4

Bagging

4.1 Introduction

In this chapter, we make a major transition. We have thus far focused on statistical learning procedures that produce a single set of results: regression coefficients, measures of fit, residuals, classifications, and others. Thus, there is but one regression equation, one set of smoothed values, or one classsification tree. Most statistical procedures operate in a similar fashion.

The discussion now shifts to statistical learning building on many sets of outputs that are aggregated to produce results. Such algorithms make a number of passes over the data. On each pass, inputs are linked to outputs just as before. But the ultimate results of interest are the collection of all the results from all passes over the data.

Bayesian model averaging may be a familiar illustration from another statistical tradition (Madigan et al., 1996; Hoeting et al., 1999). In Bayesian model averaging, there is an assumed $f(X)$; there is a "true model." A number of potentially true models, differing in the predictors selected, are evaluated. The model output is then averaged with weights determined by model uncertainty. Output from models with greater uncertainty are given less weight. From a statistical learning perspective, Bayesian model averaging has a number of problems, including the dependence that is necessarily built in across model results (Xu and Golay, 2006). We address shortly how statistical learning procedures relying on multiple results proceed rather differently.

Aggregate results can have several important benefits. Averaging over a collection of fitted values can help compensate for overfitting. That is, the averaging tends to cancel out results shaped by idiosyncratic features of the data. One can then obtain more stable fitted values and more honest assessments of how good the fit really is. Second, a large number of fitting attempts can produce very flexible fitting functions able to respond to systematic, but highly localized, features of the data. In effect, there can be a very large number of basis functions and the prospect of reducing bias in the fitted values. Third, putting the averaging and the flexible fitting functions together has the

R.A. Berk, *Statistical Learning from a Regression Perspective*,
DOI: 10.1007/978-0-387-77501-2_4, © Springer Science+Business Media, LLC 2008

potential to break the bias–variance tradeoff. Sometimes you can have your cake and eat it too.

In this chapter, we focus on bagging, which capitalizes on the averaging process. Averaging can reduce the variance. There are also some implications for bias. Later chapters consider statistical learning procedures that in different ways address more directly bias in the fitted values as well as the variance.

We emphasize categorical response variables. We are again concentrating on classifiers. The rationale is largely the same: the exposition is more effective and the step to quantitative predictors is easy to make. We begin with a return to the problem of overfitting. Although overfitting has been discussed several times in earlier chapters, it needs to be linked more directly to CART to help set the stage for our exposition of bagging.

4.2 Overfitting and Cross-Validation

A long-standing problem in the philosophy of science is whether the credibility of scientific conclusions is greater if the conclusions are evaluated through their forecasting skill or their consistency with the data on hand. That is, what weight should be given to an accurate forecast compared to a good fit? The answer is not straightforward, but in the end, accurate forecasts are likely to be more convincing. And one of the reasons is that forecasts are not vulnerable to overfitting, whether from intentional "fudging" or overzealous data exploration (Lipton, 2005).

Any attempt to summarize patterns in a dataset risks overfitting. All fitting procedures adapt to the data on hand so that even if the results are applied to a new sample from the same population, fit quality will likely decline. Hence, generalization can be somewhat risky. And to the degree that a fitting procedure is highly flexible, overfitting can be exacerbated. There is a greater opportunity to fit idiosyncratic features of the data. For example, Hastie et al. (2001: 200–203) show in a slightly different context that the unjustified "optimism increases linearly with the number of inputs or basis functions ..., but decreases as the training sample size increases." In other words, it can be highly desirable to have few parameters to be estimated and many observations with which to construct the estimates.

Consider CART as a key illustration. The basis function formulation can be instructively introduced at three points in the fitting process. First, for any given predictor being examined for its best split, overfitting will increase with the number of splits possible. In effect, a greater number of basis functions are being screened (where a given split leads to a basis function). Second, for each split, CART evaluates all possible predictors. An optimal spilt is chosen over all possible splits of all possible predictors. This defines the optimal basis function for that stage. Hence within each stage, overfitting increases as the number of candidate predictors increases. Third, for each new stage, a new optimal basis function is chosen and applied. Consequently, overfitting

increases with the number of stages, which for CART means the number of optimal basis functions, typically represented by the number of nodes in the tree.

The overfitting in CART can be misleading in a number of ways. Measures meant to reflect how well the model fits the data are likely to be too optimistic. Thus, for example, the number of classification errors may be too small. In addition, the model itself may have a structure that will not generalize well. For example, one or more predictors may be included in a tree that really do not belong. Finally, should statistical inference be introduced, standard errors can be too small. Overly narrow confidence intervals and falsely powerful tests follow.

Ideally, one would have two random samples from the same population: a training dataset and a test dataset. A tree would be built from the training data, and some measure of fit would be obtained. A simple measure might be the fraction of cases classified correctly. A more complicated measure might take the costs of false negatives and false positives into account. Then with the tree structure in place, cases from the test data would be "dropped down" the tree, and the fit computed again. It is almost certain that the fit would degrade, with how much being a measure of overfitting. The fit measure from the test data would be a better indicator of how accurate the classification process really is.

Often there is only a single dataset. Enter cross-validation. The data are split up into several randomly chosen, nonoverlapping, partitions of about the same size. That is, one samples without replacement. Ten such subsets are common. CART is applied to the data from nine of the partitions, and the results are evaluated with the remaining partition. So, if there are 1000 observations, one would build the tree on 900 randomly selected observations and evaluate the tree using the other 100 observations.

With ten partitions, the building and testing sequence could be undertaken ten times, each time with nine partitions as the training data and one partition as the test data. Each of the ten partitions would be part of the training data for nine of the ten analyses, and would serve as the test data for one of the ten analyses. From each of the ten test partitions, a measure of fit would be computed. An instructive measure of fit would be the average fit value over the ten splits. Relying on the test partitions reduces overfitting. Taken one at a time, the small test partitions can be vulnerable to sampling error. The averaging process tends to cancel out some chance variation. There is nothing magic about using ten random partitions of the data. When there are very few partitions, each training dataset will have far fewer observations than the entire sample. Insofar as the CART results are sample size dependent, substantial bias can be introduced. For example, with a smaller training sample, a less complex tree might result. However, when there are a great many partitions, largely the same data are used over and over to construct the tree, and the test datasets have very few observations. Then, the fit measure computed

from the test data can have high variance (Hastie et al., 2001: Section 7.10). Using five to ten splits seems to be a good compromise in practice.

Cross-validation is available in many implementations of CART and is discussed in the seminal book by Breiman and his colleagues (1984: Section 11.5). Often the number of splits of the data can be specified. When the number of splits is the same as the number of observations in the original sample, the process is sometimes called "leave-one-out" cross-validation. We discussed this in Chapter 1 when model evaluation was first addressed. As noted then, extensions on this basic idea using bootstrap samples are available (Efron and Tibshirani, 1993: Chapter 17).

Unfortunately, cross-validiation neglects the extracted pattern of associations between the inputs and the outputs, which may, because of overfitting, be very misleading. Although one may obtain a more honest measure of overall performance, the structure of the associations revealed by the analysis is not addressed. One may be stuck with a tree that makes little substantive sense or will not generalize well. But in the use of subsamples of the data and in averaging over subsamples, there is a very powerful idea. Bagging exploits that idea to address overfitting in a more fundamental manner.

4.3 Bagging as an Algorithm

The notion of combining fitted values from a number of fitting attempts has been suggested by several authors (LeBlanc and Tibshirani, 1996; Mojirsheibani, 1997; 1999) In an important sense, the whole becomes more than the sum of its parts. "Bagging," which stands for "Bootstrap Aggregation," is perhaps the earliest procedure to exploit a combination of fitted values based on random samples of the data (Breiman, 1996). Bagging may be best understood initially as nothing more than an algorithm.

Consider the following steps in a fitting algorithm with a dataset having N observations and a binary response variable.

1. Take a random sample of size N with replacement from the data.
2. Construct a classification tree as usual but do not prune.
3. Assign a class to each terminal node, and store the class attached to each case coupled with the predictor values for each observation.
4. Repeat Steps 1-3 a large number of times.
5. For each observation in the dataset, count the number of times over trees that it is classified in one category and the number of times over trees it is classified in the other category
6. Assign each observation to a final category by a majority vote over the set of trees. Thus, if 51% of the time over a large number of trees a given observation is classified as a "1," that becomes its classification.
7. Construct the confusion table from these class assignments.

Although there remain some important variations and details to consider, these are the key steps to produce "bagged" classification trees. The idea of classifying by averaging over the results from a large number of bootstrap samples generalizes easily to a wide variety of classifiers beyond CART. Later we show that bagging can be usefully applied for quantitative responses as well.

4.3.1 Margins

Bagging introduces some new concepts that need to be addressed, not just to deepen the understanding of bagging, but for some other procedures considered in later chapters. One of these concepts is the "margin."

Operationally, the difference between the proportion of times a case is correctly classified and the proportion of times it is incorrectly classified is sometimes called the "margin" for that case. If, over all trees, an observation is correctly classified 75% of the time and incorrectly classified 25% of the time, the margin is $.75 - .25 = .50$. Large margins are desirable because a more stable classification is implied. In a large number of random samples of the data, the class assigned to that observation is far more likely than not to be the same. Ideally, there should be large margins for all of the observations. This bodes well for generalization to new data. A more formal and extensive treatment of the concept of the "margin" is provided in the next chapter.

Recall the discussion in the previous chapter on instability in CART fitted values. Overfitting in CART tends to be more serious when for the terminal nodes the proportions of observations in each of the response variable classes tend to be similar. If the split is .51 versus 49, for instance, the movement of a little more than one percent of the cases from one class to another could change the class assigned to that node. Then, all of the cases that were correctly classified are now misclassified, and all of the cases incorrectly classified are now correctly classified. Were another sample taken, the initial node class might be reassigned, and the pattern of classification errors would change again. It follows that for any given observation in this terminal node, the margin is likely to be very small or even negative. Such observations will not be classified in a reliable manner. One might say that the vote over trees is too close to call.

Conversely, if the proportions within each terminal node are quite different, it would take the movement of relatively many cases to change the classes assigned. The bagged margins for observations across trees are likely to be larger and the classifications more stable. More reliable classifications result. One might say that the vote is a landslide.

4.3.2 Out-Of-Bag Observations

In the steps just described, the tree is built and then the data used to build the tree are used again to compute the classification error. One way to think

about this is that training data are "dropped down" the tree to determine how well the tree performs. The training data are "resubstituted" when tree performance is evaluated.

In some implementations of bagging, one can do better. For each tree, observations not included in the bootstrap sample (called "out-of-bag" observations) can be treated as a test dataset. These are then dropped down the tree instead of the data used to build the tree. A record is kept of the class with which each out-of-bag observation is labeled, as well as its values on all of the predictors. Then in the averaging process, it is these assigned values that are used as class labels, and based on these, a confusion table contructed. In other words, the averaging for a given observation over trees is done only using the trees for which that observation was not used in the fitting process. Thus, a fitting enterprise has been turned into a genuine forecasting enterprise. This leads to more honest fitted values and more honest confusion tables.

It is important to emphasize that the improvement will usually be seen in forecasting accuracy. If bagged CART results are compared to the results from a single classification tree, the single tree may seem to perform better. But this is misleading. If resubstituted values are used to construct the confusion tables for both the single tree and the bagged trees, the bagged trees should look worse. The bagging results have been adjusted for overfitting, at least in part. When out-of-bag data are used to construct the confusion table for the bagged trees, the bagged results will appear to suffer even more by comparison. A fair competition between the performance of a single tree and a set of bagged trees requires confusion tables for both procedures constructed from test data. On this level playing field, bagged trees will usually perform better than single trees. Some exceptions are considered shortly.

In summary, by assigning cases to categories using a majority vote over a set of bootstrapped classification trees, overfitting can be dramatically curtailed. Forecasting accuracy is improved because generalization error is reduced. Using the out-of-bag observations can further curb the potential overfitting.

4.4 Some Thinking on Why Bagging Works

The core of bagging's potential is found in the averaging over results from a substantial number of bootstrap samples. As a first approximation, the averaging helps to cancel out the impact of random variation. However, there is more to the story, some details of which are especially useful for understanding a number of statistical learning procedures discussed in subsequent chapters.

4.4.1 More on Instability in CART

One can get an initial sense of the need for bagging from Figures 4.1 to 4.3. The three figures are three classification trees constructed from the same data,

but each uses a different bootstrap sample (i.e., sampled with replacement from the data). The data were collected to help forecast incidents of domestic violence within households served by a sheriff's department from a large metropolitan area. For a sample of households to which sheriff's deputies were dispatched for domestic violence incidents, the deputies collected information on a series of possible predictors of future domestic violence. For example, they determined whether police officers had been called to that household in the recent past. Then, the households were followed for two months and any new incidents of domestic violence recorded. The data were used to construct a forecasting algorithm so that when information was collected on new households, forecasts of the likelihood of more domestic violence incidents could be made.

It is clear that the three figures are very different. Although each tree's initial splitting variable is the number of times the police had been called to that household before, different break points are chosen. More important, the subsequent splits vary widely across the three trees. It is clear that in this instance, CART does not produce trees that are likely to be stable under different random samples from the same population. It would follow that interpretations of the results would be unreliable.

This is a very important lesson. Interpretations from the results of a single tree can be quite risky when CART performs in this manner. And recall from the previous chapter that CART can produce unstable results because of any number of common problems: small sample sizes, heterogeneous terminal nodes, or highly correlated predictors.

Also problematic may be the classes that CART assigns to nodes. For each of the figures, the sample sizes in the terminal nodes are generally quite small. This can increase substantially the instability of the classes assigned. With node distributions such as 4 to 3, 5 to 4, or even 9 to 6, changes in the composition of the data from sample to sample could easily alter how the observations in a node are classified or even whether the node is constructed at all.

However, if over trees the different nodes in which a given case might fall tend to classify that case in the same manner, instability in tree structure does not necessarily translate into instability in the class assigned. In other words, when CART is used solely as a classification tool, the classes assigned may be relatively stable even if the tree structure is not. Experience suggests that such is sometimes the case. Recall that much the same phenomenon can be found in conventional regression when predictors are highly correlated. The regression coefficients estimated for particular predictors may be very unstable, but it does not necessarily follow that the fitted values will be unstable as well.

Finally, each tree has many terminal nodes. As a result, each tree represents a very flexible fitting function; there are a large number of conditional proportions estimated. As a result, the number of classification errors is likely to be relatively few for each tree individually. Each fit, therefore, may be quite

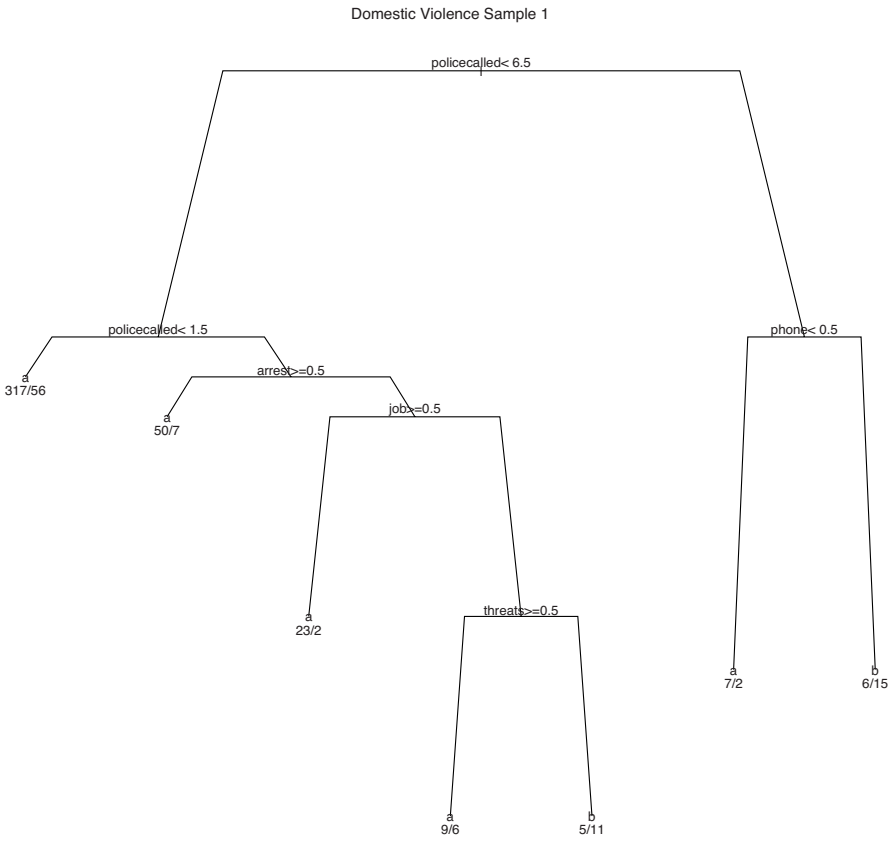

Fig. 4.1. Tree diagram for first bootstrap sample.

good. Were the original data a random sample from a well-defined population, one might be able to argue that the bias in the assigned classes is small.

At the same time, one must be clear about what is being estimated. Suppose there is a real population and CART were applied to all of the data in that population. Then, if CART is applied to a random sample of observations from that population, one might be able to grow a tree providing unbiased estimates of the splits and the fitted values. A requirement would be to have a large enough sample so that a sufficiently large tree could be grown with the sample data. That is, all of the terminal nodes in the population tree

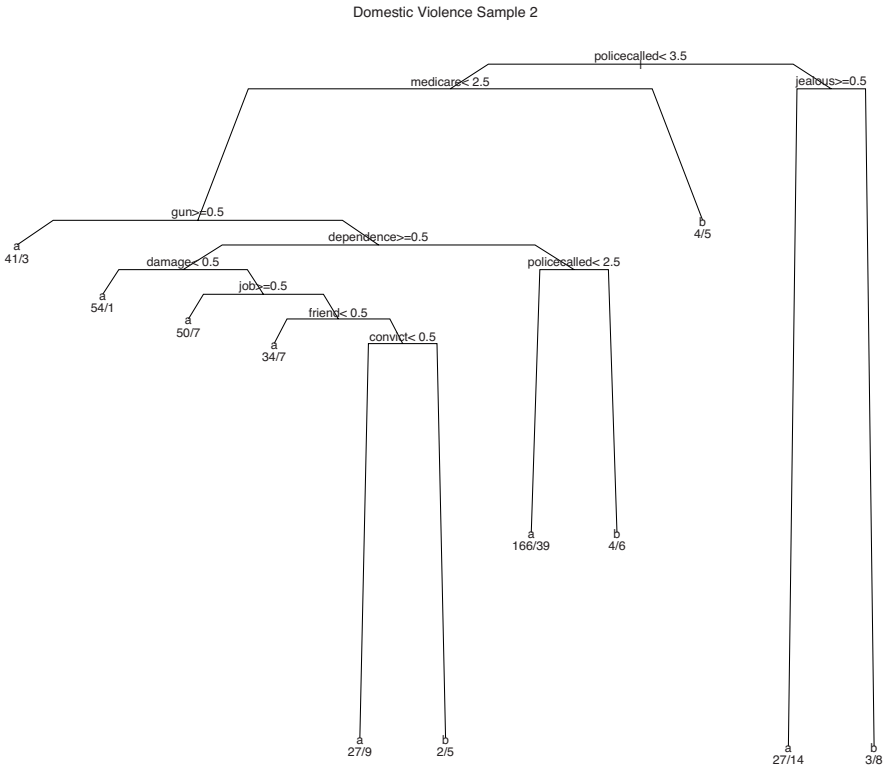

Fig. 4.2. Tree diagram for second bootstrap sample.

could be reproduced with the sample data. The deeper problem is that there is no guarantee whatsoever that the population tree, let alone the sample tree, captures the way in which the population data were generated. In the most obvious case, there may be no information in the population about all predictors that in fact were relevant.

At a more abstract level, the same concerns apply to data said to be a product of a particular stochastic process. If that stochastic process really functions through mechanisms that comport with a classification or regression tree, and if the inputs to that stochastic process are included in the dataset being analyzed, there is again the possibility of obtaining unbiased tree esti-

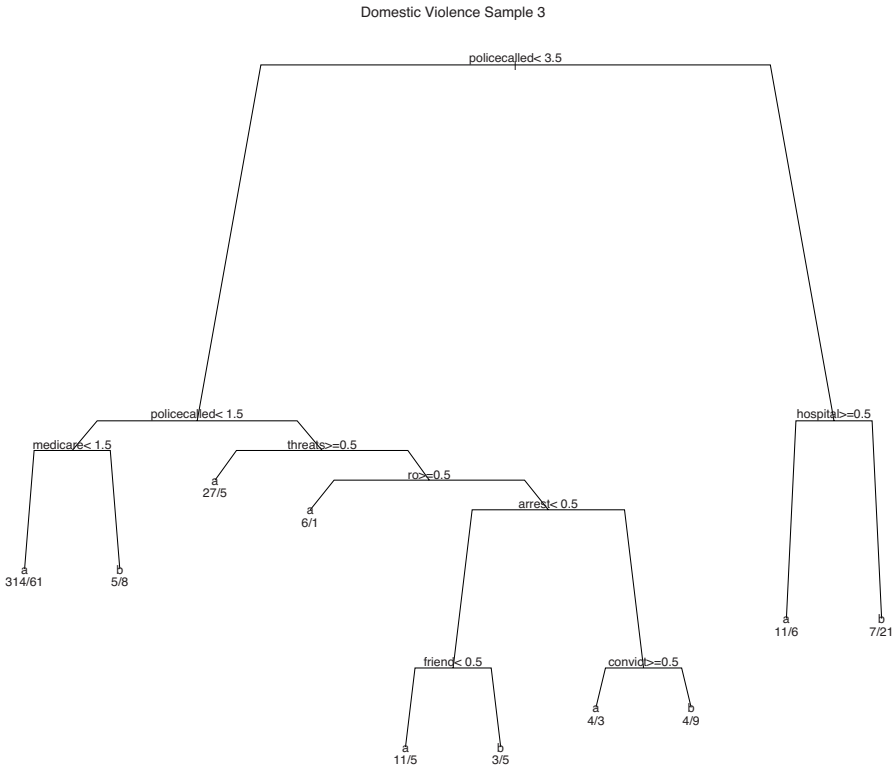

Fig. 4.3. Tree diagram for third bootstrap sample.

mates. But if the stochastic process does not comport with a classification or regression tree, or if the requisite predictors are unavailable, bias will likely result.

In short, it is not clear how much bias exists in the three trees. But it is clear that the variance across trees is large. Bagging can help with the variance.

4.4.2 How Bagging Can Help

Consider the classifications that would follow from each tree. Suppose that for each observation one averaged over trees to determine the class assigned. With a binary outcome, the averaging would take form of a vote across trees. Because there are three trees in this illustration, a majority vote would be two out of three or three out of three. If an observation were classified as having a new incident of domestic violence (i.e., "b") in two of the three trees or in three of the three trees, it would be classified as a high-risk domestic violence household. If an observation were classified as having a new incident of domestic violence in none of the three trees or one of the three trees, it would not be classified as a chronic domestic violence household.

As an averaging process, voting over trees tends to cancel out the impact of random sampling error on the classes assigned to observations (Brieman, 1996; 2000). Idiosyncratic results from tree to tree can be averaged away and more stable estimates can follow. The variance in the assigned classes can be reduced as a consequence.

The idea of independent random samples from a population must not be confused with bootstrap samples from the data. Independent random samples from a population are the conceptual foundation for conventional (frequentist) statistical inference. One works within the thought experiment of a limitless number of independent random samples from a population or a limitless number of independent realizations of a stochastic process. The definitions of the bias and variance for a statistic computed from the data on hand follow from this thought experiment.

Bootstrap samples are probability samples with replacement from the data on hand. Often such procedures are justified as an effort to simulate the thought experiment. For bagging, however, the bootstrap samples serve another purpose: they are the foundation for the averaging process by which the bias–variance tradeoff may be constructively addressed. Statistical inference is not the motivation. One can think of the averaging as a kind of shrinkage that can, as before, increase the stability of the fitted values. For each observation, fitted values are pulled toward their mean over bootstrap samples.

There is actually no requirement in bagging that the samples drawn from the training data be with replacement. It seems that in general, one can use samples without replacement and obtain virtually the same results as long as a particular relationship is maintained between the size of the larger samples with replacement and the size of the smaller samples without replacement (Buja and Stuetzle, 2006). If there are N observations in the training data, a sample without replacement of $N/2$ effectively will produce the same bagged results as a sample with replacement of size N. So, the key idea is working with a large number of random samples of the training data. Whether the sampling is with or without replacement does not by itself seem to be a critical factor.

The results developed by Buja and Steutze (2006) make clear that when sampling with replacement, the nominal sample size can be larger than N, and

as that sample size increases, the equivalent sample size for sampling without replacement can approach N. However, there is no definitive message about what the ideal sample size should be, whether with or without replacement. Therefore, the discussion that follows emphasizes a sample of size N sampling with replacement, consistent with the traditional bootstrap.

Finally, bagging can have implications for the bias (Bühlmann and Yu, 2002). The basic concern is this: bagging acts as a smoother for the step functions CART produces. If the underlying $f(X)$ is smooth, bagging will tend to reduce bias by "sanding off" the corners of the step functions. If the underlying $f(X)$ has the same jagged structure as step functions, "sanding off" the corners can increase the bias. Apparently, neither of these consequences were anticipated in the initial work on bagging but were eventually recognized as a byproduct of the averaging that bagging employs. More is said about bagging and bias shortly.

4.4.3 A Somewhat More Formal Explanation

We can now formalize these ideas a bit (Breiman, 1996) by applying concepts from conventional regression analysis. We begin with a discussion of the variance and a simple illustration to set the stage.

Bagging and Variance

Consider first a given predictor value x_0 and an associated response. Imagine a single random draw from a population, conditional on x_0. The value of the response for that draw is an unbiased estimate of the mean response for all observations in the population with the same value of x, x_0. But because that estimate is constructed from a single observation, the estimate can vary a lot from sample to sample if the response is not homogeneous at x_0. Had a random sample of, say, ten observations at x_0 been drawn instead, the mean of those 10 values would still be an unbiased estimate of the mean of the responses at x_0. But now, the sampling variability would likely be much smaller because the sample size is much larger. With more observations one can shrink the variance, and in this case, still have an unbiased estimate.

Consider now a more complicated illustration. We assume for the moment that the response variable is quantitative. We focus on a single observation. For that observation, assume there is a true function of the predictor $f(x_0)$ through which the response is related to x_0. That true function can be found in the population or in the stochastic process responsible for the data. What is the mean squared error of an observed value of the response variable y with respect to the fitted value $\hat{f}(x_0)$?

The mean squared error over repeated random samples (or realizations) can be decomposed into the sum of three parts: !1) an irreducible error, (2) the bias in the fitted value, and (3) the variance of the fitted value. More explicitly (Hastie et al., 2001: 197),

$$E[(y-\hat{f}(x_0))^2|x=x_0] = \sigma_\varepsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2. \quad (4.1)$$

There is nothing that can be done about the σ_ε^2. It reflects the variance of y around its true conditional mean at x_0. Generally, the more complex the model $\hat{f}$, the smaller the squared bias, shown in Equation 4.1 as $[E\hat{f}(x_0) - f(x_0)]^2$. A more complex model will generally fit the data better. But with greater complexity, a greater number of degrees of freedom is used up. The likely result is greater variance, shown in Equation 4.1 as $E[\hat{f}(x_0) - E\hat{f}(x_0)]^2$. Put another way, the available information in the sample is being spread more thinly over the fitted values being estimated. The bias–variance tradeoff is with us again.

Bagging can, in principle, usefully address the link between the bias and the variance. For any given amount of bias, averaging over many bootstrap samples produces a far more stable collection of fitted values than is likely from any single sample. It is as if one had a large number of samples (or realizations) generated by the frequentist thought experiment. Moreover, because bagging helps to produce more stable estimates, one is more free to fit complex functions to the data. If there is a subject matter rationale for fitting a tree with a large number of terminal nodes, for example, concerns about high variance need not automatically be a serious constraint.

Equation 4.1 should be understood as illustrative. In particular, it does not literally apply when the response variable is categorical. When models for categorical data are used, the bias of the fitted values is related to the variance of the fitted values. The simple partitioning shown in Equation 4.1 does not follow. Nevertheless, the same general implications apply.

Bagging and Bias

Having addressed the variance, we turn to the bias. Figure 4.4 illustrates how bagging can affect the bias. To keep the graph simple, there is a single predictor with the $f(X)$ the smooth S-shaped function shown linking the predictor to a binary 0/1 response. Imagine now that CART is applied one time to each of four different bootstrap samples of the data. Each time, only one break in the predictor is allowed. (Such trees are sometimes call "stumps.") The four step functions that result are overlaid.

Consider now a single value of the predictor x_0 of 14. At x_0, the value of $f(X)$ is about .8. If only the single step function on the far right were available, $\hat{f}(X)$ would be around .9. If only the single step function just to the left were available, $\hat{f}(X)$ would be around .75. Yet, the average of the two would be pretty close to .8. More generally, with a greater number of CART step functions averaged, the S-shaped $f(X)$ is better approximated. Bagging can reduce bias by what is, in effect, smoothing (Bühlmann and Yu, 2002). The key is that $f(X)$ is a smooth function to begin with.

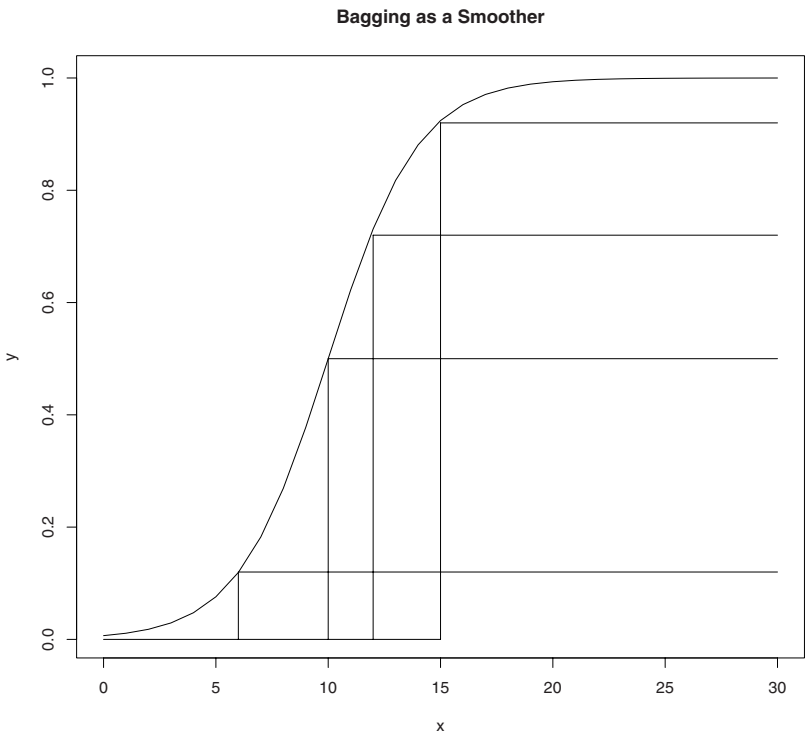

Fig. 4.4. How bagging smooths.

4.5 Some Limitations of Bagging

Bagging has been used recently in a number of interesting ways beyond classification and regression trees (Hothorn and Lausen, 2003). The principles that bagging exploits are quite general. But there are also important limitations.

4.5.1 Sometimes Bagging Does Not Help

Bagging only returns different fitted values from those that could be obtained from one pass over the original data if the fitting procedure is a nonlinear or an adaptive function of the data. For example, all of the smoothers considered earlier were, with the predictors treated as fixed, linear in the data. Recall that the fitted values were just a linear combination of the original values of the response variable. There are no gains from bagging such estimators. The fitted values from bagging would be effectively the same as the fitted values from the original data with no sampling (and identical if the number of bootstrap samples increases without limit).

4.5.2 Sometimes Bagging Can Make the Bias Worse

Look again at Figure 4.4. Suppose $f(X)$ is really very jagged, much like a step function. Then, the smoothing that bagging accomplishes can increase bias because the smoothing on the average moves the fitted values away from the correct $f(X)$. One does not want the sharp corners of the CART estimates sanded off. Classification can also be adversely affected.

Weak classifiers can also create problems, especially when the distribution of the response is highly unbalanced. Weak classifiers are sometimes defined as those that do no better than the marginal distribution. Suppose the marginal distribution of the response is unbalanced so that it is very difficult for a model using the predictors to perform better than the marginal distribution. Under those circumstances the rare class will likely be misclassified most of the time because votes will be typically be won by the class that is far more common.

To illustrate this point, suppose there is a binary response variable, and for the moment, we are interested in a single observation that happens to be a "success." For a given set of trees, that observation is classified as a success about two times out of ten. So, the classification for that observation will be wrong about 80% of the time. But if one classifies by majority vote, the class assigned would be a failure and that would be wrong 100% of the time. Because the classifier does a poor job, the majority vote produces a disappointing result. And if the other observations in the training data tend to be affected by the same difficulty, bagging will perform less well than CART. Bias is increased.

In practice, such problems will be rare if the data analyst pays attention to how the classifier performs before bagging is applied. A key question is how the estimated functions being bagged correspond to the function being estimated. If serious mismatches are avoided, one important source of bias can be reduced. In addition, one should always proceed with great caution if one has very weak classifiers. We show in later chapters that if one has weak classifiers, alternative procedures may be called for.

4.5.3 Sometimes Bagging Can Make the Variance Worse

Bagging sometimes can also perform poorly with respect to the variance (Grandvalet, 2004). Figure 4.5 shows a scatterplot with a binary outcome. The observations are represented by shaded rectangles. Two are far darker than the rest. Both are outliers in x. Consider now their role for the fitted values.

Suppose that the response is a linear function of the x. The fitted values, therefore, should also be a linear function of x. Working with a linear function makes the exposition much easier, and the general lessons from the discussion that follows apply when the response and the fitted values are a nonlinear

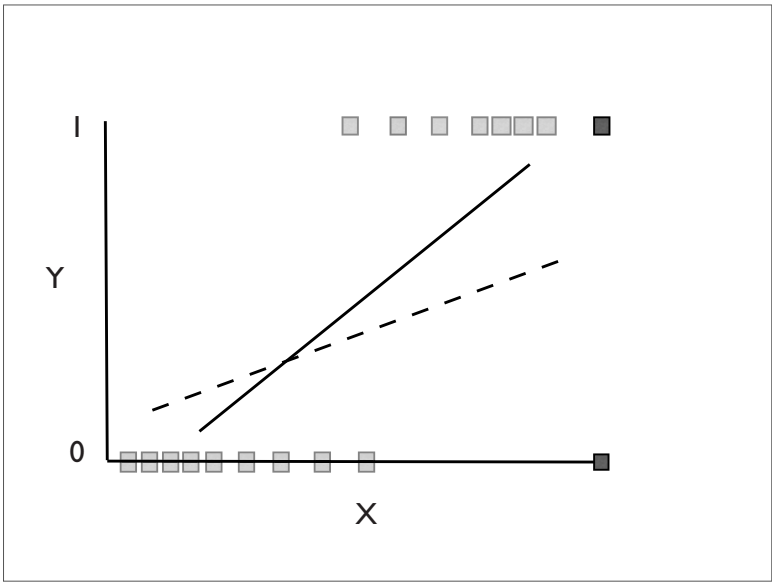

Fig. 4.5. The role of influence in bagging.

function of x. The lessons carry over as well to fitting exercises when there is more than one predictor.

The solid line shows the fitted values with the lower-right outlier excluded from the data. The dotted line shows the fitted values with the lower-right outlier included. The lines are rather different, implying that whether that value is included in the analysis alters the response function substantially. Therefore, the outlier is influential.

In contrast, whether the outlier in the upper-right part of the scatterplot is included makes little difference in the fitted values. It happens to fall very near the line generated by the other fitted values. Deleting it does not change the fit a great deal. Therefore, it is not influential.

However, because the upper-left outlier increases substantially the variance of x without increasing the variance of the residuals very much, it helps to anchor the fitted values. Within the thought experiment of independent random samples of training data from a well-defined population, the fitted values will be more stable if values such as the upper-left outlier are present. Recall that to have substantial influence, an observation needs to be away from the mass of the data in the space defined by the predictors and also needs to have a large disparity between its fitted value of the response and its actual value. For an accessible discussion in the case of linear regression see Cook and Weisberg (1999: 360) and Peña (2005).

Now think about a set of bootstrap samples of the data. If there is an observation like the lower-right outlier, the fitted values will vary a great deal

depending on whether that influential observation happens to be in the sample. But then, averaging over bootstrap samples can help to stabilize the fitted values. This means that in the canonical thought experiment, the variance over random samples from the same population will be reduced. Bagging is doing just what it is supposed to do; the bagged results are an improvement.

The outlier observation in the upper right is not influential but helps to stabilize the fitted values. As a result, bootstrap sampling tends to destabilize the fitted values. When that outlier is by chance not included in the bootstrap sample, the fitted values derived from the other observations will tend to vary more over bootstrap samples. An observation that helps to anchor the fit is absent.

In practice, instances in which bagging can increase the variance sometimes can be spotted. A good place to start is with the univariate statistics for all predictors and the usual search for outliers and highly skewed or unbalanced distributions. Insofar as outliers can be excluded from the analysis on subject matter grounds (e.g., an observation is so atypical that it probably represents some kind data of error), the risks to bagging can be reduced. In the same spirit, highly skewed distributions might be transformed toward more symmetric distributions.

For highly unbalanced, categorical predictor variables with three or more classes, it can help to collapse classes. In the binary case, sometimes it is possible to combine two or more predictors in a way that still makes subject matter sense and restores some balance. For example, people with a PhD could be combined with people holding other advanced degrees to define a new variable equal to 1 if there is any post college education, and 0 otherwise. And there is always the option of dropping highly unbalanced predictors from the analysis.

However, problems with univariate distributions may not prove to have serious consequences. The predictor in question may not figure importantly in the fit because its relationship with the response is weak. Indeed, it may be excluded from the model altogether. In addition, the region in the predictor space where the instability is most manifest may be of little subject matter interest. For example, there may be little interest in the fitted values near the tails of the predictor distribution. Finally, where the mass of the data are, the impact of the instability may be modest. Thus in Figure 4.5, the two lines are much the same toward the middle of the distribution of x. In short, some trial and error can be useful before a final decision is made to exclude outliers.

There are extensions of conventional influence statistics that can be applied to bagging before the bagging begins (Grandvalet, 2004: 267–268). Although they have yet to be battled tested, they may be able to help in finding observations that are likely to be influential. But, the problem for bagging is somewhat different. One needs to find observations that ought to be influential because they are outliers in the space defined by the predictors, but that are actually not influential because they fall on the path of the fitted values constructed from the other observations. In Grandvalet's words, such

values provide "good" influence. Unfortunately, good influence leads to "bad" bagging.

The problems with bagging just described have their analogues for quantitative responses. Bagging is at its best when the problem to overcome is instability. Bagging when the fitted values are already very stable (or when the fitted values contain large amounts of bias) can make things worse. It is important to examine the data carefully before bagging is applied.

4.5.4 Losing the Trees for the Forest

Even when bagging performs as advertised, the price for averaging over trees can be high. There is no longer a single tree structure to interpret and, therefore, no tree diagram. Consequently, there is no direct way to consider how the inputs are related to the output. This is a very serious problem to which we return in the next chapter.

With no tree to interpret, the basic output from bagging is the predicted class for each case. Commonly there is an estimate of the classification error and a cross-tabulation of the classes predicted by the classes observed. This is nothing more than a confusion table but now based on averaging over trees. In addition, there can be separate error calculations for the different response classes, and a comparison of the number of false negatives to the number of false positives. If out-of-bag data are used, the confusion table is an even more honest representation of the results. Sometimes the software will store each of the trees as well, although these are rarely of any interest because the amount of information is typically overwhelming.

4.5.5 Bagging Is Only an Algorithm

Bagging may be seen less as an extension of CART and more as an illustration of what Breiman (2001b) calls "algorithmic modeling." Algorithmic models are computer algorithms designed to solve very particular data analysis problems. Linking inputs to outputs so that classification errors are small is a key example. Although there may also be an interest in describing how the inputs are linked to outputs, there is no effort to represent in the algorithm the mechanisms by which the linkage occurs. Thus, algorithmic models are not causal models. For researchers who want causal models, bagging is not the procedure.

4.6 An Example

Table 4.1 shows the bagged confusion table for the domestic violence data. Before bagging was applied, some CART results were examined to determine if in general CART might be appropriate for these data. Taking the empirical

distribution as the prior and using the default of equal costs for false negatives and false positives, CART seemed to help.

According to the bagged results in Table 4.1, there are 516 observations overall with .29 of them misclassified. About .15 of the households are incorrectly classified as having chronic domestic violence problems, and about .82 of the households are incorrectly classified as not having chronic domestic violence problems. The proportion of incorrect no DV classifications is .22, and the proportion of incorrect DV classifications is .75. Table 4.1 uses the out-of-bag (OOB) data to construct the fitted values, so the confusion table is more honest than the CART confusion tables in which the test data are the same as the training data. A confusion table was constructed from CART output using the same data, but with no bagging applied. The proportion of misclassified cases overall was .24, down from .29. CART was a bit too optimistic.

	Predict No DV	Predict DV	Model Error
No DV	347	60	.15
DV	89	20	.82
Use Error	.22	.75	Overall Error = .29

Table 4.1. Bagged CART confusion table for estimates of domestic violence.

4.7 Bagging a Quantitative Response Variable

Bagging works by the same general principles when the response variable is quantitative. Recall that CART constructs a regression tree by maximizing the reduction in the error sum of squares at each split. Each case is placed in a terminal node with a conditional mean. That mean is the predicted value for all cases of that terminal node.

All of the concerns about overfitting apply, especially given the potential impact that outliers can have on the fitting process when the response variable is quantitative. Recall that with the sum of squares fitting function, a few cases that fall a substantial distance from the mass of the data can produce results that do not characterize well the data on hand and do not generalize well either.

At the same time, overfitting is not always a problem. The consequences of overfitting can be unimportant if

1. The number of observations is large.
2. The number of predictors is small.
3. The number of terminal nodes is small.

4. There are no observations that fall some distance away from the mass of the data for the joint distribution of response variable and the predictors.

With a numerical response variable, bagging averages over trees in much the same way it averages over trees when the response variable is categorical. For each tree, each observation is placed in a terminal node and assigned the mean of that terminal node. Then, the average of these assigned means over trees is computed for each observation. This average value for each case is the bagged fitted value used. It is an average of conditional means for a large number of regression trees. The averaging process will tend to cancel out the impact of trees producing extreme conditional means and in so doing, helps to reduce the impact of overfitting. If for each tree, it is the OOB data that are placed in terminal nodes, the overfitting problems can be reduced even more.

As just noted however, overfitting is not necessarily a problem for CART analyses. To illustrate, the CART regression analysis undertaken earlier for high school grade point average was done again with bagged regression trees. Recall that there were approximately 8000 observations. This time eight predictors were used.

There was no evidence of outliers in the joint distribution of the predictors and the response. A series of bivariate scatterplots was first examined, and no apparent outliers were spotted. However, a series of bivariate plots is not the same as a single multivariate plot. So, the model implied by the CART results was re-estimated using linear regression with appropriate interaction terms. Then Cook's distance was computed for each observation. As expected, with so large a sample and relatively few predictors, no single observation stood out as problematic. Taken together, these two approaches are not iron clad proof that all is well, but make it a reasonable working premise.

It was not surprising, therefore, that bagging did not make an important difference. The root mean squared error (i.e., the standard deviation of the residuals) was .4136 for the CART results and .4132 for the bagged CART results. The grade point average response variable ranged from 1.0 to 5.0, therefore a difference of the root mean square error in the fourth decimal place is effectively noise.

4.8 Software Considerations

In R, the bagging procedure (i.e., bagging() in the *ipred* library) can be applied to classification, regression, and survival trees. The arguments from these procedures can be passed to ipred(). For example, one set the prior in rpart() to take misclassification costs into account, and this information is used in ipred(). The library was written by Andrea Peters (Andrea.Peters@imbe.imed.uni-erlangen.de) and Torsten Hothorn (Torsten.Hothorn@rzmail.uni-erlangen.de). The package maintainer is Andrea Peters. Perhaps the key concern is that when confusion tables are constructed, or when other measures of

performance are computed, one must be clear on what is being done. There are at least three possibilities.

1. Trees are bagged as usual, and bagged classifications or bagged conditional means constructed. These are then compared to the actual classifications or response variable values in the original data from which bootstrap samples were drawn.
2. Trees are bagged as usual, and bagged classifications or bagged conditional means constructed. The software stores the predictor values leading to each terminal node. New data (from a test sample) are dropped down the bagged tree and assigned to terminal nodes based on the values of their predictors. Each of the new observations is assigned a class or conditional mean determined by the class or conditional mean of the terminal node in which it lands. These fitted values for the new data are compared to the actual values of the response in the new data.
3. CART is used to grow a single tree using a bootstrap sample of the data. As usual, classifications or conditional means are constructed for each terminal node. The set of predictor values leading to each terminal node is stored. Observations not included in the bootstrap sample are noted. These are the out-of-bag observations for that tree. The out-of-bag observations are then dropped down the tree and assigned the class or conditional mean of the terminal node in which they land. The same process is repeated for a number of trees. When the votes are cast to determine class membership or when conditional means over trees are averaged, the only trees considered for a given observation are the ones for which that observation was not in the training data. That is, for a given case i, the only trees that count are the trees for which case i was not used (i.e., it was among the out-of-bag observations). Generally about one-third of the original data are not chosen to be included in each training sample. Even with a relatively small number of trees, therefore, each of the observations will have several votes or conditional means to average.

The third method was used for analysis of grade point average, just reported. The second and third methods are generally more honest than the first because the separation between training data and the test data is more complete. But all three methods are often better than no bagging at all. The drawback to the second method is that a second random sample from the same population is needed. The drawback to the third method is that it will often be useful to construct a larger number of trees because for each tree, only about a third of the data figure in the voting. For example, although 25 trees may be good enough for methods one and two, 100 trees may be a good number for method three. But usually, 100 trees will not be a prohibitive computational burden.

A rather different set of issues can sometimes be raised by the bootstrap sampling process. In particular, a given sample may produce predictors or response variables that are constants. This is more likely when predictor or

response observations are categorical and unbalanced. For example, if in a sample of 300 inmates only 30 are Asian-Americans, a bootstrap sample may include no Asian-Americans whatsoever.

The problem for the software is what to do in such situations. One approach would be to discard samples in which any of the variables were constants. Another approach would be to throw out the offending variables. However, discarding variables is not an option if the response variable is one of the set. In any case, the worst outcome is for the software to crash. It can be well worth the time to read the software documentation especially carefully if there are highly unbalanced variables in the dataset.

4.9 Summary and Conclusions

Bagging is an important conceptual advance and a useful tool in practice. The conceptual advance is to aggregate fitted values from a large number of bootstrap samples. Ideally, many sets of fitted values, each with low bias but high variance, may be averaged in a manner than can effectively reduce the bite of the bias–variance tradeoff. Thanks to bagging, there can be a way to usefully address this long-standing dilemma in statistics. Moreover, the ways in which bagging aggregates the fitted values is the basis for other statistical learning developments.

In practice, bagging can generate fitted values that often reproduce the data well and forecast with considerable skill. Both masters are served without making unrealistic demands on available computing power. Bagging can also be usefully applied to a wide variety of fitting procedures.

But bagging also suffers from several problems. Perhaps most important, there is no way within the procedure itself to depict how the predictors are related to the response. One can obtain a more honest set of fitted values and a more honest evaluation of how good the fitted values really are. But as a descriptive device, bagging is pretty much a bust. Other tools are needed, which are considered in the next chapter.

A second problem is that because the same predictors are available from tree to tree, the sets of fitted values are not fully independent. The averaging is not as effective as it could be if the sets of fitted values were closer to independent. This too is addressed shortly.

Third, bagging can stumble badly if the fitting function is consistently and substantially inappropriate. Large and systematic errors in the fitted values are just reproduced a large number of times and do not, therefore, cancel out in the averaging process. For categorical response variables, bagging a very weak classifier can sometimes make things worse.

Fourth, the bootstrap sampling can lead to problems when categorical predictors or outcomes are highly unbalanced. For any given bootstrap sample, the unbalanced variable can become a constant. Depending on the fitting function being bagged, the entire procedure may abort.

Finally, bagging can actually increase instability if there are outliers that help to anchor the fit. Such outliers will be lost to some of the bootstrap samples. Bagging can be extended so that many of these problems are usefully addressed, even if full solutions are not available. We turn to some of these solutions in the next chapter. And in their potential solutions is found another form of statistical learning, still farther away from conventional regression analysis.

Exercises

Problem Set 1

The goal of this first exercise is to compare the performance of linear regression, CART, and bagging applied to CART. Construct the following data set in which the response is a quadratic function of a single predictor.

```
x1=rnorm(500)
x12=x1^2
y=1+(2*(x12))+(2*rnorm(500))
```

1. Plot the $1 + (2 \times \text{x12})$ against x1. This is the "true" relationship between the response and the predictor without the complication of the disturbances. This is the $f(X)$ you hope to recover from the data.

2. Proceed as if you know that the $f(X)$ is quadratic. Fit a linear model with x12 as the predictor. Then plot the fitted values against x1. You can see how well linear regression does when the functional form is known.

3. Now suppose that you do not know that the $f(X)$ is quadratic. Apply linear regression to the same response variable using x1(not x12) as the sole predictor. Construct the predicted values and plot the fitted values against x1. How do the fitted values compare to what you know to be the correct $f(X)$? (It is common to assume the functional form is linear when the functional form is unknown.)

4. Apply CART to the same response variable using rpart() and x1(not x12) as the sole predictor. Use the default settings. Construct the predicted values, using predict(). Then plot the fitted values against x1. How do the CART fitted values compare to what you know to be the correct $f(X)$? How do the CART fitted values compare to the fitted values from the linear regression with x1 as the sole predictor?

5. Apply bagging to the same response variable using ipred() and x1 as the sole predictor. Use the default settings. Construct the predicted values using predict(). Then plot the fitted values against x1. How do the bagged fitted values compare to the linear regression fitted values?

6. You know that the relationship between the response and x1 should be a smooth parabola. How do the fitted values from CART compare to the fitted values from bagging? What feature of bagging is highlighted?

Problem Set 2

Load the dataset "Freedman" from the *car* library. For 100 American cities, there are four variables: the crime rate, the population, population density, and proportion nonwhite. As before, the crime rate is the response and the other variables are predictors.

1. Use rpart() and its default values to fit a CART model. Compute the root mean square error for the model. One way to do this is to use predict.rpart() to obtain the fitted values and with the observed values for the variable "crime," compute the root mean square error in R. Then use bagging() from the library *ipred* and the out-of-bag observations to obtain a bagged value for the root mean square error for the same CART model. Compare the two estimates of fit and explain why they differ.

2. Using sd(), compute the standard deviation for the CART fitted values and the bagged fitted values. Compare the two standard deviations and explain why they differ.

Problem Set 3

Load the dataset "frogs" from the library *DAAG* Using "pres.abs" as the response build a CART model under the default settings.

1. Construct a confusion table with "pres.abs" and the predicted classes from the model. Now, using bagging() from the library *ipred*, bag the CART model using the out-of-bag observations. Construct a confusion table with "pres.abs" and the bagged predicted classes from the model. Compare the two confusion tables and explain why they differ.

2. Cross-tabulate using table() or xtab() the fitted classes from CART and the bagged CART. Examine the two cells for cases in which the two sets of fitted classes do not agree. Why is the number of observations in each about the same?