Bagging Predictors

By

Leo Breiman*

Technical Report No. 421
September 1994

*Partially supported by NSF grant DMS-9212419

Department of Statistics
University of California
Berkeley, California 94720

Bagging Predictors

Leo Breiman[1]
Department of Statistics
University of California at Berkeley

Abstract

Bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class. The multiple versions are formed by making bootstrap replicates of the learning set and using these as new learning sets. Tests on real and simulated data sets using classification and regression trees and subset selection in linear regression show that bagging can give substantial gains in accuracy. The vital element is the instability of the prediction method. If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy.

1. Introduction

A learning set of $\mathcal{L}$ consists of data $\{(y_n, \boldsymbol{x}_n), n = 1, \ldots, N\}$ where the y's are either class labels or a numerical response. We have a procedure for using this learning set to form a predictor $\varphi(\boldsymbol{x}, \mathcal{L})$ — if the input is $\boldsymbol{x}$ we predict y by $\varphi(\boldsymbol{x}, \mathcal{L})$. Now, suppose we are given a sequence of learnings sets $\{\mathcal{L}_k\}$ each consisting of N independent observations from the same underlying distribution as $\mathcal{L}$. Our mission is to use the $\{\mathcal{L}_k\}$ to get a better predictor than the single learning set predictor $\varphi(\boldsymbol{x}, \mathcal{L})$. The restriction is that all we are allowed to work with is the sequence of predictors $\{\varphi(\boldsymbol{x}, \mathcal{L}_k)\}$.

If y is numerical, an obvious procedure is to replace $\varphi(\boldsymbol{x}, \mathcal{L})$ by the average of $\varphi(\boldsymbol{x}, \mathcal{L}_k)$ over k. i.e. by $\varphi_A(\boldsymbol{x}) = E_{\mathcal{L}}\varphi(\boldsymbol{x}, \mathcal{L})$ where $E_{\mathcal{L}}$ denotes the expectation over $\mathcal{L}$, and the subscript A in φ_A denotes aggregation. If $\varphi(\boldsymbol{x}, \mathcal{L})$

[1] partially supported by NSF grant DMS-9212419.

predicts a class $j \in \{1, \ldots, J\}$, then one method of aggregating the $\varphi(\boldsymbol{x}, \mathcal{L}_k)$ is by voting. Let $N_j = \#\{k; \varphi(\boldsymbol{x}, \mathcal{L}_k) = j\}$ and take $\varphi_A(\boldsymbol{x}) = \mathrm{argmax}_j N_j$.

Usually, though, we have a single learning set $\mathcal{L}$ without the luxury of replicates of $\mathcal{L}$. Still, an imitation of the process leading to φ_A can be done. Take repeated bootstrap samples $\{\mathcal{L}^{(B)}\}$ from $\mathcal{L}$, and form $\{\varphi(\boldsymbol{x}, \mathcal{L}^{(B)})\}$. If y is numerical, take φ_B as

$$\varphi_B(\boldsymbol{x}) = av_B \varphi(\boldsymbol{x}, \mathcal{L}^{(B)}).$$

If y is a class label, let the $\{\varphi(\boldsymbol{x}, \mathcal{L}^{(B)})\}$ vote to form $\varphi_B(\boldsymbol{x})$. We call this procedure "**b**ootstrap **agg**regat**ing**" and use the acronym **bagging**.

The $\{\mathcal{L}^{(B)}\}$ form replicate data sets, each consisting of N cases, drawn at random, *but with replacement*, from $\mathcal{L}$. Each $(y_n, \boldsymbol{x}_n)$ may appear repeated times or not at all in any particular $\mathcal{L}^{(B)}$. The $\{\mathcal{L}^{(B)}\}$ are replicate data set drawn from the bootstrap distribution approximating the distribution underlying $\mathcal{L}$. For background on bootstrapping, see Efron and Tibshirani [1993]. A critical factor in whether bagging will improve accuracy is the stability of the procedure for constructing φ. If changes in $\mathcal{L}$, i.e. a replicate $\mathcal{L}$, produces small changes in φ, then φ_B will be close to φ. Improvement will occur for unstable procedures where a small change in $\mathcal{L}$ can result in large changes in φ. Unstability was studied in Breiman [1994] where it was pointed out that neural nets, classification and regression trees, and subset selection in linear regression were unstable, while k-nearest neighbor methods were stable.

For unstable procedures bagging works well. In Section 2 we bag classification trees on a variety of real and simulated data sets. The reduction in test set missclassification rates ranges from 20% to 47%. In section 3 regression trees are bagged with reduction in test set mean squared error on data sets ranging from 22% to 46%. Section 4 goes over some theoretical justification for bagging and attempts to understand when it will or will not work well. This is illustrated by the results of Section 5 on subset selection in linear regression using simulated data. Section 6 gives concluding remarks. These discuss how many bootstrap replications are useful, bagging nearest neighbor classifiers and bagging class probability estimates.

The evidence, both experimental and theoretical, is that bagging can push a good but unstable procedure a significant step towards optimality. On the other hand, it can slightly degrade the performance of stable procedures. There has been recent work in the literature with some of the flavor of bagging. In particular, there has been some work on averaging and voting over multiple trees. Buntine [1991] gave a Bayesian approach, Kwok and Carter [1990] used voting over multiple trees generated by using alternative splits, and Heath et. al. [1993] used voting over multiple trees generated by alternative oblique splits. Dieterrich [1991] showed that a method for coding many class problems into a large number of two class problems increases accuracy. There is some commonality of this idea with bagging.

2. Bagging Classification Trees

2.1. Results

Bagging was applied to classification trees using the following data sets:

waveform (simulated)
heart
breast cancer (Wisconsin)
ionosphere
diabetes
glass
soybean

All of these except the heart data are in the UCI repository (ftp ics.uci.edu /pub/machine-learning-databases). The data are briefly described in Section 2.2.

Testing was done using random divisions of each data set into a learning and test set, constructing the usual tree classifier using the learning set, and bagging this tree using 50 bootstrap replicates. This was repeated 100 times for each data set (specifics are given in Section 2.3). The average test set missclassification rate using a single tree is denoted by $\bar{e}_S$ and the bagging rate by $\bar{e}_B$. The results are:

Table 1 Missclassification Rates (Percent)

Data Set	$\bar{e}_S$	$\bar{e}_B$	**Decrease**
waveform	29.0	19.4	33%
heart	10.0	5.3	47%
breast cancer	6.0	4.2	30%
ionosphere	11.2	8.6	23%
diabetes	23.4	18.8	20%
glass	32.0	24.9	22%
soybean	14.5	10.6	27%

For the waveform data it's known that the minimal attainable rate (Bayes Rate) is 14.0%. Using this as a base, the excess error drops from 15.0% to 5.4%.

2.2. **Data Sets**

Table 2 gives a summary of the data sets and the test set sizes used.

Table 2
Data Set Summary

Data Set	# Samples	# Variables	# Classes	# Test Set
waveform	300	21	3	1500
heart	1395(823)	16(18)	2	250
breast cancer	699	9	2	100
ionosphere	351	34	2	25
diabetes	1036(768)	8	2	250
glass	214	9	6	20
soybean	307	35	19	25

The figures in parentheses are for the original data sets. These were modified for reasons described below to give the as-used numbers. In all but the simulated waveform data, the data set was randomly divided into a test set and learning set. So, for instance, in the glass data, the size of the learning set in each iteration was $194 = 214 - 20$. For the simulated waveform data, a learning set of 300 and a test set of 1500 were generated for each iteration. Brief descriptions of the data sets follows. More extended background is available in the UCI repository.

Waveform This is simulated 21 variable data with 300 cases and 3 classes each having probability 1/3. It is described in Breiman et al [1984] (a C subroutine for generating the data is in the UCI repository subdirectory /waveform).

Heart This is data from the study referred to in the opening paragraphs of the CART book (Breiman et. al. [1984]). To quote:

> At the University of California, San Diego Medical Center, when a heart attack patient is admitted, 19 variables are measured during the first 24 hours. These include blood pressure, age, and 17 other ordered and binary variables summarizing the medical symptoms considered as important indicators of the patient's condition.
>
> The goal of a recent medical study (see Chapter 6) was the development of a method to identify high risk patients (those who will not survive at least 30 days) on the basis of the initial 24-hour data.

The data base has also been studied in Olshen et al [1985]. It was gathered on a project (SCOR) headed by John Ross Jr. Elizabeth Gilpin and Richard Olshen were instrumental in my obtaining the data. The data used had 18 variables. Two variables with high proportions of missing data were deleted, together with a few other cases that had missing values. This left 779 complete cases — 77 deaths and 702 survivors. To equalize class sizes, each case of death was replicated 9 times giving 693 deaths for a total of 1395 cases.

Breast Cancer This is data given to the UCI repository by William H. Wolberg, University of Wisconsin Hospitals, Madison (see Wolberg and Mangasariam [1990]). It is two class data with 699 cases, (458 benign and 241 malignant). It has 9 variables consisting of cellular characteristics. (subdirectory /breast-cancer-wisconsin)

Ionosphere This is radar data gathered by the Space Physics Group at Johns Hopkins University (see Sigillito et. al. [1989]). There are 351 cases with 34 variables, consisting of 2 attributes for each at 17 pulse numbers. There are two classes: good = some type of structure in the ionosphere (226); bad = no structure (125). (subdirectory /ionosphere)

Diabetes This is a data base gathered among the Pima Indians by the National Institute of Diabetes and Digestive and Kidney Diseases. (See Smith et. al. [1988]). The data base consists of 768 cases, 8 variables and two classes. The variables are medical measurements on the patient plus age and pregnancy information. The classes are: tested positive for diabetes (268) or negative (500). To equalize class sizes, the diabetes cases were duplicated giving a total sample size of 1036. (subdirectory /pima-indians-diabetes)

Glass This data base was created in the Central Research Establishment, Home Office Forensic Science Service Aldermaston, Reading, Berkshire. Each case consists of 9 chemical measurements on one of 6 types of glass. There are 214 cases.

Soybean The soybean learning set consists of 307 cases, 35 variables and 19 classes. The classes are various types of soybean diseases. The variables are observation on the plants together with some climatic variables. All are categorical. Some missing values were filled in. (subdirectory /soybean/soybean_large.data)

(subdirectory /glass)

2.2. **Computations**

In all runs, the following procedure was used:

i). The data set was randomly divided into a test set $\mathcal{T}$ and learning set $\mathcal{L}$. The test sets sizes selected in the real data sets are ad hoc, mostly chosen so that $\mathcal{L}$ would be reasonably large. In simulated data, test set size was chosen comfortably large.

ii). A classification tree was constructed from $\mathcal{L}$, with selection done by 10-fold cross-validation. Running the test set $\mathcal{T}$ down this tree gives the missclassification rate $e_S(\mathcal{L}, \mathcal{T})$

iii). A bootstrap sample $\mathcal{L}_B$ is selected from $\mathcal{L}$, and a tree grown using $\mathcal{L}_B$ and 10-fold cross-validation. This is repeated 50 times giving tree classifiers $\varphi_1(\boldsymbol{x}), \ldots, \varphi_{50}(\boldsymbol{x})$.

iv). If $(j_n, \boldsymbol{x}_n) \in \mathcal{T}$, then the estimated class of $\boldsymbol{x}_n$ is that class having the plurality in $\varphi_1(\boldsymbol{x}_n), \ldots, \varphi_{50}(\boldsymbol{x}_n)$. The proportion of times the estimated class differs from the true class is the bagging missclassification rate $e_B(\mathcal{L}, \mathcal{T})$.

v) The random division of the data is repeated 100 times and the reported $\bar{e}_S$, $\bar{e}_B$ are the averages over the 100 iterations.

3. Bagging Regression Trees

3.1. Results

Bagging trees was used on 5 data sets with numerical responses.

Boston Housing	
Ozone	
Friedman #1	(simulated)
Friedman #2	(simulated)
Friedman #3	(simulated)

The computing scheme was similar to that used in classification. Learning and test sets were randomly selected, 25 bootstrap replications used, and 100 iterations. The results are:

Table 3
Mean Squared Test Set Error

Data Set	$\bar{e}_S$	$\bar{e}_B$	Decrease
Boston Housing	19.1	11.7	39%
Ozone	23.1	18.0	22%
Friedman #1	11.4	6.2	46%
Friedman #2	30,800	21,700	30%
Friedman #3	.0403	.0249	38%

3.2. Data Sets

Table 4
Summary of Data Sets

Data Set	**#Cases**	**# Variables**	**# Test Set**
Boston Housing	506	12	25
Ozone	330(366)	8(9)	15
Friedman #1	200	10	1000
Friedman #2	200	4	1000
Friedman #3	200	4	1000

Boston Housing This data became well-known through its use in the book by Belsley, Kuh, and Welsch [1980]. It has 506 cases corresponding to census tracts in the greater Boston area. The y-variable is median housing price in the tract. There are 12 predictor variables, mainly socio-economic. The data has since been used in many studies. (UCI repository/housing).

Ozone The ozone data consists of 366 readings of maximum daily ozone at a hot spot in the Los Angeles basin and 9 predictor variables — all meteorlogical, i.e. temperature, humidity, etc. It is described in Breiman and Friedman [1985] and has also been used in many subsequent studies. Eliminating one variable with many missing values and a few other cases leaves a data set with 330 complete cases and 8 variables.

Friedman #1 All three Friedman data sets are simulated data that appear in the MARS paper (Friedman [1991]). In the first data set, there are ten independent predictor variables $x_1, \ldots, x_{10}$ each of which is uniformly distributed over $[0,1]$. The response is given by

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + \epsilon$$

where ϵ is $N(0,1)$. Friedman gives results for this model for sample sizes 50, 100, 200. We use sample size 200.

Friedman #2, #3 These two examples are taken to simulate the impedance and phase shift in an alternating current circuit. They are 4 variable data with

#2 $$y = (x_1^2 + (x_2x_3 - (1/x_2x_4))^2)^{1/2} + \epsilon_2$$

#3 $$y = \tan^{-1}\left(\frac{x_2x_3 - (1/x_2x_4)}{x_1}\right) + \epsilon_3$$

where x_1, x_2, x_3, x_4 are uniformly distributed over the ranges

$$\begin{aligned} 0 &\leq x_1 \leq 100 \\ 20 &\leq (x_2/2\pi) \leq 280 \\ 0 &\leq x_3 \leq 1 \\ 1 &\leq x_4 \leq 11 \end{aligned}$$

The noise ϵ_2, ϵ_3 are distributed as $N(0, \sigma_2^2)$, $N(0, \sigma_3^2)$ with σ_2, σ_3 selected to give 3:1 signal/noise ratios. In each example, the sample sizes are 200.

3.2. Computations

The two real data sets were divided at random in a learning set $\mathcal{L}$ and test set $\mathcal{T}$. For each of the simulated data sets, a learning set $\mathcal{L}$ of 200 cases was generated and a test set of 1000 cases. A regression tree was grown using $\mathcal{L}$ and 10-fold cross-validation. The test set $\mathcal{T}$ was run down $\mathcal{L}$ and gave mean-squared-error $e_S(\mathcal{L}, \mathcal{T})$.

Then 25 bootstrap replicates $\mathcal{L}^{(B)}$ of $\mathcal{L}$ were generated. For each one, a regression tree was grown using $\mathcal{L}^{(B)}$ and 10-fold cross-validation. This gave 25 predictors $\varphi_1(\boldsymbol{x}), \ldots, \varphi_{25}(\boldsymbol{x})$. For each $(y_n, \boldsymbol{x}_n) \in \mathcal{T}$, the predicted $\hat{y}_B$ value was taken as $av_k\ \varphi_k(\boldsymbol{x}_n)$. Then $e_B(\mathcal{L}, \mathcal{T})$ is the mean-squared-error between the $\hat{y}_B$ and the true y-values in $\mathcal{T}$. This procedure was repeated 100 times and the errors averaged to give the single tree error $\bar{e}_S$ and the bagged error $\bar{e}_B$.

4. Why Bagging Works

Let each $(y, \boldsymbol{x})$ case in $\mathcal{L}$ be independently drawn from the probability distribution P. Suppose y is numerical and $\varphi(\boldsymbol{x}, \mathcal{L})$ the predictor. Then the aggregated predictor is

$$\varphi_A(\boldsymbol{x}, P) = E_{\mathcal{L}} \varphi(\boldsymbol{x}, \mathcal{L}).$$

Take Y, $\boldsymbol{X}$ to be random variables having the distribution P and independent of $\mathcal{L}$. The average prediction error e in $\varphi(\boldsymbol{x}, \mathcal{L})$ is

$$e = E_{\mathcal{L}} E_{Y,\mathbf{X}} (Y - \varphi(\boldsymbol{X}, \mathcal{L}))^2.$$

Define the error in the aggregated predictor φ_A to be

$$e_A = E_{Y,\mathbf{X}} (Y - \varphi_A(\boldsymbol{X}, P))^2.$$

Using the inequality $(EZ)^2 \leq EZ^2$ gives

$$\begin{aligned} e &= EY^2 - 2EY\varphi_A + E_{Y,\mathbf{X}} E_{\mathcal{L}} \varphi^2(\boldsymbol{X}, \mathcal{L}) \\ &\geq E(Y - \varphi_A)^2 = e_A \end{aligned}$$

Thus, φ_A has lower mean-squared prediction error than φ. How much lower depends on how unequal the two sides of

$$[E_{\mathcal{L}}\varphi(\boldsymbol{x},\mathcal{L})]^2 \leq E_{\mathcal{L}}\varphi^2(\boldsymbol{x},\mathcal{L})$$

are. The effect of instability is clear. If $\varphi(\boldsymbol{x},\mathcal{L})$ does not change too much with replicate $\mathcal{L}$ the two sides will be nearly equal, and aggregation will not help. The more highly variable the $\varphi(\boldsymbol{x},\mathcal{L})$ are, the more improvement aggregation may produce. But φ_A always improves on φ.

Now, the bagged estimate is not $\varphi_A(\boldsymbol{x},P)$, but rather

$$\varphi_B(\boldsymbol{x}) = \varphi_A(\boldsymbol{x},P_{\mathcal{L}}),$$

where $P_{\mathcal{L}}$ is the distribution that concentrates mass $1/N$ at each point $(y_n,\boldsymbol{x}_n) \in \mathcal{L}$, ($P_{\mathcal{L}}$ is called the bootstrap approximation to P). Then φ_B is caught in two currents: on the one hand, if the procedure is unstable, it can give improvement through aggregation. On the other side, if the procedure is stable, then $\varphi_B = \varphi_A(\boldsymbol{x},P_{\mathcal{L}})$ will not be as accurate for data drawn from P as $\varphi_A(\boldsymbol{x},P) \simeq \varphi(\boldsymbol{x},\mathcal{L})$.

There is a cross-over point between instability and stability at which φ_B stops improving on $\varphi(\boldsymbol{x},\mathcal{L})$ and does worse. This has a vivid illustration in the linear regression subset selection example in the next section. There is another obvious limitation of bagging. For some data sets, it may happen that $\varphi(\boldsymbol{x},\mathcal{L})$ is close to the limits of accuracy attainable on that data. Then no amount of bagging will do much improving. This is also illustrated in the next section.

In classification, a predictor $\varphi(\boldsymbol{x},\mathcal{L})$ predicts a class label $j \in \{1,\ldots,J\}$. If $\mathcal{L}$ is drawn from the distribution P, and Y, $\boldsymbol{X}$ are from P independent of $\mathcal{L}$, then the probability of correct classification for $\mathcal{L}$ fixed is;

$$\begin{aligned} r(\mathcal{L}) &= P(Y = \varphi(\boldsymbol{X},\mathcal{L})) \\ &= \sum_j P(\varphi(\boldsymbol{X},\mathcal{L}) = j|Y = j)P(Y = j). \end{aligned}$$

Denote

$$Q(j|\boldsymbol{x}) = P_{\mathcal{L}}(\varphi(\boldsymbol{x},\mathcal{L}) = j).$$

Then, averaged over $\mathcal{L}$, the probability of correct classification is

$$r = \sum_j E(Q(j|\boldsymbol{X})|Y = j)P(Y = j)$$

$$= \sum_j \int Q(j|\boldsymbol{x})P(j|\boldsymbol{x})P_X(d\boldsymbol{x})$$

where $P_X(d\boldsymbol{x})$ is the overall $\boldsymbol{x}$ distribution.

Since $\varphi_A(\boldsymbol{x}) = \arg\max_i Q(i|\boldsymbol{x})$,

$$r_A = \sum_j \int I(\arg\max_i Q(i|\boldsymbol{x}) = j)P(j|\boldsymbol{x})P_X(d\boldsymbol{x})$$

where $I(\cdot)$ is the indicator function. Consider the set

$$C = \{\boldsymbol{x}; \arg\max_j P(j|\boldsymbol{x}) = \arg\max_j Q(j|\boldsymbol{x})\}.$$

For $\boldsymbol{x} \in C$

$$\sum_j I(\arg\max_i Q(i|\boldsymbol{x}) = j)P(j|\boldsymbol{x}) = \max_j P(j|\boldsymbol{x})$$

so that

$$r_A = \int_{\boldsymbol{x}\in C} \max_j P(j|\boldsymbol{x})P_X(d\boldsymbol{x}) + \int_{\boldsymbol{x}\in C'} \sum_j I(\varphi_A(\boldsymbol{x}) = j)P(j|\boldsymbol{x})P_X(d\boldsymbol{x}).$$

The highest attainable correct classification rate is given by the predictor

$$Q^*(\boldsymbol{x}) = \arg\max_j P(j|\boldsymbol{x})$$

and has the correct classification rate

$$r^* = \int \max_j P(j|\boldsymbol{x})P_X(d\boldsymbol{x}).$$

If $\boldsymbol{x} \in C$, the sum $\sum_j Q(j|\boldsymbol{x})P(j|\boldsymbol{x})$ can be less than $\max_j P(j|\boldsymbol{x})$. Thus, even if $P_X(C) \simeq 1$, the unaggregated predictor φ can be far from optimal. But φ_A is nearly optimal. Aggregating can therefore transform good predictors into nearly optimal ones. On the other hand, unlike the numerical y situation, poor predictors can be transformed into worse ones. The same behavior regarding stability holds. Bagging unstable classifiers usually improves them. Bagging stable classifiers is not a good idea.

5. A Linear Regression Illustration

5.1. Forward Variable Selection

Subset selection in linear regression gives an illustration of the points made in the previous section. With data of the form $\mathcal{L} = \{(y_n; \boldsymbol{x}_n),\ n = 1, \ldots, N\}$ where $\boldsymbol{x} = (x_1, \ldots, x_M)$ consists of M predictor variables, a popular prediction method consists of forming predictors $\varphi_1(\boldsymbol{x}), \ldots, \varphi_M(\boldsymbol{x})$ where each φ_m is linear in $\boldsymbol{x}$ and depends on only m of the M x-variables. Then one of the $\{\varphi_m\}$ is chosen as the designated predictor. For more background, see Breiman and Spector [1993].

A common method for constructing the $\{\varphi_m\}$, and one that is used in our simulation, is forward variable entry. If the variables used in φ_k are $x_{m_1}, \ldots, x_{m_k}$, then for each $m \notin \{m_1, \ldots, m_k\}$ form the linear regression of y on $(x_{m_1}, \ldots, x_{m_k}, x_m)$, compute the residual sum-of-squares RSS(m) and take $x_{m_{k+1}}$ such that m_{k+1} minimizes RSS(m) and $\varphi_{k+1}(\boldsymbol{x})$ the linear regression based on $(x_{m_1}, \ldots, x_{m_{k+1}})$.

There are other forms of variable selection i.e. best subsets, backwards and variants thereof. What is clear about all of them is that they are unstable procedures (see Breiman [1994]). The variables are competing for inclusion in the $\{\varphi_m\}$ and small changes in the data can cause large changes in the $\{\varphi_m\}$.

5.2. Simulation Structure

The simulated data used in this section are drawn from the model.

$$y = \sum_m \beta_m x_m + \epsilon$$

where ϵ is $N(0,1)$. The number of variables $M = 30$ and the sample size is 60. The $\{x_m\}$ are drawn from a mean-zero joint normal distribution with $EX_iX_j = \rho^{|i-j|}$ and at each iteration, ρ is selected from a uniform distribution on $[0,1]$.

It is known that subset selection is nearly optimal if there are only a few large non-zero β_m, and that its performance is poor if there are many small but non-zero β_m. To bridge the spectrum, three sets of coefficients are used. Each set of coefficients consists of three clusters; one is centered at $m = 5$, one at $m = 15$ and the other at $m = 25$. Each cluster is of the form

$$\beta_m = c[(h - |m-k|)^+]^2, \quad m = 1, \ldots, 30$$

where k is the cluster center, and $h = 1, 3, 5$ for the first, second and third set of coefficients respectively. The normalizing constant C is taken so that the R^2 for the data is $\simeq .75$. Thus, for $h = 1$, there are only three non-zero $\{\beta_m\}$. For $h = 3$ there are 15 non-zero $\{\beta_m\}$, and for $h = 5$, there are 27 non-zero $\{\beta_m\}$, all relatively small.

For each set of coefficients, the following procedure was replicated 250 times:

i). Data $\mathcal{L} = \{(y_n, \boldsymbol{x}_n),\ n = 1, \ldots, \}$ was drawn from the model

$$y = \sum \beta_m x_m + \epsilon$$

where the $\{x_m\}$ were drawn from the joint normal distribution described above.

ii). Forward entry of variables was done using $\mathcal{L}$ to get the predictors $\varphi_1(\boldsymbol{x}), \ldots, \varphi_M(\boldsymbol{x})$. The mean-squared prediction error of each of these was computed giving $e_1, \ldots, e_M$.

iii). Fifty bootstrap replicates $\{\mathcal{L}^{(B)}\}$ of $\mathcal{L}$ were generated. For each of these, forward stepwise regression was applied to construct predictors $\{\varphi_1(\boldsymbol{x}, \mathcal{L}^{(B)}), \ldots, \varphi_M(\boldsymbol{x}, \mathcal{L}^{(B)})\}$. These were averaged over the $\mathcal{L}^{(B)}$ to give the bagged sequence $\varphi_1^{(B)}(\boldsymbol{x}), \ldots, \varphi_M^{(B)}(x)$. The prediction errors $e_1^{(B)}, \ldots, e_M^{(B)}$ for this sequence was computed.

These computed mean-squared-errors were averaged over the 250 repetitions to give two sequences $\{\bar{e}_m^{(S)}\}$, $\{\bar{e}_m^{(B)}\}$. For each set of coefficients, these two sequences are plotted vs. m in Figure 1a,b,c.

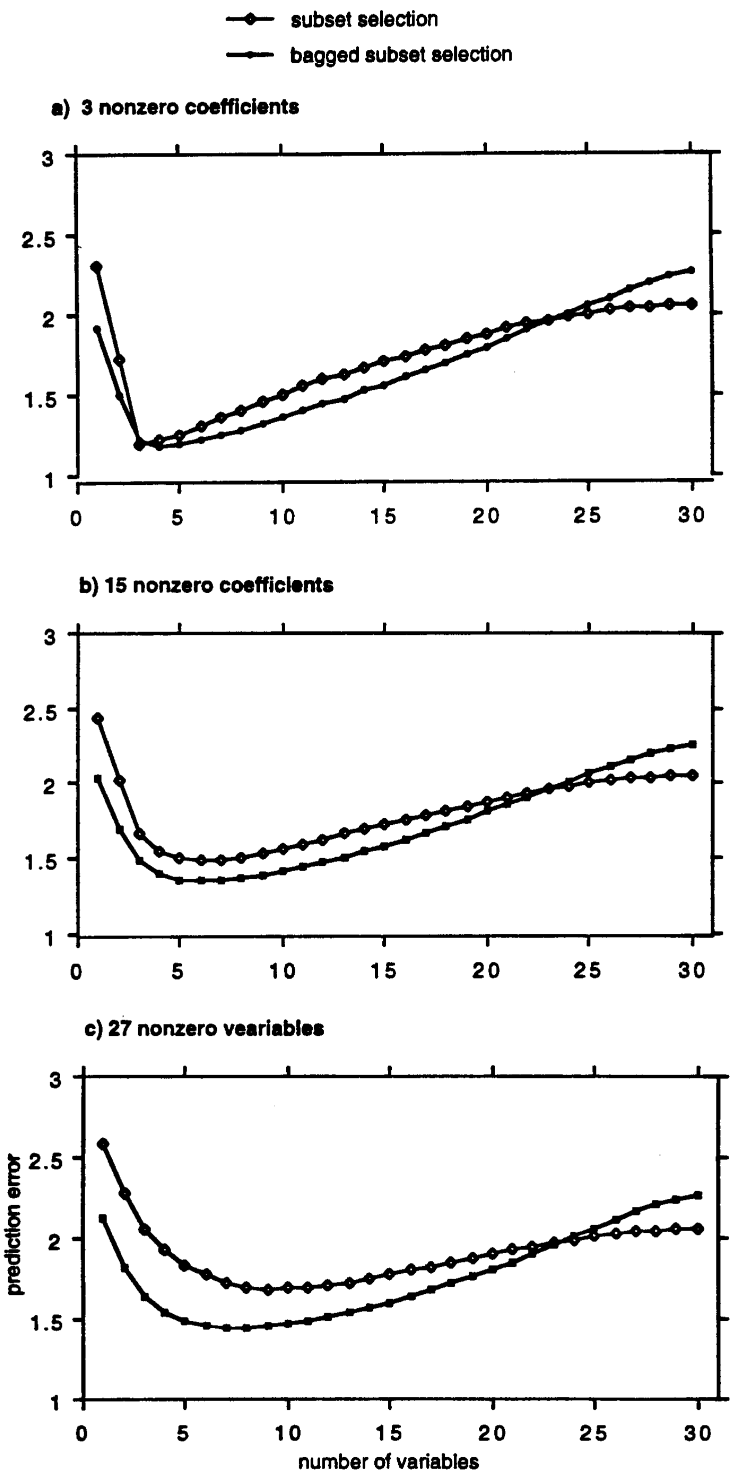
5.3. Discussion of Simulation Results

First and most obvious is that the best bagged predictor is always at least as good as the best subset predictor. When $h = 1$ and subset selection is nearly optimal, there is no improvement. For $h = 3$ and 5 there is substantial improvement. This illustrates the obvious: bagging can improve only if the unbagged is not optimal.

The second point is less obvious. Note that in all three graphs there is a point past which the bagged predictors have larger prediction error than the unbagged. The explanation is this: linear regression using *all* variables is a fairly stable procedure. The stability decreases as the number of variables used in the predictor decreases. As noted in section 4, for a stable procedure $\varphi_B = \varphi_A(\boldsymbol{x}, P_{\mathcal{L}})$ is not as accurate as $\varphi \simeq \varphi(\boldsymbol{x}, P)$. The higher values of

FIGURE 1

Prediction Error for Subset Selection and Bagged Subset Selection vs. Number of Variables



a) 3 nonzero coefficients

b) 15 nonzero coefficients

c) 27 nonzero veariables

$\bar{\varphi}_m^{(B)}$ for large m reflect this fact. As m decreases, the instability increases and there is a cross-over point of which $\varphi_m^{(B)}$ becomes more accurate than φ_m

6. Concluding Remarks

6.1. Bagging Class Probability Estimates

Some classification methods estimate probabilities $\hat{p}(j|\boldsymbol{x})$ that an object with prediction vector $\boldsymbol{x}$ belongs to class j. Then the class corresponding to $\boldsymbol{x}$ is estimated as $\arg\max_j \hat{p}(j|\boldsymbol{x})$. For such methods, a natural competitor to bagging by voting is to average the $\hat{p}(j|\boldsymbol{x})$ over all bootstrap replications, getting $\hat{p}_B(j|\boldsymbol{x})$ and then use the estimated class $\arg\max_j \hat{p}_B(j|\boldsymbol{x})$. This estimate was computed in every classification example we worked on. The resulting missclassification rate was always virtually identical to the voting missclassification rate. In some applications, estimates of class probabilities are required, instead of, or along with, the classifications. The evidence so far indicates that bagged estimates are likely to be more accurate than the single estimates. To verify this, it would be necessary to compare both estimates with the true values $p^*(j|\boldsymbol{x})$ over the $\boldsymbol{x}$ in the test set. For real data the true values are unknown. But they can be computed for the simulated waveform data, where they reduce to computing an expression involving error functions.

Using the waveform data, we did a simulation similar to that in Section 2 with learning and test sets both of size 300, and 25 bootstrap replications. In each iteration, we computed the average over the test set and classes of $|\hat{p}(j|\boldsymbol{x}) - p^*(j|\boldsymbol{x})|$ and $|\hat{p}_B(j|\boldsymbol{x}) - p^*(j|x)|$. This was repeated 50 times and the results averaged. The single tree estimates had an error of .189. The error of the bagged estimates was .124, a decrease of 34%.

6.2. How Many Bootstrap Replicates Are Enough?

In our experiments, 50 bootstrap replicates was used for classification and 25 for regression. This does not mean that 50 or 25 were necessary or sufficient, but simply that they seemed reasonable. My sense of it is that fewer are required when y is numerical and more are required with an increasing number of classes.

The answer is not too important when procedures like CART are used, because running times, even for a large number of bootstraps, are very nominal. But neural nets progress much slower and replications may require many days of computing. Still, bagging is almost a dream procedure for parallel computing. The construction of a predictor on each $\mathcal{L}^{(B)}$ proceeds with no communication necessary from the other CPU's.

To give some ideas of what the results are as connected with the number of bootstrap replicates we ran the waveform data using 10, 25, 50 and 100 replicates using the same simulation scheme as in Section 2. The results are:

Table 5.1
Bagged Missclassification Rates (%)

No. Bootstrap Replicates	Missclassification Rate
10	21.8
25	19.5
50	19.4
100	19.4

The unbagged rate is 29.0, so its clear that we are getting most of the improvement using only 10 bootstrap replicates. More than 25 bootstrap replicates is love's labor lost.

6.3. Bagging Nearest Neighbor Classifiers

Nearest neighbor classifiers were run on all the data sets described in section 2 except for the soybean data whose variables were categorical. The same random division into learning and test sets was used with 100 bootstrap replicates, and 100 iterations in each run. A Euclidean metric was used with each coordinate standardized by dividing by its standard deviation over the learning set. See Table 5 for the results:

Table 5
Missclassification Rates for Nearest Neighbor

Data Set	$\bar{e}_S$	$\bar{e}_B$
waveform	26.1	26.1
heart	6.3	6.3
breast cancer	4.9	4.9
ionosphere	35.7	35.7
diabetes	16.4	16.4
glass	21.6	21.6

Nearest neighbor is more accurate than single trees in 5 of the 6 data sets, but bagged trees are more accurate in 5 of the 6 data sets.

Cycles did not have to be expended to find that bagging nearest neighbors does not change things. Some simple computations show why. Given N possible outcomes of a trial (the N cases $(y_n, \boldsymbol{x}_n)$ in the learning set) and N trials, the probability that the nth outcome is selected $0, 1, 2, \ldots$ times is approximately Poisson distributed with $\lambda = 1$ for large N. The probability that the nth outcome will occur at least once is $1 - (1/e) \simeq .632$.

If there are N_B bootstrap repetitions in a 2-class problem, then a test case may change classification only if its nearest neighbor in the learning set is not in the bootstrap sample in at least half of the N_B replications. This probability is given by the probability that the number of heads in N_B tosses of a coin with probability .632 of heads is less than $.5N_B$. As N_B gets larger, this probability gets very small. Analogous results hold for J-class problems.

The stability of nearest neighbor classification methods with respect to perturbations of the data distinguishes them from competitors such as trees and neural nets.

6.4. **Conclusions**

Bagging goes a ways toward making a silk purse out of a sow's ear, especially if the sow's ear is twitchy. It is a relatively easy way to improve an existing method, since all that needs adding is a loop in front that selects the bootstrap sample and sends it to the procedure and back end that does the aggregation. What one loses, with the trees, is a simple and interpretable structure. What one gains is increased accuracy.

References

Belsley, D., Kuh, E., and Welsch, R. (1980) "Regression Diagnostics", John Wiley and Sons.

Breiman, L. (1994) Heuristics of instability in model selection, Technical Report, Statistics Department, University of California at Berkeley.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984) "Classification and Regression Trees", Wadsworth.

Breiman, L. and Friedman, J. (1985) Estimating optimal transformations in multiple regression and correlation (with discussion), *Journal of the American Statistical Association*, **80**, 580-619 .

Breiman, L. and Spector, P (1992) Submodel Selection and Evaluation in Regression – the X-Random Case, International Review of *Statistics*, **3**, 291-319

Buntine, W. (1991) "Learning classification trees", Artificial Intelligence Frontiers in Statistics, ed D.J. Hand, Chapman and Hall, London, 182-201.

Dietterich, T.G. and Bakiri, G. (1991) Error-correcting output codes: A general method for improving multiclass inductive learning programs, Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), Anaheim, CA: AAAI Press.

Efron, B., and Tibshirani, R. (1993) "An Introduction to the Bootstrap". Chapman and Hall.

Friedman, J. (1991) Multivariate adaptive regression splines (with discussion), *Annals of Statistics*, **19**, 1-141.

Heath, D., Kasif, S., and Salzberg, S. (1993) k-dt: a multi-tree learning method. Proceedings of the Second International Workshop on Multistrategy Learning, 1002-1007, Chambery, France, Morgan Kaufman.

Kwok, S., and Carter, C. (1990) Multiple decision trees, Uncertainty in Artificial Intelligence 4, ed. Shachter, R., Levitt, T., Kanal, L., and Lemmer, J., North-Holland, 327-335.

Olshen, R., Gilpin, A., Henning, H., LeWinter, M., Collins, D., and Ross, J. (1985) Twelve-month prognosis following myocardial infarction: Classification trees, logistic regression, and stepwise linear discrimination, Proceedings of the Berkeley conference in honor of Jerzy Neyman and Jack Kiefer, L. Le Cam;R. Olshen, (Ed), Wadsworth, 245-267.

Smith, J., Everhart, J., Dickson, W., Knowler, W., and Johannes, R. (1988) Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical

Care 261–265. IEEE Computer Society Press.

Sigillito, V. G., Wing, S. P., Hutton, L. V.,and Baker, K. B. (1989) Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, **10**, 262-266.

Wolberg, W. and Mangasarian, O (1990) Multisurface method of pattern separation for medical diagnosis applied to breast cytology, Proceedings of the National Academy of Sciences, U.S.A., Volume 87, December 1990, pp 9193-9196.