5

Random Forests

5.1 Introduction and Overview

Just as in bagging, imagine constructing a large number of trees with bootstrap samples from a dataset. But now, as each tree is constructed, take a random sample of predictors before each node is split. For example, if there are twenty predictors, choose a random five as candidates for defining the split. Then construct the best split, as usual, but selecting only from the five chosen. Repeat this process for each node. And as in bagging, do not prune. Thus, each tree is produced from a random sample of cases, and at each split a random sample of predictors. Finally, just as in bagging, classify by a majority vote of the full set of trees. Breiman calls the set of such trees a "random forest" (Breiman, 2001a).

The random forest algorithm is, therefore, very much like the bagging algorithm. Again let N be the number of observations and assume for now that the response variable is binary.

1. Take a random sample of size N with replacement from the data.
2. Take a random sample without replacement of the predictors.
3. Construct the first CART partition of the data.
4. Repeat Step 2 for each subsequent split until the tree is as large as desired. Do not prune.
5. Drop the out-of-bag data down the tree. Store the class assigned to each observation along with each observation's predictor values.
6. Repeat Steps 1–5 a large number of times (e.g., 500).
7. Using only the class assigned to each observation when that observation is not used to build the tree, count the number of times over trees that the observation is classified in one category and the number of times over trees it is classified in the other category.
8. Assign each case to a category by a majority vote over the set of trees. Thus, if 51% of the time over a large number of trees a given case is classified as a "1," that becomes its estimated classification.

R.A. Berk, *Statistical Learning from a Regression Perspective*,
DOI: 10.1007/978-0-387-77501-2_5, © Springer Science+Business Media, LLC 2008

5.1.1 Unpacking How Random Forests Works

It should be clear that random forests draws on many features of procedures discussed in the last two chapters. To begin, random forests uses CART as a key building block. An important benefit is that one can capitalize on CART's strengths and flexibility. For example, large trees can be effective tools for reducing bias, and the averaging over trees can substantially reduce instability that might otherwise result. In addition, the relative costs of false negatives and false positives can be explicitly considered. Especially for policy-related applications, this can be vital.

It should also be apparent that random forests is bagging, but more so. By working with a random sample of predictors at each possible split, the fitted values across trees are more independent. Consequently, the gains from averaging over a large number of trees can be more dramatic. But there is more to the story.

If the individual trees in a random forest are unbiased in their fitted values and estimated splits, the gains from random forest are solely with respect to the variance. In practice, of course, there is no way to know if this is true and there are usually lots of reasons for skepticism. Sometimes, therefore, random forests can be seen as helping to reduce the bias.

Perhaps most directly, random forests is able to work with a very large number of predictors, even more predictors than there are observations. In addition to conventional regression modeling, all of the statistical learning procedures considered thus far have required that the number of predictors be less than the number of observations (usually much less). An obvious gain with random forests is that more information may be brought to bear on the fitting process. More predictors can weigh in, which can reduce bias. Variables that should play a role and that otherwise would have been excluded, can participate.

A more subtle gain is that different sets of predictors can be evaluated for different splits so that different "models" can be applied as needed. To appreciate how this works recall the CART splitting criterion:

$$\Delta I(s, A) = I(A) - p(A_L)I(A_L) - p(A_R)I(A_R), \tag{5.1}$$

where $I(A)$ is the value of the parent impurity, $p(A_R)$ is the probability of a case falling in the right daughter node, $p(A_L)$ is the probability of a case falling in the left daughter node, $I(A_R)$ is the impurity of the right daughter node, and $I(A_L)$ is the impurity of the left daughter node. CART tries to find the predictor and the split for which $\Delta I(s, A)$ is as large as possible.

The key point is that the usefulness of a split is a function of the two new impurities and the probability of cases falling into either of the prospective daughter nodes. Suppose there is a predictor that could produce splits in which one of the daughter nodes is very homogeneous but has relatively few observations whereas the other node is quite heterogeneous but has relatively many observations. Suppose there is another predictor that could generate two nodes of about the same size, each of which is only moderately homogeneous. If these two predictors were forced to compete against each other, the second predictor might well be chosen, and the small local region that the first predictor would address be ignored. However, if the second predictor were not in the pool of competitors, the first might be selected instead.

Similar issues arise with predictors that are substantially correlated. There may be little difference empirically between the two so that when they compete to be a splitting variable, one might be chosen almost as easily as the other. But they would not partition the data in exactly the same way. The two partitions that would be defined would largely overlap. But each partition would have unique content as well. The unique content defined by the predictor not chosen would not be included in that step. Moreover, with the shared area now removed from consideration, the chances that the neglected predictor would be selected later would be significantly reduced. But if the two variables each had an opportunity to be selected without competing against each other, each might be able to contribute.

Both kinds of competitions are in practice likely to involve many variables, especially if there are a large number of predictors. Then, there can be a few predictors that in a procedure such as CART will dominate the fitting process because on the average they consistently perform just a bit better than their competitors. Consequently, many other predictors, which could be useful for very local features of the data, are rarely selected as splitting variables.

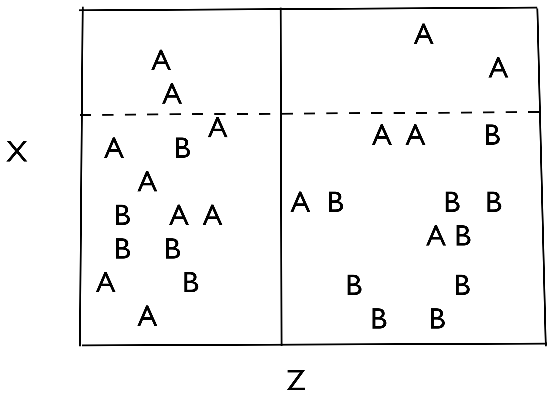
The same issues can arise for a single predictor. Recall that in CART, all predictors are evaluated, even predictors previously selected. The same predictor may be selected more than once. But each new selection for a given predictor implies the construction of a new basis function for that predictor (i.e., a different break point). In the competition between all predictors at each stage, basis functions that might be very important, but only for a small fraction of the data, risk being overlooked.

With random forests computed for a large enough number of trees, each predictor will have at least several opportunities to be the predictor defining a split. And in those opportunities, it will have very few competitors. Moreover, if there are a relatively small number of dominant predictors, much of the time a dominant predictor will not be included. As a result, predictors that might ordinarily be overlooked have the opportunity to contribute to the fit. The same implications follow for different basis functions for given predictors. With a changing mix of competitors, highly specialized basis functions may have the opportunity to define a split.

The sampling of predictors also has beneficial effects for the variance because of the averaging over trees. In effect, the predictor sampling and averaging leads to shrinkage of the impact of each predictor on the fitted values. The impacts on the fitted values when a predictor is included are averaged with the impacts on the fitted values when that predictor is not included. This regularization can help to reduce the variance beyond what would follow solely from the sampling of observations.

To help fix these ideas, Figure 5.1 shows a partitioning diagram much like the one used earlier when CART was first introduced. As before, there are two predictors, x and z. There is a choice to be made between the vertical split on z represented by the solid line and the horizontal split on x represented by the dashed line. The horizontal split might not be selected because its one very homogeneous partition has relatively few observations. But if the predictor z were not available, the horizontal split shown might well be chosen.

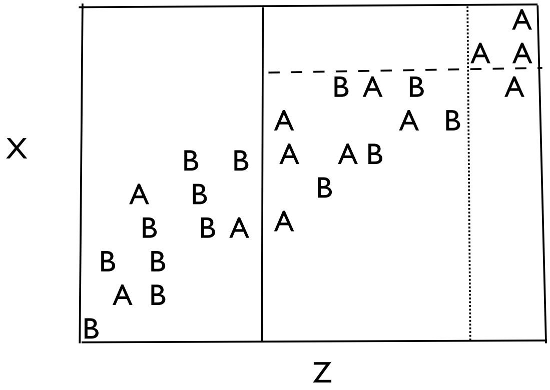
Very much the same issues arise if the vertical split is made, but then the choice is between a horizontal split and another vertical split, which would imply a second basis function for z. Consider the right vertical partition, for example. The potential horizontal split shown would have to compete against all possible vertical splits of z. But if z were not among the competitors for the second split, the horizontal split might well be chosen.



Recursive Partitioning of a Binary Outcome With Rare Cases
(where G = A or B and predictors are Z and X)

Fig. 5.1. Partitioning for some rare cases.

Consider now Figure 5.2, a reproduction of Figure 3.8. This figure was used earlier to illustrate CART instabilities when two predictors were highly correlated. Here the point is that whether the partition defined by the dashed line or the partition defined by the dotted line were selected, it would make no difference for the three As in the upper-right hand corner. But whether the fourth A, just below the dashed line, would be included depends on which partition won in the fitting competition. If the horizontal partition were to win, the fourth A would be placed in a very heterogeneous grouping. If the vertical partition were to win, the fourth A would be placed in a very homogeneous grouping. And if the former, the fact that the fourth A was very similar to its three A neighbors would probably be lost to the analysis. With the three As already in their own partition, it is unlikely that another partition would be defined later which was also very high on x and z. If x were not available when the second split were determined, the vertical slice shown might well be the one selected.



Recursive Partitioning of a Binary Outcome with High Colinearity
(where G = A or B and predictors are Z and X)

Fig. 5.2. Competition between highly correlated predictors

The goal of finding a role for highly specialized predictors is an argument for growing very large, unpruned trees. Generally, this seems to be a wise strategy. However, large trees can sometimes lead to very unstable results when there are a substantial number of predictors that at best are weakly related to the response and correlated substantially with one another (Segal, 2003). In effect, this becomes a problem with multicolinearity that the averaging over trees becomes more difficult. The instability is too large to be readily averaged out. In practice, therefore, it can occasionally be useful to work with smaller trees, especially when there are a large number of weak predictors that are strongly associated with one another. If before a data analysis is begun there are reasons to worry about such problems, tree size can be used as a tuning parameter in yet another manifestation of the bias–variance tradeoff.

Geurts and his colleagues (2006) have proposed another method for selecting predictors for enhancing independence across trees and further opening up the predictor competition. They do not build each tree from a bootstrap sample of the data. Rather, for each random sample of predictors, they select splits for each predictor at random (with equal probability), subject to some minimum number of observations in the smaller of the two partitions. Then, as in random forests, the predictor that reduces heterogeneity the most is chosen to define the two subsets of observations. They claim that this approach will reduce the overall heterogeneity at least as much as other ensemble procedures without a substantial increase in bias. However, this conclusion would seem to depend on how good the predictors really are. Moreover, if one is interested in interpreting the manner in which inputs are related to outputs, their method risks serious subject matter errors. In the averaging process over trees, model results characterized by optimal splits are weighted the same as model results characterized by random spits.

In summary, with forecasting accuracy as a criterion, bagging is in principle an improvement over CART. And by this same criterion, random forests is in principle an improvement over bagging. Indeed, random forests is among the very best classifiers invented to date (Breiman, 2001a). A key reason is the ability to consider a very large number of predictors, even more predictors than observations. This can lead to reductions in the bias and reductions in the variance.

5.2 An Initial Illustration

Table 5.1 shows some results for the domestic violence data described earlier. As before, there are a little over 500 observations, and even if just double interactions are considered, well over 100 predictors. This time, the goal is not to forecast new calls for service to the police department that likely involve domestic violence, but only those calls in which there is evidence that felony domestic violence has actually occurred. Such incidents represent about 4% of the cases. They are very small as a fraction of all domestic violence calls for service. And as such, they would normally be extremely difficult to forecast with better skill than using the marginal distribution of the response alone. One would make only four mistakes in 100 households if one classified all households as not having new incidents of serious domestic violence.

Using the response variable as the only source of information would in this case mean never correctly identifying serious domestic violence households. The policy recommendation might be for the police to assume that the domestic violence incident to which they had been called would be the last serious one for that household. This would almost certainly be an unsatisfactory result, which implies that there are significant costs from false negatives.

Using a cost ratio of 10 to 1 for false negatives to false positives favored by the police department (more on how to do that shortly), Table 5.1 shows that random forests incorrectly classifies households 13 times out of 100 overall. If equal costs were used and the empirical distribution of the response variable taken as the prior distribution, random forests would likely do at least as well as the marginal distribution (i.e. four mistakes per 100 households).

But whenever costs other than equal ones are introduced, the overall error proportion will increase, so this result is no surprise. It is also not a problem.

More instructive measures of performance are found in the row proportions. Random forests manages to correctly identify the very rare serious domestic violence households about half the time with a model error of only .30 for households without these problems. As one would expect, when a logistic regression was applied to the data, not a single incident of serious domestic violence was identified, either correctly or incorrectly. The logistic regression performed no better than the marginal distribution of the response.

	No DV Forecasted	DV Forecasted	Model Error
No DV	341	146	.30
DV	15	14	.51
Use Error	.04	.91	Overall Error = .13

Table 5.1. Confusion table for the ten-to-one random forest model for new domestic violence incidents.

The use errors (i.e., column proportions) also look promising. When no future incidents of domestic violence are forecasted, that forecast is correct about 96 times out of 100. When future incidents of domestic violence are forecasted, that forecast is correct about 1 time in 10. Although that might seem to be disappointing, it is a reflection of the costs assigned. The results imply that the police department is prepared to live with nine false positives for every true positive. If that tradeoff is indeed acceptable, then the forecasting exercise works as intended.

5.3 A Few Formalities

With some initial material on random forests behind us, it is useful to take a bit more formal look at the procedure. We build on an exposition by Breiman (2001a). The concepts considered make more rigorous some ideas that we have used in the past two chapters, and provide important groundwork for material to come. As before, we focus on categorical, and especially binary, response variables.

We also need to change notation just a bit. Bold type is used for vectors and matrices. Capital letters are used for random variables.

5.3.1 What Is a Random Forest?

With categorical response variables, a random forest is a classifier. More than two classes can be used. The intent is to assign classes to observations using information contained in a set of predictors. A random forest is constructed from a set of K classification trees, each based in part on chance mechanisms.

We formally represent the random forest classifier as a collection of tree-structured classifiers $\{f(\mathbf{x}, \Theta_k), k = 1, \dots\}$, where $\mathbf{x}$ is an input vector of P predictor values used to assign a class, and k is an index for a given tree. Each Θ_k is a random vector constructed for the kth tree so that it is independent of past random vectors $\Theta_1, \dots, \Theta_{k-1}$, and is generated from the same distribution. For bagging, it is the means by which observations are selected at random with replacement from the training data. For random forests, it is also the means by which subsets of predictors are sampled without replacement for each potential split. In both cases, Θ_k is a collection of integers. Integers for both sampling procedures can be represented by Θ_k.

But we are getting ahead of ourselves. The output from a given classifier is an assigned class for each observation, determined by the input values $\mathbf{x}$. In CART, for example, the class assigned to an observation is the class associated with the terminal node in which an observation falls. With random forests, the class assigned to each observation is determined by a vote over the set of tree classifiers for which that observation was not part of the training dataset. That is, classes are assigned to observations much as they are in bagging. It is conceptually very important to distinguish between the class assigned by the kth classifier and the class ultimately assigned by a vote.

5.3.2 Margins and Generalization Error for Classifiers in General

Suppose there is a training dataset with predictor values and associated values for a categorical response. A training dataset has been drawn at random. This means that the data on hand can be treated as a realization of two kinds of random variables: a set of predictors and a response variable.

Consider now a single data point from the training data. If the training data have, for instance, 250 observations and 20 variables, that data point might be the 27th row in a 250 by 20 data matrix. The values in this row will change from sample to sample. Consequently, the set of P predictor values can be represented by the random variable $\mathbf{X}$. The actual class for that data point will be represented by the random variable Y.

Suppose there is an ensemble of K classifiers, $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_K(\mathbf{x})$. For the moment, we do not consider how these different classifiers are constructed. The margin function at the data point $\mathbf{X}, Y$ is then defined as

$$mg(\mathbf{X}, Y) = av_k I(f_k(\mathbf{X}) = Y) - \max_{j \neq Y} av_k I(f_k(\mathbf{X}) = j), \tag{5.2}$$

where $I(.)$ is an indicator function, j is an incorrect class, av_k denotes averaging over the set of classifiers for a single realized data point, and max denotes the largest value.

For a given set of x-values and the associated observed class, the margin is the average number of votes for the correct observed class minus the maximum average number of votes for any other incorrect class. Thus, the margin is the smallest spread between the number of correct and incorrect votes for a given data point. When there are only two classes, there is only one spread to worry about, and the word "maximum" is unnecessary.

From the definition of the margin function, generalization error is then

$$g = P_{\mathbf{X},Y}(mg(\mathbf{X}, Y) < 0), \tag{5.3}$$

where P means probability (easily confused with p, which can be the number of predictors.

The probability is over the space represented by the random variables $\mathbf{X}, Y$, so that generalization error addresses what happens to the margin over different realizations of the data point. Generalization error, therefore, is the probability over realizations of the data point that the classification vote will be overturned (i.e., go from positive to negative), and the assigned class changed. A low probability indicates the class assigned is likely to be stable over random samples from the same population.

This definition can be somewhat confusing. Generalization error, sometimes called "prediction error"or "test error," is more typically defined as the expected loss over all sources of randomness built into the full set of fitted values. Then, generalization error can be written as

$$E[L(Y, f(\mathbf{X}))]. \tag{5.4}$$

The loss function $L(Y, f(\mathbf{X}))$, is derived from disparities between the values of the response predicted and the values of the response observed. Sometimes a "hat" is placed over the "f" to indicate that it is an estimated function.

Here, we are using a "1–0 loss" because the assigned class is compared to the observed class via the indicator function. It would also be possible to compare the observed class to the predicted probability of membership in that class. Then, a popular loss function is the deviance. But using the deviance is more appropriate when constructing a given classifier, and if used instead of the 1–0 loss here would fundamentally change random forests.

5.3.3 Generalization Error for Random Forests

Now, suppose the classifier $f_k(\mathbf{X}) = f(\mathbf{X}, \Theta_k)$; the classifier is a random forest. Breiman proves (2001a) that as the number of trees increases, the estimated generalization error converges to the population generalization error, which is

$$P_{\mathbf{X},y}(P_\Theta(f(\mathbf{X}, \Theta) = Y) - \max_{j \neq Y} P_\Theta(f(\mathbf{X}, \theta) = j) < 0). \tag{5.5}$$

The arguments for $P_{\mathbf{X},Y}$ are (1) the probability of the correct classification over trees, and (2) the maximum probability over trees of a wrong classification. The number of these trees increases without limit. Then the estimated generalization error converges to the probability over $\mathbf{X}, Y$ that a vote will be overturned.

There is a lot going on here. The data used for a given tree are a bootstrap sample of the training dataset. The training dataset is a realization of the random variables. Thus, two different chance mechanisms are involved, the first reflected in P_{Θ} and the second reflected in $P_{\mathbf{X},Y}$.

The importance of the convergence is that demonstrably random forests does not overfit as more trees are grown. One might think that with more trees one would get an increasingly false sense of how well the results would generalize. Breiman proves that this is not true. Given all of the concern about the problem of overfitting, this is an important result.

It is also important to appreciate the limitations of what is being claimed. The convergence is to a value for the generalization error in the population. (This implies that the data are a random sample from that population or a random realization from a specified stochastic process.) This says nothing about the quality of the population classifier responsible for that generalization error. That classifier could well generate fitted values some distance from the actual response variable's observations, and these fitted values could contain substantial systematic error. As a result, the target generalization error could be some distance from the "true" generalization error were the classifier in the population the "correct" classifier (Traskin, 2008). In short, Breiman does not prove that the classifier estimated from the data is a consistent estimator of the $f(X)$, even if one has access to all of the necessary predictors perfectly measured. We return to this point in a slightly different context shortly.

As noted earlier, however, if there is no postulated $f(X)$, or if the data analyst is prepared to accept that with the data available the $f(X)$ will not be well estimated, then concerns about whether the population classifier is "correct" are moot. Breiman's proof remains relevant nevertheless because one might otherwise believe that growing lots of trees in order to assemble a random forest would lead to overfitting.

5.3.4 The Strength of a Random Forest

The margin function for a given realized data point in a random forest (not just any classifier) is defined as

$$mr(\mathbf{X}, Y) = P_{\Theta}(f(\mathbf{X}, \Theta) = Y) - \max_{j \neq Y} P_{\Theta}(f(\mathbf{X}, \Theta) = j), \qquad (5.6)$$

where $f(\mathbf{X}, \Theta)$ is a set of classifications that varies because of the chance mechanism represented by Θ. Thus, considering all possible trees and that realized data point, the margin function of a random forest is the probability that a classification will be correct minus the maximum probability that it will be incorrect.

The margin function for a given random forest takes the training data as fixed. If one allows for different realizations of the data point, the margin function for a random forest will vary from realization to realization. But if we take the expected value of Equation 5.6, over realizations of the data point, the strength of a random forest is defined as

$$s = E_{\mathbf{X},\mathbf{y}} mr(\mathbf{X}, \mathbf{y}). \tag{5.7}$$

Thus, the strength of a random forest is essentially the average margin over randomly drawn training data. Clearly, the larger this expected value is, the better.

5.3.5 Dependence

For a given data point, the trees from which a random forests is constructed differ from each other because of the chance mechanism represented in Θ_k. Recall that for both the selection of observations in the bootstrap sample and the selection of potential predictors at each split, the chance mechanism generates a set of integers.

For the bootstrap sample, the chance mechanism generates the number of times each observation in the training data is selected. For the predictors, the chance mechanism generates integers denoting which predictors are chosen. For example, if there are 10 predictors $1, \ldots, 10$, and a decision is made to select three of them, the integers 1, 3, and 8 might be drawn at random.

Ideally, both chance mechanisms should perform so that the output from each tree is as independent as possible. That allows the averaging to occur most effectively. But what output in particular?

For the binary outcome case, there is a relatively straightforward result. Draw an observation from the relevant population. For any one tree, classify that realized data point. If the classification is correct, record a 1. If the classification is incorrect, record a 0. Repeat the process beginning with random sampling of a single observation. The average correlation between these coded values, which represent whether the classification is correct, over samples and trees, is the appropriate measure of dependence. Ideally, this correlation should be small. The probability that a given observation chosen at random is classified correctly should be unrelated to whether any other randomly chosen observation is classified correctly.

5.3.6 Implications

The central concept in this more technical discussion is the margin. From the margin comes the definition of generalization error. Generalization error, in turn, depends on the strength of the collection of classifiers and the dependence between them. When the dependence between trees is small and the strength of the trees is large, the generalization error will be small. More precisely, Breiman shows that the upper bound for the generalization error is

$$g^* = \frac{\bar{\rho}(1 - s^2)}{s^2}, \tag{5.8}$$

where $\bar{\rho}$ is the average correlation over realizations of the data and trees, and s is the strength of the set of trees. We want the former to be small and the latter to be large.

In addition, generalization error will not increase with the number of trees. Indeed, more trees are generally better than fewer trees because the true value of the generalization error will be more accurately approximated. In practice, this means that unless one is limited by the computer being used, a random forests can usefully include several thousand trees.

Another practical lesson is that a useful random forest classifier will on the average produce large margins as observations are classified. Put another way, one should have less confidence in random forest results when the margins tend to be small. This point and related ones have been made informally in earlier discussions.

A final lesson is that the random selection of predictors helps to make random forests more desirable than bagging because dependence is reduced. Other advantages of random forests are considered shortly.

5.4 Random Forests and Adaptive Nearest Neighbor Methods

A conceptual link was made earlier between CART and adaptive nearest neighbor methods. Not surprisingly, similar links can be made between random forests and adaptive nearest neighbor methods. But for random forests, there are a number of more subtle issues (Lin and Jeon, 2006). These are important not just for a deeper understanding of random forests, but for some extensions of random forests considered at the end of this chapter.

Recall that in CART, each terminal node represented a region of nearest neighbors. The boundaries of the neighborhood were constructed adaptively when the best predictors and their best splits were determined. With the neighborhood defined, all of the observations inside were used to compute a mean or proportion. This value became the measure of central tendency for the response within that neighborhood. In short, each terminal node and the neighborhood represented had it own conditional mean or conditional proportion. Figure 3.1 might be a useful memory refresher.

Consider the case in which equals costs are assumed. This makes for a much easier exposition, and no key points are lost. The calculations that take place within each terminal node, in effect, rely on a weight given to each value of the response variable. (Meinshausen, 2006; Lin and Jeon, 2006). For a given terminal node, all observations not in that node play no role when the mean or proportion is computed. Consequently, each such observation has a weight of zero. For a given terminal node, all of the observations are used when the mean or proportion is computed. Consequently, each value of the response variable in that node has a weight equal to $1/n_\tau$, where n is the number of observations in terminal node τ. Once the mean or proportion for a terminal node is computed, that mean or proportion can serve as a fitted value for all cases that fall in that terminal node.

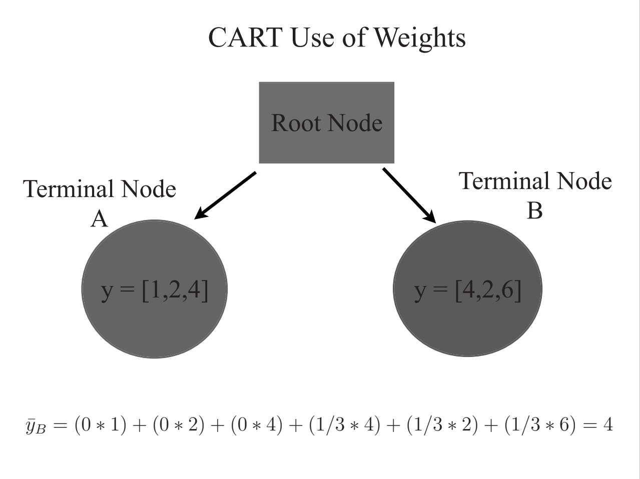

CART Use of Weights
Root Node
Terminal Node A
Terminal Node B
y = [1,2,4]
y = [4,2,6]
$\bar{y}_B = (0 * 1) + (0 * 2) + (0 * 4) + (1/3 * 4) + (1/3 * 2) + (1/3 * 6) = 4$

Fig. 5.3. CART weighting.

Figure 5.3 shows a toy rendering of this account. The tree has but a single partitioning of the data. There are three values of the response variable in each terminal node. Consider terminal node B. The mean for terminal node B is 4, computed with weights of 1/3 for the values in that node and weights of 0 otherwise. Each of the three observations landing in terminal node B can be assigned a value of 4 as their fitted value. If the response variable had been binary, the numbers in the two terminal nodes would have been replaced by 1s and 0s. Then a conditional proportion for each terminal node would be the outcome of the weighted averaging. And from this an assigned class could be determined as usual.

A bit more formally, a conditional mean or proportion for any terminal node τ is

$$\bar{y}_\tau | x = \sum_{i=1}^{N} w_{(i,\tau)} y_i, \tag{5.9}$$

where the sum is taken over the entire training dataset, and w_i is the weight for each y_i. The sum of the weights over all observations is 1.0. In practice, most of the weights will be zero because they are not associated with the terminal node in question. This is really no different from the manner in which nearest neighbor methods can work when summary measures of a response variable are computed.

It may be important to underscore that each observation in the training dataset has a set of weights, one weight for each y_i. So, if there are 150 observations in the training data, there will be 150 weight values. It is these weights and y-values from which the mean or proportion for each terminal node is computed. Then each observation can have one fitted value depending on the node in which it comes to rest.

When a classification or regression tree becomes part of a random forest, there are a large number of trees with which to contend. For a given observation, there will now be a set of weights produced by each tree. Once again, most of the weights will be 0, with the rest fractions. Because of the stochastic nature of each tree, the weights will vary.

Consider a given observation $\mathbf{x}_0$ characterized by a set of predictor values and a value for the response variable. From tree to tree, the observations in the terminal nodes in which $\mathbf{x}_0$ falls will vary. Consequently, the node mean will likely vary as well. It follows that the set of weights used to compute the fitted value for $\mathbf{x}_0$ will change. In effect, random forests averages these weights for $\mathbf{x}_0$ (and for all other observations). Then, the average weight replaces the tree-specific weight in Equation 5.9. Weights greater than zero are sometimes called "voting points" when the mean or proportion for $\mathbf{x}_0$ is computed (Lin and Jeon, 2006: 579–580). The sum of the average weights over all observations is 1.0. Clearly, the links to adaptive nearest neighbors remain. It is just that each weight is an average weight over trees.

To help firm up these ideas, Table 5.2 carries on with the toy example for a single case. There are six observations in the dataset represented in the table as columns. A very small random forest of three trees is grown. To keep the table format manageable, assume that despite sampling with replacement, each observation appears just once in each bootstrap sample. The cell entries for the first three rows are the CART weights for each of the three trees. The last row is the average of each column. Each row sums to 1.0 and the average weights sum to 1.0 (except for rounding error).

Tree	i=1	i=2	i=3	i=4	i=5	i=6
1	0	.33	.33	0	.33	0
2	.5	.5	0	0	0	0
3	.25	.25	0	.25	0	.25
Average	.25	.36	.11	.083	.11	.083

Table 5.2. Weights from three random trees for a single observation.

Table 5.2 contains three sets of CART weights and their average for a given observation. When in random forests the fitted value is computed for that observation, the average weights in the last row are used. Each value of $y_i, i = 1, 2, \ldots, 6$, is multiplied by its weight and summed. There would be a corresponding table for each of the six observations in the dataset.

There are at least three interesting implications that follow from formulating the random forest fitted values in terms of weights. First, a plot of the weights in the space defined by the predictors can be instructive. To begin, consider a given $\mathbf{x}_0$ and all of the nonzero weights for all of the observations associated with that target point. Then, find each x-value in the predictor space and mark that point with a symbol for the weight. Figure 5.4 is an example for CART in which just two predictors are shown: age and income. The response might be years of education. For the moment, assume that age and income are unrelated. For ease of exposition, we can ignore the other predictors used in the analysis such as gender or ethnicity. For simplicity, we also ignore the weight values.

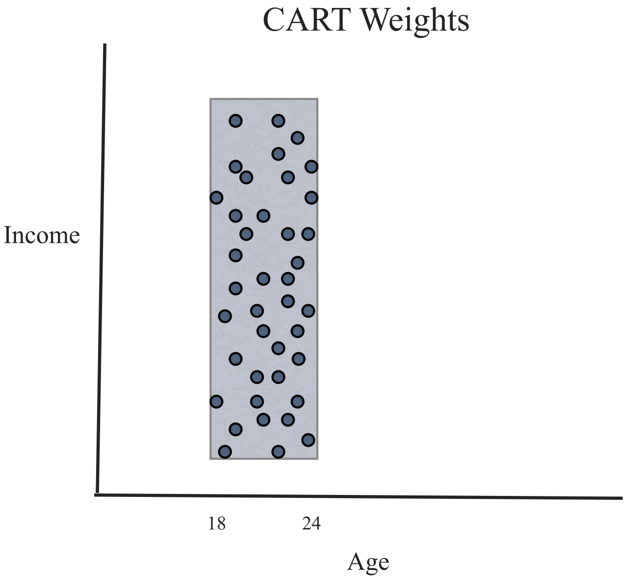


Fig. 5.4. Weights with two unrelated predictors.

Predictors that define the terminal node in which $\mathbf{x}_0$ falls will have all of nonzero weights (1) clustered along the dimensions of $\mathbf{x}$ used to define $\mathbf{x}_0$'s terminal node and (2) spread out along the dimensions of $\mathbf{x}$ not used to define

$\mathbf{x}_0$'s terminal node. Suppose, for example, age alone is used to define the terminal node. The terminal node only includes individuals between 18 and 24 years of age. Then, the plotted weights would be found within the range of 18 to 24 years old.

For Figure 5.4, income is not a defining feature of the node; it was not used to define the relevant splits. Then, the weights would be spread out over its range. Income is not by itself important for how the terminal node is defined and plays no systematic role in how the node mean or proportion is computed. In other words, such plots are characterized by tight clustering along some dimensions and little clustering at all along others, much as in Figure 5.4.

Suppose now that age and income are related. Perhaps older people tend to have higher incomes. The basic ideas just presented still apply, but the plot will be less dramatic. There will now be some clustering along the income dimension as well because of the association between the two predictors.

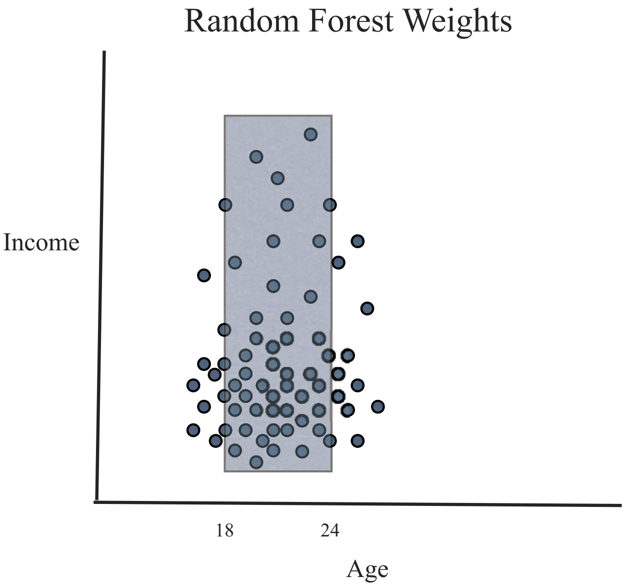


Fig. 5.5. Weights with two related predictors.

Random forests leads to somewhat different looking plots. As Figure 5.5 shows, the distinction between predictors that are included and predictors that are excluded is far less dramatic. One reason is that even weak predictors may be selected on occasion for use in partitioning the data if the predictors against which it is competing are even weaker. Another reason is that depending on the competing predictors selected by chance for consideration, a splitpoint for strong predictors may sometimes be chosen that would otherwise be ignored.

Consequently, although there will be more nonzero weights clustering within regions defined by stronger predictors than elsewhere, there will also be instances where weights for weak predictors or weak regions within strong predictors will be greater than zero. Notice that in Figure 5.5 there is clustering in the vertical direction implying that now there is a cutpoint separating moderate to low incomes from high incomes. When income does not compete with age, a useful partitioning of the data tends to be found. Notice also that there are now voting points for years of age outside of 18 to 24.

It follows that information from income that would have been pushed aside under CART can be brought to bear in random forests. The fitting burden is shared, at least a bit, between age and income because nonzero weights are more widely distributed. In this context, income is an illustration of a highly specialized predictor that random forests is able to exploit. This is also another way to think about how random forests regularizes the fitted values.

Compared to CART more generally, random forests can include a wider variety of locations in the predictor space when conditional means or proportions are computed. There can be more voting points for a larger set of predictors and partitions within predictors. It is this ability to make better use of predictors that in part explains random forests' successes.

A second implication of the weighting formulation is that once one recognizes that random forests computes its fitted values as weighted averages, other uses can be made of the weights (Meinshausen, 2006). In particular, when one has a quantitative response variable (more on this later), the weights can be used to construct the cumulative distribution of the response values for each configuration of x-values. And with this in hand, one can compute for quantitative variables any conditional quantiles of interest such as the median or the 90th percentile. In effect, one can do random forests quantile regression.

Table 5.3 provides a simple illustration for a given target value $\mathbf{x}_0$. There are ten response values available for $\mathbf{x}_0$, listed in order, that have nonzero average weights across trees. The mean is computed by multiplying each response value by its average weight and adding the products. In this case, the mean is 83. However, quantiles are also available. The 10th quantile is 66. The 50th quantile (the median) is 82. The 90th quantile is 92. In short, if information such as that found in Table 5.3 is available, one is not limited to the mean of each $\mathbf{x}_0$.

There will sometimes be interest in the fitted values of conditional medians to "robustify" random forest results or to consider a central tendency measure unaffected by the tails of the distribution. Sometimes, there is subject-matter interest in learning about the conditional distribution of a very high or very low quantile, especially if those conditional distributions differ from one another and from the conditional distribution for the median.

For example, in today's world of school accountability based on standardized tests, perhaps students who score especially poorly on standardized tests

Average Weight	Response Value	Cumulative Weight
.10	66	.10
.11	71	.21
.12	74	.33
.08	78	.41
.09	82	.50
.10	85	.60
.13	87	.73
.07	90	.80
.11	98	.91
.09	99	1.0

Table 5.3. Weights and cumulative weights for a target value $\mathbf{x}_0$

.

respond better to smaller classroom sizes than students who excel on standardized tests. The performance distribution on standardized tests, conditioning on classroom size, differs for good versus poor performers. Random forests quantile regression can address such concerns. A real illustration is provided later in this chapter.

A final implication is that the weights provide another way to think about how a correct classifier would perform. In the population or stochastic process responsible for the data, the $f(X)$ leads to weights that in turn generate as a weighted average the systematic part of the response variable. A desirable classifier working with a random sample from this population or a random realization from the stochastic process might provide unbiased, or at least consistent, estimates of those weights. From these estimated weights, unbiased or consistent estimates of the fitted values could be constructed. By this standard, CART and random forest do not measure up.

5.5 Taking Costs into Account in Random Forests

Just as in CART, there is a need to consider the relative costs of false negatives and false positives. Otherwise, for each tree, one again has to live with the default values of equal costs and a prior distribution for the response variable that is the same as its empirical distribution in the data.

Perhaps the most conceptually direct method would be to allow for a cost matrix just as CART does. To date, this option is not available in random forest software, and there are suspicions that it might not work effectively if it were.

There are four approaches that have been seriously considered for the binary class case. They differ by whether costs are imposed on the data before each tree is built, as each tree itself is built, or at the end when classes are assigned.

1. Just as in CART, one can use a prior distribution to capture costs when each tree is built. This has the clear advantages of being based on the mechanics of CART and a straightforward way in the binary case to translate costs into an appropriate prior.
2. After all of the trees are built, one can differentially weight the classification votes over trees. For example, one vote for classification in the less common category might count the same as two votes for classification in the more common category. This has the advantage of being easily understood.
3. After all of the trees are built, one can abandon the majority vote rule and use thresholds that reflect the relative costs of false negatives and false positives. For instance, rather than classifying as "1" all observations for which the vote is larger than 50%, one might classify all observations as "1" when the vote is larger than 33%. This too is easy to understand.
4. When each bootstrap sample is drawn before a tree is built, one can oversample one class of cases relative to the other class of cases in much the same spirit as disproportional stratified sampling used for data collection (Thompson, 2002: Chapter 11). Before a tree is built, one oversamples the cases for which forecasting errors are relatively more costly. Conceptually, this is a lot like altering the prior distribution.

All four approaches share the problem that the actual ratio of false negatives to false positives in the confusion table may not sufficiently mirror the cost ratio. In practice, this means that whatever method is used to introduce relative costs, that method is simply considered a way to "tune" the results. With some trial and error, an appropriate ratio of false negatives to false positives can usually be achieved.

Although some very tentative experience suggests that in general all four methods can tune the results as needed, there may be some preference for tuning by the prior or by stratified bootstrap sampling. Both of these methods will affect the confusion table through the trees themselves. The structure of the trees themselves responds to the costs introduced. Changing the way votes are counted or the thresholds used only affects the classes assigned, and leaves the trees unchanged. The defaults of equal costs and the empirical prior remain in effect. It would seem that by allowing the trees themselves to respond to cost considerations, more responsive forecasts should be produced. Moreover, any output beyond a confusion table will reflect the role of costs. More is said about such output shortly.

There is one very important situation in which the stratified sampling approach is likely to be superior to the other three approaches. If the response variable is highly unbalanced (e.g., a 95–5 split), any given bootstrap sample may fail to include enough observations for the rare category. Then, a useful tree will be difficult to construct. As observed earlier, it will often be difficult under these circumstances for CART to move beyond the marginal distribution of the response. Oversampling rare cases when the bootstrap sample is drawn will generally eliminate this problem. Using a prior that makes the rare observations less rare can also help, but that help applies in general and will not be sufficient if a given bootstrap sample makes the rare cases even more rare. We consider some examples in depth shortly. But a very brief illustration is provided now to prime the pump.

5.5.1 A Brief Illustration

	Forecast No Misconduct	Forecast Misconduct	Model Error
No Misconduct	3311	1357	.29
Misconduct	58	80	.42
Use Error	.02	.94	Overall Error = .29

Table 5.4. Confusion table for forecasts of serious prison misconduct.

Table 5.4 was constructed using data from the prison misconduct study described earlier. In this example, the response is incidents of very serious misconduct, not the garden-variety kind. As noted previously, such misconduct is relatively rare. Less than about 3% of the inmates had such reported incidents. So, just as for the domestic violence data shown in Table 5.1, it is difficult to do better than the marginal distribution under the usual CART defaults.

Suppose that the costs of forecasting errors for the rare cases were substantially higher than the costs of forecasting errors for the common cases. These relative costs can be effectively introduced by taking a stratified bootstrap sample, oversampling the rare cases. And by making the rare cases less rare, problems that might follow from the highly unbalanced response variable can sometimes be overcome.

For Table 5.4, the bootstrap samples for each of the response categories was set to equal 100. The "50–50" bootstrap distribution was selected by trial and error to produce a cost ratio of false negatives to false positives of about 20 to 1. This may be too high for real policy purposes, but it is still within the range considered reasonable by prison officials.

Why 100 cases each? Experience to date suggests that the sample size for the less common response category should equal about two-thirds of the number of cases in the class. If a larger fraction of the less common cases is sampled, the out-of-bag sample size may be too small.

With the number of bootstrap observations for the less common category determined to be 100, the 50–50 constraint leads to 100 cases being sampled for the more common response category. In practice, one determines the sample size for the less common outcome and then adjusts the sample size of the more common outcome as needed.

Table 5.4 can be interpreted just as any of the earlier confusion tables. For example, the overall proportion of cases incorrectly identified is .29. Random forests forecasts 42% of the incidents of misconduct incorrectly and 29% of the no misconduct cases incorrectly. Were prison officials to use these results for forecasting, a forecast of no serious misconduct would be wrong only 2 times out of 100, and a forecast of serious misconduct would be wrong 94 times out of 100. But for very serious inmate misconduct, having 1 true positive for about 20 false positives may be an acceptable tradeoff. The misconduct represented can include homicide, assault, sexual assault, and narcotics trafficking.

To summarize, random forests provides several ways to take the costs of false negatives and false positives into account. Ignoring these options does not mean that costs are not affecting the results. The default is equal to the costs and the use of the marginal distribution of the response variable as the empirical prior. However, there is to date no formal justification for preferring one costing method over the others, and the early hands-on experience is far from conclusive.

5.6 Determining the Importance of the Predictors

Just as for bagging, random forests leaves behind so many trees that collectively they are useless for interpretation. Yet, a central goal of statistical learning is to explore how inputs are related to outputs. Exactly how best to do this is currently unresolved, but there are several useful options available. We begin with a discussion of variable “importance.”

5.6.1 Contributions to the Fit

One approach to predictor importance is to record the decrease in the fitting measure (e.g., Gini index) each time a given variable is used to define a split. The sum of these reductions for a given tree is a measure of importance for that variable when that tree is built. For random forests, one can average this measure of importance over the set of trees.

As with variance partitions, however, reductions in the fitting criterion ignore the forecasting skill of a model, which many statisticians treat as the gold standard. Fit measures are computed with the data used to build the classifier. They are not computed from test data.

Moreover, it can be difficult to translate contributions to a fit statistic into practical terms. Simply asserting that a percentage contribution to a fit statistic is a measure of importance is circular. Importance must be defined outside of the procedure used to measure it. And what is it about contributions to a measure of fit that makes a predictor more or less important? Even if an external definition is provided, is a predictor important if it can account for, say, 10% of the reduction in impurity?

In any case, one must be fully clear that contributions to the fit by themselves are silent on what would happen if in the real world a predictor is manipulated. Causality can only be established by how the data were generated, and causal interpretations depend on there being a real intervention altering one or more predictors (Berk, 2003).

5.6.2 Contributions to Forecasting Skill

Breiman (2001a) has suggested another form of randomization to assess the role of each predictor. This method is implemented in the R version of random forests. It is based on the reduction in predictive accuracy when a predictor is shuffled so that it cannot make a systematic contribution to a forecast. Reductions in predictive accuracy can be translated into practical terms. Would a reduction of, say, 10% in forecasting accuracy matter in real applications? In contrast to fit statistics, forecasting skill has direct implications for actual decisions.

Breiman's approach has much in common with the concept of Granger causality (Granger and Newbold, 1986: Section 7.3). Imagine two times series, Y_t and X_t. If the future conditional distribution of Y given current past values of Y is the same as the future conditional distribution of Y given current and past values of Y and X, X does not Granger cause Y. If the two future conditional distributions differ, X is a *prima facie* cause of Y.

These ideas generalize so that for the baseline conditional distribution, one can condition not just on current and past values of Y but on current and past values of other predictors (but not X). Then X Granger causes Y, conditional on the other predictors, if including X as a predictor changes the future conditional distribution of Y. In short, the idea of using forecasting skill as a way to characterize the performance of predictors has been advanced in both the statistical and econometrics literature.

Breiman's importance measure of forecasting skill differs perhaps most significantly from Granger's in that Breiman does not require time series data and randomly shuffles the values of predictors rather than dropping (or adding) predictors from the forecasting model. The latter has some important implications discussed shortly.

For Breiman's approach a categorical response variable is constructed using the following algorithm.

1. Construct a measure prediction error ν for each tree as usual by dropping the out-of-bag (OOB) data down the tree. Note that this is a real forecasting enterprise because data not used to build the tree are used to evaluate its predictive skill.
2. If there are p predictors, repeat Step 1 p times, but each time with the values of a given predictor randomly shuffled. The shuffling makes that predictor on the average unrelated to the response and all other predictors. For each shuffled predictor j, compute new measures of prediction error, ν_j.

3. For each of the p predictors, average over trees the difference between the prediction error with no shuffling and the prediction error with the jth predictor shuffled.

The average increase in the forecasting error when a given predictor j is shuffled represents the importance of that predictor for forecasting skill. That is,

$$\mathrm{I}_j = \sum_{k=i}^{K} \left[\frac{1}{K}(\nu_j - \nu) \right], \quad j = 1, \ldots, p, \tag{5.10}$$

where there are K trees, ν_j is the forecasting error with predictor j shuffled, and ν is the forecasting error with none of the predictors shuffled. It is sometimes possible for forecasting accuracy to improve slightly when a variable is shuffled because of the randomness introduced. A negative measure of forecasting importance follows. Negative forecasting importance can be treated as no decline in accuracy or simply can be ignored.

As written, Equation 5.10 is somewhat open-ended. The measures of forecasting error (ν and ν_j) are not defined. One could imagine using the number of forecasting errors, the percentage of cases forecasted incorrectly, the change in the margin, or some other measure. Currently, the preferred measure is the same as the one used when confusion tables are constructed: the proportion (or percentage) of cases misclassified. This has the advantage of allowing direct comparisons between predictor importance and either the row or column totals in the table. In addition, all of the other measures considered to date have been found less satisfactory for one reason or another. For example, some measures are misleadingly sensitive; small changes in the number of classification errors can lead to large changes in the importance measure.

One significant complication is that Equation 5.10 will almost always produce different importance measures for given predictors for different categories of the response. That is, there will be for any given predictor a measure of importance for each class forecasted, and the measures will not generally be the same. For example, if there are three response classes, there will be three measures of importance for each predictor that will generally not be the same. Moreover, this can lead to different rankings of predictors in the forecasting importance depending on which response category is being considered. Although this may seem odd, it follows directly from the fact that the number of observations in each response class and the margins for each class will typically differ. Consequently, a given increase in the number of misclassifications can have different impacts. A detailed illustration is be presented shortly.

Partly in response to such complications, one can standardize the declines in forecasting skill. The standard deviation of Equation 5.10 can be computed over the K trees. In effect, one has a bootstrap estimate over trees of the standard error associated with the increase in forecasting error, which can be used as a descriptive measure of stability. Larger values imply less stability.

Then, one can divide Equation 5.10 by this value. The result can be interpreted as a z-score so that importance measures are now all on the same scale. And with a bit of a stretch, confidence intervals can be computed and conventional hypothesis tests performed. It is a stretch because the sampling distribution of the predictor importance measure is usually not known. Perhaps more important, the descriptive gains from standardization are modest at best, as the illustrations that follow make clear.

One of the drawbacks of the shuffling approach to variable importance is that only one variable is shuffled at a time. There is no role for joint importance over several predictors. This can be an issue when predictors are not independent. There will be a contribution to forecasting skill that is uniquely linked to each predictor and a joint contributions shared between two or more predictors.

There is currently no option in the random forest software to shuffle more than one variable at a time. However, it is relatively easy to apply the prediction procedure in random forests using as input the original dataset with two or more of the predictors shuffled. Then, Equation 5.10 can be employed as before, where j would now be joined by other predictor subscripts. The main problem is that the number of potential joint contributions can be very large. In practice, some subset selection procedure is likely to be needed, perhaps based on substantive considerations.

It might seem that Granger's approach of examining forecasting skill with and without a given predictor included is effectively the same as Breiman's shuffling approach. And if so, one might consider, for instance, dropping sets of predictors to document their joint contribution. But actually, the two strategies are somewhat different. In Granger's approach, dropping or adding predictors to the model means that the model itself will be re-estimated each time. So, the comparisons Granger favors are the result of different predictors being included and different models. Under Breiman's approach, the model is not reconstructed. The shuffling is undertaken as an additional procedure with the model fixed.

In summary, for many scientists the ability to forecast accurately is the gold standard of a model's worth. If one cannot forecast well, it means that the model cannot usefully reproduce the empirical world. It follows that such a model has little value. And as now stressed a number of times, a model that fits the data well will not necessarily forecast well. The take-home message is simple: if forecasting skill is the gold standard (or even just a very important criterion by which to evaluate a model), then a predictor's contribution to that skill is surely one reasonable measure of that predictor's importance.

Some Examples

Figures 5.6 to 5.9 show how predictor importance can be represented as a reduction in forecasting accuracy. The data are, once again, from the prison study with the response variable very serious misconduct in prison. In each figure, importance is on the horizontal axis and predictor names are on the vertical axis.

Figure 5.6 displays predictor importance for the “misconduct” response category. Importance is just the average decline over trees in forecasting accuracy. It is, therefore, “unscaled” or “unstandardized.” Taking all the variables into account, Table 5.4 indicates that serious misconduct is correctly forecasted about 58% of the time. That accuracy drops to about 51% (i.e., about 7%) if the variable “Term” is shuffled. It drops to about 52% (i.e., about 6%) if the variable “Gang” is shuffled. None of the other predictors meaningfully affect forecasting accuracy. The two most important predictors are “Term” and “Gang.” Recall that “Term” refers to sentence length and “Gang” refers to street or prison gang activity.

It is useful to keep in mind that the importance represented is the decline in forecasting accuracy uniquely attributable to each predictor. Predictive skill shared between predictors is not included. Thus, for example, the sum of the declines in forecasting accuracy for all predictors is usually less, often far less, than overall forecasting accuracy.

Figure 5.7 shows the unscaled importance of the predictor for the no misconduct response category. Now the base is different because the absence of misconduct is forecast with about 71% accuracy. Moreover, there is no particular reason why the predictors that play a major role in forecasting misconduct should play a major role in forecasting no misconduct. Because this may seem to be counterintuitive, some discussion is warranted.

Recall how classification is accomplished in random forests. The class is assigned by majority vote. Two features of those votes are especially relevant here: the margin and the number of actual class members.

Consider a simple example. Suppose a given inmate receives a vote of 25 to 24 to be assigned to the misconduct class category. Suppose that in fact that inmate has a reported incident of serious misconduct; the forecast is correct. Now a predictor is shuffled. The vote might be very different. But suppose it is now 24 to 25. Only one vote has changed. Yet, the inmate is now placed in the no misconduct class. This increases the forecasting error by one inmate.

Is that one inmate increase enough to matter? It depends on how many inmates were correctly predicted to have a serious misconduct incident and how many were incorrectly predicted to have a serious misconduct incident. Suppose 10 inmates actually had an incident of serious misconduct, with 7 correctly predicted to have an incident of serious misconduct and 3 incorrectly predicted to have an incident of serious misconduct. Changing just one inmate from a true positive to a false negative reduces forecasting accuracy from 70% to 60%. If there had been 100 inmates who actually had incidents of serious misconduct with 70 true positives and 30 true negatives, changing that one inmate would reduce forecasting accuracy from 70% to 69%.

In summary, if the margins tend to be small, dropping a predictor can easily change the class assigned for a significant number of cases. Then, if the number of true class members is small as well, the change of even a few

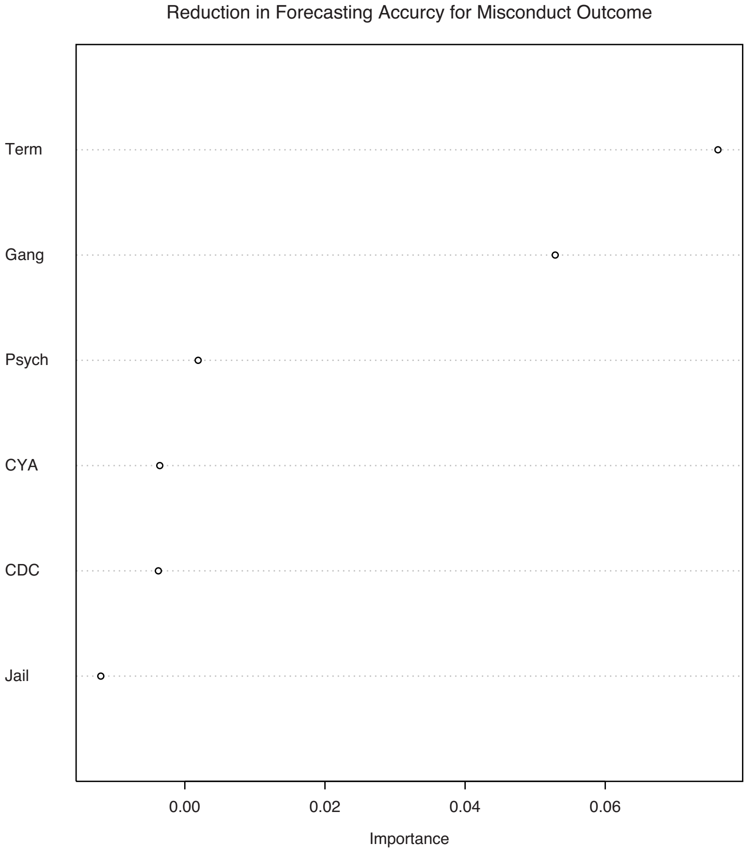


Fig. 5.6. Unscaled forecasting importance for misconduct

cases from one assigned class to another can dramatically affect the proportion of cases whose class membership was accurately forecasted. Because both the margins and the class sizes can differ depending on which response category is considered, forecasting importance of the predictors can differ as well. Thus, Figure 5.6 looks somewhat different from Figure 5.7. Comparisons such as these seem to argue for some kind of standardization, perhaps through the z-scores mentioned earlier.

In Figure 5.7, none of the predictors affect forecasting skill very much. Yet, "Term" is still the most important predictor. A previous sentence in a state juvenile facility (CYA) now comes in second. Gang activity drops to third place.

The differences between Figures 5.6 and 5.7 illustrate the kinds of complications just noted. As an empirical matter, the number of no misconduct observations is large. Other things being equal, many cases would have to change from true negative to false positive for forecasting accuracy to decline a substantial amount. But it is more complicated than that because margins for each case will likely differ from the margins when misconduct is the response category.

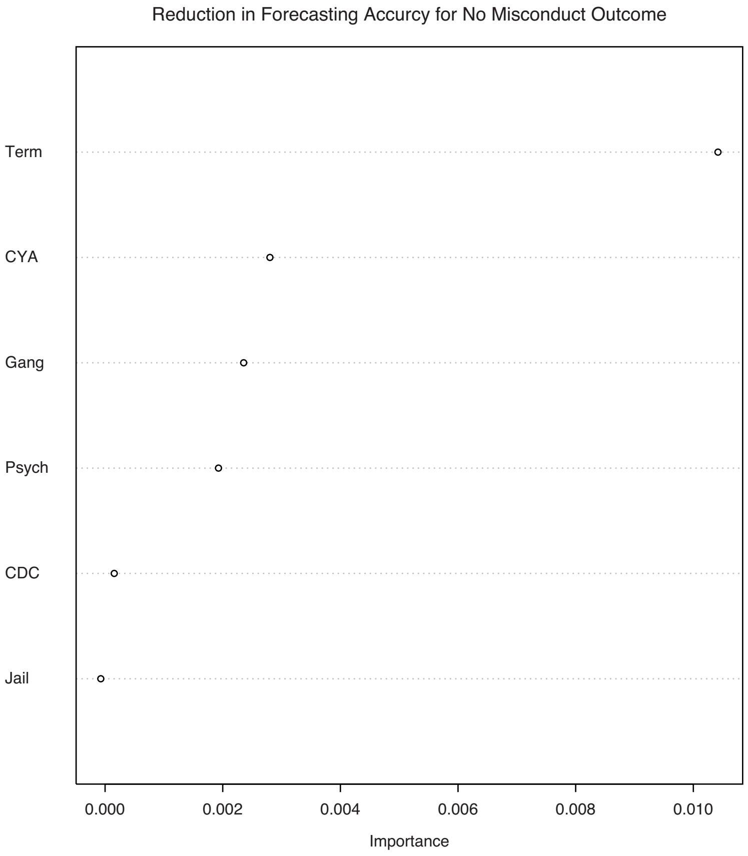


Fig. 5.7. Unscaled forecasting importance for no misconduct.

Figures 5.8 and 5.9 replay the same analysis in standardized scores. The horizontal axis is now in z-scores. Because forecasting importance is scaled to be in units of the same size, the two figures can be more easily compared. However, we already know that forecasting skill declines very little for the no misconduct class when predictors are shuffled. Those are the facts. It is not clear, therefore, how standardizing helps if the goal is to characterize predictors by their forecasting skill. Two predictors may be deemed strong and equally important by the z-score metric when in fact one substantially affects forecasting skill and the other does not.

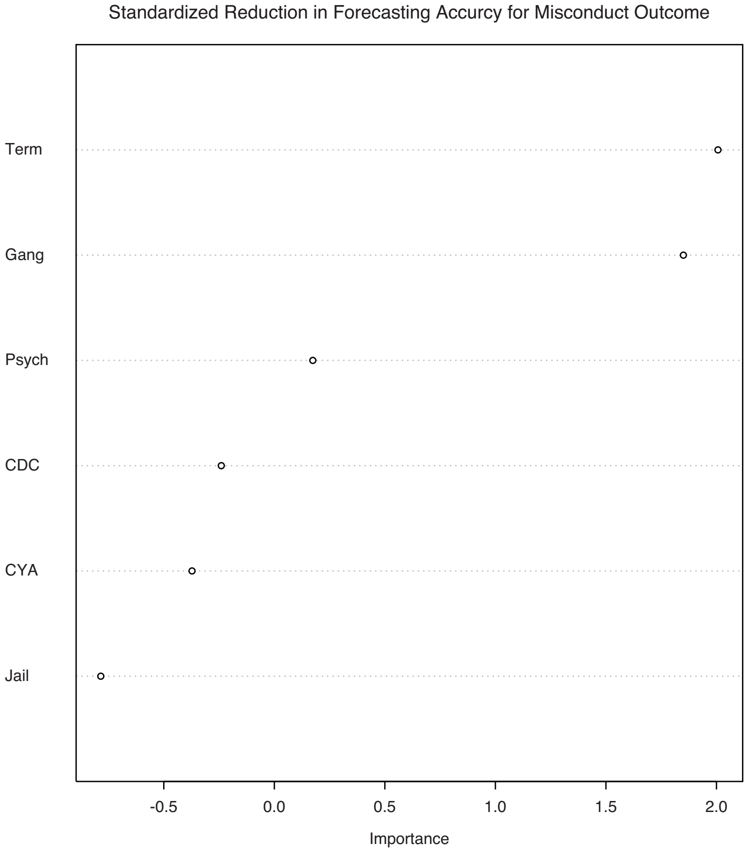


Fig. 5.8. Scaled forecasting importance for misconduct.

Let's return to the matter of stability over trees and take "Term" for the misconduct outcome as an example. The standard deviation over trees of the measure of forecasting importance is about .03. Thus for term length, one can say that although mean importance over trees is about .07, importance will typically vary from about .04 to about .10. If, however, the standard deviation over trees were .10, importance would typically vary from the lower bound of 0.0 to about .17. Clearly, one has a much worse fix on how important term length really is for the second case.

Insofar as the distribution of raw importance scores over trees is approximately normal, formal hypothesis tests and confidence intervals can follow.

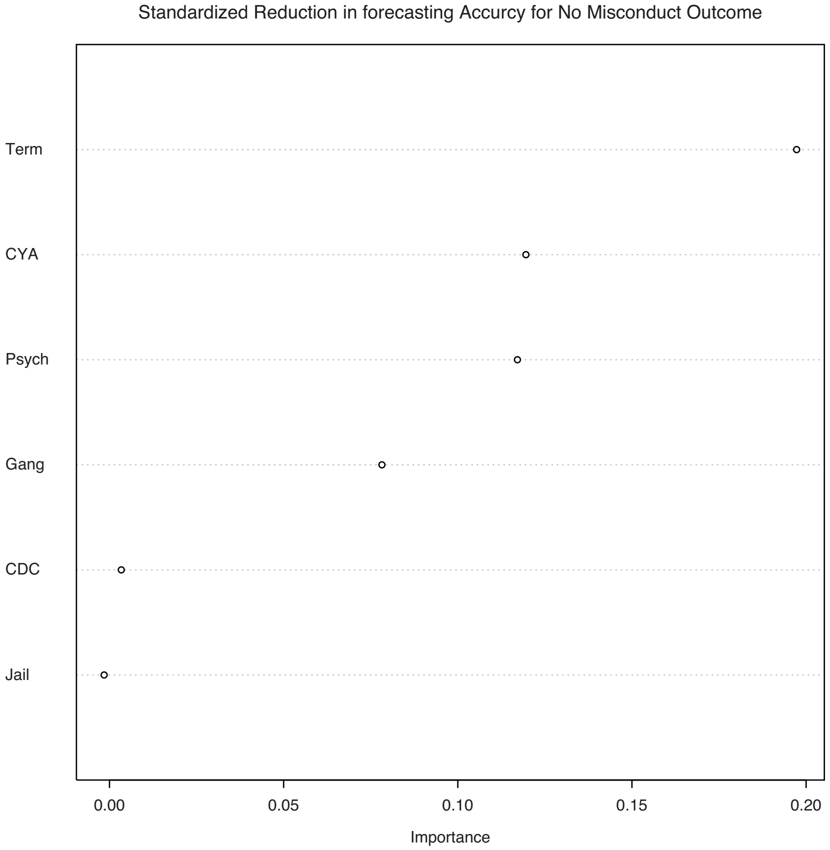


Fig. 5.9. Scaled forecasting importance for no misconduct.

For example, if term length had a z-score of over 2.0, one might justifiably take that as a rejection of the null hypothesis at the .05 level that the importance of term length is 0.0. If one is not prepared to bet that the distribution of importance is approximately normal, one can in principle resort to resampling tests directly over the set of trees. This capability is currently not available in R's random forests software, but with very modest changes in the code it could be.

One must be clear that the uncertainty being assessed comes from the bootstrap sampling and predictor sampling within random forests itself. Nothing whatsoever is being said about the stability of importance measures over sets of training data selected by probability sampling. Random forests outputs a single measure of importance for each predictor as an average over trees. If one were interested in the overall uncertainty in this single measure for each predictor, one would at least need to address in addition the implications of random samples of training data. A possible approach would be to embed the random forest procedure in bootstrap samples of the existing training data.

In summary, the decision to standardize or not standardize raises the old saw of substantive versus statistical significance. The unstandardized measure of forecasting importance addresses substantive significance. The standardized measure of forecasting importance addresses (ideally) statistical significance. Both can be important, but they are different.

The implications of these illustrations generalize to response variables with more than two categories. There can be scaled or unscaled plots for each response category. This underscores a point made earlier. Moving from two response categories to more than two does not change anything fundamental. But analysis complexity will increase dramatically with each additional response category.

5.7 Response Functions

Predictor importance is only part of the story. In addition to knowing the importance of each predictor, it can be very useful to have a description of how each predictor is related to the response. The set of response functions needs to be described.

One useful solution, based on an earlier suggestion by Breiman and his colleagues (1984) is "partial dependence plots" (Friedman, 2001; Hastie et al., 2001: Section 10.13.2). For tree-based approaches such as CART, one proceeds as follows.

1. Grow a forest.
2. Suppose x_1 is the initial predictor of interest, and it has v distinct values in the training data. Construct v data sets as follows.
   a) For each of the v values of x_1, make up a new dataset where x_1 only takes on that value, leaving all other variables untouched.
   b) For each of the v datasets, predict the response using random forests. There will be a single value averaged over all observations.
   c) Average each of these predictions over the trees.
   d) Plot the average prediction for each value for each of the v datasets against the v values of x_1
3. Go back to Step 2 and repeat for each predictor.

Partial dependence plots show the relationship between a given predictor and the response averaged within the joint values of the other predictors as they are represented in a tree structure. In this way, the other predictors are being "held constant" by matching. Consequently, no assumptions are made about how the predictors are related to one another or to the response variable. One price for this approach is that interaction effects are not represented unless the appropriate interaction variables are constructed in advance and included among the set of predictors.

Unlike the plots of fitted values constructed by smoothers (e.g., from the generalized additive model), partial dependence plots impose no smoothness constraints, and the underlying tree structure tends to produce somewhat bumpy results. In practice, one usually imposes an "eyeball" smoother when the plot is interpreted. Alternatively, it is often possible to overlay a smoother if the software stores the requisite output.

Partial plots can be constructed for quantitative responses and for responses with more than two categories. For quantitative response variables, the units represented on the vertical axis usually are the natural units of the response, whatever they happen to be. For categorical response variables, the units of the response represented on the vertical axis are unconventional and easily misunderstood. Because partial dependence plots can be so very useful for applied work, the metric used needs to be examined in some detail. We begin with the binomial case.

It is common to see a logistic regression equation written as

$$\log\left(\frac{p}{1-p}\right) = \mathbf{X}\boldsymbol{\beta}, \tag{5.11}$$

where p is the probability of success. The term on the left-hand side is the log of the odds of a success, often called the "logit." The change in the response for a unit change in a predictor is in "logits."

For the multinomial case, the most common approach to logistic regression builds up from the familiar binary formulation. If there are K response categories, there are $K-1$ equations, each of the same general form as Equation 5.11. However, one equation of the K possible equations is redundant because the response categories are exhaustive and mutually exclusive. Thus, if an observation does not fall in categories $1,\ldots,K-1$, it must fall in the Kth category. This implies that a single category can be chosen as the reference category, just as in the binomial case (i.e., there are two possible outcomes and one equation). Then, for each of the $K-1$ equations, the logit is the log of the odds for a given category compared to the reference category.

Suppose there are four response categories, and the fourth is chosen as the reference category. There would then be three equations with three different responses, one for $\log(p_1/p_4)$, one for $\log(p_2/p_4)$, and one for $\log(p_3/p_4)$. The predictors would be the same for each equation, but each equation would have its own set of regression coefficients differing in values across equations.

One might think that partial dependence plots would follow a similar convention. But they don't. The choice of the reference category determines which logits will be used, and the logits used affect the regression coefficients that result. Although the overall fit is the same no matter what the reference category, and although one can compute from the set of estimated regression coefficients what the regression coefficients would be were another reference category used, the regression coefficients reported are still different when different reference categories are used.

There is no statistical justification for choosing one reference category or another. The choice is usually made on subject matter grounds to make interpretations easier, and the choice can easily vary from data analyst to data analyst. So, the need for a reference category can complicate interpretations of the results and means that a user of the results has to undertake considerable additional work if regression coefficients using another reference category are desired.

In response to these complications, partial dependence plots are based on a somewhat different approach. There are K, rather than $K-1$, response functions, one for each response variable class. For the logistic model, these take the form of

$$p_k(X) = \frac{e^{f_k(X)}}{\sum_{k=1}^{K} e^{f_k(X)}}. \tag{5.12}$$

There is still a redundancy problem to solve. The solution employed by partial dependence plots is to constrain $\sum_{k=1}^{K} f_k(X) = 0$. This leads to the multinomial deviance loss function and the use of a rather different kind of baseline.

Instead of using a given category as the reference, the unweighted mean of the proportions in the K categories is used as the reference. In much the same spirit as analysis of variance, the response variable units are then in deviations from a mean. More specifically, we let

$$f_k(X) = \log[\mathrm{p}_k(X)] - \frac{1}{K}\sum_{k=1}^{K} \log[\mathrm{p}_k(X)]. \tag{5.13}$$

Thus, the response is the disparity between the logged proportion for category k and the average of the logged proportions for all K categories. The units are essentially logits but with the mean over the K classes as the reference. Consequently, each response category can have its own equation and, therefore, its own partial dependence plot. This approach is applied even when there are only two response categories, and the conventional logit formulation might not present interpretive problems.

To help fix these ideas, consider an example of a single data point for a binary outcome. Here, a single data point is defined by a single value of a given predictor. For that data point, the partial dependence algorithm classifies all of the observations as described earlier. For each of the response variable categories, there is a proportion of observations assigned. Then Equation 5.13 is applied.

To illustrate, consider once again the prison data. Suppose term length in years was the predictor whose relationship with the binary misconduct response variable was of interest. And suppose for a term length of say, 1 year, the proportion of inmates engaging in an incident of misconduct was

.20 (computed using the partial dependence algorithm). If the proportion of success is .20, the value plotted is $\log(.2)-[\log(.2)+\log(.8)]/2 = -0.693$ (using natural logarithms). This is the value that would be plotted on the vertical axis for the term length value on the horizontal axis of 1.0.

The same approach could be used for the proportion of inmates with no misconduct. The proportion of failures is necessarily .80, so that value plotted for failures is $\log(.8)-[\log(.2)+\log(.8)]/2 = 0.693$, also associated with a term length of 1.0. In the binary case, essentially the same information is obtained no matter which response class is examined. The partial dependence function is centered on 0.0 because of the constraint that the sum of the response functions is zero; these are deviation scores in logit units. So, the distance above zero for the response function of one class is the same as the distance below zero for the response function of the other class. One response function is the mirror image of the other. Thus, one partial dependence plot is the mirror image of the other partial dependence plot, and only one of the two is required for interpretation.

In the binary case, it is easy to get back into more familiar logit units. The values produced by Equation 5.13 are half the usual log of the odds. And from there, one can easily get back to the relevant probabilities. For example, multiplying –.693 by 2 and exponentiating yields an odds of .25. Then, solving for the numerator probability results in a value of .20. We are back where we started.

Equation 5.13 would be applied for each value of term length. Thus, for 1.5 years, the proportion of inmates engaging in misconduct might be .25. Then the value plotted on the horizontal axis would be 1.5, and the value on the vertical axis would be $\log(.25) - [\log(.25) + \log(.75)]/2 = -.549$.

The value of –.549 is at a region where the response function is increasing. With .5 units (i.e., six months) increase in term length, the value of the response increases .144 (i.e., from –.693 to –.549). And all other values produced for different term lengths can be interpreted in a similar way. Consequently, one can get a sense of how the response variable changes with changes in a given predictor, all other predictors held constant.

For more than two response variable categories, each of the response categories can be usefully plotted. Suppose, for example, there are three response categories: no misconduct, minor misconduct, and serious misconduct. And suppose the respective proportions when term length is 1.0 years are .70, .20, and .10. The three values computed for the three response categories are respectively 1.066, –.187, and –.880. Note that as before the sum of the values is again 0.0. Each of these values would likely change as the value of the predictor of interest changed. For each, the sum would still be zero. But the changing values could not be represented as a set of mirror images. Three partial dependence plots would follow.

To summarize, the vertical axis in partial dependence plots is the response function as defined by Equation 5.13. It is derived from Equation 5.12 with the constraint that the sum of response functions $f_k(X)$ is equal to zero. Thus, the sum of the values from Equation 5.13 for the categories of the response variable is zero as well.

5.7.1 An Example

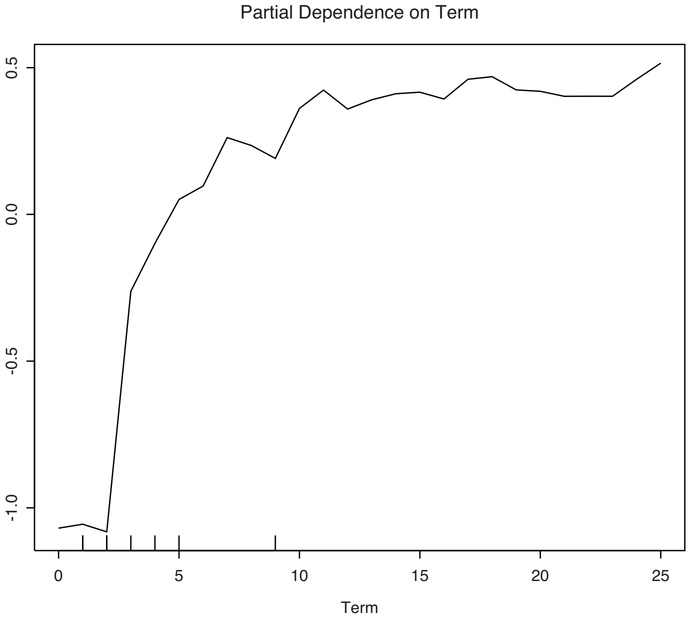


Fig. 5.10. Response function for misconduct and term.

Figures 5.10 and 5.11 show partial dependence plots constructed from the prison data. Figure 5.10 is for the misconduct response category and Figure 5.11 is for the no misconduct response category. For both, term length in years is the predictor. In both cases the vertical axis is in the logit units just discussed, and the horizontal axis is in years. From the discussion just completed, one plot should be the mirror image of the other. For the binary case, one partial plot is sufficient.

Both plots indicate that the odds of serious misconduct generally increase with term length. The increase is relatively rapid for terms from two to ten years. There seems to be no relationship between term length and misconduct for terms less than two years, and the rate of increase is relatively slow for terms greater than about ten years.

In order to get a practical sense of whether misconduct varies a lot with term length, it can be useful to transform the logits back to their underlying

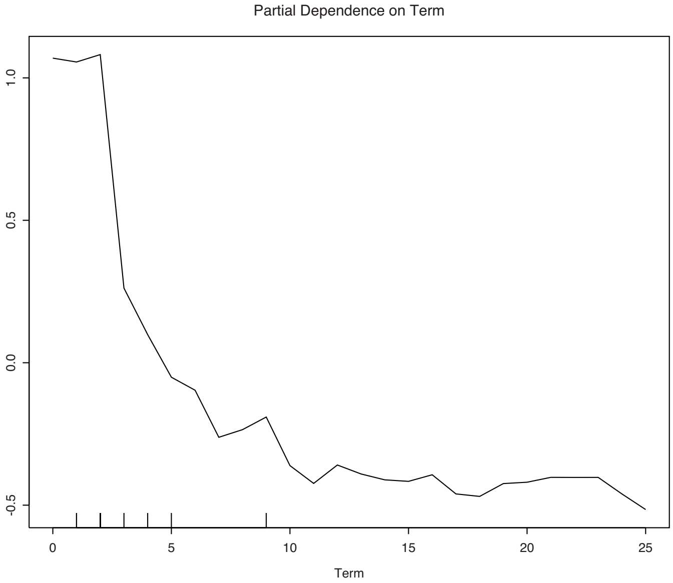


Fig. 5.11. Response function for no misconduct and term.

probabilities. For example, a logit value of −1 is equal to a probability of about .12. A logit value of .5 is equal to a probability of about .73. Because −1 and .5 represent the approximate minimum and maximum values of the response in logit units, the probability of serious misconduct increases from about .12 to about .72 as term length increases from two years to about ten years. This is a large effect in practical terms. Note that this increase is in deviation scores and that, therefore, it is the difference that matters, not the values themselves.

Finally, it is important to keep in mind that the response functions displayed in partial dependence plots reflect the relationship between a given predictor and the response, conditioning on all other predictors. All other predictors are being "held constant" in a manner that is equivalent to matching. That is why the plots are called partial dependence plots. Consequently, Figures 5.10 and 5.11 show how term length is related to serious misconduct, with gang activity, age at the time of admission to prison, and all other predictors included in the analysis held constant.

Figures 5.12 – 5.14 show the response functions for three classes of inmate misconduct and sentence length. The three classes, as before, are no misconduct, minor misconduct, and serious misconduct. Three partial dependence plots are necessary because although the values of the three response functions sum to zero, no plot is the mirror image of another. The baseline is, in the sense discussed above, the typical proportion of inmates over the three response classes.

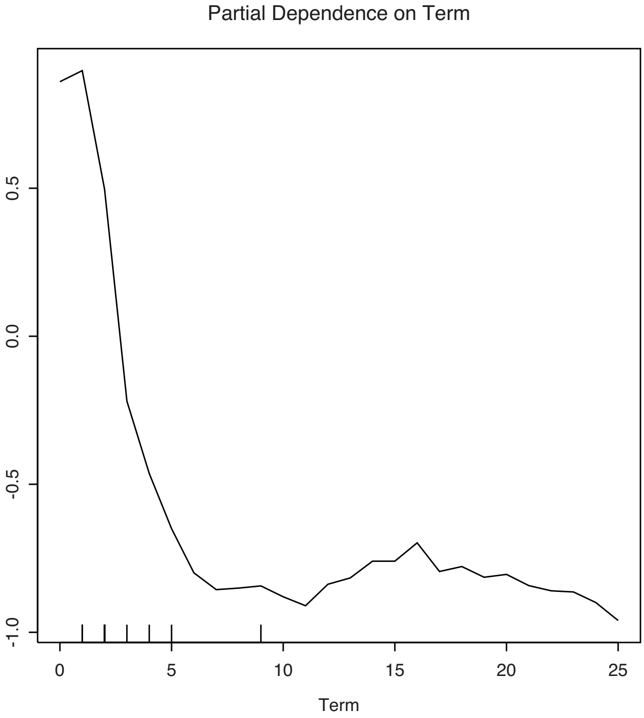


Fig. 5.12. Response function for no misconduct and term for a three class response.

Figure 5.12 shows the partial dependence of no misconduct on sentence length. The proportion of inmates with no reported incidents of misconduct decreases rapidly for sentences up to five years. Then the response function becomes flat.

Figure 5.13 shows the partial dependence of minor misconduct on sentence length. The proportion of inmates with reported incidents of minor misconduct increases rapidly for sentences up to about five years, levels off, and then declines for sentences of more than ten years. The downward trend ends with sentences of about 18 years. After that, it may even increase a bit.

Figure 5.14 shows the partial dependence of serious misconduct on sentence length. The proportion of inmates with reported incidents of serious misconduct increases rapidly up to a sentence of about five years and then increases much less rapidly thereafter.

Viewed as a group, the three figures complement one another and show associations that are large in practical terms. With increasing sentence length, the proportion of inmates who engage in no misconduct drops off rapidly until a sentence of about five years. Over those same shorter sentences, the propor-

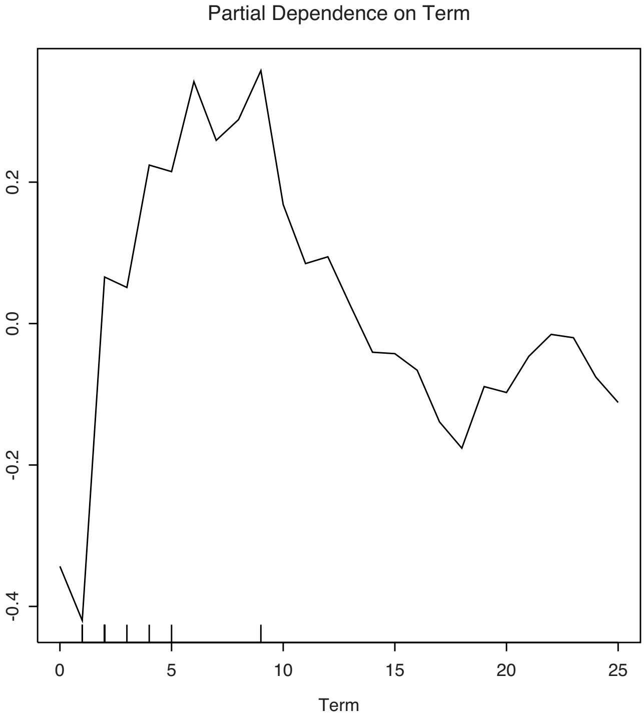


Fig. 5.13. Response function for minor misconduct and term for a three class response.

tion of inmates who engage in minor misconduct increases commensurately. And over the same short sentences, the proportion of inmates who engage in serious misconduct also increases commensurately.

But for sentences of around ten years or more, the proportion of inmates who engage in minor misconduct falls off, and the proportion of inmates who engage in serious misconduct continues to increase. Longer sentences are associated with increases in the likelihood of both minor and serious misconduct, but for very long sentences, the association is only with serious misconduct. One interpretation is that with very long sentences, inmates who might commit acts of minor misconduct now commit acts of serious misconduct.

5.8 The Proximity Matrix

It can be useful to determine the degree to which individual observations tend to be classified alike. In random forests, this information is contained in the "proximity matrix." The proximity matrix is constructed as follows.

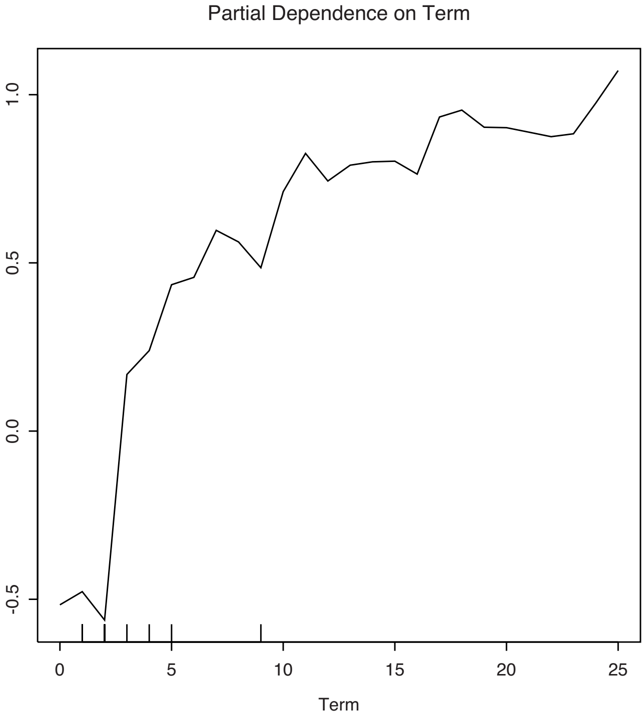


Fig. 5.14. Response function for serious misconduct and term for a three class response.

1. Grow a tree as usual.
2. Drop all the training data (in-bag and out-of-bag) down the tree.
3. For all possible pairs of cases, if a pair lands in the same terminal node, increase their proximity by one.
4. Repeat Steps 1–4 until the designated number of trees has been grown.
5. Normalize by dividing by the number of trees.

The result is an $n \times n$ matrix with each cell showing the proportion of trees for which each pair of observations winds up in the same terminal node. The higher that proportion, the more alike those observations are in how CART places them, and the more “proximate” they are.

However, it can be very demanding to store an $n \times n$ matrix and even more demanding to operate on it. The storage problem can be partly addressed by only storing the upper or lower triangle, but working with tens of thousands of proximity values (or more) remains a serious difficulty. When feasible, it can help to work with a random sample of the training data instead of the full set of observations. Another work around is to store only the largest proximity values. There are currently efforts under way to find more elegant and formally defensible solutions.

The proximity matrix is also usually far too large to be directly examined in a meaningful manner. But helpful information can be extracted from the proximity matrix in several different ways. We consider three applications that are sufficiently well developed to be of some practical use.

5.8.1 Clustering by Proximity Values

The proximity matrix can be treated as a similarity matrix and subjected to multidimensional scaling. Plots of the observations in the first two dimensions extracted can help show whether the data tend to cluster in the space defined by predictors and whether those clusters tend to differ by the class to which the observations in each cluster belong. This information can give an initial sense of whether a classification exercise is likely to be successful.

The existing methods for displaying the scaling results are currently in some flux. A lot depends on first solving the computational problems associated with the proximity matrix. In addition, the current graphic display will no doubt be refined as hands-on experiences accumulates.

5.8.2 Using Proximity Values to Impute Missing Data

There are two ways in which random forests can impute missing data. The first and quick method relies on a measure of location. If a predictor is quantitative, the median of the available values is used. If the predictor is categorical, the modal category from the available data is used. If there are small amounts of missing data, this method may be satisfactory, especially given the computational demands of the second method.

The second method capitalizes on the proximity matrix in the following manner.

1. The “quick and dirty” method of imputation is first applied to the training data, a random forest is constructed, and the proximity values computed.
2. If the missing value is from a quantitative variable, the weighted average of the values of the nonmissing cases for that variable is used. The proximity values between that missing observation and all of the nonmissing observations are used as the weights. So, cases that are more like the cases with the missing data are given greater weight. All missing values for that variable are computed in the same fashion.
3. If the missing value is from a categorical value, the inputed value is the most common nonmissing value for the variable, with the frequencies weighted by proximity. Again, cases more like the case with the missing data are given greater weight. All missing values for that variable are are computed in the same fashion.

The step using proximity values is then iterated several times. Experience to date suggests that four to six iterations is sufficient. But the use of imputed values tends to make the OOB measures of fit too optimistic. There is really less information being brought to bear in the analysis than the random forest algorithm knows about. The computational demands are also quite daunting and may be impractical for many datasets until more efficient ways to handle the proximities are found.

5.8.3 Using Proximities to Detect Outliers

The proximity matrix can be used to spot outliers in the space defined by the predictors. The basic idea is that outliers are observations whose proximities to all other observations in the data are small. Currently, the procedures in R's version of random forests to detect outliers are not implemented for quantitative response variables. For categorical response variables, outliers are defined within categories of the response variable. For each observed outcome class, each observation is given a value for its "outlyingness" computed as follows.

1. For a given observation, compute the sum of the squares of the proximities with all of the other observations in the same outcome class. Then take the inverse. A large value will indicate that on the average the proximities are small for that observation. Do the same for all other observations in that class. One can think of these values as unstandardized.
2. Compute the median and mean absolute deviation around the median of the unstandardized values.
3. Subtract the median from each of the unstandardized values and divide by the mean absolute deviation. In this fashion, the unstandardized values are standardized.
4. Values less than zero are set to 0.0.

These steps are then repeated for each category of the response variable. Observations with values larger than about ten can be considered outliers.

Especially if the number of observations overall is modest (e.g., less than 100), it can be instructive to drop the outliers from the training data, repeat the random forest analysis, and see if the results change by a meaningful amount. If the number of observations is large, it is very unlikely that a few outliers will make an important difference in the results.

When the data analyst considers dropping one or more outlying cases, a useful diagnostic tool can be a cross-tabulation of the classes assigned for the set of observations that the two random forest analyses have in common. If the observations are, by and large, classified in the same way in both analyses, the outliers do not make an important difference to the classification process.

5.9 Quantitative Response Variables

There is not very much new that needs to be said about quantitative response variables once one appreciates that random forests handles quantitative response variables much as CART does. Recall that for CART, impurity when trees are constructed is defined as the within-node error sum of squares. A new partition of the data is determined by the split that would most reduce the within-node error sum of squares. Predicted values are determined by the mean of the response variable in each of the terminal nodes. For each observation, the mean of its terminal node is the value assigned.

For regression trees, therefore, there are no classification errors, only residuals. Concerns about false negatives and positives and their costs are no longer relevant. There are no confusion tables and no measures of importance based on predictor errors.

To turn a regression tree into a fully operational random forest, there are several operations required.

1. Just as in the classification case, each tree is constructed from a random sample (with replacement) of the training data.
2. Just as in the classification case, at each potential partitioning of the data, a random sample (without replacement) of predictors is used.
3. Just as in the classification case, the out-of-bag data are used to construct predicted values. After a tree is built, the OOB observations are dropped down the tree. From these observations, a mean is computed for each terminal node. These means serve as the predicted values for the observations in their respective terminal nodes. The predicted values are not (and cannot be) membership in a particular class.
4. Then, random forest averages in much the way as it does for classification problems. For a given observation, the average of the tree-by-tree predicted values is computed using only the predicted values from trees in which that observation was not used to build the tree. This is the predicted value that random forest returns. Then the deviations between these over-tree predicted values and the observed values are used to construct the mean square error reported for the collection of trees that constitutes a random forest. The value of the mean square error can be used to compute a pseudo R^2 as $(1 - \mathrm{MSE})/\mathrm{Var}(Y)$.
5. Construction of partial dependence plots is done in the same manner as for classification trees, but now the fitted response is the set of conditional means for different predictor values, not a set of transformed fitted proportions.
6. Importance is computed using the shuffling approach as before. And as before there is a "resubstitution" measure and a forecasting measure. For the resubstitution measure, consider a single tree. Each time a given variable is used to define a partitioning of the data, the reduction in the within-node error sum of squares is recorded. When the tree is complete,

the reductions are summed. The result is the error sum of squares that can be attributed to each predictor. These totals, one for each predictor, are then averaged over trees.

7. The forecasting measure uses the OOB observations. For each tree, the OOB observations are used to compute the predicted values and the within-node mean square error around them. Then a given predictor is shuffled, and the OOB predicted values and mean square error computed again. An increase in this mean square error is a decrease in accuracy. These decreases are averaged over trees to get an average decrease in accuracy for that predictor. The standard deviation of these decreases over trees can be used to standardize the average decrease, if that is desirable.

Despite the tight connection between regression trees and random forests, there are a few features found in some implementations of regression trees that have yet to be introduced into random forests. Perhaps most important, random forests is currently limited to the normal regression model. There are, for instance, no accommodations for count data, where some form of Poisson regression might be appropriate. Likewise, there are no accommodations for bounded response variable distributions such as might be found for survival data. However, such generalizations are likely to come soon.

For example, there is a new procedure in R called quantregForest() that computes for each terminal node quantiles of the user's choosing. Instead of storing only the mean of each terminal node as trees are grown, the entire distribution is stored. Recall the earlier discussion surrounding Table 5.3. Once the user decides which quantiles are of interest, they can be easily computed.

If one is worried about the impact of within-node outliers on the conditional mean, the conditional median can be used instead. If for substantive reasons there is interest in the first or third quartile, those can be used. Perhaps most interestingly, the quantile option provides an interesting way to take the costs of forecasting errors into account. For example, if the 75th quantile is chosen, the consequences of underestimates are three times more costly than the consequences of overestimates. However, such calculations only affect what is done with the information contained in the terminal nodes across trees. This approach does not require that the trees themselves be grown again with a linear loss function, let alone a loss function with asymmetric costs. In other words, the trees grown under quadratic loss are not changed. As a result, the quantile adjustments are not complete. An example is discussed later in this chapter.

5.10 Tuning Parameters

Despite the complexity of the random forest algorithm and the large number of potential tuning parameters, most of the usual defaults work well in practice. The tuning parameters most likely to require some manipulation are the following.

1. *Node Size*—Unlike in CART, the number of observations in the terminal nodes of each tree can be very small. The goal is to grow trees with as little bias as possible. The high variance that would result can be tolerated because of the averaging over a large number of trees. In the R implementation of random forests, the default sample sizes for the terminal nodes are one for classification and five for regression. These seem to work well. But one must also keep in mind the concerns raised earlier when there are a large number of predictors weakly related to the response and at least moderately related to each other. If such predictors are not dropped, it is usually wise to grow smaller trees. If one is interested in estimating a quantile, such as in quantile random forests, then terminal node sizes about twice as large will often be necessary. If there are only five observations in a terminal node, for instance, it will be difficult to get a good read on, say, the 90th percentile.
2. *Number of Trees*—The number of trees used to constitute a forest needs to be at least several hundred and probably no more that several thousand. In practice, 500 trees is often a good compromise. It sometimes makes sense to do most of the intitial development (see below) with about 500 trees and then confirm the results with a run using about 3000 trees.
3. *Number of Predictors Sampled*—The number of predictors sampled at each split would seem to be a key tuning parameter that should affect how well random forests performs. Although it may be somewhat surprising, very few predictors need to be randomly sampled at each split, and with sensible bounds on the number sampled, it does not seem to matter much for the OOB error estimates. With a large number of trees, each predictor will have an ample opportunity to contribute, even if very few are drawn for each split. For example, if the average tree in a random forest has ten terminal splits, and if there are 500 trees in the random forest, there will be 5000 chances for predictors to weigh in. Sampling two or three each time should then be adequate.

   But a lot depends on the number of predictors and whether all have good potential or whether some do and some don't. In the manual for the FORTRAN version of random forests, Breiman recommends starting with the number of predictors sampled equal to the square root of the number of predictors available. Then, trying a few more or a few less as well can be instructive.

   In the R implementation of random forests, one can search for the best number of predictors to sample using the OOB error statistic as a criterion. This is an excellent tool in principle. In practice, large differences in performance are rarely found. Also, one must be careful not to overtune and introduce the overfitting that random forests is designed to prevent.

The feature of random forests that will usually make the biggest difference in the results is how the costs of false negatives and false positives are handled.

These costs have already been extensively discussed and are not reconsidered now. At the same time, costs are not really a tuning parameter, but a key aspect of how the data are to be analyzed.

5.11 An Illustration Using a Binary Response Variable

Industrialized fishing is dramatically reducing the stock of predatory fish throughout the oceans of the world. Large-scale commercial fishing affects not just the target species but other species that become the “bycatch.” The impact on dolphin populations of commercial fishing for tuna is perhaps the most visible illustration and has been the subject of a National Research Council committee report (Committee on Reducing Porpoise Mortality from Tuna Fishing, 1992). Over the past decade, international cooperation to reduce dolphin mortality has led to efforts to monitor tuna fishing practices and penalize offenders. The political and technical issues are very complex.

Dolphin are put at risk in tuna fishing because they are often used to locate large schools of tuna. For reasons that are not fully understood, dolphin are often found swimming above schools of tuna and because the dolphin typically swim close to the surface, they can be seen by fishermen some distance away. Then, when large nets are deployed to catch the tuna, the dolphin can be caught as well. Over the past two decades fishing technology and procedures have been changed so that dolphin mortality can be dramatically reduced, but the mortality is far from zero.

The Inter-American Tropical Tuna Commission (ITTC), which oversees the international purse-seine fishery for tuna in the eastern Pacific Ocean, has provided data on dolphin mortality. The dataset includes over 100,000 observations. An observation is a “set,” defined as placing a large net into the water to encircle a school of tuna. There are over 200 predictors. Here, the intent is to determine the circumstances under which dolphin mortality is likely to be high.

For example, a major cause of dolphin deaths apparently is whether the net “collapses” as it is drawn to the boat. A net collapse is a relatively rare event, but one to be actively avoided. For similar reasons, it would be helpful to learn which other predictors are associated with dolphin mortality so that preventive actions might be taken by fishermen.

Sanctions can be applied to fishermen if any dolphin are killed. There is zero tolerance for any dolphin mortality. This suggests treating the response as a binary outcome: whether any dolphin are killed or not. And this is how we proceed here.

For this illustration, we use the predictors listed below. In discussions with the ITCC, these predictors were singled out as by far the most promising. We could have included well over 100 predictors, but some of the graphics would have been unnecessarily cluttered and difficult to explain.

1. capskill: Captain skill coded "0 for less than 30 dolphin sets/year and "1" for 30 or more.
2. biomass: The number of animals encircled in the net.
3. cwtunay: Catch weight of yellowfin tuna in metric tons.
4. cwtunao: Catch weight of other tuna in metric tons.
5. encircle: Duration of the encirclement phase of the set in decimal hours.
6. netretrieval: Duration of the prebackdown net retrieval in decimal hours.
7. backdown: Duration of the backdown procedure in decimal hours.
8. netcanopy: Coded "1" if a net canopy present and "0" if a net canopy is not present. Net canopies are associated with collapsed nets.
9. diver: Coded "1" if divers were used to help dolphin escape and "0" if not.

Using a random sample of 10,000 observations, we consider first the results under the default costs. A failure to identify correctly a set in which dolphin were killed has the same costs as a failure to identify correctly a set in which no dolphin were killed. In addition, the prior used is the empirical distribution of the binary response variable.

	Predict No deaths	Predict deaths	Model Error
No Deaths	7859	142	.02
Deaths	797	202	.78
Use Error	.09	.41	Overall Error = .10

Table 5.5. Confusion table for forecasting dolphin deaths using equal costs.

Table 5.5 shows the confusion table. Overall, random forests is able to forecast with about 90% accuracy. Given the unbalanced nature of the response, this is not a very impressive feat. It is clear that most of this accuracy comes from the predictions of true negatives (i.e., no dolphin deaths), which random forests incorrectly identifies only about 2 times out of 100. About 78 times out of 100, random forests incorrectly identifies true positives (i.e., dolphin deaths). Were the random forest results used for forecasting by ITTC administrators, ship captains or on-board observers, they would be wrong only about 9 times out of 100 when they forecasted no dolphin deaths, but about 41 times out of 100 when they forecasting dolphin deaths.

Figure 5.15 shows a plot of predictor importance for the response category in which dolphin were killed. It is clear that the presence of a net canopy is the dominant predictor, followed by the length of the backdown procedure, and then two measures of the size of the tuna catch. Given that random forests identifies correctly on 22% of the time sets in which dolphin are killed, the accuracy reductions are large. Shuffling the net canopy variable reduces the model's forecasting accuracy from .22 to .15.

Figure 5.16 shows a plot of predictor importance for the response category in which no dolphin are killed. It is here that random forests stumbles badly.

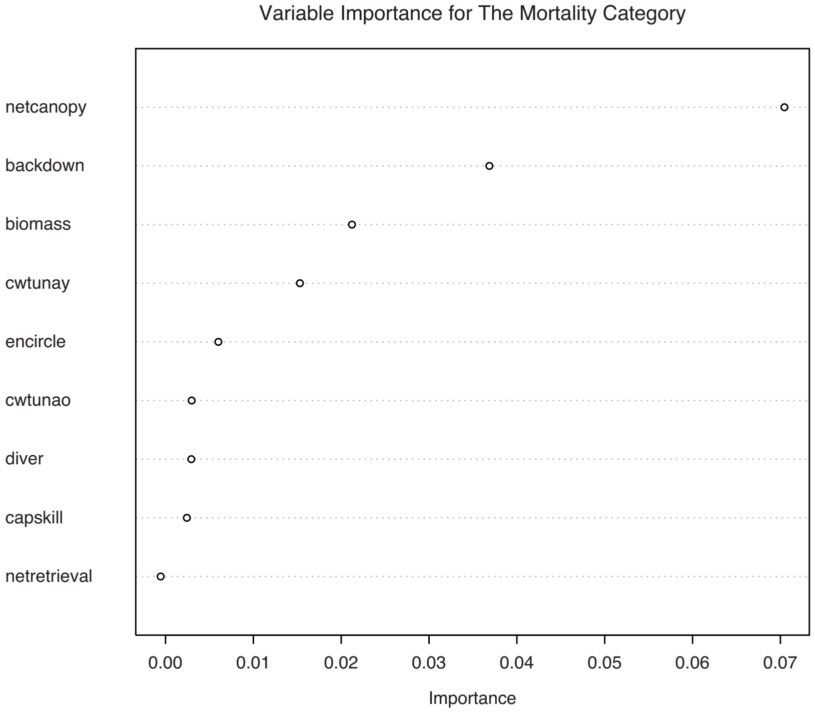


Fig. 5.15. Variable importance when the outcome is dolphin deaths and the costs are equal.

In general the same predictors are important, but their contributions to forecasting accuracy are trivial. A key reason is that it is very difficult for random forests to do better than always concluding that no dolphin were killed.

Partial dependence plots can be constructed for each of the predictors. We will consider just one to illustrate the sorts of insights that can be obtained and to highlight some important limitations. Thus, Figure 5.17 shows the empirical response function for the predictor backdown time. Backdown time is how long it takes for the net, once the tuna are encircled, to be drawn to the boat. Long backdown times are thought to be dangerous for dolphin because they increase the risk of dolphin getting caught in the net and drowning.

Figure 5.17 suggests that for very short backdown times, which are rare and probably reflect serious reporting errors, increases in backdown time are associated with decreases in dolphin mortality. This is a result that should not be taken seriously. For backdown times between about 15 minutes and an hour, increases in backdown time are, as expected, associated with substantial increases in dolphin mortality. Beyond an hour, where again the number of

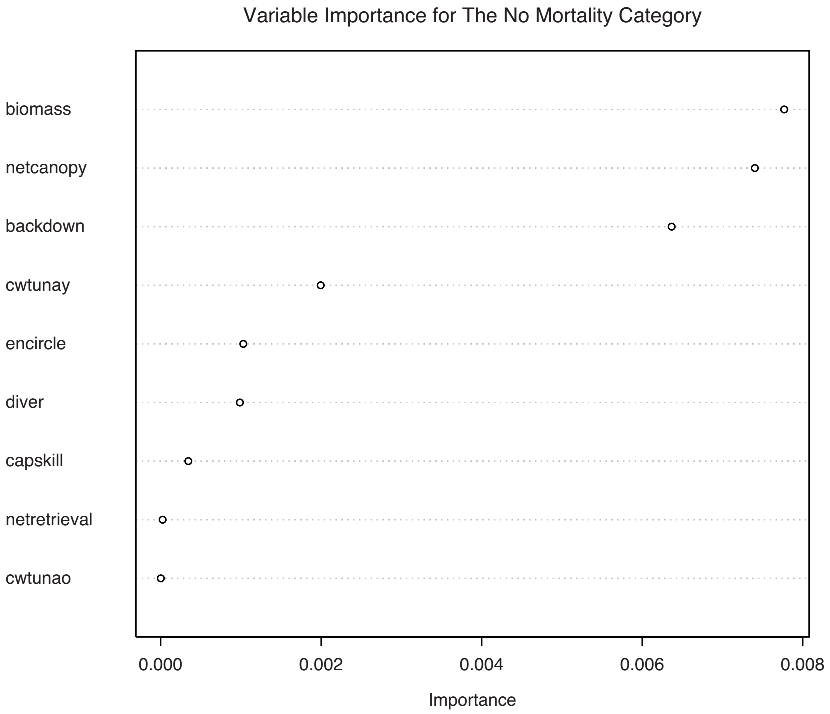


Fig. 5.16. Variable importance when the outcome is no dolphin deaths and the costs are equal.

observations is very few, the relationship essentially become flat. This too should not be taken seriously.

Recall that the units on the vertical axis are not probabilities or conventional logits. Thus, it is difficult to judge in subject matter terms whether the changes in the response shown are large enough to be important. But as shown earlier, the logit units can be transformed back into probabilities, and for Figure 5.17, the change in the logits is large enough to be very significant. When the backdown time is under 20 minutes, the chances of any dolphin deaths are less than 1 in 20 below what is typical. When the backdown time is over 40 minutes, the chances of death are around 15 in 20 above what is typical. So, the probabilities are increased a maximum of about .70.

From the off-diagonal cells in Table 5.5, one can see that there are a little over five false negatives for every false positive. Therefore, false positives are being treated as about five times more costly than false negatives. Discussion with representatives of the ITTC led to the conclusion that false negatives were actually much more costly than false positives. There were few harmful

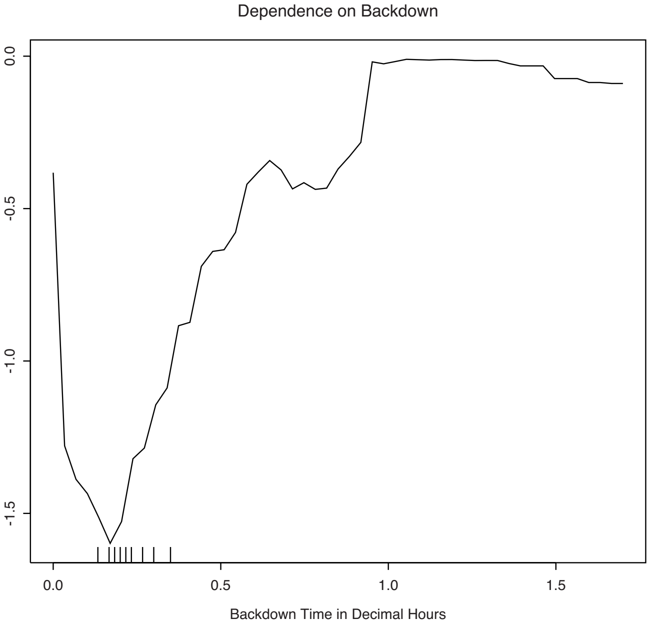


Fig. 5.17. Partial dependence on backdown time when the costs are equal.

consequences from failing to identify sets with no dolphin mortality, but many from failing to identify sets in which dolphin were killed. They suggested a ratio of about one to ten for the ratio of false negatives to false positives.

	Predict No Deaths	Predict Deaths	Model Error
No Deaths	5468	2533	.32
Deaths	271	728	.27
Use Error	.04	.78	Overall Error = .31

Table 5.6. Confusion table for forecasting dolphin deaths using a cost ratio of one to ten.

Table 5.6 shows the confusion table when the one to ten cost ratio is used. The changes are substantial. Just as one would expect, overall forecasting error increases substantially from .10 to .31.

Consistent with the cost applied, random forests now does a much better job identifying sets in which dolphin are killed. The proportion of sets incorrectly identified drops from .78 to .27. At the same time, random forests does much worse identifying sets in which dolphin are not killed. The proportion of sets incorrectly identified increases from .02 to .32. These changes stem from the new ratio of false negatives to false positives, which by intent is about one to ten.

If forecasts of no dolphin deaths are made, they will be incorrect about 4 times out of 100. If forecasts of dolphin deaths are made they will be incorrect about 78 times out of 100. This properly reflects the new cost ratio. Decision-makers are now more prepared to predict sets in which dolphin will be killed because the costs of false positives are relatively low. Table 5.6 indicates that there will be a bit more than three false positives for every true positive.

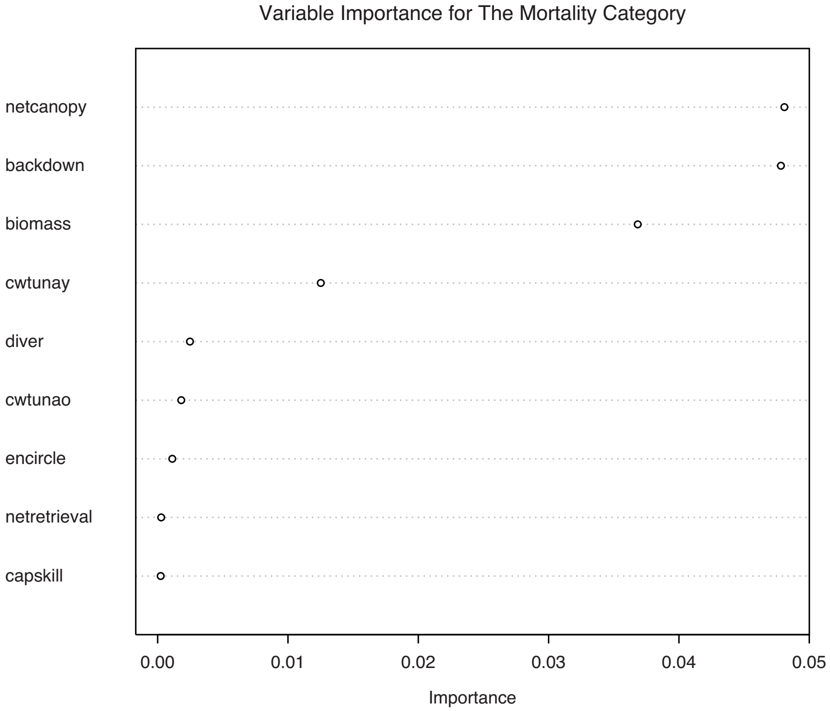


Fig. 5.18. Variable importance when the outcome is dolphin deaths and the costs are one to ten.

Figures 5.18 and 5.19 show the two importance plots. It is again apparent that the predictors matter far more for forecasts of dolphin deaths than for forecasts of no dolphin deaths. But the change in a cost ratio of one to ten has altered a bit the importance of some variables. For example, in predictions of dolphin deaths, the presence of a net canopy and backdown time are now about equally important. Under equal costs, backdown time was a little less

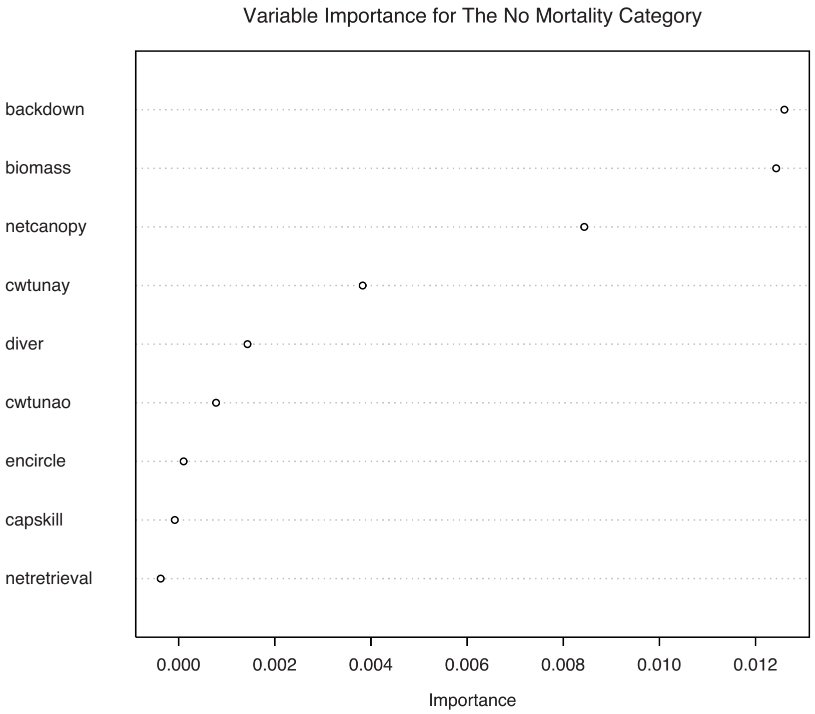


Fig. 5.19. Variable importance when the outcome is no dolphin deaths and the costs are one to ten.

important. And under the new cost ratio, the biomass of the tuna caught has moved up somewhat in importance.

It is not unusual to see the importance of variables change with changes in relative costs. In effect, there is a new weighting of observations. Variables that predict well the observations now given more weight will increase in importance.

Finally, Figure 5.20 shows the partial dependence plot for backdown time under the one to ten cost ratio. The overall shape of the curve is basically the same, but the increase in dolphin mortality is not quite as large.

5.12 An Illustration Using a Quantitative Response Variable

A recent effort was made to count the number of homeless in Los Angeles County (Berk et al., 2008). There are over 2000 census tracts in the county, and enumerators were sent to a sample of a little over 500. The details of the

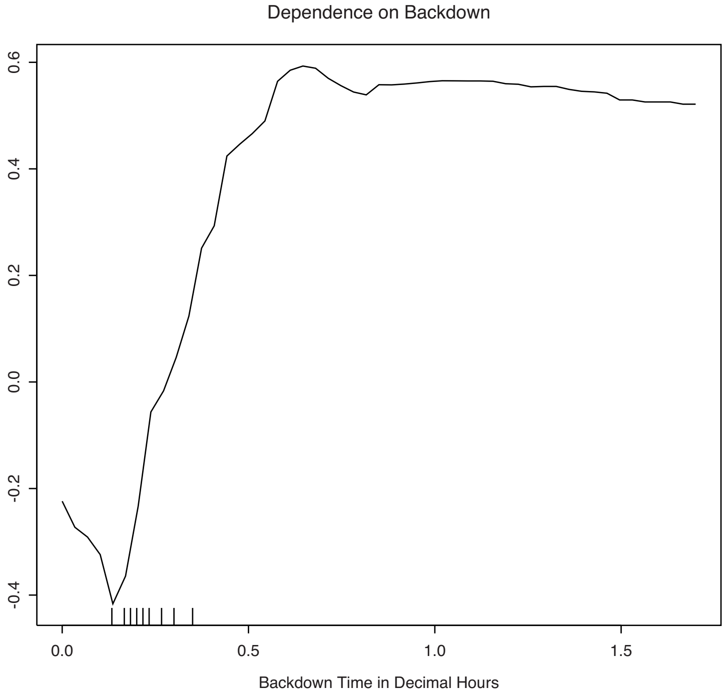


Fig. 5.20. Partial dependence on backdown time when the costs are one to ten

sampling need not trouble us here, and in the end the overall county total was estimated to be about 90,000.

In addition to countywide totals, there was a need to have estimated counts for tracts not visited. Various stakeholders might wish to have estimates at the tract level for areas to which enumerators were not sent. Random forests was used with tract-level predictors to impute the homeless counts for these tracts. About 21% of the variance in the homeless counts was accounted for by the random forests model.

Figure 5.21 is an importance plot for three of the most useful predictors. When the response variable is quantitative, the "external" and "internal" measures of importance differ from when the response variable is qualitative. The external measure, based on the OOB observations, is the average percentage increase in mean square forecasting error over trees when a given predictor is randomly shuffled. The internal measure is the average reduction in the error sum of squares over trees when a given predictor is used to define a split, which can also be called the increase in node purity.

For example, when median household income is shuffled, the mean square forecasting error increases about 7%. For the percentage of dwellings in a tract

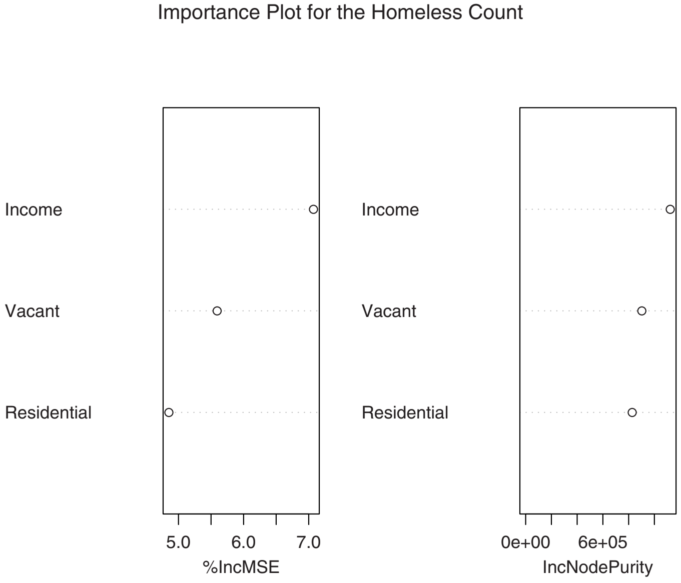


Fig. 5.21. Variable importance when the outcome is the number of homeless in a census tract.

that is vacant, shuffling increases forecasting error about 6%. For the percentage of land that is devoted to residential use, shuffling increases forecasting error about 5%. In this case, therefore, all three variables have about the same impact on forecasting skill. The rank ordering of importance is the same when the contribution to fit is used, but it is far more difficult to tell whether the contributions are large or small. It is difficult to think in raw error sum of squares units.

Figures 5.22 to 5.24 show the partial dependence plots for each predictor. For a quantitative response, the vertical axis is the conditional mean of the response for different values of the predictor in question, with all other variables fixed. Compared to the categorical response variable case, the only feature of the partial dependence algorithm that has changed is the units in which the response is represented.

From Figure 5.22, one can see that the fitted values for the number of homeless individuals in a census tract drops from a high of around 150 when median income is less than about $20,000 a year to around 30 when median income is $50,000 or more. Overall, the relationship is strongly negative. But the drop is precipitous, implying what some have called a “tipping effect.”

The visual story is much the same in Figure 5.23. When most of the land in a census tract is not used for residential dwellings, the number of homeless individuals is about 130. That figure drops to about 30 when a quarter of

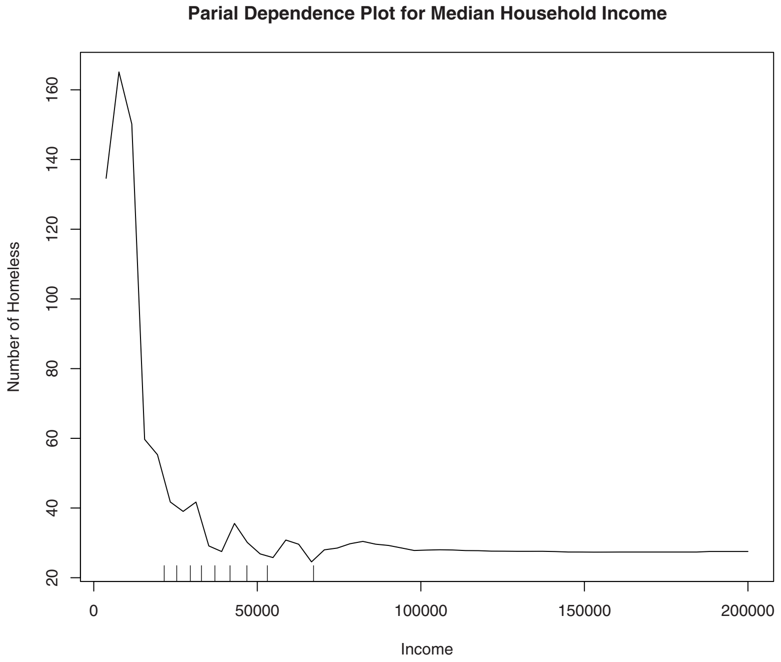


Fig. 5.22. The response function when median household income is the predictor.

the land or more is used for residents. Overall, the relationship is strongly negative with more evidence of a tipping effect.

Figure 5.24 shows a positive association between the percentage of the residential dwellings that are vacant and the number of homeless. When vacancy is near zero, the average number of homeless is about 10 per tract. When the vacancy percent is above approximately 10%, the average count increases to between 60 and 70 (with a spike right around 10%). Once again the change is very rapid.

In summary, a larger number of homeless are to be found in low income census tracts with relatively few occupied dwellings. Perhaps more interesting is that the transition from tracts with few homeless individuals to tracts with many homeless individuals occurs over a very small range of predictor values. An important methodological point is that the highly nonlinear response functions would not likely have been found using conventional regression procedures unless there were a strong a priori belief in a tipping effect and the the ability to specify a functional form that would find it. Given the sharp transition for predictor values that would have been difficult to anticipate, determining the functional form would have probably been difficult.

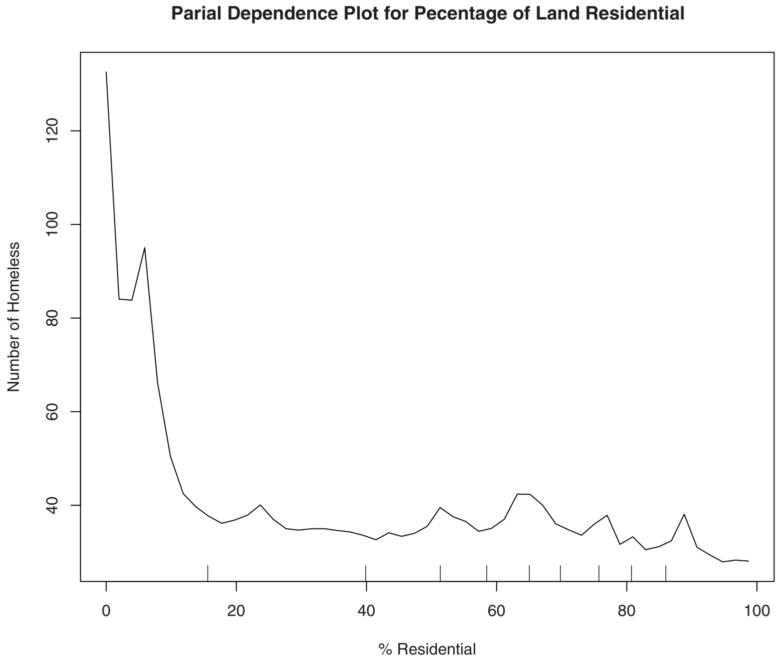


Fig. 5.23. The response function when residential land use is the predictor.

Unfortunately, even with the power of random forests, there is a marked tendency to underestimate the few very largest homeless counts. These census tracts matter a great deal in the overall amount of resources allocated to take care of homeless individuals and how those resources are allocated. As suggested earlier, this is precisely where quantiles might be more instructive than the mean.

Figure 5.25 shows a plot of the actual census tract counts against the fitted census tract counts using the conditional .05 quantile. Overestimates are being treated as far more important than underestimates: about 19 to 1. A 1-to-1 line is overlaid.

The mean absolute disparity between the fitted values and the actual values is 29.4. This is quite large considering that most of the measured homeless counts are under 50. Ideally, moreover, all of the points should fall to the 1-to-1 line. Most of the points fall above the 1-to-1 line, indicating underestimated counts much of the time. Finally, one can see that although the actual counts are sometimes larger than 400, the largest fitted count is a little over 80.

Figure 5.26 shows a plot of the actual census tract counts against the fitted census tract counts using the conditional .50 quantile: the median. Using the

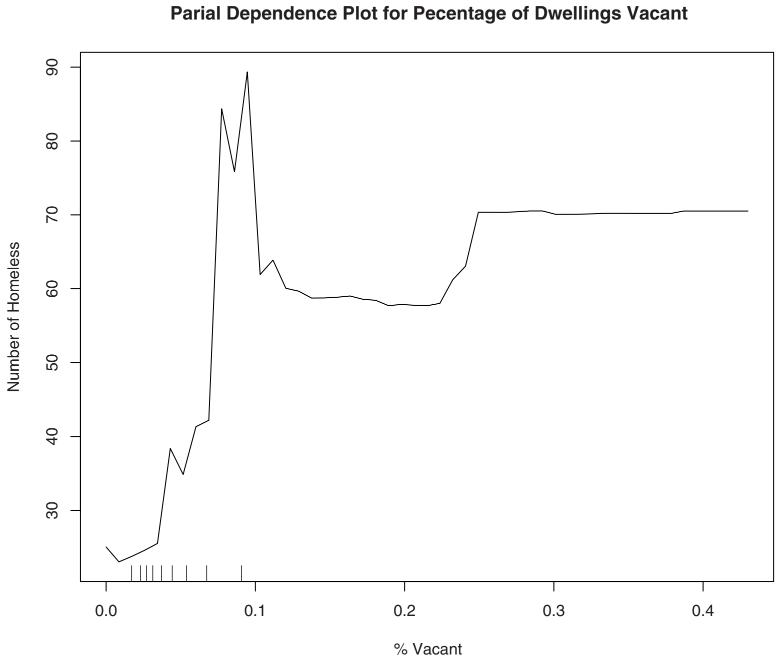


Fig. 5.24. The response function when vacant dwellings is the predictor.

median implies that the costs of overestimates are the same as the costs of underestimates. Again, a 1-to-1 line is overlaid.

Overall the fit looks quite good. The mean absolute disparity is only 3.5. In addition, the fitted counts can now be as large as about 450, which is a clear improvement if large underestimates are a serious concern. However, the very largest counts are still substantially underestimated. Allowing the overestimates and underestimates to have the same costs produces results much like those produced by the conditional mean.

Figure 5.27 shows a plot of the actual census tract counts against the fitted census tract counts using the conditional .95 quantile. Now the costs of underestimates are 19 times larger than the costs of overestimates. A 1-to-1 line is again overlaid.

Virtually all of the points fall below the 1-to-1 line, and the mean absolute disparity is 36.0. Overestimates dominate the plot. There are several fitted counts in excess of 800, which in most cases are also overestimates. On the other hand, the very highest count is fitted perfectly. Clearly, one can have very different fitted values depending on which quantiles are used.

Much as in our earlier discussion, there is no purely statistical way to determine what costs should be used. The costs need to be determined by how

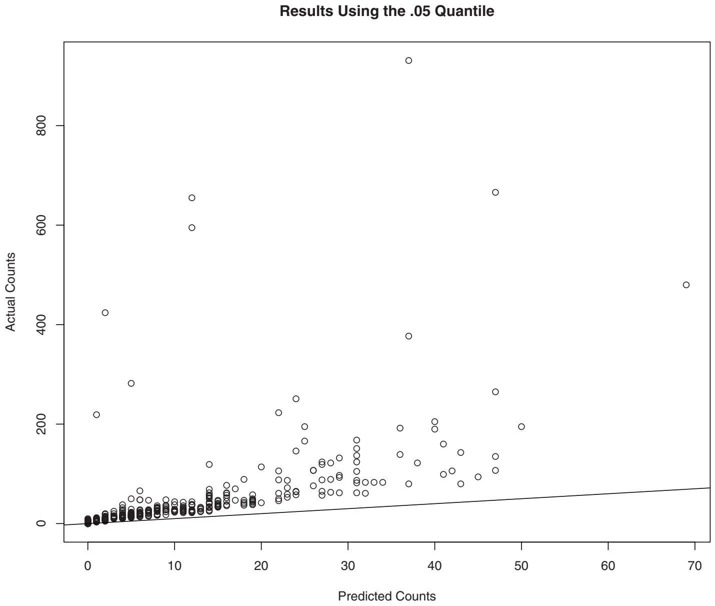


Fig. 5.25. Actual counts plotted against fitted counts using the .05 quantile: MAD = 29.4.

the results are to be used and by how various stakeholders view the consequences of those uses. For example, although using the conditional median minimizes the sum of the absolute deviations between the actual values and the fitted values, the mean absolute deviation assumes that underestimates have the same costs as overestimates. As such, it will be a misleading measure of fit when equal costs do not apply.

Figure 5.28 shows the results when the conditional .80 quantile is used. Underestimates are taken to be four times more costly than overestimates. Stakeholders might find these results the most congenial. The mean absolute disparity of 7.4 is relatively small, and the very largest counts are fitted values about as well as possible, given their variability. For our purposes, however, the statistical point is that quantile regression provides a way to employ asymmetric cost functions with random forests.

Quantile random forests is hardly the final answer to the need for asymmetric cost functions in statistical learning for quantitative response variables. As noted earlier, the trees are grown as usual using the random forests algorithm. Also, one has to be happy with linear loss. Finally, the importance plots and partial dependence plots currently available in the quantile random forests procedure are still those from the underlying random forest algorithm.

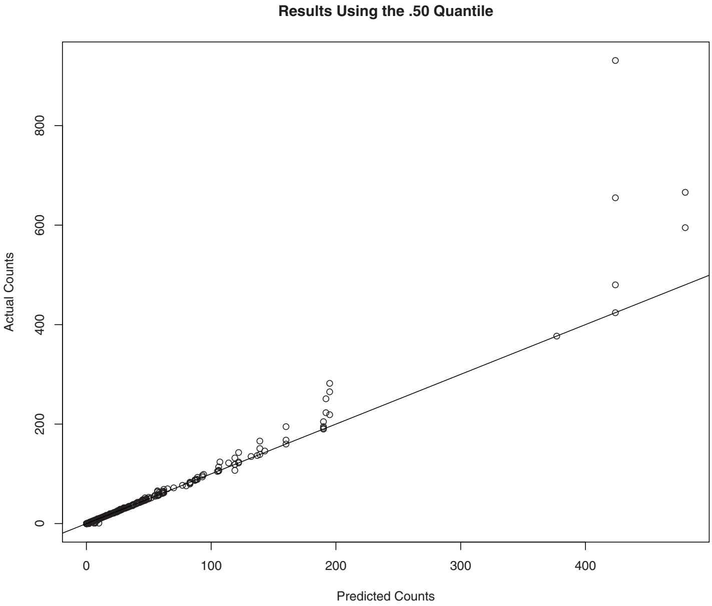


Fig. 5.26. Actual counts plotted against fitted counts using the .50 quantile: MAD = 3.5.

5.13 Software Considerations

At the moment, there are three species of random forests available. One can obtain from Leo Breiman's Web site (http://stat-www.berkeley.edu/users/breiman/RandomForests/) a FORTRAN version of random forests and some supporting documentation. Leo Breiman and Adele Cutler are the software's authors. There are some features that are otherwise not available and some features available elsewhere that are not included. It is perhaps the least user-friendly of the three.

The version of random forests available in R is far more user friendly, has better documentation, and has the key advantage of an R computing environment. It has features that are unique but lacks some of the highly experimental tools found in the FORTRAN version. The R port was undertaken by Andy Liaw and Matthew Weiner. Andy Liaw (andy_liaw@merck.com) is the maintainer.

Quantile random forests, which draws so heavily on conventional random forests, is also an R-based procedure. Quantile random forests (quantregForest) was written and is maintained by Nicolai Meinshausen.

The version of random forests available from Salford Systems (http://www.salford-systems.com/) is by far the most user-friendly. But, the user

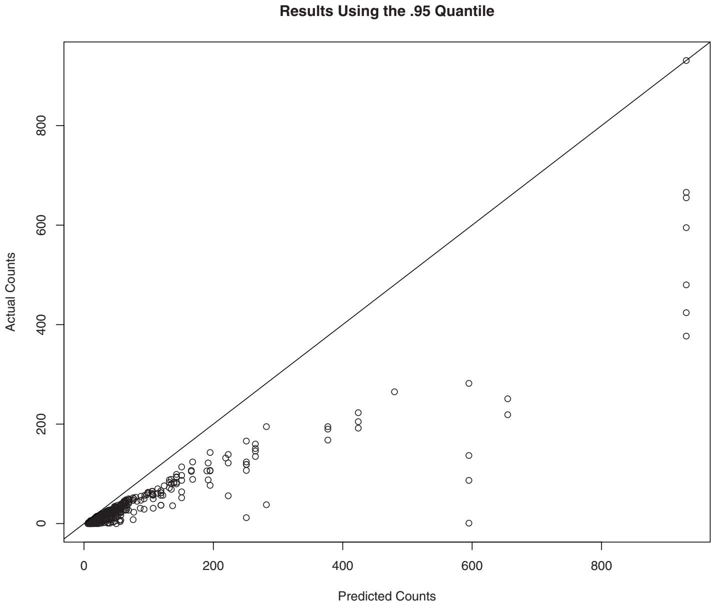


Fig. 5.27. Actual counts plotted against fitted counts using the .95 quantile: MAD = 36.0.

gives up considerable control and in general, the features included are several iterations behind those that can be found in the R version. And unlike the FORTRAN and R implementations, there is a substantial charge for the software.

It is important to stress that random forests is a new procedure still very much under development. Although it is unlikely that its main algorithm will change significantly, lots of the special routines and displays of output will. If history is any measure, one can expect significant updates of random forests at least once a year, often sooner. And unfortunately, it is often difficult for the documentation to keep up. There are also likely to be spinoffs from random forests, such as quantile random forests and others. One can imagine some of these being very handy for certain kinds of problems.

For example, Geurts and his colleagues (2006) propose what they call "extremely randomized trees." No bootstrap sample is used; the full training sample is used to grow each tree. Then the algorithm proceeds as follows.

1. For each potential partitioning, choose a random sample of predictors without replacement.
2. For these selected predictors, choose the break points at random.
3. Compute the reduction in heterogeneity for each predictor at its randomly chosen break point.

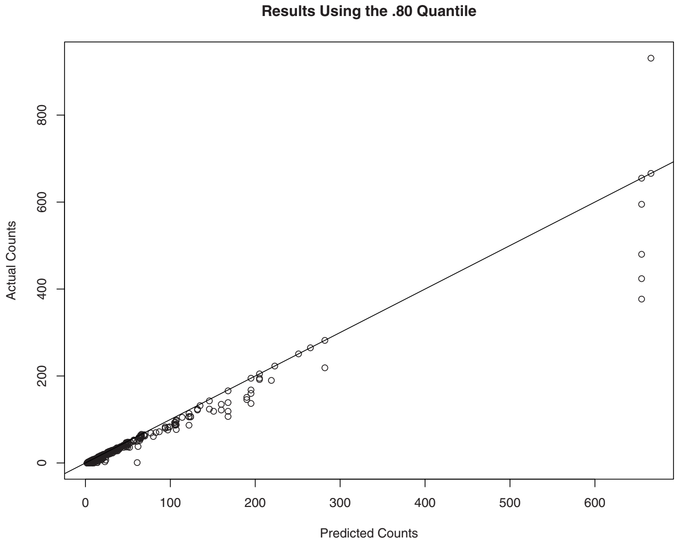


Fig. 5.28. Actual counts plotted against fitted counts using the .80 quantile: MAD = 7.4.

4. Choose the predictor that reduces the heterogeneity the most.
5. Repeat Steps 1–4 for each subsequent split.
6. Average over trees as usual.

The underlying rationale is that by selecting break points at random, greater independence is achieved across the trees in a forest compared to conventional random forests. That is, the sets of fitted values will be less dependent. As a result, instability is more effectively controlled. The price for this reduction in instability can be an increase in the bias because the random break points are not likely to be the optimal break points. Ideally, the reduction in the instability will more than offset the increase in the bias.

When the set of predictors is weak, extremely randomized trees may perform at least as well as random forests; the tradeoff works because random break points do not perform much worse than optimal break points. When the predictors are not weak, extremely randomized trees is not likely be a good choice.

There will also in the future likely be important improvements in how the results from random forests are visualized. For example, it might be useful to have a receiver operating characteristic (ROC) curve that would depend on such things as the relative costs for false negatives to false positives or the values of tuning parameters. Such a plot would have the number of true positives on the vertical axis and the number of false positives on the horizontal axis. The best result would fall in the upper-left hand corner: all true positives and no false positives. The worst result would be in the lower-right hand corner: no true positives and all false positives. Locations between these extremes would represent different tradeoffs between the two, and one could see how these tradeoffs changed with alterations of the model's features. Drummond and Holte (2006) have suggested an interesting alternative, consistent with much of the earlier discussion, in which on the vertical axis is the (normalized) expected total cost of the classification errors. This too could come in handy.

5.14 Summary and Conclusions

There is growing evidence that random forests is a very powerful statistical learning tool. If forecasting accuracy is one's main performance criterion, there are no other tools that have been shown to consistently perform any better. We consider a chief competitor in the next chapter.

Random forests seems to get its leverage from five features of the algorithm:

1. Growing large, low bias trees
2. Using bootstrap samples as training data when each tree is grown
3. Using random samples of predictors for each partitioning of the data
4. Constructing fitted values and output summary statistics from the out-of-bag data
5. Averaging over trees.

At the same time, very few of random forest's formal properties have been proven, and there remains the nettlesome problem that if one is interested in knowing the $f(X)$, a random forest estimate is not consistent. At a deeper level, the precise reasons why random forests performs so well and why it does better with some datasets than others is fully understood. There is some hard work ahead for theoretical statisticians.

Exercises

5.14.1 Problem Set 1

The goal of this first exercise is to compare the performance of linear regression, CART, and random forests. Construct the following dataset in which the response is a quadratic function of a single predictor.

```
x1=rnorm(500)
x12=x1^2
y=1+(-5*x12)+(5*rnorm(500))
```

1. Plot the $1 + (-5 \times \text{x12})$ 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 relationship between the response and the predictor is quadratic. Fit a linear model with x12 as the predictor. Then plot the fitted values against x1. The results show how the linear model can perform when you know the correct function form.

3. Now suppose you do not know that the relationship between the response and the predictor is quadratic. Apply CART to the same response variable using rpart() 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 CART fitted values compare to the linear regression fitted values? How well does CART seem to capture the true $f(X)$?

4. Apply random forests to the same response variable using randomForests() 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 random forest fitted values compare to the linear regression fitted values? How well does random forests seem to capture the true $f(X)$?

5. How do the fitted values from CART compare to the fitted values from random forests? What feature of random forests is highlighted?

6. Construct a partial dependence plot with x1 as the predictor. How well does the plot seem to capture the true $f(X)$?

7. Why in this case does the plot of the random forest fitted values and the partial dependence plot look so similar?

5.14.2 Problem Set 2

Load the dataset SLID from the *car* library. Learn about the data set using the help() command. Treat the variable "wages" as the response and all other variables as predictors. The data have some missing values you will want to remove. Try using na.omit().

1. Using the default settings, apply random forests and examine the fit quality.

2. Set the argument *mtry* at 4. Apply random forests again and examine fit quality. What if anything of importance has changed?

3. Now set ntrees at 100 and then at 1000 applying random forests both times. What if anything of importance has changed?

4. Going back to the default settings, apply random forests and examine the variable importance plots with no scaling for each predictor's standard deviation. Explain what is being measured on the horizontal axis on both plots when no scaling for the standard deviation is being used. Interpret both plots. If they do not rank the variables in the same way, why might that be? Now scale the permutation-based measure and reconstruct that plot. Interpret the results. If the ranks of the variables differ from the unscaled plot, why might that be? Focusing on the permutation-based measures (scaled and unscaled) when might it be better to use one rather than the other?

5. Construct partial dependence plots for each predictor and interpret them.

5.14.3 Problem Set 3

Load the *MASS* library and the dataset called Pima.tr. Read about the data using help().

1. Apply random forests to the data using the diagnosis of diabetes as the response. Use all of the predictors and random forest default settings. Study the confusion table.
   a) How accurately does the random forests procedure forecast overall?
   b) How accurately does the random forests procedure forecast each of the two outcomes?
   c) If the results were used to forecast either outcome, what proportions of the time would each of the forecasts be incorrect?

2. Construct variable importance plots for each of the two outcomes. Use the unscaled plots of forecasting accuracy. Compare the two plots.
   a) Which predictors are the three most important in forecasts of the presence of diabetes compared to forecasts of the absence of diabetes? Why might they not be the same?
   b) Why are forecasting contributions for the less common outcome generally larger than the forecasting contributions for the more common outcome?

3. Construct and interpret partial dependence plots of each predictor.

4. Suppose now that medical experts believe that the costs of failing to identify future cases of diabetes are four times larger than the costs of falsely identifying future cases of diabetes. For example, if the medical treatment is to get overweight individuals to lose weight, that would likely be beneficial even if the individuals were not at high risk for diabetes. But failing to prescribe a weight loss program for an overweight individual might be an error with very serious consequences. Repeat the analysis just completed but now taking the costs into account by using the stratified bootstrap sampling option in random forests.

a) How has the confusion table changed?
b) How have the two variable importance plots changed?
c) How have the partial dependence plots changed?