 5000 examples test set and 100J example training set is generated, and the following ratios evaluated:

$$R_1 = 100*E_{NR}/(E_{NR}+E_V),$$
$$R_2 = 100*(E_{NB}-E_{NR})/(E_{NR}+E_V),$$
$$R_3 = 100*E_{NB}/(E_{NB}+E_V),$$
$$R_4 = 100*E_B/E_V.$$

Thus, R_1 is the percent of the total error due to node estimation when the resubstitution method of estimating $\mathbf{p}^*(\mathbf{t})$ is used: R_2 is the percent of reduction in the total error when the bagging estimate $\mathbf{p}^B(\mathbf{t})$ is used. When $\mathbf{p}^B(\mathbf{t})$ is used, R_3 is the percent of the total error due to node estimation. Finally, R_4 is the ratio*100 of the error using the pointwise bagging estimate of $\mathbf{p}^*$

to the error using estimates constant over the terminal nodes but using the optimal node estimate $\mathbf{p}^*(\mathbf{t})$. Table 5 gives the results.

Table 5 Error Ratios(%) in Estimating Class Probabilities

Data Set	R_1	R_2	R_3	R_4
waveform	37	29	11	46
twonorm	28	21	8	46
threenorm	37	29	10	59
ringnorm	27	18	11	63

For these synthetic data sets, the values for R_1 shows that within-node variability accounts for about two-thirds of the error. The second column (R_2) shows that use of the bagging estimate $\mathbf{p}^B(\mathbf{t})$ eliminates most of the errors due to node estimation, but that the reduction is relatively modest because of the strong contribution of within-node variability. The results for R_3 show that when $\mathbf{p}^B(\mathbf{t})$ is used, only about 10% of the total error is due to node estimation, so that we are close to the limit of what can be accomplished using estimates of $\mathbf{p}^*$ constant over nodes.

The final column gives the good news that using the pointwise bagging estimates $\mathbf{p}^B(\mathbf{x})$ gives about 50% reduction as compared to the best possible node estimate. It is a bit disconcerting to see how much accuracy is lost by the averaging of estimates over nodes. Smyth et.al.[1996] avoid this averaging by using a kernel density method to estimate variable within-node densities. However, pointwise bagging estimates may give comparable or better results. Care must be taken in generalizing from these synthetic data sets as I suspect they may have more within-node variability than typical real data sets.

5. Why it works--the pointwise bias-variance decomposition

Suppose there is some underlying function $h^*(\mathbf{x})$ that we want to estimate using the training set T and that we have some method $h(\mathbf{x},T)$ for estimating $h^*(\mathbf{x})$. Then for $\mathbf{x}$ fixed, we can write, using E_T to denote expectation over replicate training sets of the same size drawn from the same distribution

$$E_T(h^*(\mathbf{x})-h(\mathbf{x},T))^2=(h^*(\mathbf{x})-E_T h(\mathbf{x},T))^2+E_T(h(\mathbf{x},T)-E_T h(\mathbf{x},T))^2 .$$

This is a pointwise in $\mathbf{x}$ version of the now familar bias-variance decomposition. The interesting thing that it shows is that at each point $\mathbf{x}$, $E_T h(\mathbf{x},T)$ has lower squared error than does $h(\mathbf{x}, T)$--it has zero variance but the same bias. That is, averaging $h(\mathbf{x},T)$ over replicate training sets improves performance at each individual values of $\mathbf{x}$.

Bagging tries to get an estimate of $E_T h(\mathbf{x},T)$ by averaging over the values of $h(\mathbf{x},T_{k,B})$. Now $E_T h(\mathbf{x},T)$ is computed assuming $\mathbf{x}$ is held fixed and T is chosen in a way that does not depend on $\mathbf{x}$. But if $\mathbf{x}$ is in the training set, then the $T_{k,B}$ often contain $\mathbf{x}$, violating the assumption. A better imitation of $E_T h(\mathbf{x},T)$ would be to leave $\mathbf{x}$ out of the training set and do bagging on the deleted training set. But this is exactly what out-of-bag estimation does resulting in more accurate estimates of $h^*(\mathbf{x})$ at every example in the training set. When these are averaged over any terminal node $\mathbf{t}$, more accurate estimates $h^B(\mathbf{t})$ of $h^*(\mathbf{t})=E(h^*(\mathbf{X}) \mid \mathbf{X}\in \mathbf{t})$ result.

6. Trees using out-of-bag output estimates.

In regression, for y,$\mathbf{x}$ an example in the training set, define the out-of-bag estimate y^B for the output y to be the average of $f(\mathbf{x},T_{k,B})$ over all k such that $\mathbf{x}$ is not in $T_{k,B}$. The out-of-bag output estimates will generally be less noisy than the original outputs. This suggests the

possibility of growing a tree using the y^B as the outputs values for the training set. We did this using the data sets described in Section 3, and followed the procedure in Breiman[1996].

With the real data sets we randomly subdivided them so that 90% served as the training set, and 10% as a test set. A tree was grown and pruned using the original training set, and the 10% test set used to get a mean-squared error estimate. Then we did 100 bootstrap iterations and computed the y^B. Finally a single tree was grown using the y^B as outputs and its error measured using the 10% test set. The random subdivsion was repeated 50 times and the test set errors averaged. With the three synthetic data sets, a training set of 200 and a test set of 2000 were freshly generated in each of the 50 runs. The results are given in Table 6.

Table 6. Mean Square Test Set Error

Data Set	Error--Original Outputs	Error--O-B Outputs	
Friedman #1	11.8	10.6	
Freidman #2*	31.2	26.8	* x1000
Friedman #3**	42.1	41.2	**/1000
Boston	20.4	18.7	
Ozone	25.5	21.2	

These decreases are not as dramatic as those given by bagging. On the other hand, they involve prediction by a single tree, generally of about the same size as those grown on the original training set. If the desire is to increase accuracy while retaining interpretability, then using the out-of-bag outputs does quite well.

The story in classification is that using the out-of-bag output estimates gives very little improvement in accuracy and can actually result in less accurate trees. The out-of-bag output estimates consist of the probability vectors $\mathbf{p}^B(\mathbf{x})$. CART was modified to accept probability vectors as outputs in tree construction, and a procedure similar to that used in regression was tried on a number of data sets with disappointing results. The problem is two-fold. First, while the probability vector estimates may be more accurate, the classification depends only on the location of the maximum component. Second, for data sets with substantial missclassification rates, the out-of-bag estimates may produce more distortion than the original class labels.

7. Out-of-bag estimates for bagged predictors.

In this section, we reinforce the work by Tibshirani [1996] and Wolpert and Macready[1996], both of whom proposed using out-of-bag estimates as an ingredient in estimates of generalization error. Wolpert and Macready worked on regression type problems and proposed a number of methods for estimating the generalization error of bagged predictors. The method they found that gave best performance is a special case of the method we propose. Tibshirani used out-of-bag estimates of variance to estimate generalization error for arbitrary classifiers. We explore estimates of generalization error for bagged predictors. For classification, our results are new.

As Wolpert and Macready point out in their paper, cross-validating bagged predictors may lead to large computing efforts. The out-of-bag estimates are efficient in that they can be computed in the same run that constructs the bagged predictor with little additional effort. Our experiments below also give evidence that these estimates are close to optimal.

Suppose again, that we have a training set T consisting of examples with an output variable $\mathbf{y}$ that can be a multidimensional vector with numerical or categorical coordinates, and corresponding input $\mathbf{x}$. A method is used to construct a predictor $\mathbf{f}(\mathbf{x},T)$, and a given loss function

L(**y, f**) measures the error in predicting **y** by **f.** Form bootstrap training sets $T_{k,B}$, predictors **f**(**x**, $T_{k,B}$) and aggregate these predictors in an appropriate way to form the bagged predictor $\mathbf{f}_B(\mathbf{x})$. For each **y,x** in the training set, aggregate the predictors only over those k for which $T_{k,B}$ does not containing **y,x.** Denote these out-of-bag predictors by $\mathbf{f}^{OB}$ Then the out-of-bag estimate for the generalization error is the average of L(**y,** $\mathbf{f}^{OB}$(**x**)) over all examples in the training set.

Denote by e^{TS} the test set error estimate and by e^{OB} the out-of-bag error estimate. In all of the runs in Sections 2 and 3, we also accumulated the average of e^{TS}, e^{OB} and $|e^{TS} - e^{OB}|$. We can also compute the expected value of $|e^{TS} - e^{OB}|$ under the assumption that e^{OB} is computed using a test set of the same size as the training set and independent of the actual test set used (see Appendix). We claim that this expected value is a lower bound for how well we could use the training set to estimate the generalization error of the bagged predictor.

Our reasoning is this: given that the training set is used to construct the predictor, the most accurate estimate for its error is a test set independent of the training set. If we used a test set of the same size as the training set, this is as well as can be done using this number of examples. Therefore, we can judge the efficiency of any generalization error estimate based on the training set by comparing its accuracy to the estimate we would get using a test set of the size of the training set.

Table 7 contains the results for the classification runs for both the real and synthetic data. The last column is the ratio of the experimentally observed $|e^{TS} - e^{OB}|$ to the expected value if the training set were an independent test set. The closer this ratio is to one, the closer to optimal e^{OB} is. Table 8 gives the corresponding results for regression.

Table 7 Estimates of Generalisation Error (% Missclassification)

Data Set	Av e^{TS}	Av e^{OB}	Av $\|e^{TS} - e^{OB}\|$	Ratio
waveform	19.6	19.4	2.7	1.41
twonorm	8.4	9.1	1.8	1.10
threenorm	21.3	21.8	3.3	1.35
ringnorm	11.8	12.2	2.6	1.33
breast-cancer	4.4	4.4	1.5	1.11
diabetes	25.6	26.2	3.6	1.28
vehicle	26.3	27.0	2.3	.96
satellite	13.7	14.1	1.3	1.01
dna	7.5	7.6	1.2	.96

Table 8 Estimates of Generalization Error (Mean Squared Error)

Data Set	Av e^{TS}	Av e^{OB}	Av $\|e^{TS} - e^{OB}\|$	Ratio	
Friedman #1	8.6	8.0	1.3	1.33	
Friedman #2	24.7*	23.6*	3.6	1.21	* x1000
Friedman #3	32.8**	30.5**	6.7**	1.06	**/1000
Boston	19.6	17.8	6.8	1.07	
Ozone	20.7	19.5	3.5	1.02	

Tables 7 and 8 show that the out-of-bag estimates are remarkably accurate. On the whole, the ratio values are close to one, reflecting the accuracy of the out-of-bag estimates of the generalization error of the bagged predictors. In classification, the out-of-bag estimates

appear almost unbiased, i.e. the average of e^{OB} is almost equal to the average of e^{TS}. But the estimates in regression may be systematically low. The two values slightly less than one in Table 6 we attribute to random fluctuations. The denominator in the ratio column depends on a parameter which has to be estimated from the data. Error in this parameter estimate may drive the ratio low)see Appendix for details).

References

Breiman, L. [1996a] Bagging Predictors , Machine Learning 26, No. 2, 123-140

Breiman, L. [1996b] Bias, Variance, and Arcing Classifiers, submitted to Annals of Statistics, ftp ftp.stat.berkeley.edu pub/breiman/arcall.ps

Breiman, L., Friedman, J., Olshen R., and Stone, C. [1984] Classification and Regression Trees, Wadsworth

Smyth, P.,Gray, A. and Fayyad, U.[1996} Retrofitting Tree Classifiers Using Kernel Density Estimation, Proceedings of the 1995 Conference on Machine Learning, San Francisco, CA: Morgan Kaufmann, 1995, pp.506-514.

Tibshirani, R.[1996] Bias, Variance, and Prediction Error for Classification Rules, Technical Report, Statistics Department, University of Toronto

Walker, Michael G. (1992) Probability Estimation for ClassificationTrees and Sequence Analysis. Stan-CS-92-1422. Ph.D. Dissertation. Departments of Computer Science and Medicine, Stanford University,Stanford, CA.

Wolpert, D.H. and Macready, W.G.[1996] An Efficient Method to Estimate Bagging's Generalization Error

Appendix

I. Adjustment to E_2 in classification

With moderate sized test sets we can get, at best, only noisy estimates of $\mathbf{p}^*$. Using these estimates to compute error rates may lead to biases in the results. However, a simple adjustment is possible in the E_2 error criterion. For $\{\mathbf{p(t)}\}$ probability estimates in the terminal nodes $\mathbf{t}$ depending only on the examples in T, E_2 is defined by

$$E_2^2 = \sum_{\mathbf{t}} q(\mathbf{t})\|\mathbf{p}^*(\mathbf{t})-\mathbf{p}(\mathbf{t})\|^2 \qquad (2.1)$$

Let $\mathbf{p}'(\mathbf{t})$ be the class proportions of test set examples in node $\mathbf{t}$ so that $\mathbf{p'}(\mathbf{t})$ is an estimate of $\mathbf{p}^*(\mathbf{t})$. We assume that the examples in S and T are independent. Let N be the number of test set examples falling into terminal node **t.** Conditional on N, $p'(j|\mathbf{t})$ times N has a binomial distribution $B(p^*(j|\mathbf{t}),N)$. Hence, $p'(j|\mathbf{t})$ has expectation $p^*(j|\mathbf{t})$ and variance $p^*(j|\mathbf{t})(1-p^*(j|\mathbf{t})$)/N. Write

$$\|\mathbf{p}'-\mathbf{p}\|^2=\|\mathbf{p}^*-\mathbf{p}\|^2+\|\mathbf{p}'-\mathbf{p}^*\|^2+2(\mathbf{p}^*-\mathbf{p},\mathbf{p}'-\mathbf{p}^*).$$

Taking expectations of both sides with respect to the examples in S holding N constant gives

$$N\cdot E\|\mathbf{p}'-\mathbf{p}\|^2=N\|\mathbf{p}^*-\mathbf{p}\|^2+1-\|\mathbf{p}^*\|^2 \qquad (A.1)$$

Putting $\mathbf{p}$=0 in (A.1) gives

$$N\cdot E\|\mathbf{p}'\|^2=N\|\mathbf{p}^*\|^2+1-\|\mathbf{p}^*\|^2 \qquad (A.2$$

Solving (A.1) and (A.2) for $N\|\mathbf{p}^*-\mathbf{p}\|^2$ gives

$$N\|\mathbf{p}^*-\mathbf{p}\|^2 = N\cdot E\|\mathbf{p}'-\mathbf{p}\|^2 - \frac{N}{N-1}(1-E\|\mathbf{p}'\|^2)$$

Thus, we estimate the error measure E_2 as:

$$E_2^2 = \sum_{\mathbf{t}} \mathbf{q}'(\mathbf{t})[\|\mathbf{p}'(\mathbf{t})-\mathbf{p}(\mathbf{t})\|^2 - (1-\|\mathbf{p}'\|^2)/(N(\mathbf{t})-1)] \qquad (A.3)$$

where N(**t**) is the number of test set examples in node **t**, q'(**t**) is N(**t**)/N, and we define the second term in the brackets to be zero if N(**t**)=1. This term in (A.3) is the adjustment. In our examples, it had only a small effect on the results.

II. Adjustment to E_2 in regression

In regression, E_2 is defined as the square root of

$$R = \sum_{\mathbf{t}} q^*(\mathbf{t})(e^*(\mathbf{t})-e(\mathbf{t}))^2.$$

Since e*(**t**) is unknown, we estimate it as the square root of the average over all test set (y,**x**) falling in **t** of (y- $\bar{y}(\mathbf{t})$)^2 and denote this estimate by e'(**t**). Let

$$R' = \sum_{\mathbf{t}} q^*(\mathbf{t})(e'(\mathbf{t})-e(\mathbf{t}))^2 .$$

Then we want to adjust R' so that it is an unbiased estimate of R, i.e. so that ER' = R, when the expectation is taken over the test set examples. To simplify this computation, we assume that:

i) test set output values in **t** are normally distributed with true mean y*(**t**).
ii) q*(**t**) | y*(**t**)- $\bar{y}(\mathbf{t})$ | is small.

Now,

$$e'(\mathbf{t})^2 = \sum_n (y_n' - \bar{y}(\mathbf{t}))^2 / N(\mathbf{t})$$

where the sum is over all test set (y,**x**) falling into **t.** By the Central Limit Theorem, e'(**t**)^2 = e*(**t**)^2 + $Z/\sqrt{N(\mathbf{t})}$ where Z is approximately normally distributed with mean zero and variance equal to the variance of (Y- $\bar{y}(\mathbf{t})$)^2 conditional on **X** in **t**. It follows that Ee'(**t**)^2 = e*(**t**)^2 , and to first order in N(**t**), Ee'(**t**) = e*(**t**) - $EZ^2/(8e^*(\mathbf{t})^3 N(\mathbf{t}))$. Recall that for a normally distributed random variable U with variance σ^2,the variance of U^2 is $2\sigma^4$. Using assumption i) gives EZ^2 = 2(var(Y- $\bar{y}(\mathbf{t})$))^2.By assumption ii) the variance of Y- $\bar{y}(\mathbf{t})$ on **t** can be approximated by e*(**t**)^2 . Thus,

$$Ee'(\mathbf{t}) = e^*(\mathbf{t})[1-.25/N(\mathbf{t})] \qquad (A.4)$$

Write

.

$$(e'(\mathbf{t})-e(\mathbf{t}))^2=(e'(\mathbf{t})-e^*(\mathbf{t}))^2+(e^*(\mathbf{t})-e(\mathbf{t}))^2+2(e'(\mathbf{t})-e^*(\mathbf{t}))(e^*(\mathbf{t})-e(\mathbf{t})). \quad \text{(A.4)}$$

Taking expectation of (A.5) with respect to the test set and using (A.4) gives

$$E(e'(\mathbf{t})-e(\mathbf{t}))^2\approx E(e^*(\mathbf{t})-e(\mathbf{t}))^2+e^*(\mathbf{t})e(\mathbf{t})/2N(\mathbf{t}) \quad \text{(A.5)}$$

Approximating e*(**t**)e(**t**) by e'(**t**)^2 gives the adjusted measure:

$$E_2^2=\sum_t q'(\mathbf{t})[(e'(\mathbf{t})-e(\mathbf{t}))^2-.5e'(\mathbf{t})^2/N(\mathbf{t})]$$

Again, the adjustment contributes a relatively small correction in our runs.

III. Lower bound for training set accuracy.

Suppose we have a classifier Q(**x**) with true generalization missclassification rate P*. That is, for the distribution of the r.v. Y,**X,** e*= P(Y≠ Q(**X**)). Two sets of data T'= {(y'$_n$,**x'**$_n$), n=1, ... ,N'} and T= {(y$_n$,**x**$_n$), n=1, ... ,N} are independently drawn from the underlying distribution of Y,**X** and run through the classifier. The first has has classification error rate e' and the second e. We evaluate g(N',N) =E|e'-e|.

In the context of the experiments on out-of-bag estimation, the first set of data is the given test set. The second set is the training set. The training set is used both to form the bagged classifier and the out-of-bag estimate of the generalization error. Suppose we had an untouched test set of the same size as the training set and used this new test set to estimate the generalization error. Certainly, we would do better than any way of using the training set over again to do the same thing. Thus g(N',N) is a lower bound for the accuracy of generalization error estimates using a training set of N examples, when it is being compared to a test set using N' examples.

Now, e is given by

$$e=\sum_n V_n/N \quad \text{(A.6)}$$

where V_n is one if the nth case in T is missclssified and zero otherwise. By the Central Limit Theorem, e=e* + $Z/\sqrt{N}$, where Z is approximately normal with mean zero and variance P*(1-P*). Similarly, e'=e* + $Z'/\sqrt{N'}$ where Z' is approximately normal mean zero also with variance e*(1-e*) and independent of Z. So e'-e is normal with variance

$$s=e^*(1-e^*)[(1/N)+(1/N')].$$

The expectation E|U| of a mean zero normal variable U with variance s is $\sqrt{2s/\pi}$ and it was this last expression that was used as a comparison in Section 6, with s estimated using e' in place of e*.

In regression , the predictor is a numerical-valued function f(**x**). The true generalization error is e* = E(Y-f(**X**))^2 . Expression (A.6) holds with $V_n=(y_n-f(\mathbf{x}_n))^2$. Again, e=e*+ $Z/\sqrt{N}$, where Z is approximately normal with mean zero and variance $c/\sqrt{N}$ with c the variance of (Y-f(**X**))^2 . Repeating the argument above, e'- e is a normal variable with mean zero and variance

s=c[(1/N)+(1/N') so E|e'-e| equals $\sqrt{2s/\pi}$. The variance of $(Y\text{-}f(\mathbf{X}))^2$ is given by $E(Y\text{-}f(\mathbf{X}))^4$ - $(E(Y\text{-}f(\mathbf{X}))^2)^2$. This value is approximated by the using the corresponding moments over the test set, leading to the evaluation of the lower bound.