Comment

Bruce Hoadley

INTRODUCTION

Professor Breiman's paper is an important one for statisticians to read. He and *Statistical Science* should be applauded for making this kind of material available to a large audience. His conclusions are consistent with how statistics is often practiced in business. This discussion will consist of an anecdotal recital of my encounters with the algorithmic modeling culture. Along the way, areas of mild disagreement with Professor Breiman are discussed. I also include a few proposals for research topics in algorithmic modeling.

CASE STUDY OF AN ALGORITHMIC MODELING CULTURE

Although I spent most of my career in management at Bell Labs and Bellcore, the last seven years have been with the research group at Fair, Isaac. This company provides all kinds of decision support solutions to several industries, and is very well known for credit scoring. Credit scoring is a great example of the problem discussed by Professor Breiman. The input variables, $\mathbf{x}$, might come from company databases or credit bureaus. The output variable, y, is some indicator of credit worthiness.

Credit scoring has been a profitable business for Fair, Isaac since the 1960s, so it is instructive to look at the Fair, Isaac analytic approach to see how it fits into the two cultures described by Professor Breiman. The Fair, Isaac approach was developed by engineers and operations research people and was driven by the needs of the clients and the quality of the data. The influences of the statistical community were mostly from the nonparametric side—things like jackknife and bootstrap.

Consider an example of behavior scoring, which is used in credit card account management. For pedagogical reasons, I consider a simplified version (in the real world, things get more complicated) of monthly behavior scoring. The input variables, $\mathbf{x}$, in this simplified version, are the monthly bills and payments over the last 12 months. So the dimension of $\mathbf{x}$ is 24. The output variable is binary and is the indicator of no severe delinquency over the next 6 months. The goal is to estimate the function, $f(\mathbf{x}) = \log(\Pr\{y = 1|\mathbf{x}\}/\Pr\{y = 0|\mathbf{x}\})$. Professor Breiman argues that some kind of simple logistic regression from the data modeling culture is not the way to solve this problem. I agree. Let's take a look at how the engineers at Fair, Isaac solved this problem—way back in the 1960s and 1970s.

The general form used for $f(\mathbf{x})$ was called a segmented scorecard. The process for developing a segmented scorecard was clearly an algorithmic modeling process.

The first step was to transform $\mathbf{x}$ into many interpretable variables called prediction characteristics. This was done in stages. The first stage was to compute several time series derived from the original two. An example is the time series of months delinquent—a nonlinear function. The second stage was to define characteristics as operators on the time series. For example, the number of times in the last six months that the customer was more than two months delinquent. This process can lead to thousands of characteristics. A subset of these characteristics passes a screen for further analysis.

The next step was to segment the population based on the screened characteristics. The segmentation was done somewhat informally. But when I looked at the process carefully, the segments turned out to be the leaves of a shallow-to-medium tree. And the tree was built sequentially using mostly binary splits based on the best splitting characteristics—defined in a reasonable way. The algorithm was manual, but similar in concept to the CART algorithm, with a different purity index.

Next, a separate function, $f(\mathbf{x})$, was developed for each segment. The function used was called a scorecard. Each characteristic was chopped up into discrete intervals or sets called attributes. A scorecard was a linear function of the attribute indicator (dummy) variables derived from the characteristics. The coefficients of the dummy variables were called score weights.

This construction amounted to an explosion of dimensionality. They started with 24 predictors. These were transformed into hundreds of characteristics and pared down to about 100 characteristics. Each characteristic was discretized into about 10 attributes, and there were about 10 segments. This makes $100 \times 10 \times 10 = 10,000$ features. Yes indeed, dimensionality is a blessing.

Dr. Bruce Hoadley is with Fair, Isaac and Co., Inc., 120 N. Redwood Drive, San Rafael, California 94903-1996 (e-mail: BruceHoadley@FairIsaac.com).

What Fair, Isaac calls a scorecard is now elsewhere called a generalized additive model (GAM) with bin smoothing. However, a simple GAM would not do. Client demand, legal considerations and robustness over time led to the concept of score engineering. For example, the score had to be monotonically decreasing in certain delinquency characteristics. Prior judgment also played a role in the design of scorecards. For some characteristics, the score weights were shrunk toward zero in order to moderate the influence of these characteristics. For other characteristics, the score weights were expanded in order to increase the influence of these characteristics. These adjustments were not done willy-nilly. They were done to overcome known weaknesses in the data.

So how did these Fair, Isaac pioneers fit these complicated GAM models back in the 1960s and 1970s? Logistic regression was not generally available. And besides, even today's commercial GAM software will not handle complex constraints. What they did was to maximize (subject to constraints) a measure called divergence, which measures how well the score, S, separates the two populations with different values of y. The formal definition of divergence is $2(E[S|y = 1] - E[S|y = 0])^2/(V[S|y = 1] + V[S|y = 0])$. This constrained fitting was done with a heuristic nonlinear programming algorithm. A linear transformation was used to convert to a log odds scale.

Characteristic selection was done by analyzing the change in divergence after adding (removing) each candidate characteristic to (from) the current best model. The analysis was done informally to achieve good performance on the test sample. There were no formal tests of fit and no tests of score weight statistical significance. What counted was performance on the test sample, which was a surrogate for the future real world.

These early Fair, Isaac engineers were ahead of their time and charter members of the algorithmic modeling culture. The score formula was linear in an exploded dimension. A complex algorithm was used to fit the model. There was no claim that the final score formula was correct, only that it worked well on the test sample. This approach grew naturally out of the demands of the business and the quality of the data. The overarching goal was to develop tools that would help clients make better decisions through data. What emerged was a very accurate and palatable algorithmic modeling solution, which belies Breiman's statement: "The algorithmic modeling methods available in the pre-1980s decades seem primitive now." At a recent ASA meeting, I heard talks on treed regression, which looked like segmented scorecards to me.

After a few years with Fair, Isaac, I developed a talk entitled, "Credit Scoring—A Parallel Universe of Prediction and Classification." The theme was that Fair, Isaac developed in parallel many of the concepts used in modern algorithmic modeling.

Certain aspects of the data modeling culture crept into the Fair, Isaac approach. The use of divergence was justified by assuming that the score distributions were approximately normal. So rather than making assumptions about the distribution of the inputs, they made assumptions about the distribution of the output. This assumption of normality was supported by a central limit theorem, which said that sums of many random variables are approximately normal—even when the component random variables are dependent and multiples of dummy random variables.

Modern algorithmic classification theory has shown that excellent classifiers have one thing in common, they all have large margin. Margin, M, is a random variable that measures the comfort level with which classifications are made. When the correct classification is made, the margin is positive; it is negative otherwise. Since margin is a random variable, the precise definition of large margin is tricky. It does not mean that $E[M]$ is large. When I put my data modeling hat on, I surmised that large margin means that $E[M]/\sqrt{V(M)}$ is large. Lo and behold, with this definition, large margin means large divergence.

Since the good old days at Fair, Isaac, there have been many improvements in the algorithmic modeling approaches. We now use genetic algorithms to screen very large structured sets of prediction characteristics. Our segmentation algorithms have been automated to yield even more predictive systems. Our palatable GAM modeling tool now handles smooth splines, as well as splines mixed with step functions, with all kinds of constraint capability. Maximizing divergence is still a favorite, but we also maximize constrained GLM likelihood functions. We also are experimenting with computationally intensive algorithms that will optimize any objective function that makes sense in the business environment. All of these improvements are squarely in the culture of algorithmic modeling.

OVERFITTING THE TEST SAMPLE

Professor Breiman emphasizes the importance of performance on the test sample. However, this can be overdone. The test sample is supposed to represent the population to be encountered in the future. But in reality, it is usually a random sample of the

current population. High performance on the test sample does not guarantee high performance on future samples, things do change. There are practices that can be followed to protect against change.

One can monitor the performance of the models over time and develop new models when there has been sufficient degradation of performance. For some of Fair, Isaac's core products, the redevelopment cycle is about 18–24 months. Fair, Isaac also does "score engineering" in an attempt to make the models more robust over time. This includes damping the influence of individual characteristics, using monotone constraints and minimizing the size of the models subject to performance constraints on the current test sample. This score engineering amounts to moving from very nonparametric (no score engineering) to more semiparametric (lots of score engineering).

SPIN-OFFS FROM THE DATA MODELING CULTURE

In Section 6 of Professor Breiman's paper, he says that "multivariate analysis tools in statistics are frozen at discriminant analysis and logistic regression in classification ...." This is not necessarily all that bad. These tools can carry you very far as long as you ignore all of the textbook advice on how to use them. To illustrate, I use the saga of the Fat Scorecard.

Early in my research days at Fair, Isaac, I was searching for an improvement over segmented scorecards. The idea was to develop first a very good global scorecard and then to develop small adjustments for a number of overlapping segments. To develop the global scorecard, I decided to use logistic regression applied to the attribute dummy variables. There were 36 characteristics available for fitting. A typical scorecard has about 15 characteristics. My variable selection was structured so that an entire characteristic was either in or out of the model. What I discovered surprised me. All models fit with anywhere from 27 to 36 characteristics had the same performance on the test sample. This is what Professor Breiman calls "Rashomon and the multiplicity of good models." To keep the model as small as possible, I chose the one with 27 characteristics. This model had 162 score weights (logistic regression coefficients), whose P-values ranged from 0.0001 to 0.984, with only one less than 0.05; i.e., statistically significant. The confidence intervals for the 162 score weights were useless. To get this great scorecard, I had to ignore the conventional wisdom on how to use logistic regression.

So far, all I had was the scorecard GAM. So clearly I was missing all of those interactions that just had to be in the model. To model the interactions, I tried developing small adjustments on various overlapping segments. No matter how hard I tried, nothing improved the test sample performance over the global scorecard. I started calling it the Fat Scorecard.

Earlier, on this same data set, another Fair, Isaac researcher had developed a neural network with 2,000 connection weights. The Fat Scorecard slightly outperformed the neural network on the test sample. I cannot claim that this would work for every data set. But for this data set, I had developed an excellent algorithmic model with a simple data modeling tool.

Why did the simple additive model work so well? One idea is that some of the characteristics in the model are acting as surrogates for certain interaction terms that are not explicitly in the model. Another reason is that the scorecard is really a sophisticated neural net. The inputs are the original inputs. Associated with each characteristic is a hidden node. The summation functions coming into the hidden nodes are the transformations defining the characteristics. The transfer functions of the hidden nodes are the step functions (compiled from the score weights)—all derived from the data. The final output is a linear function of the outputs of the hidden nodes. The result is highly nonlinear and interactive, when looked at as a function of the original inputs.

The Fat Scorecard study had an ingredient that is rare. We not only had the traditional test sample, but had three other test samples, taken one, two, and three years later. In this case, the Fat Scorecard outperformed the more traditional thinner scorecard for all four test samples. So the feared overfitting to the traditional test sample never materialized. To get a better handle on this you need an understanding of how the relationships between variables evolve over time.

I recently encountered another connection between algorithmic modeling and data modeling. In classical multivariate discriminant analysis, one assumes that the prediction variables have a multivariate normal distribution. But for a scorecard, the prediction variables are hundreds of attribute dummy variables, which are very nonnormal. However, if you apply the discriminant analysis algorithm to the attribute dummy variables, you can get a great algorithmic model, even though the assumptions of discriminant analysis are severely violated.

A SOLUTION TO THE OCCAM DILEMMA

I think that there is a solution to the Occam dilemma without resorting to goal-oriented arguments. Clients really do insist on interpretable functions, $f(\mathbf{x})$. Segmented palatable scorecards are very interpretable by the customer and are very accurate. Professor Breiman himself gave single trees an A+ on interpretability. The shallow-to-medium tree in a segmented scorecard rates an A++. The palatable scorecards in the leaves of the trees are built from interpretable (possibly complex) characteristics. Sometimes we can't implement them until the lawyers and regulators approve. And that requires super interpretability. Our more sophisticated products have 10 to 20 segments and up to 100 characteristics (not all in every segment). These models are very accurate and very interpretable.

I coined a phrase called the "Ping-Pong theorem." This theorem says that if we revealed to Professor Breiman the performance of our best model and gave him our data, then he could develop an algorithmic model using random forests, which would outperform our model. But if he revealed to us the performance of his model, then we could develop a segmented scorecard, which would outperform his model. We might need more characteristics, attributes and segments, but our experience in this kind of contest is on our side.

However, all the competing models in this game of Ping-Pong would surely be algorithmic models. But some of them could be interpretable.

THE ALGORITHM TUNING DILEMMA

As far as I can tell, all approaches to algorithmic model building contain tuning parameters, either explicit or implicit. For example, we use penalized objective functions for fitting and marginal contribution thresholds for characteristic selection. With experience, analysts learn how to set these tuning parameters in order to get excellent test sample or cross-validation results. However, in industry and academia, there is sometimes a little tinkering, which involves peeking at the test sample. The result is some bias in the test sample or cross-validation results. This is the same kind of tinkering that upsets test of fit pureness. This is a challenge for the algorithmic modeling approach. How do you optimize your results and get an unbiased estimate of the generalization error?

GENERALIZING THE GENERALIZATION ERROR

In most commercial applications of algorithmic modeling, the function, $f(\mathbf{x})$, is used to make decisions. In some academic research, classification is used as a surrogate for the decision process, and misclassification error is used as a surrogate for profit. However, I see a mismatch between the algorithms used to develop the models and the business measurement of the model's value. For example, at Fair, Isaac, we frequently maximize divergence. But when we argue the model's value to the clients, we don't necessarily brag about the great divergence. We try to use measures that the client can relate to. The ROC curve is one favorite, but it may not tell the whole story. Sometimes, we develop simulations of the client's business operation to show how the model will improve their situation. For example, in a transaction fraud control process, some measures of interest are false positive rate, speed of detection and dollars saved when 0.5% of the transactions are flagged as possible frauds. The 0.5% reflects the number of transactions that can be processed by the current fraud management staff. Perhaps what the client really wants is a score that will maximize the dollars saved in their fraud control system. The score that maximizes test set divergence or minimizes test set misclassifications does not do it. The challenge for algorithmic modeling is to find an algorithm that maximizes the generalization dollars saved, not generalization error.

We have made some progress in this area using ideas from support vector machines and boosting. By manipulating the observation weights used in standard algorithms, we can improve the test set performance on any objective of interest. But the price we pay is computational intensity.

MEASURING IMPORTANCE—IS IT REALLY POSSIBLE?

I like Professor Breiman's idea for measuring the importance of variables in black box models. A Fair, Isaac spin on this idea would be to build accurate models for which no variable is much more important than other variables. There is always a chance that a variable and its relationships will change in the future. After that, you still want the model to work. So don't make any variable dominant.

I think that there is still an issue with measuring importance. Consider a set of inputs and an algorithm that yields a black box, for which x_1 is important. From the "Ping Pong theorem" there exists a set of input variables, excluding x_1 and an algorithm that will yield an equally accurate black box. For this black box, x_1 is unimportant.

IN SUMMARY

Algorithmic modeling is a very important area of statistics. It has evolved naturally in environments with lots of data and lots of decisions. But you can do it without suffering the Occam dilemma; for example, use medium trees with interpretable GAMs in the leaves. They are very accurate and interpretable. And you can do it with data modeling tools as long as you (i) ignore most textbook advice, (ii) embrace the blessing of dimensionality, (iii) use constraints in the fitting optimizations (iv) use regularization, and (v) validate the results.

Comment

Emanuel Parzen

1. BREIMAN DESERVES OUR APPRECIATION

I strongly support the view that statisticians must face the crisis of the difficulties in their practice of regression. Breiman alerts us to systematic blunders (leading to wrong conclusions) that have been committed applying current statistical practice of data modeling. In the spirit of "statistician, avoid doing harm" I propose that the first goal of statistical ethics should be to guarantee to our clients that any mistakes in our analysis are unlike any mistakes that statisticians have made before.

The two goals in analyzing data which Leo calls prediction and information I prefer to describe as "management" and "science." Management seeks *profit*, practical answers (predictions) useful for decision making in the short run. Science seeks *truth*, fundamental knowledge about nature which provides understanding and control in the long run. As a historical note, Student's t-test has many scientific applications but was invented by Student as a management tool to make Guinness beer better (bitter?).

Breiman does an excellent job of presenting the case that the practice of statistical science, using only the conventional data modeling culture, needs reform. He deserves much thanks for alerting us to the algorithmic modeling culture. Breiman warns us that "if the model is a poor emulation of nature, the conclusions may be wrong." This situation, which I call "the right answer to the wrong question," is called by statisticians "the error of the third kind." Engineers at M.I.T. define "suboptimization" as "elegantly solving the wrong problem."

Emanuel Parzen is Distinguished Professor, Department of Statistics, Texas A&M University, 415 C Block Building, College Station, Texas 77843 (e-mail: eparzen@stat.tamu.edu).

Breiman presents the potential benefits of algorithmic models (better predictive accuracy thandata models, and consequently better information about the underlying mechanism and avoiding questionable conclusions which results from weak predictive accuracy) and support vector machines (which provide almost perfect separation and discrimination between two classes by increasing the dimension of the feature set). He convinces me that the methods of algorithmic modeling are important contributions to the tool kit of statisticians.

If the profession of statistics is to remain healthy, and not limit its research opportunities, statisticians must learn about the cultures in which Breiman works, *but also* about many other cultures of statistics.

2. HYPOTHESES TO TEST TO AVOID BLUNDERS OF STATISTICAL MODELING

Breiman deserves our appreciation for pointing out generic deviations from standard assumptions (which I call bivariate dependence and two-sample conditional clustering) for which we should routinely check. "Test null hypothesis" can be a useful algorithmic concept if we use tests that diagnose in a model-free way the directions of deviation from the null hypothesis model.

Bivariate dependence (correlation) may exist between features [independent (input) variables] in a regression causing them to be proxies for each other and our models to be unstable with different forms of regression models being equally well fitting. We need tools to routinely test the hypothesis of statistical independence of the distributions of independent (input) variables.

Two sample conditional clustering arises in the distributions of independent (input) variables to discriminate between two classes, which we call the conditional distribution of input variables X given each class. Class I may have only one mode (cluster) at low values of X while class II has two modes