CHAPTER 34

# An Introduction to Statistical Learning from a Regression Perspective

RICHARD BERK

## INTRODUCTION

Statistical learning is a loose collection of procedures in which key features of the final results are determined inductively. There are clear historical links to exploratory data analysis. There are also clear links to techniques in statistics such as principle components analysis, clustering, and smoothing, and to long-standing concerns in computer science, such as pattern recognition and edge detection. But statistical learning would not exist were it not for recent developments in raw computing power, computer algorithms, and theory from statistics, computer science, and applied mathematics. It can be very computer intensive. Extensive discussions of statistical learning can be found in Hastie et al. (2009) and Bishop (2006). Statistical learning is also sometimes called machine learning or reinforcement learning, especially when discussed in the context of computer science.

In this chapter, we consider statistical learning as a form of nonparametric regression analysis.[1] The advantage is that novel concepts can be introduced in a setting that many readers will find familiar. The risk is that the advances that statistical learning represents may not be fully appreciated and that important statistical learning procedures not included within a regression perspective will be overlooked. Yet, in a short overview, this is a useful tradeoff and will provide plenty of material. The approach taken relies heavily on a recent book-length treatment by Berk (2008).

Some background concepts will be considered first. Three important statistical learning procedures are then be discussed in turn: random forests, boosting, and support vector

[1] Statisticians commonly define regression analysis so that the aim is to understand "as far as possible with the available data how the conditional distribution of some response variable varies across subpopulations determined by the possible values of the predictor or predictors" (Cook and Weisberg 1999: 27). Interest centers on the distribution of the response variable conditioning on one or more predictors. Often the conditional mean is the key parameter.

A.R. Piquero and D. Weisburd (eds.), *Handbook of Quantitative Criminology*,
DOI 10.1007/978-0-387-77650-7_34, © Springer Science + Business Media, LLC 2010

machines. Although the exposition makes little use of formal mathematical expressions, some important ideas from mathematical statistics are necessarily assumed. The chapter also necessarily assumes some prior exposure to smoothers and classification and regression trees (CART), and a solid background in the generalized linear model. An attempt is made in several footnotes to provide additional didactic material.

## THE BASIC FRAMEWORK

An important, although somewhat fuzzy, distinction is sometimes made between a confirmatory data analysis and an exploratory data analysis. For a confirmatory data analysis, the form of the statistical model is determined before looking at the data. All that remains is to estimate the values of some key parameters. For an exploratory data analysis, there is no model. Statistical tools are used to extract patterns in the data. The distinction is fuzzy because in practice, few statistical analyses are truly confirmatory. More typically, some important parts of the model are unknown before the data are examined. For example, from a large set of potential explanatory variables, a subset may be selected for the "final model" through a procedure such as stepwise regression.

It is sensible, therefore, to think about a continuum from confirmatory to exploratory analyses. This is especially important for an introduction to statistical learning. Let's begin at the confirmatory side of the continuum.

### Confirmatory Data Analysis

The poster child for confirmatory data analysis is the ubiquitous linear regression model, which takes the following form.

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_{pi} + \varepsilon_i, \tag{34.1}$$

where $i$ is the index for each of $N$ cases, $Y_i$ is the quantitative response variable, $X_1$ through $X_p$ are the $p$ well-measured predictors, $\beta_0$ through $\beta_p$ are the $p+1$ the regression parameters, and $\varepsilon_i$ is a random disturbance. Each $\varepsilon_i$ is assumed to behave as if drawn independently of the predictors and of one another from a distribution with a mean of 0 and a variance of $\sigma^2$. As a model, (34.1) is a very explicit theory about how the response variable is generated. If a researcher believes that theory, data can be used to estimate the values of the regression coefficients in an unbiased manner. The value of $\sigma^2$ can also be properly estimated. Statistical inference can naturally follow.

If one assumes in addition that the disturbances in (34.1) behave as if drawn at random from a normal distribution (34.1) becomes the normal regression special case of the generalized linear model (McCullagh and Nelder 1989). Logistic regression, probit regression, and Poisson regression are other special cases that depend on the nature of the response variable, how the response variable can be transformed, and the distribution assumed for the disturbances. Then as before, the model is known up to the values of the regression coefficients. Estimates of those parameters are obtained from data. Statistical inference can follow as a matter of course.

Equation (34.1) and its generalizations may be interpreted as causal models. One needs to assert that each of the predictors can be independently manipulated. Then, each regression coefficient coveys how much on the average the response variable changes when its associated predictor is changed by one unit. Equation (34.1) is absolutely silent on whether such casual interpretations are appropriate (Freedman 2004). Such claims must be justified with information outside of the model, usually through social science theory, and occasionally because a real experiment was undertaken.

## Confirmatory Data Analysis with Some Important Exploratory Components

In a wide variety of applications in criminology and other social sciences, there is a substantial disconnect between (34.1) and how the data were actually generated (Berk 2003; Freedman 2005; Morgan and Winship 2007). The model is substantially wrong. The same problem can arise for the entire generalized linear model. Although this is old news (e.g., Holland 1986), practice has changed slowly.

There have been many suggestions about the way one might tinker with models like (34.1). Transforming the response and/or the predictors, winnowing down the set of predictors, and removing outlier observations are examples (Cook and Weisberg 1999). However, these methods are commonly neglected and are often too little and too late in any case (Berk 2003). For instance, procedures that are meant to provide the proper transformation for $Y_i$ require that all other features of the model are effectively correct. This is a very demanding and often unrealistic requirement.

A bolder step is to make the systematic part of (34.1) far more flexible. One can rewrite (34.1) as

$$Y_i = f(\mathbf{X}_i) + \varepsilon_i, \tag{34.2}$$

where $\mathbf{X}$ is an $N \times p$ matrix of predictors, and $f(\mathbf{X}_i)$ represents how $\mathbf{X}_i$ is related to $Y_i$. Typically, $\varepsilon_i$ is assumed to have the same properties as in (34.1).

The key point is that the $f(\mathbf{X}_i)$ is determined empirically from the data. In other words, the researcher supplies the correct set of well measured predictors, and a computer does the rest. No particular functional forms need to be assumed. The approach is sometimes called function estimation. Just like (34.1), (34.2) can be altered to include the full generalized linear model, but with the $f(\mathbf{X}_i)$ determined by the data.

Equation (34.2), may also be interpreted as a causal model. However, it differs from (34.1) and its generalizations in that there are no longer regression coefficients representing causal effects. Casual effects have to be represented in other ways. We consider this matter later. But, the same basic requirements and concerns apply.

## Exploratory Data Analysis

For many applications, the step taken from (34.1) to (34.2) is not bold enough. For example, several important predictors may not be in the data set or may be in the data set but poorly measured. Then, assumptions made about $\varepsilon_i$ can be unreasonable, or at least not easily justified. And, because $\varepsilon_i$ is unobservable, it is difficult to bring data to bear.

It often makes sense, therefore, to treat (34.1) or (34.2) as descriptions of how in the data on hand, the conditional mean of the response differs depending on the values of each predictor. There is no model stating how the data were generated. Treating regression analysis solely as a descriptive analysis is always formally correct.

The enterprise is now exploratory data analysis. The enterprise is also fully consistent with the definition of a regression analysis. There is nothing in the definition of a regression analysis that requires a model of how the data were generated, let alone a causal model. Further discussion of using regression analysis for description can be found in Berk (2003).

## A Few More Definitions

Equation (34.2) represents the kind of statistical learning we will consider. It can be used for function estimation or description, and is called "supervised learning" because there is a response variable. When there is no response variable – one only has predictors – the name is "unsupervised learning.[2]" Unsupervised learning will not be considered in the chapter. Because the most visible and early statistical learning methods built on many passes through the data, "learning" became a metaphor for how the fitted values can improve as the data are revisited.[3]

There are a number of statistical procedures consistent with (34.2) that are not usually considered statistical learning. Smoothers are perhaps the most common example. Classification and regression trees is another instance. As statisticians and computer scientists have examined the properties of statistical learning procedures, however, many earlier approaches have been recast as special cases of statistical learning. Consequently, features of statistical learning once thought to be distinctive are now seen as matters of degree. This chapter will emphasize the more recent and more novel developments that a look to be the most promising for practice. But, there will not be a clear boundary between statistical learning and a number of earlier techniques. A comprehensive examination can be found in the fine book by Hastie et al. (2009).

## Statistical Inference

Because there is no model before the data are analyzed, there cannot be any formal statistical inference applied to the $f(\mathbf{X}_i)$ that results. If statistical inference is forced on some feature of the output nevertheless, it is unlikely that the sampling distribution will be known, even asymptotically. This is the same problem that is associated with all model selection procedure (Leeb and Pötscher 2006).

As key problem is that because the definition of a regression parameter depends on the particular model in which it is embedded, and because that model is unknown, conventional statistical criteria such as unbiasedness and consistency are not defined. Moreover, the process of model selection is a form of data snooping, which is well known to compromise statistical tests and confidence intervals. There are solutions to these problems, but they are only practical in certain situations (Berk et al. 2009b) and are beyond the score of this chapter in any case. At this point, statistical inference is usually a suspect and secondary aspect of statistical learning.

[2] Under these circumstances, statistical learning is in the tradition of principal components analysis and clustering.

[3] This may seem a lot like standard numerical methods, such as when the Newton–Raphson algorithm is applied to logistic regression. We will see that it is not. For example, in logistic regression, the form of the relationships between the predictors and the response are determined before the algorithm is launched. In statistical learning, one important job of the algorithm is to determine these relationships.

## RANDOM FORESTS

Random forests provides a very instructive introduction to statistical learning from a regression perspective. It represents an explicit challenge to modeling by adopting an "algorithmic" approach (Breiman 2001b). There is no model linking inputs to outputs. There are, therefore, no model parameters whose values need to be estimated. Rather, a computer program attempts to directly associate predictors with the response in a manner that is highly sensitive to features of the data. The forecasting accuracy of this approach is impressive (Breiman 2001a; Berk 2008: Chapter 5; Hastie et al. 2009: Chapter 15; Berk et al. 2009a).

Because there is no equivalent of regression parameters to estimate, links between predictors and the response are shown through two other algorithms. These algorithms are considered below. They build directly on a forecasting framework and have intuitively clear meaning.

The random forests algorithm is an integration of several earlier statistical procedures. It uses classification and regression trees (Breiman et al. 1984) as a building block and then takes ideas from the bootstrap (Efron and Tibshirani 1993) and from bagging (Breiman 1996). But, the manner in which the algorithm preforms is not easily anticipated from its component parts.

The seminal paper on random forests is by Leo Breiman (2001a). An interesting interpretation is provided by Lin and Jeon (2006). A very accessible textbook treatment can be found in Berk (2008: Chapter 5). A more formal exposition can be found in Hastie et al. (2009: Chapter 15). Random forest is available in a Fortran program written by Leo Breiman and Ann Cutler, in the programming language R, and in a very user friendly form from Salford Systems (http://www.salford-systems.com/). The random forests procedure in R is the Breiman and Cutler program with some new features. The Salford Systems version of random forests also builds on the Brieman and Cutler program.

### The Random Forests Algorithm

Consider now the steps in the random forests algorithm. It is well worth studying carefully. There is a training sample with $N$ observations. There is a response variable and a set of $p$ predictors. Then, the algorithm proceeds in the following sequence.[4]

1. *Draw a random sample of size $N$ with replacement from the training data.*
   Comment: The observations not selected are saved as the "out-of-bag" (OOB) data. These are the test data for the given tree. On the average, about a third of the training data set becomes OOB data. This is the bootstrap step. Some very recent work suggests that the sample size should be a bit smaller than $N$ (Traskin 2008).
2. *Draw a small random sample of predictors without replacement (e.g., three predictors).*
   Comment: Usually the random sample of predictors is much smaller than the number of predictors overall.

[4] To provide exact instructions would require formal mathematical notation or the actual computer code. That level of precision is probably not necessary for this chapter and can be found elsewhere (e.g., in the source code for the procedure *randomForest* in R). There are necessarily a few ambiguities, therefore, in the summary of this algorithm and the two to follow. Also, only the basics are discussed. There are extensions and variants to address special problems.

3. *Using the observations sampled, subset the data using CART as usual into two subsets. If the response variable is categorical, the split is chosen to minimize the Gini index. If the response is quantitative, the split is chosen to minimize the residual sum of squares.*
   Comment: This is the first step in growing a classification or regression tree. If the response is categorical, a classification tree is grown. If the response is quantitative, a regression tree is grown.
4. *Repeat Steps 2 and 3 for all later subsets until additional partitions do not improve the model's fit.*
   Comment: The result is a regression or classification tree. There is some debate about how large to grow the tree. Current opinion favors growing the tree as large as possible.
5. *For a classification tree, compute as usual the class to be assigned to each terminal node. For a regression tree, compute as usual the conditional mean of each terminal node.*
   Comment: Both can be thought of as the fitted values from the tree.
6. *"Drop" the OOB data down the tree. For a categorical response variable, assign the class associated with the terminal node in which an observation falls. For a quantitative response variable, assign the conditional mean of the terminal node in which an observation falls.*
   Comment: The key point is that the OOB data were not used to grow the tree. True forecasts are the result.
7. *Repeat Steps 1–6 many times (e.g., 500) to produce many classification or regression trees.*
8. *For a categorical response variable, classify each observation by majority vote over all trees when that observation was OOB. For a quantitative response variable, assign the average conditional mean over all trees when that observation was OOB.*
   Comment: These are the forecasts for each observation averaged over trees.

The output of random forests is a set of forecasts, one for each observation in the training data. From this and the observed value of the response for each observation, one is able to compute various measure of forecasting accuracy. In the categorical case, the measure might be the proportion of cases forecasted incorrectly. For a quantitative response, the measure might be the mean squared error. In practice, however, it is important to unpack these overall measures. With a categorical response variable, for example, it is common to examine separately the proportion in each class (i.e., category) correctly forecasted. Often, some classes are forecasted better than others.

Random forests have some very important assets when compared with the conventional regression models. The CART building blocks are very flexible so that unanticipated nonlinear relationships between the predictors and the response can be inductively discovered. However, individual trees are known to be very unstable. Averaging over trees tends to cancel out the instabilities. The averaging is a form of shrinkage associated with estimators such as the "lasso" (Tibshirani 1996) and a way to smooth the step functions that CART naturally constructs.[5]

[5] Shrinkage estimators have a long history starting with empirical Bayes methods and ridge regression. The basic idea is to force a collection estimated values (e.g., a set of conditional means) toward a common value. For regression applications, the estimated regression coefficients are "shrunk" toward zero. A small amount of bias is introduced

Another asset is that by sampling predictors, the fitted values are made more independent across trees. This enhances the impact of the averaging and is another form of shrinkage. It also gives predictors that would otherwise be neglected a chance to contribute to the fitting process. Competition between predictors is greatly reduced with the result that the fitted values can be especially sensitive to highly nonlinear relationships. That is, predictors that are important for only a few observations are not necessarily shouldered aside by predictors that are more important overall. All predictors can help out.[6]

Yet, another asset is that as the number of trees in the random forest increases without limit, the estimate of population generalization error is consistent (Breiman 2001a: 7). That is, one obtains for the binary case a consistent estimate of the probability of a correct classification over trees minus the probability of an incorrect classification over trees. Thus, random forests does not overfit when a large number of trees is grown. This is in important contrast to other highly exploratory methods.[7]

Finally, for categorical response variables, random forests can introduce the relative costs of forecasted false negatives and false positives directly into the algorithm. Most other regression procedures assume that the costs are the same. For example, the costs of failing to identify an individual who will likely commit a serious felony while on parole are assumed to be the same as the costs of falsely identifying an individual as someone who will likely commit a serious felony while on parole. Equal costs are unrealistic in a wide variety of settings and introducing unequal costs can dramatically alter the forecasts that result.

To help fix these ideas, consider the following illustration for a binary response variable. Table 34.1 is a reconstruction from an article in which for individuals in Philadelphia on probation or parole, a forecast was made for whether or not they would be charged with a homicide or attempted homicide within 2 years of intake (Berk et al. 2009a). Predictors included the usual information available at intake. Table 34.1 is essentially a cross-tabulation of the actual outcome against the forecasted outcome. The cell entries are counts with the exception of those on the far right, which are proportions. For ease of labeling, charges of a homicide or an attempted homicide will be denoted by "homicide" in Table 34.1.

into the estimates to gain a substantial reduction in the variance of the estimates. The result can be an overall reduction in mean squared error. Shrinkage can also be used for model selection when some regression coefficients are shrunk to zero and some are not. Shrinkage is discussed at some length in the book by Hastie et al. (2009: 61–69) and Berk (2008: 61–69, 167–174). Shrinkage is closely related to "regularization" methods.

[6] Recall that when predictors in a regression analysis are correlated, the covariate adjustments ("partialling") can cause predictors that are most strongly related to the response to dominate the fitting process. Predictors that are least strongly related to the response can play almost no role. This can be a particular problem for nonlinear response functions because the predictors that are more weakly related to the response overall may be critical for characterizing a small but very essential part of the nonlinear function. By sampling predictors, random forests allows the set of relevant predictors to vary across splits and across trees so that some of the time, weak predictors only have to "compete" with other weak predictors.

[7] Overfitting occurs when a statistical procedure responds to idiosyncratic features of the data. As a result, the patterns found do not generalize well. With a new data set, the story changes. In regression, overfitting becomes more serious as the number of parameters being estimated increases for a fixed sample size. The most widely appreciated example is stepwise regression in which a new set of regression coefficients is estimated at each step. Overfitting can also be a problem in conventional linear regression as the number of regression coefficient estimates approaches the sample size. And, overfitting can also occur because of data snooping. The researcher, rather than some algorithm, is the guilty party. The best way to take overfitting into account is to do the analysis on training data and evaluate the results using test data. Ideally, the training data and test data are random samples from the same population. One might well imagine that overfitting would be a problem for random forests. Breiman proves that this is not so.

**Table 34.1. Random forests confusion table for forecasts of homicide or attempted homicide using the training sample and out-of-bag observations ($N = 30{,}000$)**

| | Forecast of no homicide | Forecast of homicide | Forecasting error |
|---|---|---|---|
| No homicide observed | 27,914 | 1,764 | 0.06 |
| Homicide observed | 185 | 137 | 0.57 |

Charges of homicide or attempted homicide are rare events. About 1 in 100 individuals were charged with a homicide or attempted homicide within 2 years of intake. It was very difficult, therefore, to improve on the marginal distribution. If one predicted that all individuals under supervision would not commit such crimes, that forecast would be correct about 99% of time. With so unbalanced a marginal distribution, it is not surprising that when logistic regression was applied to the data, only two cases were forecasted to be charged with a homicide or attempted homicide, and one was a false positive. In fact, 322 individuals were so charged.

Random forests does better. To begin, stakeholders thought that false negatives were about ten times more costly than false positives. The costs of 10 individuals falsely labeled as prospective murderers were about the same as the costs of one prospective murderer who was not labeled as such. This cost ratio was built into the random forests forecasts. Thus, the ratio in Table 34.1 of false negatives to false positives is around 10 to 1 (1,764/185). That ratio, in turn, helps shape the actual forecasts.

When an individual was actually not charged with a homicide or an attempted homicide, 6% of the cases were forecasted incorrectly. When an individual was actually charged with a homicide or an attempted homicide, 57% of the cases were forecasted incorrectly; about 43% of the time when the random forest algorithm forecasted a homicide or attempted homicide, it was correct.[8] Stakeholder found this level of accuracy useful.

## Predictor Importance

Although random forests earns it keep through its fitted values, there is naturally an interest in which predictors are most important for forecasting accuracy. This information is obtained through a second algorithm using the following instructions.

1. *For categorical response variables, compute the predicted class over all trees for each case when it is OOB. For quantitative response variables, compute the conditional mean over all trees for each case when it is OOB.*
2. *For categorical response variables, compute the proportion of cases misclassified for each response class. For quantitative response variables, compute the mean squared error.*
   Comment: These serve as a baselines for forecasting accuracy.
3. *Randomly permute all values of a given predictor.*
   Comment: Shuffling makes the predictor unrelated to the response (and all other predictors) on the average.
4. *Repeat Step1.*

[8] Table 34.1 raises several other important issues, but they are beyond the scope of this review (see Berk et al. 2009a).

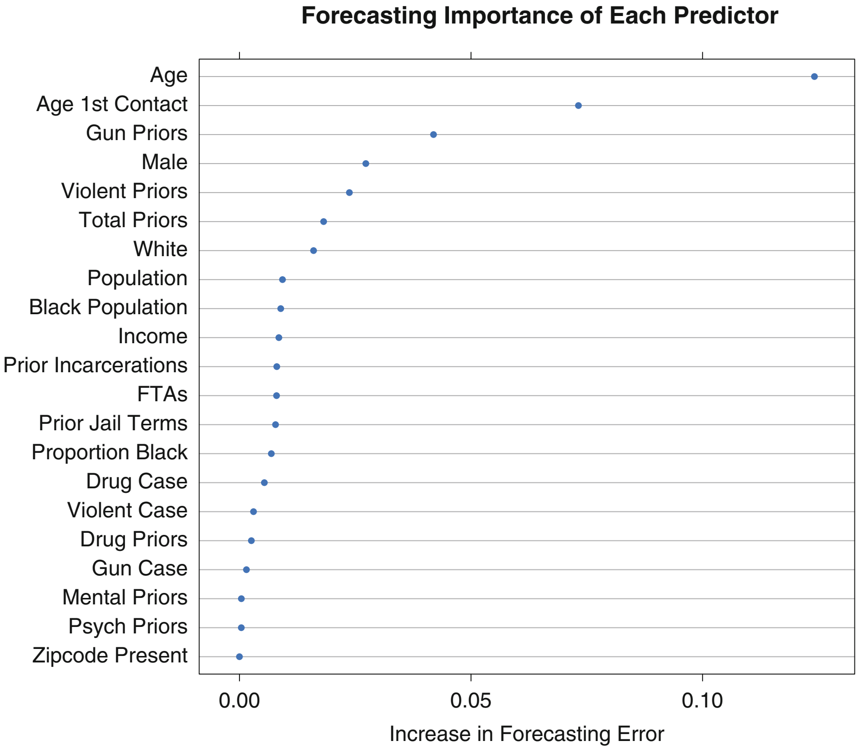

FIGURE 34.1. Predictor importance for forecasting accuracy.

5. *Repeat Step 2.*
   Comment: Steps 4 and 5 provide a performance measure after shuffling.
6. *Compute the increase in forecasting error by comparing the results of Step 5 to the results of step 2.*
   Comment: The increase is a measure of forecasting importance.
7. *Repeat Step 3 through Step 6 for each predictor.*

Figure 34.1 provides importance measures for the predictors used to forecast a charge of murder or attempted murder. Age is the most important predictor. When age was shuffled, the proportion of homicide or attempted homicide cases incorrectly forecasted increased about 12% points (i.e., from 0.57 to 0.69). The increase for the age of first contact with the adult courts was the second most important predictor with an increase of about 8% points. None of the predictors below race ("White") in the table were individually important for forecasting accuracy.

## Response Functions

Knowing a predictor's contribution to forecasting accuracy is useful but insufficient. It can also be important to learn how each predictor is related to the response with all other predictors held constant. A third algorithm is required.

1. *For a given predictor with $V$ values, construct $V$ special data sets, setting that predictor's values to each value $v$ in turn and fixing the other predictors at their existing values.*
   Comment: For example, if the predictor is years of age, $V$ might be 40, and there would be 40 data sets, one for each year of 40 years of age. In each dataset, age would be set to one of the 40 age values for all observations (e.g., 21 years old), whether that age was true or not. The rest of the predictors would be fixed at their existing values. The values of the other predictors do not change over data sets and in that fundamental sense are held constant. There are no covariance adjustments as in conventional regression.
2. *Using a constructed data set with a given $v$ (e.g., 26 years of age) and the random forest output, compute the fitted value for each case.*
3. *Average the fitted values over all cases.*
   Comment: This provides the average fitted response for a given $v$, all other predictors fixed at their actual values. For categorical response variables, the average fitted value is a conditional proportion or a transformation of it. For quantitative response variables, the average fitted value is a conditional mean. The fitted values are analogous to the $\hat{y}_i$ in conventional linear regression.
4. *Repeat Steps 2 and 3 for each of the $V$ values.*
5. *Plot the average fitted values from Step 4 for each $v$ against the $V$ values of the predictor.*
   Comment: This step produces a partial response plot (alternatively called a "partial dependence plot") showing how the average response for a given predictor varies with values of that predictor, all other predictors held constant.
6. *Repeat Steps 1–5 for each predictor.*

To illustrate, Fig. 34.2 shows the partial dependence plot of the age at first contact with the adult court system. The outcome is a charge of homicide or attempted homicide in centered logits. The reasoning for these units is beyond the scope of this chapter but is discussed in Berk's book on statistical learning (2008: 222–226). The circles are the average fitted values, and a smoother is overlaid.

The message is largely in the shape of the response curve. The odds of a charge of homicide or attempted homicide drop precipitously from age 12 to age 30. There is then a gradual and modest increase from age 30 to age 50. Probationers or parolees who get into serious trouble at a young age are at much higher risk to be "shooters." This means that an armed robbery at age 15 is a strong indicator of later violence. That same armed robbery at age 30 is not. The increase between 30 and 50 years of age may represent a different violence etiology, perhaps associated with domestic violence.

The negative slope overall is certainly no surprise. The shape of the response curve, however, was not anticipated by current theory in criminology. Moreover, the strongly nonlinear form was thoroughly misspecified by the logistic regression. Most of the other important quantitative predictors also had dramatically nonlinear response functions, which helps explain why the logistic regression fared so poorly.[9]

[9] Although one can construct partial dependence plots for categorical predictors, all one can see is a bar chart. For each category, the height of the bar is the average response value. The order of the bars along the horizontal axis and their distance from one another are necessarily arbitrary.

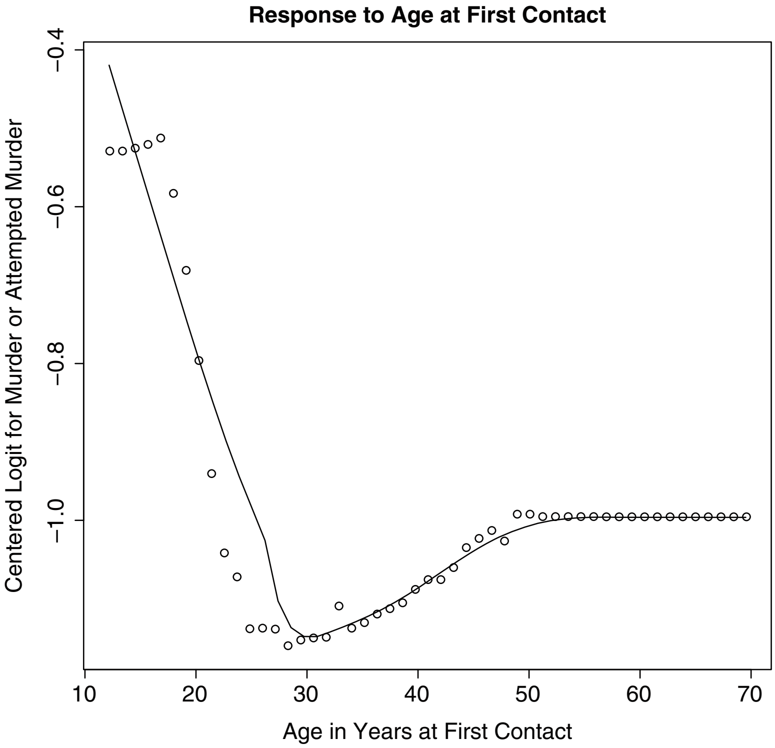

FIGURE 34.2. Partial dependence plot for age at first adult court contact.

## BOOSTING

Boosting is often seen as a good alternative to random forests and in practice, it also performs very well. The mathematics behind boosting, however, is somewhat more demanding, and the details of the algorithms require more exposition than can be accommodated here. Fortunately, the broad fundamentals of boosting can be summarized quite easily.

There are many kinds of boosting. But the earliest and most well-known boosting procedure was developed in computer science and in particular, through the work of Freund and Schapire (1996), Schapire (1999, 2002). It is called Adaboost and was originally intended for categorical response variables. We begin with Adaboost. Later, we consider briefly extensions and recent reconceptualizations of boosting by statisticians.

Consider a binary response coded as 1 or $-1$.[10] The Adaboost algorithm then has the following structure (Hastie et al. 2009: 339). As before, there are $N$ observations in the training data.

1. *Initialize a set of observation weights* $w_i = 1/N, i = 1, 2, \ldots, N$.
   Comment: So, if $N = 1{,}000$, the initial weight for each observation is 1/1,000.
2. *For* $m = 1$ to $M$*:*

[10] A coding of 1 and 0 can work too. But the 1 or $-1$ coding leads to the convenient result that the sign of the fitted value determines class membership.

Comment: There will be $M$ passes through the data.

a. *Fit a classifier* $G_m(x)$ *to the training data using the weights* $w_i$.
Comment: In this context, a "classifier" is essentially any regression statistical procedure for categorical response variables. The subscript $m$ denotes the iteration, and $x$ represents a set of predictors.

b. *Compute:* $\mathrm{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{n} w_i}$.
Comment: This is just the weighted proportion of cases misclassified for the $m$th iteration.

c. *Compute* $\alpha_m = \log[(1 - \mathrm{err}_m)/\mathrm{err}_m]$.
Comment: This is a logit transformation of the weighted proportion of cases misclassified for the $m$th iteration.

d. *Set* $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))], i = 1, 2, \ldots, N$.
Comment: $I$ is an indicator ("dummy") variable constructed from the operation within the parenthesis. Thus, new weights are constructed so that cases misclassified in the prior pass through the data are given a larger weight in the next pass through the data.

3. *Output* $G(x) = \mathrm{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.
Comment: After $M$ passes through the data, the class assigned to each case is determined by the sign of its averaged fitted values.

Like random forests, Adaboost makes many passes through the data (e.g. 1,000), and each pass through the data produces a set of fitted values. Adaboost also averages its sets of fitted values. But, there is no sampling of the data and no sampling of predictors. And, the averaging is a weighted average, with weights a function of overall fitting accuracy. That is, the better the fit of the output from a given pass through the data, the more weight given to that set of fitted values in the averaging. Adaboost can classify and forecast very accurately, although the forecasting must be undertaken in a subsequent step; it is not built into the algorithm as it is in random forests.

There are now a large number of boosting variants. Perhaps, the most influential of these have been produced by statisticians who have shown that some important features of Adaboost, and boosting in general, can be formulated as a more conventional statistical regression procedure. That is, boosting is "just" another means to minimize a regression loss function and many different loss functions are possible.[11] Thus, Hastie et al. (2009: 343–350) show that Adaboost is doing something very much like logistic regression, but with a somewhat different loss function.

Building on this insight, many different kinds of boosting can be formulated (Friedman 2001, 2002; Friedman et al. 2000, 2004; Bühlmann and Yu 2006), and some properties of boosting can be proved (Mannor et al. 2002; Zhang and Yu 2005). One of the most flexible boosting methods is stochastic gradient boosting (Friedman 2002), which builds on the generalized linear model.[12] There are analogs to logistic regression, Poisson regression, normal regression, Cox proportional hazard regression, quantile regression and others. Software for stochastic gradient boosting can be found in R.

[11] Recall that the loss function of least squares regression, for example, is the sum of the squared residuals.

[12] Stochastic gradient boosting samples the training data in the same spirit as random forests.

The statistical view of boosting is not the final word. There are some important features of boosting that at least so far are not “statistical” and are not yet well understood (Wyner 2003; Mease et al. 2007; Mease and Wyner 2008). For example, Adaboost does a fine job capturing the class to which a case belongs (e.g., a reoffender or not), but can easily lead to extremely inaccurate values for the *probability* of belonging to a given class (Buja et al., 2005).

Just like random forests, there is no direct way within the basic algorithm to understand how predictors are related to the response. Just like for random forests, additional algorithms are needed. To date, these algorithms are much like that those used with random forests. There are boosting versions of importance measures and partial dependence plots. Boosting output has much the same look and feel as random forests output.

Perhaps, the major drawback of boosting is that the costs of forecasting errors are not a routine part of the algorithm. But, some potential remedies have been proposed. For example, Kriegler and Berk (2009) develop a form of boosting based on quantile regression in which the quantile specified for the response variable determines the differential costs of forecasting errors.[13]

## SUPPORT VECTOR MACHINES

Support vector machines is another very popular statistical learning procedure with roots in computer science. Its rationale is clever and relatively simple. The mathematics is cleverer, but not simple at all. Support vector machines is associated most strongly with the work of Vladimir Vapnick (1996). A discussion of support vector machines from a statistical perspective can be found in Hastie et al. (2009: 423–437). A very accessible exposition is provided by Berk (2008: Chapter 7).

The basic idea is this. Suppose there is a binary outcome: failure on parole or not. Suppose, also there is a risk score serving as a single predictor (although in practice there is usually more than one predictor). For very large risk score values, all of the offenders in fact fail on parole. For very low risk score values, none of the offenders in fact fails on parole. It is for the middle range values of the risk score that the outcomes are mixed. Some offenders fail and some do not. Indeed, some of the offenders in the middle range who fail have lower risk scores than some other offenders in the middle range who do not fail. For the very high and very low risk scores, classification is trivial because for the very low or very high risk scores, the offenders are homogeneous on the outcome. It makes good sense, therefore, to focus on the middle ranges where accurate classification is nontrivial.

Support vector machines builds on this basic logic in two ways. First, its loss function ignores the cases that are easily classified correctly. What matters is how the procedure performs for the cases that are difficult to accurately classify. This is broadly similar to the rationale for boosting. And like boosting, forecasting must be undertaken in a subsequent step. Unlike random forests, it is not part of the basic algorithm. Second, like boosting and random forests, support vector machines can inductively construct highly nonlinear functions for accurate classification. However, the building blocks for these functions are very different for support vector machines. How they are different need not trouble us here. A formal

[13] In quantile regression, the fitted values are conditional quantiles, not the conditional mean. For example, the 50th quantile is the conditional median (Keonker 2005). Then, if, say, the 75th quantile is used, overestimates are three times more important (75/25) than underestimates.

discussion of these issues can be found in Hastie et al. (2009: 423–426) and Bishop (2006: 294–299). A more intuitive treatment can be found in Berk (2008: 312–314).

There has been less interest among developers of support vector machines for representing the way predictors are related to the response. Accurate classification and forecasting are the primary goals. Consequently, there is to date little that is comparable to measures of variable important and partial dependence plots. However, there seems to be no principled reason why such procedures would not be helpful. It is likely that importance measures and partial dependence plots will be available in the near future.

Support vector machine can have several operational problems. Most of the developmental work has focused on categorical response variables. For quantitative response variables, support vector machines are less well developed and somewhat harder to justify. In addition, still under development are effective ways to build in the costs of forecasting errors. Finally, support vector machines have several important tuning parameters that can dramatically affect the results. It is not yet clear how best to select the values for these tuning parameters.

Software for support vector machines can be found in a number of stand-alone procedures available over the internet. At last count, there were also three different support vector machines procedures in R. If it does not already exist, there will soon be support vector machines software commercially available. It can be a very powerful procedure.

## SUMMARY AND CONCLUSIONS

Statistical learning used as a form of regression analysis has several distinctive features. There is no need for a model in the sense that criminologist use the concept. In addition, how the predictors are related to the response is determined inductively from the data. And, the need to respond to the data in a sensitive fashion leads to computer-intensive algorithms that can respond to highly local, but nonetheless important, features of the data. Finally, statistical learning output differs from conventional regression output. In particular, there are no regression coefficients. Taken together, these features can make statistical inference and casual inference highly problematic. Of course, they are often highly problematic for conventional regression as well although practitioners too often proceed nevertheless. More discussion of statistical inference and causal inference for statistical learning can be found in Berk (2008).

Statistical learning is most model-like when the goal is function estimation. But, the process by which the function is determined falls in the middle ground between confirmatory and exploratory data analysis. In practice, moreover, it will be difficult to make the case that the disturbances have the requisite properties. The enterprise, therefore, is usually very close to exploratory data analysis.

Some might find the inability to proceed in a confirmatory manner a very serious constraint. However, if the form of the model is known with confidence, much of the rationale for statistical learning evaporates. Some form of parametric regression will suffice. And, if the form of the model is unknown, or known only with great uncertainty, exploratory tools can be very liberating.

At a deeper level, statistical learning takes researchers back to first principles. The algorithms stay very close to the data and to the immediate goals of the analysis. There is no model and its attending complications standing between the researcher and the data. There is no model-based recipe either. Rather, the goal is knowledge discovery, and in practice the process is fundamentally inductive.

**Acknowledgments** Work on this paper was supported in part by a grant from the National Science Foundation: SES-0437169, “Ensemble Methods for Data Analysis in the Behavioral, Social and Economic Sciences.” That support is gratefully acknowledged.

## REFERENCES

Berk RA (2003) Regression analysis: a constructive critique. Sage Publications, Newbury Park, CA

Berk RA (2008) Statistical learning from a regression perspective. Springer, New York

Berk RA, Sherman L, Barnes G, Kurtz E, Lindsay A (2009a) Forecasting murder within a population of probationers and parolees: a high stakes application of statistical forecasting. J R Stat Soc Ser A 172(part 1):191–211

Berk RA, Brown L, Zhao L (2009b) Statistical inference after model selection. Working Paper, Department of Statistics, University of Pennsylvania

Bishop CM (2006) Pattern recognition and machine learning. Springer, New York

Breiman L (1996) Bagging predictors. Mach Learn J 26:123–140

Breiman L (2001a) Random forests. Mach Learn 45:5–32

Breiman L (2001b) Statistical modeling: two cultures (with discussion). Stat Sci 16:199–231

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth Press, Monterey, CA

Buja A, Stuetzle W, Shen Y (2005) Loss functions for binary class probability estimation and classification: structure and applications. Unpublished Manuscript, Department of Statistics, The Wharton School, University of Pennsylvania

Bühlmann P, Yu B (2006) Sparse boosting. J Mach Learn Res 7:1001–1024

Cook DR, Weisberg S (1999) Applied regression including computing and graphics. Wiley, New York

Efron B, Tibshirani R (1993) Introduction to the bootstrap. Chapman & Hall, New York

Freedman DA (2004) Graphical models for causation and the identification problem. Eval Rev 28:267–293

Freedman DA (2005) Statistical models: theory and practice. Cambridge University Press, Cambridge

Freund Y, Schapire R (1996) Experiments with a new boosting algorithm. In: Machine learning: proceedings for the 13th international conference. Morgan Kaufmann, San Francisco, pp 148–156

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29:189–1232

Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38:367–378

Friedman JH, Hastie T, Tibsharini R (2000) Additive logistic regression: a statistical view of boosting (with discussion). Ann Stat 28:337–407

Friedman JH, Hastie T, Rosset S, Tibsharini R, Zhu J (2004) Discussion of boosting papers. Ann Stat 32:102–107

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, 2nd edn. Springer, New York

Holland P (1986) Statistics and causal inference. J Am Stat Assoc 8:945–960

Keonker R (2005) Quantile regression. Cambridge University Press, Cambridge

Kriegler B, Berk RA (2009) Estimating the homeless population in Los Angeles: an application of cost-sensitive stochastic gradient boosting. Working paper, Department of Statistics, UCLA

Leeb H, Pötscher BM (2006) Can one estimate the conditional distribution of post-model-selection estimators? Ann Stat 34(5):2554–2591

Lin Y, Jeon Y (2006) Random forests and adaptive nearest neighbors. J Am Stat Assoc 101:578–590

Mannor S, Meir R, Zhang T (2002) The consistency of greedy algorithms for classification. In: Kivensen J, Sloan RH (eds) COLT 2002. LNAI, vol 2375. pp 319–333

McCullagh P, Nelder JA (1989) Generalized linear models, 2nd edn. Chapman & Hall, New York

Mease D, Wyner AJ (2008) Evidence contrary to the statistical view of boosting. J Mach Learn 9:1–26

Mease D, Wyner AJ, Buja A (2007) Boosted classification trees and class probability/quantile estimation. J Mach Learn 8:409–439

Morgan SL, Winship C (2007) Counterfactuals and causal inference: methods and principle for social research. Cambridge University Press, Cambridge

Schapire RE (1999) A brief introduction to boosting. In: Proceedings of the 16th international joint conference on artificial intelligence

Schapire RE (2002) The boosting approach to machine learning: an overview. In: MSRI workshop on non-linear estimation and classification

Tibshirani RJ (1996) Regression shrinkage and selection via the LASSO. J R Stat Soc Ser B 25:267–288

Traskin M (2008) The role of bootstrap sample size in the consistency of the random forest algorithm. Technical Report, Department of Statistics, University of Pennsylvania

Vapnick V (1996) The nature of statistical learning theory. Springer, New York

Wyner AJ (2003) Boosting and exponential loss. In: Bishop CM, Frey BJ (eds) Proceedings of the 9th annual conference on AI and statistics, Jan 3–6, Key West, FL

Zhang T, Yu B (2005) Boosting with early stopping: convergence and consistency. Ann Stat 33(4):1538–1579