Chapter 9
Broader Implications and a Bit of Craft Lore

9.1 Some Integrating Themes

Over the past decade, the number of statistical learning procedures that can be viewed as a form of regression has grown. By and large, they are variants on, or extensions of, the procedures discussed in earlier chapters. The major advances are to be found in deeper understandings of the underlying mechanisms and increasingly, some common themes.

The major players, random forests, boosting, and support vector machines, share with niche players like neural networks and Bayesian additive regression trees the use of linear basis expansions to provide a rich collection of predictors. How this is done can vary. Random forests arrives at its basis expansions by building inductively over a large number of regression or classification trees, sampling the training data and predictors. Stochastic gradient boosting proceeds by sampling and reweighting the data with each iteration. Support vector machines get the job done by constructing rich predictor kernels in advance of the data analysis. Neural networks imposes nonlinear transformations of the predictors through its hidden layers. BART generates an ensemble of trees by stochastic decision rules while treating the data as fixed. It should not be surprising that when properly implemented, one can often get similar performance across these statistical learning methods.

The reliance on complicated linear basis expansions usually leads to blackbox procedures. One can get fitted values that perform very well, but the role of the predictors responsible is typically obscure. There have been recent efforts to develop auxiliary algorithms that can help, and more such advances are in the offing. But the blackbox problem underscores that statistical learning procedures depend on algorithms not models in which ends can justify means. If one's primary data analysis

The original version of this chapter was revised: See the "Chapter Note" section at the end of this chapter for details. The erratum to this chapter is available at https://doi.org/10.1007/978-3-319-44048-4_10.

R.A. Berk, *Statistical Learning from a Regression Perspective*,
Springer Texts in Statistics, DOI 10.1007/978-3-319-44048-4_9

goal is to explain, statistical learning is not likely to be helpful, and formal causal inference is typically off the table. When feasible, one is better off doing experiments.

Each of the procedures discussed can be represented as $Y = f(X) + \varepsilon$, where $\hat{f}(X)$ is arrived at by minimizing some loss function. The introduction of ε and loss function optimization can be seen as recasting machine learning as statistical learning. In practice, however, any of the procedures we have discussed properly can sail under either flag.

The formulation relying on $Y = f(X) + \varepsilon$ does not imply that $f(X)$ is the true response surface. It is called an approximation for good reason. When there is estimation, the target is an acknowledged approximation. The goal is to arrive at an effective approximation with the understanding that there will be bias and variance separating the estimate from the "truth." In the end, statistical learning earns its keep by explicitly constructing approximations of the true response surface that by several criteria are usually better than the unacknowledged approximations constructed by conventional models.

There remains hard work to be done understanding why statistical learning procedures work so well. Margin maximization, loss function optimization, and interpolation all play some role. But the accounts are at best incomplete. For example, there is somewhat limited understanding of why certain kernels work well in certain settings but not others.

All of the statistical learning procedures we have discussed are conceptually and operationally challenged by statistical inference, statistical tests, and confidence intervals. Bayesian additive regression trees tackles the problem head on, but in a manner that many find unsatisfactory. All of the other methods have their greatest success when the training data can be seen credibly as IID realizations from a joint probability distribution and when there are test data to provide honest performance assessments.

9.2 Some Practical Suggestions

Just as for any other set of statistical procedures, practice is guided significantly by craft lore. In that spirit, we turn to a bit of craft lore about the use of statistical learning. It is important to keep in mind, however, craft lore can change dramatically with experience, and the experience with statistical learning to date is somewhat spotty.

9.2.1 Choose the Right Procedure

Recall Breiman's distinction between two cultures: a "data modeling culture" and an "algorithmic modeling culture" (2001b). The data modeling culture favors the generalized linear model and its various extensions. A data analysis begins with a mathematical expression meant to represent the mechanisms by which nature works. Estimation serves to fill in the details. The algorithmic modeling culture is concerned solely with linking inputs to outputs. The subject-matter mechanisms connecting the two are not represented and there is, therefore, no a priori vehicle by which inputs are transformed into outputs. A data analysis is undertaken to invent such a vehicle so that a good fit results. There is no requirement whatsoever that the vehicle reveals nature's machinery.

But, there is in practice no clear distinction between procedures that belong in the data modeling culture and procedures that belong in the algorithmic modeling culture. In both cultures, information extracted from data is essential. Even for a correct regression model, parameter estimates are obtained from data. Rather, there is a continuum characterized by how much the results depend on substantively informed constraints imposed on the analysis. For conventional regression, at one extreme, there are extensive constraints meant to represent the machinery by which nature proceeds. At the other extreme, random forests and stochastic gradient boosting mine associations in the data with virtually no substantively informed restrictions. Many procedures, such as those within the generalized additive model, fall in between.

How then should a data analysis tool be selected? As a first cut, the importance of explicitly representing nature's machinery should be determined. If explanation is the dominant data analysis motive, procedures from the data modeling culture should be favored. If prediction is the dominant data analysis motive, procedures from the algorithmic modeling culture should be favored. If neither is dominant, procedures should be used that are a compromise between the two extremes.

If one is working within the data modeling culture, the choice of procedures is determined primarily by the correspondence between subjective-matter information available and features of a candidate modeling approach. The correspondence should be substantial. For example, if nature is known to proceed through a linear combination of causal variables, a form of conventional regression may well be appropriate.

Working within the algorithmic modeling culture, the choice of procedures ideally is primarily determined by out-of-sample performance. One might hope that through formal mathematics and forecasting contests, clear winners and losers could be identified. Unfortunately, the results are rarely definitive. One major problem is that forecasting performance is typically dataset specific; accuracy depends on particular features of data that can differ across datasets. A winner on one forecasting task will often be a loser on another forecasting task. Another major problem is how to tune the procedures so that each is performing as well as it can on a given dataset. Because the kinds and numbers of tuning parameters vary across algorithmic methods, there is usually no way to ensure that fair comparisons are being made. Still another problem is that a lot depends on exactly how forecasting performance is measured. For example, the area under an ROC curve will often pick different winners from those evaluated by cost-weighted classification error.

However, all of the algorithmic methods emphasized in earlier chapters can perform well in a wide range of applications. In practice, perhaps the best strategy is for a data analyst to select a method that he or she adequately understands, that has features responding best to the application at hand, and that has the most instructive output. For example, only some of the procedures discussed can easily adapt to forecasting errors that have asymmetric costs, and some can handle very large datasets better than others. The procedures can also differ by whether there are, for example, partial dependence plots and how variable importance is measured. Forecasting accuracy is but one of several criteria by which algorithmic procedures can be compared. Among these other criteria are:

1. *ease of use* — A combination of the procedure itself and the software with which it is implemented;
2. *readily available software* — R usually a good place to start in part because commercial packages are often several years behind;
3. *good documentation* — for both the procedure and the software (be wary of commercial products that hide the details for their procedures by calling them proprietary);
4. *adaptability* — the procedure and its software should be easily adapted to unanticipated circumstances such as the need for test data;
5. *processing speed* — a function of the nature of the procedure, the number of observations, the number of variables, and the quality of the code (e.g., parallelization);
6. *ease of dissemination* — some procedures and some kinds of output are easier to explain to users of the results;
7. *special features of the procedure* — examples include the ability to handle classification with more than two classes, ways to introduce asymmetric costs from fitting errors, and tools for peering into the blackbox; and
8. *cost* — some commercial products can be quite pricey.

If there is no clear winner, it can always be useful to apply more than one procedure and report more than one set of results.

9.2.2 Get to Know Your Software

There is not yet, and not likely to be in the near future, a consensus on how any of the various statistical learning procedures should be implemented in software. For example, a recent check on software available for support vector machines found working code for over a half dozen procedures. There is, as well, near anarchy in naming conventions, and notation. Thus, the term “cost,” for instance, can mean several different things, and a symbol such as γ can be a tuning parameter in one derivation and a key argument in another derivation.

One cannot assume that a description of a procedure in a textbook (including this one) or journal article corresponds fully to software using the very same name, even by the same authors. Consequently, it is very important to work with software for which there is good technical documentation on the procedure and algorithms being used. There also needs to be clear information on how to introduce inputs, obtain outputs, and tune the procedure. Descriptions of two computer programs can use the same name for different items, or use very different names for the same item. And in either case, the naming conventions may not correspond to the naming conventions in the technical literature.

Even when the documentation looks to be clear and complete, a healthy dose of skepticism is useful. There are sometimes errors in the documentation, or in the software, or in both. So, it is usually important to "shake down" any new software with data that have previously been analyzed properly to determine if the new results come out as expected. In addition, it is usually helpful to experiment with various tuning parameters to see if the results make sense. In short, *caveat emptor*.

It is also very important keep abreast of software updates, which can come as often as five or six times a year. As a routine matter, new features are added, existing features are deleted, bugs fixed, and documentation rewritten. These changes are often far more than cosmetic. Working with an older version of statistical learning software can lead to unnecessary problems.

Finally, a key software decision is whether to work primarily with shareware such as found in R or Python, or with commercial products. The tradeoffs have been discussed earlier at various points. Cost is certainly an issue, but perhaps more important is the tension between having the most current software and having the most stable software and documentation. Shareware is more likely to be on the leading edge, but often lacks the convenience and stability of commercial products. One possible strategy for individuals who are unfamiliar with a certain class of procedures is to begin with a good commercial product, and then once some hands-on skill has been developed, migrate to shareware.

9.2.3 Do Not Forget the Basics

It is very easy to get caught up in the razzle-dazzle of statistical learning and for any given data analysis, neglect simple fundamentals. All data analyses must start with an effort to get "close" to the data. This requires a careful inspection of elementary descriptive statistics: means, standard deviations, histograms, cross-tabulations, scatterplots, and the like. It also means understanding how the data were generated and how the variables were measured. Moving into a statistical learning procedure without this groundwork can lead to substantial grief. For example, sometimes numeric values are given to missing data. Treating these values as legitimate values can seriously distort any data analysis, including ones undertaken with statistical learning.

It will often be helpful to apply one or more forms of conventional regression analysis before moving to statistical learning. One then obtains an initial sense of how good the fit is likely to be, of the likely signs of key relationships between predictors and the response, and of problems that might be more difficult to spot later (e.g., does one really have a weak learner?). An important implication is that it will often be handy to undertake statistical learning within a software environment in which a variety of statistical tools can be applied to the same data. This can weigh against single-purpose, statistical learning software.

To take one simple example, a tuning parameter in random forests may require a distinct value for each response class. But the order in which those arguments are entered into the expression for the tuning parameter may be unclear. In the binary case, for example, which category comes first? Is it $\omega = c(1, 0)$ or $\omega = c(0, 1)$? A wrong choice is easily made. Random forests runs just the same and generates sensible-looking output. But the analysis has not been tuned as it should have been. It can be difficult to spot such an error unless one knows the marginal distribution of the response variable and the likely sign of relationships between each predictor and the response. A few cross-tabulations and a preliminary regression analysis can help enormously.

Finally, one must not forget that preliminary analyses of the data can introduce data snooping, especially if relationships between potential predictors and potential responses are examined. This does not mean that one should avoid these analyses. What it means is that often, test data are essential.

9.2.4 Getting Good Data

As noted many times, there is no substitute for good data. The fact that boosting, for example, can make weak classifiers stronger, does not mean that boosting can make weak data stronger. There are no surprises in what properties good data should have: a large number of observations, little measurement error, a rich set of predictors, and a reasonably well-balanced response variable distribution. The clear message is that it is very important to invest time and resources in data collection. One cannot count on statistical learning successfully coming to the rescue. Indeed, some forms of statistical learning can be quite fussy and easily pulled off course by noisy data, let alone data that have systematic measurement error.

The case for having legitimate test data can be quite strong. Statistical learning procedures that use out-of-bag data or the equivalent may not formally need a test dataset. The out-of-bag observations can serve that purpose. But most statistical learning procedures currently are not designed to work with random samples of the data, even when that might make a lot of sense. Therefore, having access to test data is usually very important.

Even for random forests, test data beyond the out-of-bag observations can come in handy. Comparisons between how random forests performs and how other approaches (including conventional regression) perform are often very instructive. For example, one might learn that the key relationships are linear and that it is not worth losing degrees of freedom fitting more complex functions. Yet such comparisons cannot be undertaken unless there are test data shared by all of the statistical procedures in play. Finally, having a true test dataset can help a great deal if random forests is applied repeatedly to the same training data after changes in the tuning parameters. At the very end of the tuning process, there is still the opportunity to get a more honest measure of performance from data that until that moment have not been used.

9.2.5 *Match Your Goals to What You Can Credibly Do*

Much of the literature on statistical modeling is formulated around some $f(X)$. There is a real mechanism by which the data were generated. An essential goal of a data analysis is to recover the data-generation function from a dataset. It can be very tempting, therefore, to frame all data analyses in a similar manner.

But, one of the themes of this book has been that in reality, more modest goals are likely to be appropriate. Perhaps most important, statistical learning is not built around a regression model of the data generation process. The data are realized from a joint probability distribution and analyzed by algorithmic methods. In addition, one will usually not have access to all of the requisite predictors, let alone predictors that are all well measured. Finally, various kinds of data snooping will often be impossible to avoid. For these and other reasons, a level I analysis will be the primary enterprise.

But there also will be circumstances when a level II analysis can be justified and properly undertaken. These circumstances are addressed in various sections of the book. Perhaps the major take-home message is that level II analyses are never routine. They require clear and careful thought. For example, the vote proportions produced by random forests are not probabilities and do not represent the chances that a given observation falls into a particular outcome class. They are a measure of the internal reliability of the random forest algorithm.

Although causal thinking can be important as the research task is being formulated and the data are being collected, statistical learning procedures are not designed for Level III analyses. It can be very tempting to use some forms of statistical learning output, such as variable importance plots, to make causal statements. But the various definitions of importance do not comport well with the canonical definition of a causal effect, and the output is not derived from a causal model.

An important implication is that using a statistical learning procedure to do variable selection can lead to a conceptual swamp. If the purpose is to screen for important causal variables, it is not apparent how the statistical learning output is properly used for that purpose. This does not preclude dimension reduction in service of other ends. For example, regularization is an essential tool when the intent is to improve the stability of statistical learning output.

9.3 Some Concluding Observations

Over the past decade, statistical learning has become one of the more important tools available to applied statisticians and data analysts. But, the hype in which some procedures are wrapped can obscure important limitations and lead to analyses undertaken without sufficient care.

Statistical learning properly done will often require a major attitude adjustment. One of most difficult obstacles to effective applications is letting go of premises from conventional modeling. This will be especially difficult for experienced data analysts trained in traditional methods. One of the most common errors is to overlay statistical learning on top of model-based conceptions. Statistical learning is not just more of the same.

Finally, users of results from statistical learning must proceed with care. There is lots of money to be made and professional reputations to be built with statistical razzle-dazzle that is actually voodoo statistics. It can be very important to have access to technical advice from knowledgeable individuals who have no skin in the game.