Prediction, Estimation, and Attribution

Bradley Efron*
Stanford University

Abstract

The scientific needs and computational limitations of the Twentieth Century fashioned classical statistical methodology. Both the needs and limitations have changed in the Twenty-First, and so has the methodology. Large-scale prediction algorithms — neural nets, deep learning, boosting, support vector machines, random forests — have achieved star status in the popular press. They are recognizable as heirs to the regression tradition, but ones carried out at enormous scale and on titanic data sets. How do these algorithms compare with standard regression techniques such as ordinary least squares or logistic regression? Several key discrepancies will be examined, centering on the differences between prediction and estimation or prediction and attribution (significance testing.) Most of the discussion is carried out through small numerical examples.

Keywords: surface plus noise, random forests, ephemeral predictors, black box

*The author's work is supported in part by an award from the National Science Foundation, DMS 1608182.

1 Introduction

Statistical regression methods go back to Gauss and Legendre in the early 1800s, and especially to Galton in 1877. During the 20th Century, regression ideas were adapted to a variety of important statistical tasks: the prediction of new cases, the estimation of regression surfaces, and the assignment of significance to individual predictors, what I've called "attribution" in the title of this article. Many of the most powerful ideas of 20th Century statistics are connected with regression: least squares fitting, logistic regression, generalized linear models, ANOVA, predictor significance testing, regression to the mean.

The 21st Century has seen the rise of a new breed of what can be called "pure prediction algorithms" — neural nets, deep learning, boosting, support vector machines, random forests — recognizably in the Gauss–Galton tradition, but able to operate at immense scales, with millions of data points and even more millions of predictor variables. Highly successful at automating tasks like online shopping, machine translation, and airline information, the algorithms (particularly deep learning) have become media darlings in their own right, generating an immense rush of interest in the business world. More recently, the rush has extended into the world of science, a one-minute browser search producing "Deep learning in biology"; "Computational linguistics and deep learning"; and "Deep learning for regulatory genomics".

How do the pure prediction algorithms relate to traditional regression methods? That is the central question pursued in what follows. A series of salient differences will be examined– differences of assumption, scientific philosophy, and goals. The story is a complicated one, with no clear winners or losers; but a rough summary, at least in my mind, is that the pure prediction algorithms are a powerful addition to the statistician's armory, yet substantial further development is needed for their routine scientific applicability. Such development is going on already in the statistical world, and has provided a welcome shot of energy into our discipline.

This article, originally a talk, is written with a broad brush, and is meant to be descriptive of current practice rather than normative of how things have to be. No previous knowledge of the various prediction algorithms is assumed, though that will sorely underestimate many

readers.

This is not a research paper, and most of the argumentation is carried out through numerical examples. These are of small size, even miniscule by current prediction standards. A certain giganticism has gripped the prediction literature, with swelling prefixes such as *tera-*, *peta-*, and *exa-* bestowing bragging rights. But small data sets can be better for exposing the limitations of a new methodology.

An excellent reference for prediction methods, both traditional and modern, is Hastie, Tibshirani, and Friedman (2009). Very little will be said here about the mechanics of the pure prediction algorithms: just enough, I hope, to get across the idea of how radically different they are from their traditional counterparts.

2 Surface plus noise models

For both the prediction algorithms and traditional regression methods, we will assume that the data $\boldsymbol{d}$ available to the statistician has this structure:

$$\boldsymbol{d} = \{(x_i, y_i),\ i = 1, 2, \ldots, n\}; \tag{2.1}$$

here x_i is a p-dimensional vector of predictors taking its value in a known space $\mathcal{X}$ contained in $\mathbb{R}^p$, and y_i is a real-valued response. The n pairs are assumed to be independent of each other. More concisely we can write

$$\boldsymbol{d} = \{\boldsymbol{x}, \boldsymbol{y}\}, \tag{2.2}$$

where $\boldsymbol{x}$ is the $n \times p$ matrix having x_i^t as its ith row, and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^t$. Perhaps the most traditional of traditional regression models is

$$y_i = x_i^t \beta + \epsilon_i \qquad (i = 1, 2, \ldots, n), \tag{2.3}$$

$\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, i.e., "ordinary least squares with normal errors." Here β is an unknown p-dimensional parameter vector. In matrix notation,

$$\boldsymbol{y} = \boldsymbol{x}\beta + \boldsymbol{\epsilon}. \tag{2.4}$$

For any choice of x in $\mathcal{X}$, model (2.3) says that the response y has expectation $\mu = x^t\beta$, so $y \sim \mathcal{N}(\mu, \sigma^2)$. The linear surface $\mathcal{S}_\beta$,

$$\mathcal{S}_\beta = \{\mu = x^t\beta,\ x \in \mathcal{X}\}, \tag{2.5}$$

contains all the true expectations, but the truth is blurred by the noise terms ϵ_i.

More generally, we might expand (2.3) to

$$y_i = s(x_i, \beta) + \epsilon_i \qquad (i = 1, 2, \ldots, n), \tag{2.6}$$

where $s(x, \beta)$ is some known functional form that, for any fixed value of β, gives expectation $\mu = s(x, \beta)$ as a function of $x \in \mathcal{X}$. Now the surface of true expectations, i.e., the *regression surface*, is

$$\mathcal{S}_\beta = \{\mu = s(x, \beta),\ x \in \mathcal{X}\}\,. \tag{2.7}$$

Most traditional regression methods depend on some sort of *surface plus noise* formulation (though "plus" may refer to, say, binomial variability). The surface describes the scientific truths we wish to learn, but we can only observe points on the surface obscured by noise. The statistician's traditional estimation task is to learn as much as possible about the surface from the data $\boldsymbol{d}$.

Figure 1 shows a small example, taken from a larger data set in Efron and Feldman (1991): $n = 164$ male doctors volunteered to take the cholesterol-lowering drug cholostyramine. Two numbers were recorded for each doctor,

$$x_i = \text{normalized compliance} \quad \text{and} \quad y_i = \text{observed cholesterol decrease.} \tag{2.8}$$

Compliance, the proportion of the intended dose actually taken, ranged from zero to 100%, -2.25 to 1.97 on the normalized scale, and of course it was hoped to see larger cholesterol decreases for the better compliers.

A normal regression model (2.6) was fit, with

$$s(x_i, \beta) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3, \tag{2.9}$$

in other words, a cubic regression model. The black curve is the estimated surface

$$\widehat{\mathcal{S}} = \left\{ s\left(x, \hat{\beta}\right) \text{ for } x \in \mathcal{X} \right\}, \tag{2.10}$$

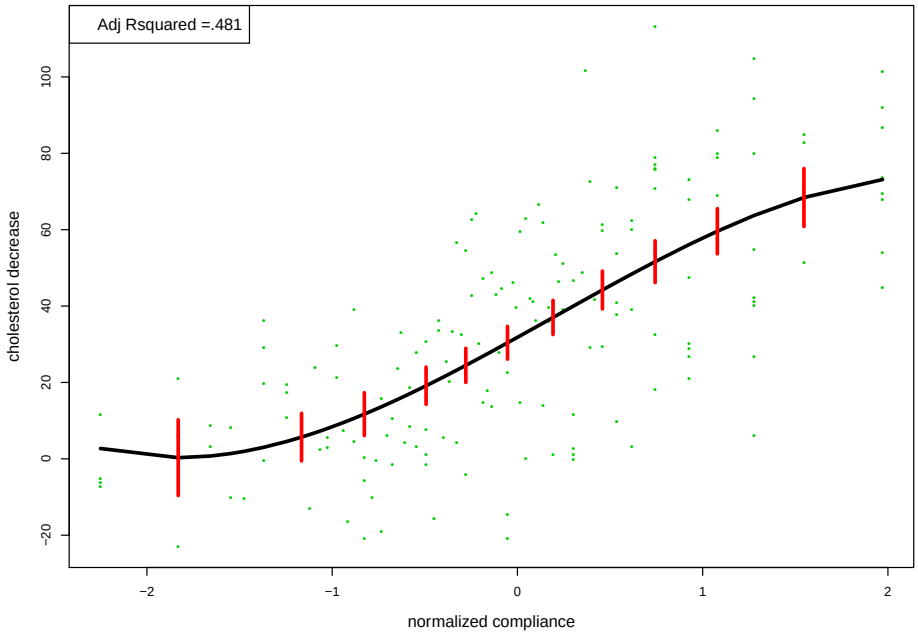

Figure 1: Black curve is OLS fitted regression to cholostyramine data (dots); vertical bars indicate ± one standard error estimation.

fit by maximum likelihood or, equivalently, by ordinary least squares (OLS). The vertical bars indicate ± one standard error for the estimated values $s(x, \hat{\beta})$, at 11 choices of x, showing how inaccurate $\widehat{\mathcal{S}}$ might be as an estimate of the true $\mathcal{S}$.

That is the estimation side of the story. As far as attribution is concerned, only $\hat{\beta}_0$ and $\hat{\beta}_1$ were significantly nonzero. The adjusted R^2 was 0.482, a traditional measure of the model's predictive power.

Another mainstay of traditional methodology is logistic regression. Table 1 concerns the *neonate data* (Mediratta et al., 2019): $n = 812$ very sick babies at an African facility were observed over the course of one year, 605 who lived and 207 who died. Eleven covariates were measured at entry: gestational age, body weight, apgar score, etc., so x_i in (2.1) was 11-dimensional, while y_i equaled 0 or 1 as the baby lived or died. This is a surface plus noise model, with a linear logistic surface and Bernoulli noise.

The 11 predictor variables were standardized to have mean 0 and variance 1, after which logistic regression analysis was carried out. Table 1 shows some of the output. Columns 1 and 2 give estimates and standard errors for the regression coefficients (which amount to a description of the estimated linear logistic surface $\widehat{\mathcal{S}}$ and its accuracy).

Table 1: Logistic regression analysis of neonate data. Significant two-sided p-values indicated for 6 of 11 predictors; estimated logistic regression made 18% prediction errors.

	estimate	st. error	p-value
Intercept	−1.549	.457	.001***
gest	−.474	.163	.004**
ap	−.583	.110	.000***
bwei	−.488	.163	.003**
resp	.784	.140	.000***
cpap	.271	.122	.027*
ment	1.105	.271	.000***
rate	−.089	.176	.612
hr	.013	.108	.905
head	.103	.111	.355
gen	−.001	.109	.994
temp	.015	.124	.905

Column 3 shows standard two-sided p-values for the 11 variables, six of which are significantly nonzero, five of them strongly so. This is the attribution part of the analysis. As far as prediction is concerned, the fitted logistic regression model gave an estimated probability p_i of death for each baby. The prediction rule

$$\text{if} \quad \begin{matrix} p_i > 0.25 \\ p_i \le 0.25 \end{matrix} \quad \text{predict} \quad \begin{matrix} \text{dies} \\ \text{lives} \end{matrix} \tag{2.11}$$

had an empirical error rate of 18%. (Threshold 0.25 was chosen to compensate for the smaller proportion of deaths.)

All of this is familiar stuff, serving here as a reminder of how traditional regression analyses typically begin: a description of the underlying scientific truth (the "surface") is formulated, along with a model of the errors that obscure direct observation. The pure prediction algorithms follow a different path, as described in Section 3.

The left panel of Figure 2 shows the surface representation of a scientific icon, Newton's

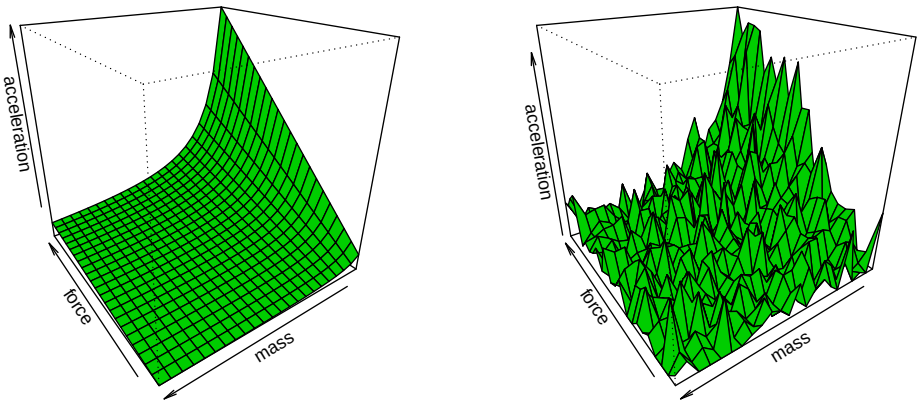

Figure 2: On left, a surface depicting Newton's Second Law of motion, acceleration = force/mass; on right, a noisy version.

Second Law of motion,

$$\text{acceleration} = \frac{\text{force}}{\text{mass}}. \tag{2.12}$$

It is pleasing to imagine the Second Law falling full-born out of Newton's head, but he was a master of experimentation. The right panel shows a (fanciful) picture of what experimental data might have looked like.[1]

In the absence of genius-level insight, statistical estimation theory is intended as an instrument for peering through the noisy data and discerning a smooth underlying truth. Neither the cholostyramine nor the neonate examples is as fundamental as Newton's Second Law but they share the goal of extracting dependable scientific structure in a noisy environment. The noise is ephemeral but the structure, hopefully, is eternal, or at least long-lasting (see Section 8).

3 The pure prediction algorithms

The 21st Century[2] has seen the rise of an extraordinary collection of prediction algorithms: *random forests*, *gradient boosting*, *support vector machines*, *neural nets* (including *deep learning*), and others. I will refer to these collectively as the "pure prediction algorithms" to dif-

[1] A half-century earlier, Galileo famously used inclined planes and a water clock to estimate the acceleration of falling objects.

[2] Actually, the "long 21st Century," much of the activity beginning in the 1990s.

ferentiate them from the traditional prediction methods illustrated in the previous section. Some spectacular successes — machine translation, iPhone's Siri, facial recognition, championship chess and Go programs — have elicited a tsunami of public interest. If media attention is the appropriate metric, then the pure prediction algorithms are our era's statistical stars.

The adjective "pure" is justified by the algorithms' focus on prediction, to the neglect of estimation and attribution. Their basic strategy is simple: to go directly for high predictive accuracy and not worry about surface plus noise models. This has some striking advantages and some drawbacks, too. Both advantages and drawbacks will be illustrated in what follows.

A prediction algorithm is a general program for inputting a data set $\boldsymbol{d} = \{(x_i, y_i), i = 1, 2, \ldots, n\}$ (2.1) and outputting a rule $f(x, \boldsymbol{d})$ that, for any predictor vector x, yields a prediction

$$\hat{y} = f(x, \boldsymbol{d}). \tag{3.1}$$

We hope that the *apparent error rate* of the rule, for classification problems the proportion of cases where $\hat{y}_i \neq y_i$,

$$\widehat{\text{err}} = \#\left\{f(x_i, \boldsymbol{d}) \neq y_i\right\}/n \tag{3.2}$$

is small. More crucially, we hope that the *true error rate*

$$\text{Err} = E\left\{f(X, \boldsymbol{d}) \neq Y\right\} \tag{3.3}$$

is small, where (X, Y) is a random draw from whatever probability distribution gave the (x_i, y_i) pairs in $\boldsymbol{d}$; see Section 6. Random forests, boosting, deep learning, etc. are algorithms that have well-earned reputations for giving small values of Err in complicated situations.

Besides being very different from traditional prediction methods, the pure prediction algorithms are very different from each other. The least intricate and easiest to describe is *random forests* (Breiman, 2001). For dichotomous prediction problems, like that for the neonate babies, random forests depends on ensembles of *classification trees*.

Figure 3 shows a single classification tree obtained by applying the R program `Rpart`[3] to the neonates. At the top of the tree all 812 babies were divided into two groups: those with cpap (an airway blockage signifier) less than threshold 0.6654 were put into the more-favorable prognosis group to the left; those with cpap $\geq$ 0.6654 were shunted right into

[3]A knockoff of CART (Breiman, Friedman, Olshen, and Stone, 1984).

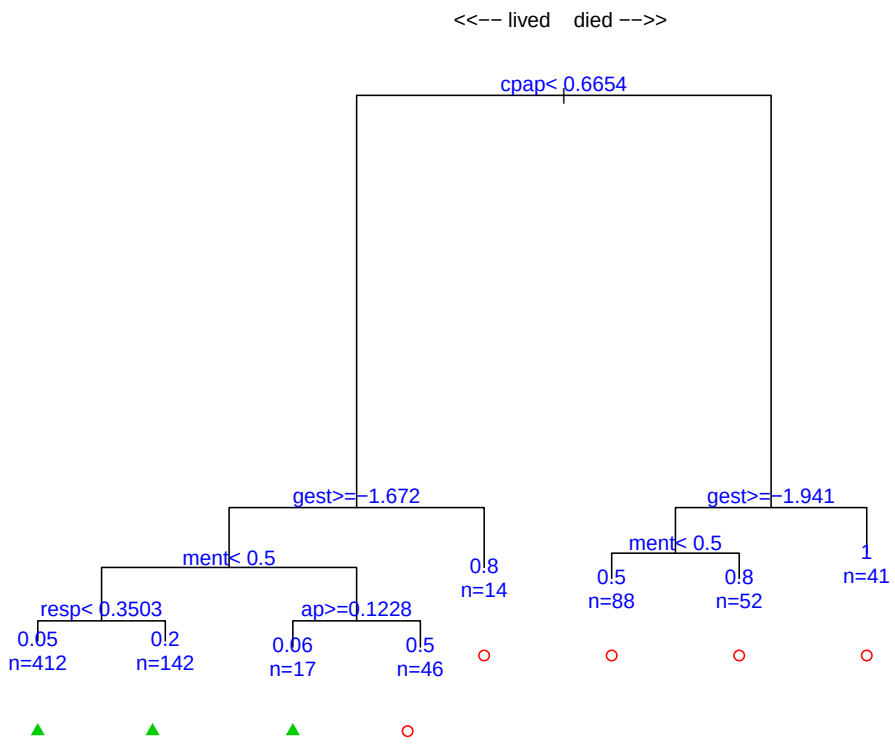

Figure 3: Classification tree for neonate data. Triangled terminal nodes predict *baby lives*, circled predict *baby dies*; the rule has apparent prediction error rate 17% and cross-validated rate 18%.

the less-favorable prognosis group. The predictor cpap and threshold 0.6654 were chosen to maximize, among all possible (predictor, threshold) choices, the difference in observed death rates between the two groups.[4] Next, each of the two groups was itself split in two, following the same Gini criterion. The splitting process continued until certain stopping rules were invoked, involving very small or very homogeneous groups.

At the bottom of Figure 3, the splitting process ended at eight *terminal nodes*: the node at far left contained 412 of the original 812 babies, only 5% of which were deaths; the far right node contained 41 babies, all of which were deaths. Triangles indicate the three terminal nodes having death proportions less than the original proportion 25.5%, while circles indicate proportions exceeding 25.5%. The prediction rule is "lives" at triangles, "dies" at circles.

[4]More precisely: if $n_{\rm L}$ and $n_{\rm R}$ are the numbers in the left and right groups, and $\hat{p}_{\rm L}$ and $\hat{p}_{\rm R}$ the proportions of deaths, then the algorithm minimized the *Gini criterion* $n_{\rm L}\hat{p}_{\rm L}(1-\hat{p}_{\rm L}) + n_{\rm R}\hat{p}_{\rm R}(1-\hat{p}_{\rm R})$. This equals $n\hat{p}(1-\hat{p}) - (n_{\rm L}n_{\rm R}/n)(\hat{p}_{\rm L}-\hat{p}_{\rm R})^2$, where $n = n_{\rm L}+n_{\rm R}$ and $\hat{p} = (n_{\rm L}\hat{p}_{\rm L}+n_{\rm R}\hat{p}_{\rm R})/n$, so that minimizing Gini's criterion is equivalent to maximizing $(\hat{p}_{\rm L}-\hat{p}_{\rm R})^2$, for any given values of $n_{\rm L}$ and $n_{\rm R}$.

If a new baby arrived at the facility with vector x of 11 measurements, the doctors could predict life or death by following x down the tree to termination.

This prediction rule has apparent error rate 17%, taking the observed node proportions, 0.05, etc., as true. Classification trees have a reputation for being greedy overfitters, but in this case a ten-fold cross-validation analysis gave error rate 18%, nearly the same. The careful "traditional" analysis of the neonate data in Mediratta et al. (2019) gave a cross-validated error rate of 20%. It is worth noting that the splitting variables in Figure 3 agree nicely with those found significant in Table 1.

So far so good for regression trees, but with larger examples they have earned a reputation for poor predictive performance; see Section 9.2 of Breiman (2001). As an improvement, Breiman's *random forest* algorithm relies on averaging a large number of bootstrap trees, each generated as follows:

1. Draw a nonparametric bootstrap sample $\boldsymbol{d}^*$ from the original data $\boldsymbol{d}$, i.e., a random sample of n pairs (x_i, y_i) chosen *with* replacement from $\boldsymbol{d}$.

2. Construct a classification tree from $\boldsymbol{d}^*$ as before, but choose each split using only a random subset of p^* predictors chosen independently from the original p ($p^* \doteq \sqrt{p}$).

Having generated, say, B such classification trees, a newly observed x is classified by following it down to termination in each tree; finally, $\hat{y} = f(x, \boldsymbol{d})$ is determined by the majority of the B votes. Typically B is taken in the range 100 to 1000.

Random forests was applied to the neonate prediction problem, using R program `randomForest`, with the results graphed in Figure 4. The prediction error rate[5] is shown as a function of the number of bootstrap trees sampled. In all, $B = 501$ trees were used but there was not much change after 200. The overall prediction error rate fluctuated around 17%, only a small improvement over the 18% cross-validated rate in Figure 3. Random forests is shown to better advantage in the microarray example of Section 4.

Random forests begins with the p columns of x as predictors, but then coins a host of new predictors via the splitting process (e.g., "cpap less than or greater than 0.6654"). The new variables bring a high degree of interaction to the analysis, for instance between cpap

[5]These are "out of bag" estimates of prediction error, a form of cross-validation explained in the Appendix.

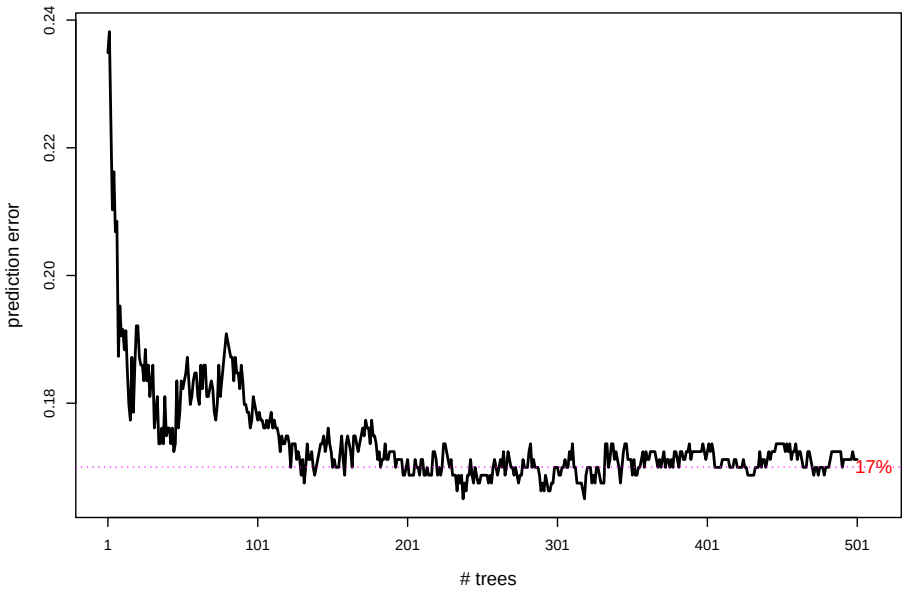

Figure 4: Random forest prediction error rate for neonate data, as a function of number of bootstrapped trees; it has cross-validated error rate 17%.

and gest in Figure 3. Though carried out differently, high interactivity and fecund coinage of predictor variables are hallmarks of all pure prediction algorithms.

4 A microarray prediction problem

Newsworthy breakthroughs for pure prediction algorithms have involved truly enormous data sets. The original English/French translator tool on Google, for instance, was trained on millions of parallel snippets of English and French obtained from Canadian and European Union legislative records. There is nothing of that size to offer here but, as a small step up from the neonate data, we will consider a microarray study of prostate cancer.

The study involved $n = 102$ men, 52 cancer patients and 50 normal controls. Each man's genetic expression levels were measured on a panel of $p = 6033$ genes,

$$x_{ij} = \text{activity of } j\text{th gene for } i\text{th man}, \tag{4.1}$$

$i = 1, 2, \ldots, 102$ and $j = 1, 2, \ldots, 6033$. The $n \times p$ matrix $\boldsymbol{x}$ is much wider than it is tall in this case, "wide data" being the trendy name for $p \gg n$ situations, as contrasted with the $p \ll n$ "tall" data sets traditionally favored.

Random forests was put to the task of predicting *normal* or *cancer* from a man's microarray measurements. Following standard procedure, the 102 men were randomly divided into *training* and *test* sets of size 51,[6] each having 25 normal controls and 26 cancer patients.

The training data $\boldsymbol{d}_{\text{train}}$ consists of 51 (x, y) pairs, x a vector of $p = 6033$ genetic activity measurements and y equal 0 or 1 indicating a normal or cancer patient. R program `randomForest` yielded prediction rule $f(x, \boldsymbol{d}_{\text{train}})$. This rule was applied to the test set, yielding predictions $\hat{y}_i = f(x_i, \boldsymbol{d}_{\text{train}})$ for the 51 test subjects.

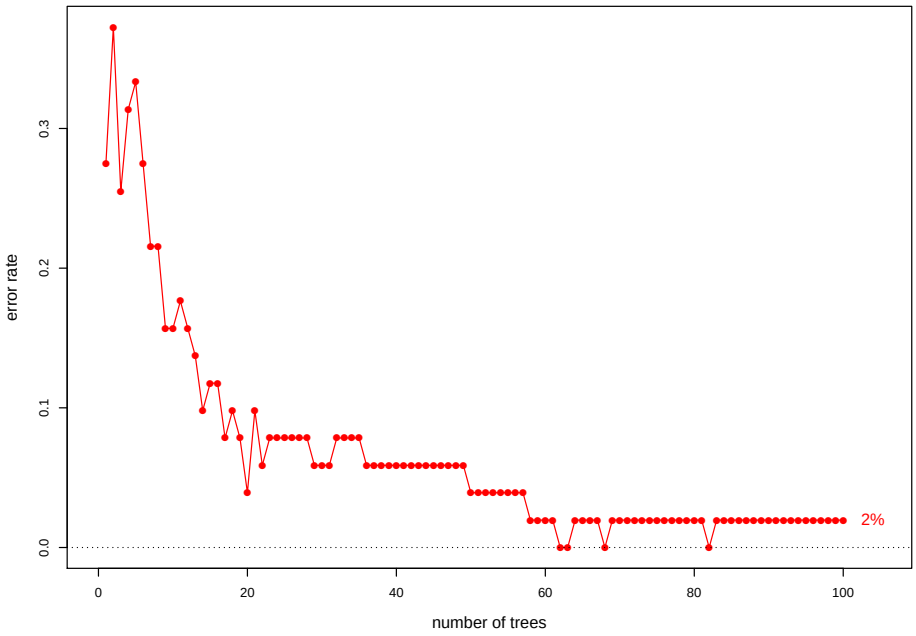

Figure 5: Test set error rate for random forests applied to prostate cancer microarray study, as a function of number of bootstrap trees.

Figure 5 graphs the test set error rate as the number of random forest trees increased. After 100 trees, the test set error rate was 2%. That is, $\hat{y}_i$ agreed with y_i, the actual outcome, for 50 of the 51 test set subjects: an excellent performance by anyone's standards! This wasn't a particularly lucky result. Subsequently, random training/test set splits were carried out 100 times, each time repeating the random forest calculations in Figure 5 and counting the number of test set errors. The modal number of errors was 2, as seen in Table 2, with "1 prediction error" occuring frequently.

[6]It would be more common to choose, say, 81 training and 21 test, but for the comparisons that follow it will be helpful to have larger test sets.

Table 2: Number of random forest test set errors in 100 random training/test splits of prostate data.

errors	0	1	2	3	4	5	7
frequency	3	26	39	12	5	4	1

A classification tree can be thought of as a function $f(x)$ taking values 0 or 1 for x in its sample space $\mathcal{X}$. The tree in Figure 3 partitions the 11-dimensional space $\mathcal{X}$ into 8 rectangular regions, three of which having $y = 0$ and five having $y = 1$. A simpler function is obtained by stopping the division process after the first split, in which case $\mathcal{X}$ is divided into just two regions, cpap < 0.6654 and cpap ≥ 0.6654. Such simple trees are picturesquely known as "stumps".

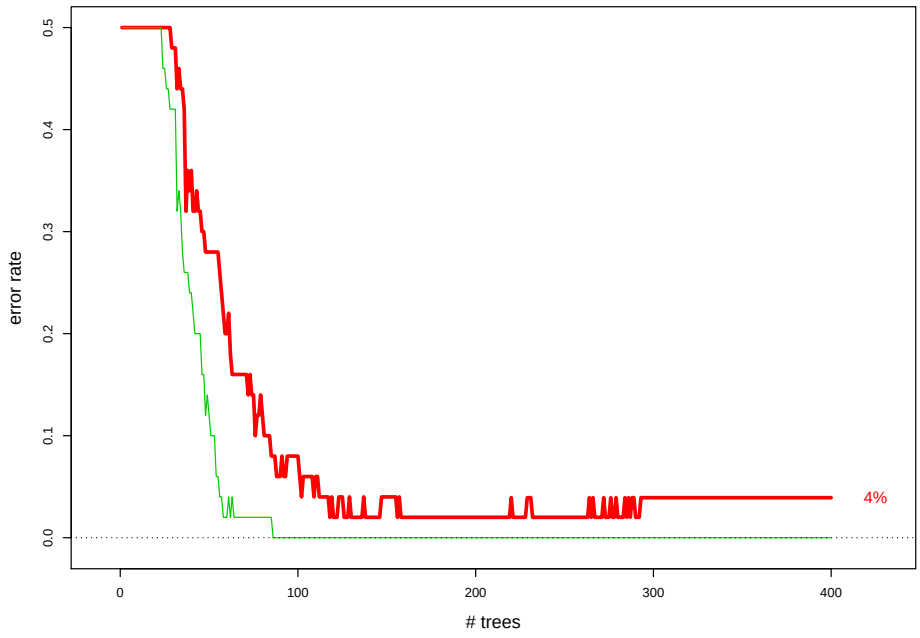

Figure 6: Test set error for boosting algorithm `gbm` applied to prostate cancer data. Thin curve is training set error, which went to zero at step 86.

This brings up another prominent pure prediction method, *boosting*. Figure 6 shows the results of applying the R program `gbm` (for gradient boosting modeling) to the prostate

cancer prediction problem.[7] Gbm sequentially fits a weighted sum of classification trees,

$$\sum_{k=1}^{K} w_k f_k(x), \tag{4.2}$$

at step $k+1$ choosing tree $f_{k+1}(x)$ to best improve the fit. The weights w_k are kept small to avoid getting trapped in a bad sequence. After 400 steps, Figure 6 shows a test sample error of 4%, that is, two mistakes out of 51; once again an impressive performance. (The examples in Hastie et al. (2009) show gbm usually doing a little better than random forests.)

In the evocative language of boosting, the stumps going into Figure 6's construction are called "weak learners": any one of them by itself would barely lower prediction errors below 50%. That a myriad of weak learners can combine so effectively is a happy surprise and a central advance of the pure prediction enterprise. In contrast, traditional methods focus on *strong* individual predictors, as with the asterisks in Table 1, a key difference to be discussed in subsequent sections.

The light curve in Figure 6 traces the gbm rule's error rate on its own training set. It went to zero at step 86 but training continued on, with some improvement in test error. Cross-validation calculations give some hint of when to stop the fitting process — here we would have done better to stop at step 200 — but it's not a settled question.

The umbrella package `keras` was used to apply neural nets/deep learing to the prostate data. Results were poorer than for random forests or gbm: 7 or 8 errors on the test set depending on the exact stopping rule. A suppport vector machine algorithm did worse still, with 11 test set errors.

The deep learning algorithm is much more intricate than the others, reporting "780,736 parameters used", these being internally adjusted *tuning parameters* set by cross-validation. That this is possible at all is a tribute to modern computing power, the underlying enabler of the pure prediction movement.

[7] Applied with $d = 1$, i.e., fitting stumps, and shrinkage factor 0.1.

5 Advantages and disadvantages of prediction

For those of us who have struggled to find "significant" genes in a microarray study,[8] the almost perfect prostate cancer predictions of random forests and gbm have to come as a disconcerting surprise. Without discounting the surprise, or the ingenuity of the prediction algorithms, a contributing factor might be that prediction is an easier task than either attribution or estimation. This is a difficult suspicion to support in general, but a couple of examples help make the point.

Regarding estimation, suppose that we observe 25 independent replications from a normal distribution with unknown expectation μ,

$$x_1, x_2, \ldots, x_{25} \overset{\text{ind}}{\sim} \mathcal{N}(\mu, 1), \tag{5.1}$$

and consider estimating μ with either the sample mean $\bar{x}$ or the sample median $\breve{x}$. As far as squared error is concerned, the mean is an overwhelming winner, being more than half again more efficient,

$$E\{(\breve{x} - \mu)^2\}/E\{(\bar{x} - \mu)^2\} \doteq 1.57. \tag{5.2}$$

Suppose instead that the task is to predict the value of a new, independent realization $X \sim \mathcal{N}(\mu, 1)$. The mean still wins, but now by only 2%,

$$E\{(X - \breve{x})^2\}/E\{(X - \bar{x})^2\} = 1.02. \tag{5.3}$$

The reason, of course, is that most of the prediction error comes from the variability of X, which neither $\bar{x}$ nor $\breve{x}$ can cure.[9]

Prediction is easier than estimation, at least in the sense of being more forgiving. This allows for the use of inefficient estimators like the gbm stumps, that are convenient for mass deployment. The pure prediction algorithms operate nonparametrically, a side benefit of not having to worry much about estimation efficiency.

[8] See Figure 15.5 of Efron and Hastie (2016).

[9] This imagines that we have a single new observation to predict. Suppose instead that we have m new observations $X_1, X_2, \ldots, X_m \overset{\text{iid}}{\sim} \mathcal{N}(\mu, 1)$, and that we wish to predict their mean $\bar{X}$. With $m = 10$ the efficiency ratio is $E\{(\bar{X} - \breve{x})^2\}/E\{(\bar{X} - \bar{x})^2\} = 1.16$; with $m = 100$, 1.46; and with $m = \infty$, 1.57. One can think of estimation as the prediction of future mean values.

For the comparison of prediction with attribution we consider an idealized version of a microarray study involving n subjects, $n/2$ healthy controls and $n/2$ sick patients: any one subject provides a vector of measurements on N genes, $\boldsymbol{X} = (X_1, X_2, \ldots, X_N)^t$, with

$$X_j \overset{\text{ind}}{\sim} \mathcal{N}(\pm\delta_j/2c, 1) \qquad \left(c = \sqrt{n/4}\right), \tag{5.4}$$

for $j = 1, 2, \ldots, N$, "plus" for the sick and "minus" for the healthy; δ_j is the effect size for gene j. Most of the genes are null, $\delta_j = 0$, say N_0 of them, but a small number N_1 have δ_j equal a positive value Δ,

$$N_0 : \delta_j = 0 \quad \text{and} \quad N_1 : \delta_j = \Delta. \tag{5.5}$$

A new person arrives and produces a microarray of measurements $\boldsymbol{X} = (X_1, X_2, \ldots, X_N)^t$ satisfying (5.4) but without us knowing the person's healthy/sick status; that is, without knowledge of the $\pm$ value. *Question*: How small can N_1/N_0 get before prediction becomes impossible? The answer, motivated in the Appendix, is that asymptotically as $N_0 \to \infty$, accurate prediction is possible if

$$N_1 = O(N_0^{1/2}), \tag{5.6}$$

but not below that.

By contrast, the Appendix shows that effective attribution requires

$$N_1 = O(N_0). \tag{5.7}$$

In terms of "needles in haystacks" (Johnstone and Silverman, 2004), attribution needs an order of magnitude more needles than prediction. The prediction tactic of combining weak learners is not available for attribution, which, almost by definition, is looking for *strong* individual predictors. At least in this example, it seems fair to say that prediction is much easier than attribution.

The three main regression categories can usefully be arranged in order

$$\text{prediction} \quad \cdots \quad \text{estimation} \quad \cdots \quad \text{attribution}, \tag{5.8}$$

with estimation in a central position and prediction and attribution more remote from each other. Traditionally, estimation is linked to attribution through p-values and confidence intervals, as in Table 1. Looking in the other direction, good estimators, when they are

available, are usually good predictors. Both prediction and estimation focus their output on the n side of the $n \times p$ matrix $\boldsymbol{x}$, while attribution focuses on the p side. Estimation faces both ways in (5.8).

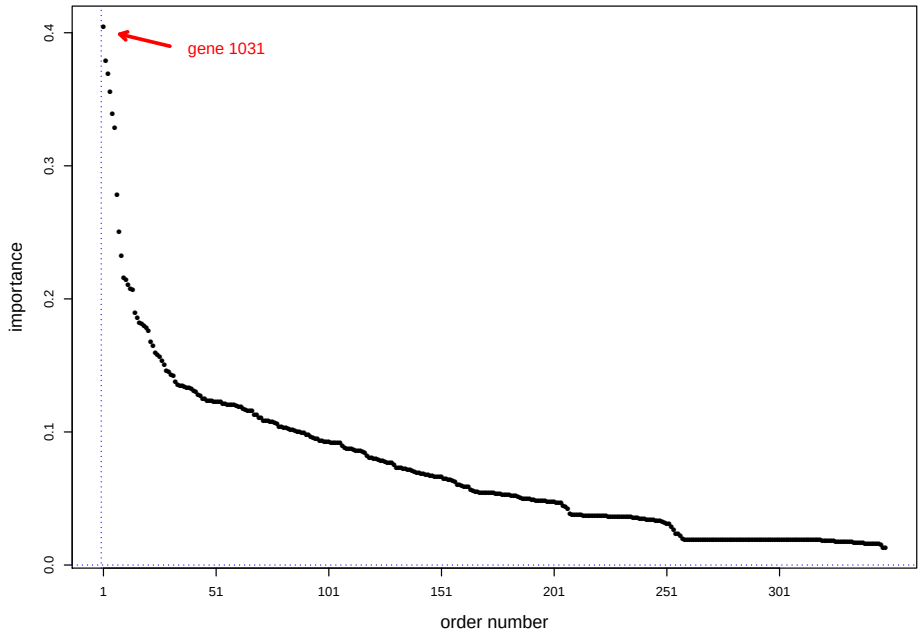

Figure 7: Random forest importance measures for prostate cancer prediction rule of Figure 5, plotted in order of declining importance.

The `randomForest` algorithm *does* attempt to connect prediction and attribution. Along with the predictions, an *importance measure*[10] is computed for each of the p predictor variables. Figure 7 shows the ordered importance scores for the prostate cancer application of Figure 5. Of the $p = 6033$ genes, 348 had positive scores, these being the genes that *ever* were chosen as splitting variables. Gene 1031 achieved the most importance, with about 25 others above the sharp bend in the importance curve. Can we use the importance scores for attribution, as with the asterisks in Table 1?

In this case, the answer seems to be no. I removed gene 1031 from the data set, reducing the data matrix $\boldsymbol{x}$ to 102×6032, and reran the `randomForest` prediction algorithm. Now

[10]There are several such measures. The one in Figure 7 relates to Gini's criterion, Section 3. At the conclusion of the algorithm we have a long list of all the splits in all the bootstrap trees; a single predictor's importance score is the sum of the decreases in the Gini criterion over all splits where that predictor was the splitting variable.

Table 3: Number of test set errors for prostate cancer random forest predictions, removing top predictors shown in Figure 7.

# removed	0	1	5	10	20	40	80	160	348
# errors	1	0	3	1	1	2	2	2	0

the number of test set prediction errors was zero. Removing the most important five genes, the most important 10, ..., the most important 348 genes had similarly minor effects on the number of test set prediction errors, as shown in Table 3.

At the final step, *all* of the genes involved in constructing the original prediction rule of Figure 5 had been removed. Now $\boldsymbol{x}$ was 102×5685, but the random forest rule based on the reduced data set $\boldsymbol{d} = \{\boldsymbol{x}, \boldsymbol{y}\}$ still gave excellent predictions. As a matter of fact, there were *zero* test set errors for the realization shown in Table 3. The prediction rule at the final step yielded 364 "important" genes, disjoint from the original 348. Removing all $712 = 348 + 364$ genes from the prediction set — so now $\boldsymbol{x}$ was 102×5321 — still gave a random forest prediction rule that made only one test set error.

The "weak learners" model of prediction seems dominant in this example. Evidently there are a great many genes weakly correlated with prostate cancer, which can be combined in different combinations to give near-perfect predictions. This is an advantage if prediction is the only goal, but a disadvantage as far as attribution is concerned. Traditional methods of attribution operate differently, striving as in Table 1 to identify a small set of causal covariates (even if strict causality can't be inferred).

The pure prediction algorithms' penchant for coining weakly correlated new predictors moves them in the opposite direction from attribution. Section 9 addresses *sparsity* — a working assumption of there being only a few important predictors — which is not at all the message conveyed by Table 3.

6 The training/test set paradigm

A crucial ingredient of modern prediction methodology is the training/test set paradigm: the data $\boldsymbol{d}$ (2.1) is partitioned into a training set $\boldsymbol{d}_{\text{train}}$ and a test set $\boldsymbol{d}_{\text{test}}$; a prediction rule $f(x, \boldsymbol{d}_{\text{train}})$ is computed using only the data $\boldsymbol{d}_{\text{train}}$; finally, $f(x, \boldsymbol{d}_{\text{train}})$ is applied to the cases in $\boldsymbol{d}_{\text{test}}$, yielding an honest estimate of the rule's error rate. But honest doesn't mean perfect.

This paradigm was carried out in Section 4 for the prostate cancer microarray study, producing an impressively small error rate estimate of 2% for random forests.[11] This seemed extraordinary to me. Why not use this rule to diagnose prostate cancer based on the vector of a new man's 6033 gene expression measurements? The next example suggests how this might go wrong.

The training and test sets for the prostate cancer data of Section 4 were obtained by *randomly* dividing the 102 men into two sets of 51, each with 25 normal controls and 26 cancer patients. Randomization is emphasized in the literature as a guard against bias. Violating this advice, I repeated the analysis, this time selecting for the training set the 25 normal controls and 26 cancer patients with the lowest ID numbers. The test set was the remaining 51 subjects, those with the highest IDs, and again contained 25 normal controls and 26 cancer patients.

In the re-analysis `randomForest` didn't perform nearly as well as in Figure 5: $f(x, \boldsymbol{d}_{\text{train}})$ made 12 wrong predictions on $\boldsymbol{d}_{\text{test}}$ with error rate 24%, rather than the previous 2%, as graphed in Figure 8. The boosting algorithm `gbm` was just as bad, producing prediction error rate 28% (14 wrong predictions) as shown in Figure 9.

Why are the predictions so much worse now? It isn't obvious from inspection but the prostate study subjects might have been collected in the order listed,[12] with some small methodological differences creeping in as time progressed. Perhaps all those weak learners going into `randomForest` and `gbm` were vulnerable to such differences. The prediction literature uses *concept drift* as a label for this kind of trouble, a notorious example being the Google flu

[11] Taking account of the information in Table 2, a better error rate estimate is 3.7%.

[12] A singular value decomposition of the normal-subject data had second principal vector sloping upwards with ID number, but this wasn't true for the cancer patient data.

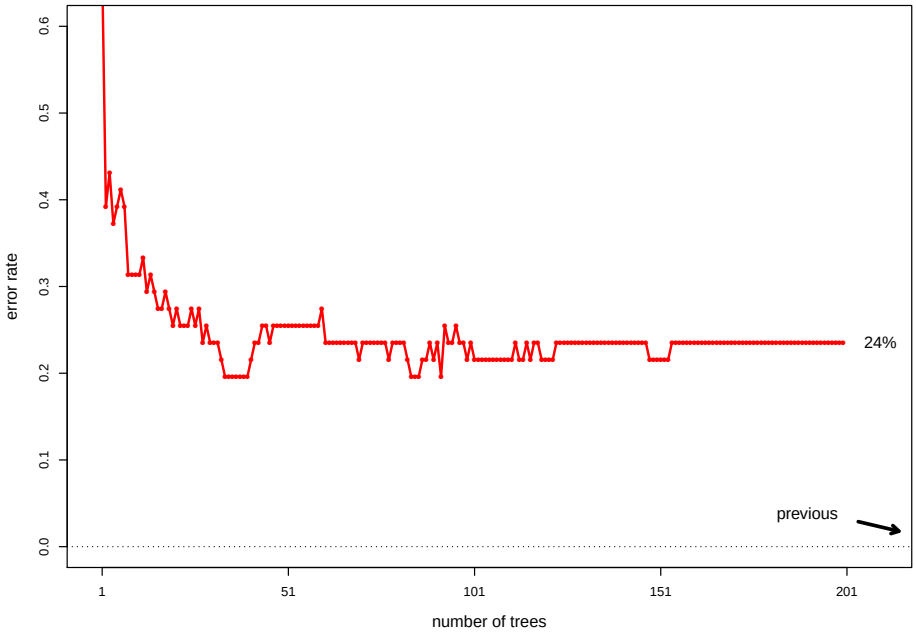

Figure 8: `randomForest` test set error for prostate cancer microarray study, now with training/test sets determined by early/late ID number. Results are much worse than in Figure 5.

predictor, which beat the CDC for a few years before failing spectacularly.[13] Choosing one's test set by random selection sounds prudent but it is guaranteed to hide any drift effects.

Concept drift gets us into the question of what our various regression methods, new and old, are supposed to be telling us. Science, historically, has been the search for the underlying truths that govern our universe: truths that are supposed to be eternal, like Newton's laws. The eternal part is clear enough in physics and astronomy — the speed of light, $E = mc^2$, Hubble's law — and perhaps in medicine and biology, too, e.g., DNA and the circulation of blood. But modern science has moved on to fields where truth may be more contingent, such as economics, sociology, and ecology.

Without holding oneself to Newtonian standards, traditional estimation and attribution usually aim for long-lasting results that transcend the immediate data sets. In the surface plus noise paradigm of Section 2, the surface plays the role of truth–at least eternal enough to justify striving for its closest possible estimation.

In the neonate example of Table 1 we would hope that starred predictors like gest and

[13]The CDC itself now sponsors annual internet-based flu forecasting challenges (Schmidt, 2019); see their past results at `predict.cdc.gov`.

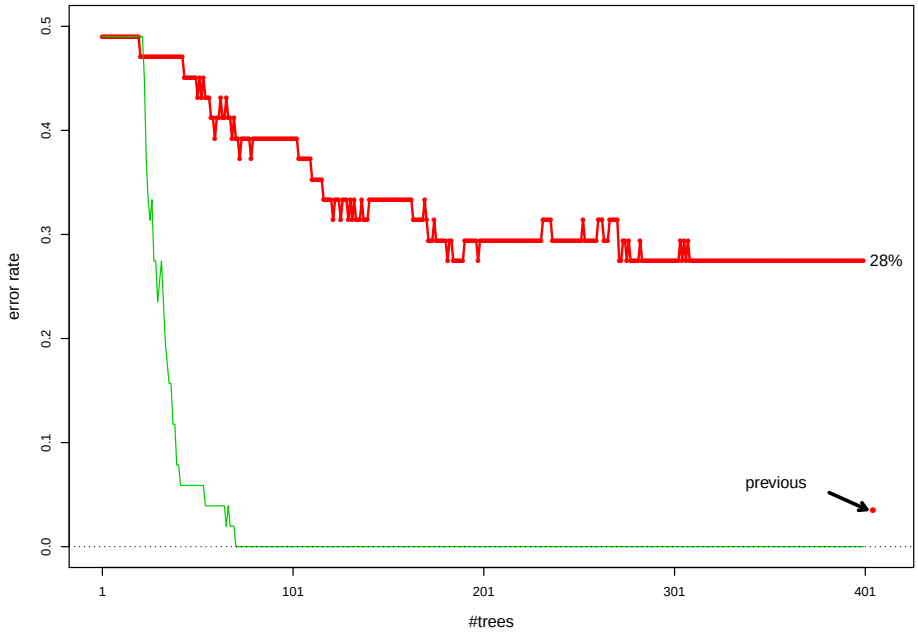

Figure 9: `gbm` test set error, early/late division; compare with Figure 6. Going on to 800 trees decreased error estimate to 26%. Training set error rate, thin curve, was zero after step 70 but test error rate continued to decline. See the brief discussion in Criterion 5 of Section 8.

ap would continue to show up as important in future studies. A second year of data was in fact obtained, but with only $n = 246$ babies. The same logistic regression model was run for the year 2 data and yielded coefficient estimates reasonably similar to the year 1 values; see Table 4. Newton wouldn't be jealous, but something of more than immediate interest seems to have been discovered.

Nothing rules out eternal truth-seeking for the pure prediction algorithms, but they have been most famously applied to more ephemeral phenomena: credit scores, Netflix

Table 4: Comparing logistic regression coefficients for neonate data for year 1 (as in Table 1) and year 2; correlation coefficient 0.79.

	gest	ap	bwei	resp	cpap	ment	rate	hr	head	gen	temp
year 1	−.47	−.58	−.49	.78	.27	1.10	−.09	.01	.1	.00	.02
year 2	−.65	−.27	−.19	1.13	.15	.41	−.47	−.02	−.2	−.04	.16

movie recommendations, facial recognition, *Jeopardy!* competitions. The ability to extract information from large heterogeneous data collections, even if just for short-term use, is a great advantage of the prediction algorithms. Random selection of the test set makes sense in this setting, as long as one doesn't accept the estimated error rate as applying too far outside the limited range of the current data.

Here is a contrived microarray example where *all* the predictors are ephemeral: $n = 400$ subjects participate in the study, arriving one per day in alternation between Treatment and Control; each subject is measured on a microarray of $p = 200$ genes. The 400×200 data matrix $\boldsymbol{x}$ has independent normal entries

$$x_{ij} \overset{\text{ind}}{\sim} \mathcal{N}(\mu_{ij}, 1) \qquad \text{for } i = 1, 2, \ldots, 400 \text{ and } j = 1, 2, \ldots, 200. \tag{6.1}$$

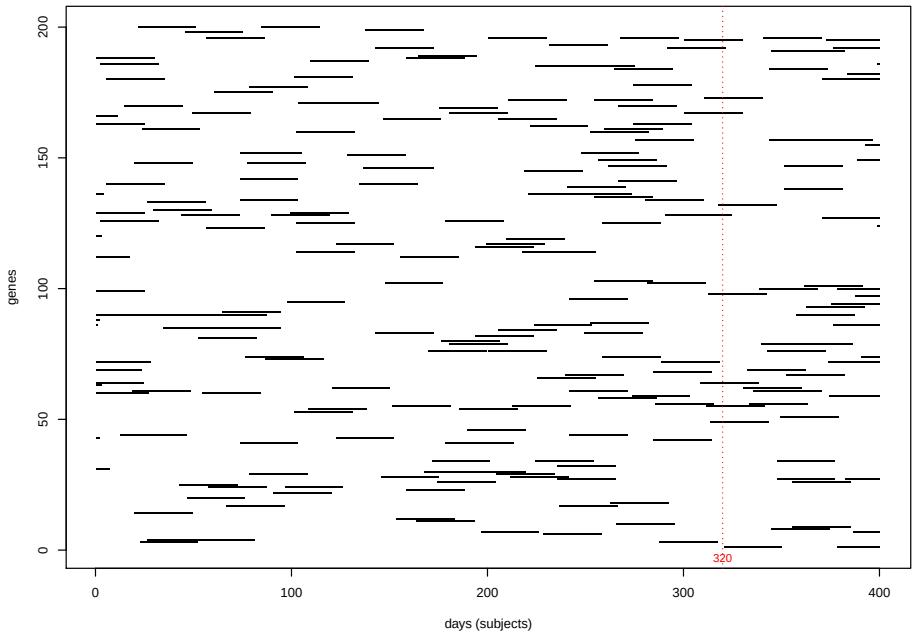

Figure 10: Black line segments indicate active episodes in the hypothetical microarray study. (Matrix transposed for typographical convenience.)

Most of the μ_{ij} are null, $\mu_{ij} = 0$, but occasionally a gene will have an active episode of 30 days during which

$$\mu_{ij} = 2 \text{ for Treatment} \quad \text{and} \quad -2 \text{ for Control} \tag{6.2}$$

for the entire episode, or

$$\mu_{ij} = 2 \text{ for Control} \quad \text{and} \quad -2 \text{ for Treatment} \tag{6.3}$$

for the entire episode. The choice between (6.2) and (6.3) is random, as is the starting date for each episode. Each gene has expected number of episodes equal 1. The black line segments in Figure 10 indicate all the active time periods.

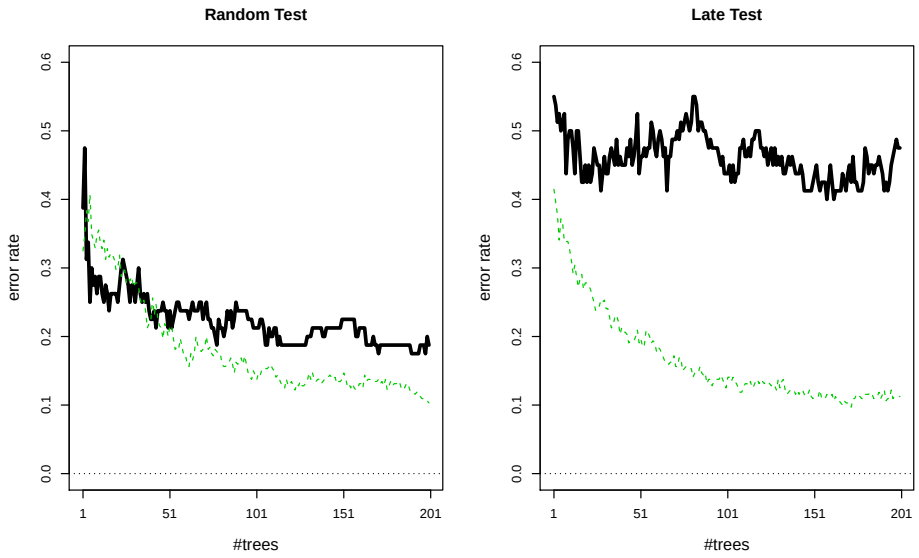

Figure 11: `randomForest` prediction applied to contrived microarray study pictured in Figure 10. *Left panel*: Test set of size 80, selected randomly from 400 days; heavy black curve shows final estimated test error rate of 19%. *Right panel*: Test set days 321 to 400; now error rate estimate is 45%. Light dotted curves in both panels are training set errors.

The 400 hypothetical subjects were *randomly* divided into a training set of 320 and a test set of 80. A `randomForest` analysis gave the results seen in the left panel of Figure 11, with test set error rate 19%. A second `randomForest` analysis was carried out, using the subjects from days 1 to 320 for the training set and from days 321 to 400 for the test set. The right panel of Figure 11 now shows test set error about 45%.

In this case it is easy to see how things go wrong. From any one day's measurements it is possible to predict Treatment or Control from the active episode responses on nearby days. This works for the random training/test division, where most of the test days will be intermixed with training days. Not so for the early/late division, where most of the test

days are far removed from training set episodes. To put it another way, prediction is easier for interpolation than extrapolation.[14]

What in general can we expect to learn from training/test set error estimates? Going back to formulation (2.1), the usual assumption is that the pairs (x_i, y_i) are independent and identically distributed (i.i.d.) from some probability distribution F on $(p+1)$-dimensional space,

$$(x_i, y_i) \overset{\text{iid}}{\sim} F \qquad \text{for } i = 1, 2, \ldots, n. \tag{6.4}$$

A training set $\boldsymbol{d}_0$ of size n_0 and a test set $\boldsymbol{d}_1$ of size $n_1 = n - n_0$ are chosen (*how* is irrelevent under model (6.4)), rule $f(x, \boldsymbol{d}_0)$ is computed and applied to $\boldsymbol{d}_1$, generating an error estimate

$$\widehat{\text{Err}}_{n_0} = \frac{1}{n_1} \sum_{\boldsymbol{d}_1} L\left(y_i, f(x_i, \boldsymbol{d}_0)\right), \tag{6.5}$$

L some loss function like squared error or counting error. Then, under model (6.4), $\widehat{\text{Err}}_{n_0}$ is an unbiased estimate of

$$\text{Err}_{n_0}(F) = E_F\left\{\widehat{\text{Err}}_{n_0}\right\}, \tag{6.6}$$

the average prediction error of a rule[15] $f(x, \boldsymbol{d}_0)$ formed from n_0 draws from F.

Concept drift can be interpreted as a change in the data-generating mechanism (6.4), say F changing to some new distribution $\tilde{F}$, as seems the likely culprit in the prostate cancer example of Figure 8 and Figure 9.[16] Traditional prediction methods are also vulnerable to such changes. In the neonate study, the logistic regression rule based on the year 1 data had a cross-validated error rate of 20% which increased to 22% when applied to the year 2 data.

The story is more complicated for the contrived example of Figure 10 and Figure 11, where model (6.4) doesn't strictly apply. There the effective predictor variables are ephemeral, blooming and fading over short time periods. A reasonable conjecture (but no more

[14] Yu and Kumbier (2019) propose the useful distinction of "internal testing" versus "external testing".

[15] An important point is that "a rule" means one formed according to the algorithm of interest and the data-generating mechanism, not the specific rule $f(x, \boldsymbol{d}_0)$ at hand; see Figure 12.3 of Efron and Hastie (2016).

[16] Cox, in his discussion of Breiman (2001), says of the applicability of model (6.4): "However, much prediction is not like this. Often the prediction is under quite different ... conditions ... [for example] what would be the effect on annual incidence of cancer in the United States of reducing by 10% the medical use of x-rays? etc."

than that) would say the weak learners of the pure prediction algorithms are prone to ephemerality, or at least are more prone than the "main effects" kind of predictors favored in traditional methodology. Whether or not this is true, I feel there is some danger in constructing training/test sets by random selection, and that their error estimates must be taken with a grain of statistical salt. To put things operationally, I'd worry about recommending the random forests prediction rule in Figure 5 to a friend concerned about prostate cancer.

This is more than a hypothetical concern. In their 2019 article, "Deep neural networks are superior to dermatologists in melanoma image classification", Brinker et al. demonstrate just what the title says; the authors are justifiably cautious, recommending future studies for validation. Moreover, they acknowledge the limitations of using a randomly selected test set, along with the possible ephemerality of some of the algorithm's predictor variables. Frequent updating would be necessary for serious use of any such diagnostic algorithm, along with studies to show that certain subpopulations weren't being misdiagnosed.[17]

7 Smoothness

It was not just a happy coincidence that Newton's calculus accompanied Newton's laws of motion. The Newtonian world, as fleshed out by Laplace, is an infinitely smooth one in which small changes in cause yield small changes in effect; a world where derivatives of many orders make physical sense. The parametric models of traditional statistical methodology enforce the smooth-world paradigm. Looking back at Figure 1 in Section 2, we might not agree with the exact shape of the cholostyramine cubic regression curve but the smoothness of the response seems unarguable: going from, say, 1 to 1.01 on the compliance scale shouldn't much change the predicted cholesterol decrease.

Smoothness of response is not built into the pure prediction algorithms. The left panel of Figure 12 shows a `randomForest` estimate of cholesterol decrease as a function of normalized compliance. It roughly follows the OLS cubic curve but in a jagged, definitely unsmooth fashion. Algorithm `gbm`, in the right panel, gave a less jagged "curve" but still with substantial local discontinuity.

[17] Facial recognition algorithms have been shown to possess gender, age, and race biases.

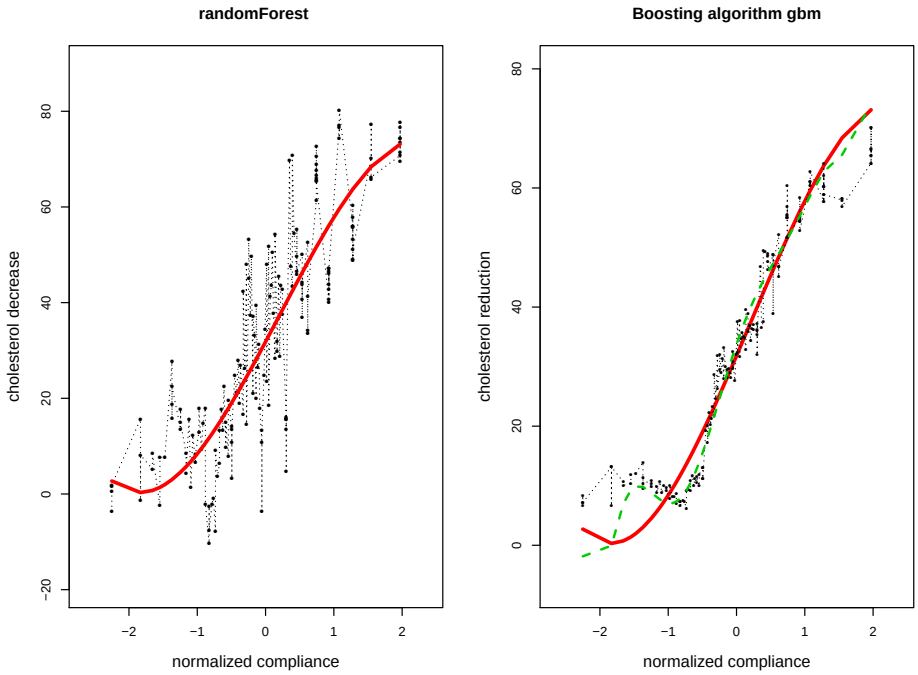

Figure 12: `randomForest` and `gbm` fits to the cholostyramine data of Figure 1, Section 2. Heavy curve is cubic OLS; dashed curve in right panel is 8th degree OLS fit.

The choice of *cubic* in Figure 1 was made on the basis of a Cp comparison of polynomial regressions degrees 1 through 8, with cubic best. Both `randomForest` and `gbm` in Figure 12 began by taking $\boldsymbol{x}$ to be the 164×8 matrix `poly(c,8)` (in R notation), with $\mathbf{c}$ the vector of adjusted compliances–an 8th degree polynomial basis. The light dashed curve in the right panel is the 8th degree polynomial OLS fit, a pleasant surprise being how the `gbm` predictions follow it over much of the compliance range. Perhaps this is a hopeful harbinger of how prediction algorithms could be used as nonparametric regression estimates, but the problems get harder in higher dimensions.

Consider the *supernova data*: absolute brightness y_i has been recorded for each of $n = 75$ supernovas, as well as x_i a vector of spectral energy measurements at $p = 25$ different wavelengths, so the data set is

$$\boldsymbol{d} = \{\boldsymbol{x}, \boldsymbol{y}\}, \tag{7.1}$$

with $\boldsymbol{x}$ 75×25 and $\boldsymbol{y}$ a 75-vector. After some preprocessing, a reasonable model is

$$y_i \overset{\text{ind}}{\sim} \mathcal{N}(\mu_i, 1). \tag{7.2}$$

It is desired to predict μ_i from x_i.

Our data $\boldsymbol{d}$ is unusually favorable in that the 75 supernovas occured near enough to Earth to allow straightforward determination of y_i without the use of x_i. However, this kind of determination isn't usually available, while x_i is always observable; an accurate prediction rule

$$\hat{y}_i = f(x_i, \boldsymbol{d}) \tag{7.3}$$

would let astronomers better use Type 1a supernovas as "standard candles" in determining the distances to remote galaxies.[18] In this situation, the smoothness of $f(x, \boldsymbol{d})$ as a function of x would be a given.

Algorithms `randomForest` and `gbm` were fit to the supernova data (7.1). How smooth or jagged were they? For any two of the 75 cases, say i_1 and i_2, let $\{x_\alpha\}$ be the straight line connecting x_{i_1} and x_{i_2} in R^{25},

$$\{x_\alpha = \alpha x_{i_1} + (1-\alpha)x_{i_2} \text{ for } \alpha \in [0,1]\}\,, \tag{7.4}$$

and $\{\hat{y}_\alpha\}$ the corresponding predictions. A linear model would yield linear interpolation, $y_\alpha = \alpha y_{i_1} + (1-\alpha)y_{i_2}$.

Figure 12 graphs $\{y_\alpha\}$ for three cases: $i_1 = 1$ and $i_2 = 3$, $i_1 = 1$ and $i_2 = 39$, and $i_1 = 39$ and $i_2 = 65$. The `randomForest` traces are notably eccentric, both locally and globally; `gbm` less so, but still far from smooth.[19]

There is no need for model smoothness in situations where the target objects are naturally discrete: movie recommendations, credit scores, chess moves. For scientific applications, at least for some of them, smoothness will be important to a model's plausibility. As far as I know, there is no inherent reason that a pure prediction algorithm must give jagged

[18] The discovery of dark energy and the cosmological expansion of the universe involved treating Type 1a supernovas as always having the same absolute brightness, i.e., as being perfect standard candles. This isn't exactly true. The purpose of this analysis is to make the candles more standard by regression methods, and so improve the distance measurements underlying cosmic expansion. Efron and Hastie (2016) discuss a subset of this data in their Chapter 12.

[19] The relatively smoother results from `gbm` have to be weighed against the fact that it gave much worse predictions for the supernova data, greatly overshrinking the $\hat{y}_i$ toward zero.

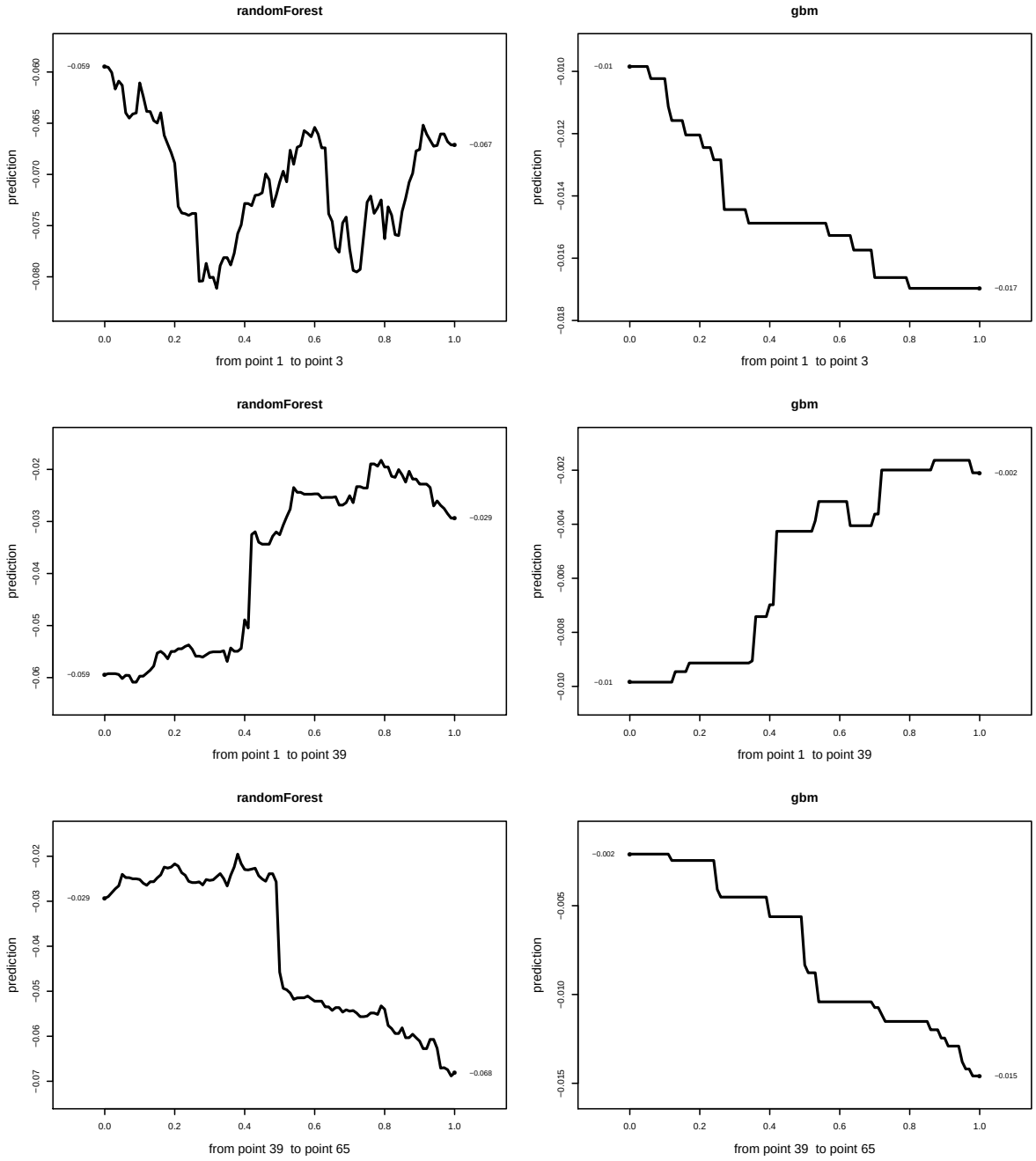

Figure 13: Interpolation between pairs of points in supernova data. Left side is `randomForest`, right side is `gbm`.

results. Neural networks, which are essentially elaborate logistic regression programs, might be expected to yield smoother output.

8 A comparison checklist

Prediction isn't the same as estmation, though the two are often conflated. Much of this paper has concerned the differences. As a summary of what has gone before as well as a springboard for broader discussion, this section presents a checklist of important distinctions and what they mean in terms of statistical practice.

The new millenium got off to a strong start on the virtues of prediction with Leo Breiman's 2001 *Statistical Science* publication, "Statistical modeling: The two cultures." An energetic and passionate argument for the "algorithmic culture" — what I have been calling the pure prediction algorithms — in this work Leo excoriated the "data modeling culture" (i.e., traditional methods) as of limited utility in the dawning world of Big Data. Professor David Cox, the lead discussant, countered with a characteristically balanced defense of mainstream statistics, not rejecting prediction algorithms but pointing out their limitations. I was the second discussant, somewhat skeptical of Leo's claims (which were effusive toward random forests, at that time new) but also somewhat impressed.

Breiman turned out to be more prescient than me: pure prediction algorithms have seized the statistical limelight in the Twenty-First Century, developing much along the lines Leo suggested. The present paper can be thought of as a continued effort on my part to answer the question of how prediction algorithms relate to traditional regression inference.

Table 5 displays a list of six criteria that distinguish traditional regression methods from the pure prediction algorithms. My previous "broad brush" warning needs to be made again: I am sure that exceptions can be found to all six distinctions, nor are the listed properties written in stone, the only implication being that they reflect current usage.

Criterion 1. Surface plus noise models are ubiquitous in traditional regression methodology, so much so that their absence is disconcerting in the pure prediction world. Neither surface nor noise is required as input to `randomForest`, `gbm`, or their kin. This is an enormous advantage for easy usage. Moreover, you can't be using a wrong model if there is no model.

Table 5: A comparison checklist of differences between traditional regression methods and pure prediction algorithms. See commentary in the text.

	Traditional regressions methods	Pure prediction algorithms
(1)	Surface plus noise models (continuous, smooth)	Direct prediction (possibly discrete, jagged)
(2)	Scientific truth (long-term)	Empirical prediction accuracy (possibly short-term)
(3)	Parametric modeling (causality)	Nonparametric (black box)
(4)	Parsimonious modeling (researchers choose covariates)	Anti-parsimony (algorithm chooses predictors)
(5)	$\boldsymbol{x}$ $p \times n$: with $p \ll n$ (homogenous data)	$p \gg n$, both possibly enormous (mixed data)
(6)	Theory of optimal inference (mle, Neyman–Pearson)	Training/test paradigm (Common Task Framework)

A clinician dealing with possible prostate cancer cases will certainly be interested in effective prediction, but the disease's etiology will be of greater interest to an investigating scientist, and that's where traditional statistical methods come into their own. If random forests had been around since 1908 and somebody just invented regression model significance testing, the news media might now be heralding an era of "sharp data".

Eliminating surface-building from inference has a raft of downstream consequences, as discussed in what follows. One casualty is smoothness (Section 7). Applications of prediction algorithms have focused, to sensational effect, on discrete target spaces — Amazon recom-

mendations, translation programs, driving directions — where smoothness is irrelevant. The natural desire to use them for scientific investigation may hasten development of smoother, more physically plausible algorithms.

Criterion 2. The two sides of Table 5 use similar fitting criteria — some version of least squares for quantitative responses — but they do so with different paradigms in mind. Following a two-hundred-year-old scientific path, traditional regression methods aim to extract underlying truth from noisy data: perhaps not eternal truth but at least some takeaway message transcending current experience.

Without the need to model surface or noise mechanisms, scientific truth fades in importance on the prediction side of the table. There may not be any underlying truth. Prediction methods can be comfortable with ephemeral relationships that need only remain valid until the next update. To quote Breiman, "The theory in this field shifts focus from data models to the properties of algorithms," that is, from the physical world to the computer. Research in the prediction community, which is an enormous enterprise, is indeed heavily focused on computational properties of algorithms — in particular, how they behave as n and p become huge — and less on how they relate to models of data generation.

Criterion 3. Parametric modeling plays a central role in traditional methods of inference, while the prediction algorithms are nonparametric, as in (6.4). ("Nonparametric", however, can involve hosts of *tuning parameters*, millions of them in the case of deep learning, all relating to the algorithm rather than to data generation.) Lurking behind a parametric model is usually some notion of causality. In the cholostyramine example of Figure 1, Section 2, we are likely to believe that increased ingestion of the drug cholostyramine causes cholesterol to decrease in a sigmoidal fashion, even if strict causality is elusive.[20]

Abandoning mathematical models comes close to abandoning the historic scientific goal of understanding nature. Breiman states the case bluntly:

> Data models are rarely used in this community [the algorithmic culture]. The approach is that nature produces data in a black box whose insides are complex,

[20]Efron and Feldman (1991) struggled to make a causality argument, one not accepted uncritically by subsequent authors.

mysterious, and at least partly unknowable.[21]

The black-box approach has a scientifically anti-intellectual feeling but, on the other hand, scientific understanding may be beside the point if prediction is the only goal. Machine translation offers a useful case study, where there has been a several-decade conflict between approaches based on linguistic analysis of language structure and more-or-less pure prediction methods. Under the umbrella name of statistical machine translation (SMT), this latter approach has swept the field, with Google Translate, for example, currently using a deep learning prediction algorithm.

Traditional statistical education involves a heavy course of probability theory. Probability occupies a smaller portion of the nonparametric pure-prediction viewpoint, with probabilistically simple techniques such as cross-validation and the bootstrap shouldering the methodological burden. Mosteller and Tukey's 1977 book, *Data Analysis and Regression: A Second Course in Statistics*, favored a nonprobabilistic approach to inference that would be congenial to a modern course in machine learning.

Criterion 4. The eleven neonate predictor variables in Table 1 were winnowed down from an initial list of 81, following a familiar path of preliminary testing and discussions with the medical scientists. Parsimonious modeling is a characteristic feature of traditional methodology. It can be crucial for estimation and, especially, for attribution, where it is usually true that the power of discovery decreases as the list of predictors grows.

The pure prediction world is anti-parsimonious. Control of the predictor set, or the "features" as they are called, passes from the statistician to the algorithm, which can coin highly interactive new features such as random forests' tree variables. "The more predictor variables, the more information," said Breiman, an especially accurate forecast of the deep learning era.

I was doubtful. My commentary on Breiman's paper began: "At first glance Leo Breiman's stimulating paper looks like an argument against parsimony and scientific insight, and in favor of black boxes with lots of knobs to twiddle. At second glance it still looks that way, but the paper *is* stimulating ...." Well–impressive results like the `randomForest` and

[21] Cox counters: "Formal models are useful and often almost, if not quite, essential for incisive thinking."

`gbm` predictions for the prostate cancer data, Figure 5 and Figure 6 of Section 4, certainly back up Leo's claim. But it is still possible to have reservations. The coined features seem here to be of the weak learner variety, perhaps inherently more ephemeral than the putative strong learners of Table 1.

This is the suggestion made in Section 6. If the prediction algorithms work by clever combinations of armies of weak learners, then they will be more useful for prediction than estimation or, especially, for attribution (as suggested in Section 5). "Short-term science" is an oxymoron. The use of prediction algorithms for scientific discovery will depend on demonstrations of their longer-term validity.

Criterion 5. Traditional applications ask that the $n \times p$ data matrix $\boldsymbol{x}$ (n subjects, p predictors) have n substantially greater than p, perhaps $n > 5 \cdot p$, in what is now called "tall data". The neonate data with $n = 812$ and $p = 12$ (counting the intercept) is on firm ground; less firm is the supernova data of Section 7, with $n = 75$ and $p = 25$. On the other side of Table 5 the pure prediction algorithms allow, and even encourage, "wide data", with $p \gg n$. The prostate cancer microarray study is notably wide, with $n = 102$ and $p = 6033$. Even if we begin with tall data, as with the cholostyramine example, the prediction algorithms widen it by the coining of new features.

How do the prediction algorithms avoid overfitting in a $p \gg n$ situation? There are various answers, none of them completely convincing: first of all, using a test set guarantees an honest assessment of error (but see the discussion of Criterion 6). Secondly, most of the algorithms employ cross-validation checks during the training phase. Finally, there is an active research area that purports to show a "self-regularizing" property of the algorithms such that even running one of them long past the point where the training data is perfectly fit, as in Figure 9 of Section 6, will still produce reasonable predictions.[22]

Estimation and, particularly, attribution work best with homogeneous data sets, where the (x, y) pairs come from a narrowly defined population. A randomized clinical trial, where the subjects are chosen from a specific disease category, exemplifies strict homogeneity. *Not*

[22]For instance, in an OLS fitting problem with $p > n$ where the usual estimate $\hat{\beta} = (\boldsymbol{x}^t\boldsymbol{x})^{-1}\boldsymbol{x}^t y$ is not available, the algorithm should converge to the $\hat{\beta}$ that fits the data perfectly, $\boldsymbol{y} = \boldsymbol{x}\hat{\beta}$, *and* has minimum norm $\|\hat{\beta}\|$; see Hastie, Montanari, Rosset, and Tibshirani (2019).

requiring homogeneity makes prediction algorithms more widely applicable, and is a virtue in terms of generalizability of results, but a defect for interpretability.

The impressive scalability of pure prediction algorithms, which allows them to produce results even for enormous values of n and p, is a dangerous virtue. It has led to a lust for ever larger training sets. This has a good effect on prediction, making the task more interpolative and less extrapolative (that is, more like Figure 5 and Figure 6, and less like Figure 8 and Figure 9) but muddies attempts at attribution.[23]

Traditional regression methods take the matrix of predictions $\boldsymbol{x}$ as a fixed ancillary statistic. This greatly simplifies the theory of parametric regression models; $\boldsymbol{x}$ is just as random as $\boldsymbol{y}$ in the pure prediction world, the only probability model being the i.i.d. nature of the pairs $(x, y) \sim F$. Theory is more difficult in this world, encouraging the empirical emphasis discussed in Criterion 6. Bayesian statistics is diminished in the a-probabilistic prediction world, leaving a tacit frequentist basis as the theoretical underpinning.

Criterion 6. Traditional statistical practice is based on a century of theoretical development. Maximum likelihood estimation and the Neyman–Pearson lemma are optimality criteria that guide applied methodology. On the prediction side of Table 5, theoretical efficiency is replaced by empirical methods, particularly training/test error estimates.

This has the virtue of dispensing with theoretical modeling, but the lack of a firm theoretical structure has led to "many flowers blooming": the popular pure prediction algorithms are completely different from each other. During the past quarter-century, first neural nets then support vector machines, boosting, random forests, and a reprise of neural nets in their deep learning form have all enjoyed the prediction spotlight. In the absence of theoretical guidance we can probably expect more.

In place of theoretical criteria, various prediction competitions have been used to grade

[23] An experienced statistician will stop reading an article that begins, "Over one million people were asked. . . ," knowing that a random sample of 1,000 would be greatly preferable. This bit of statistical folk wisdom is in danger of being lost in the Big Data era. In an otherwise informative popular book titled, of course, *Big Data*, the authors lose all equilibrium on the question of sample size, advocating for $n =$ all: all the flu cases in the country, all the books on Amazon.com, all possible dog/cat pictures. "Reaching for a random sample in the age of big data is like clutching at a horsewhip in the era of the motor car." In fairness, the book's examples of $n =$ all are actually narrowly defined, e.g., all the street manholes in Manhattan.

algorithms in the so-called "Common Task Framework". The common tasks revolve around some well known data sets, that of the Netflix movie recommendation data being best known. None of this is a good substitute for a so-far nonexistent theory of optimal prediction.[24]

Test sets are an honest vehicle for estimating prediction error, but choosing the test set by random selection from the full set $\boldsymbol{d}$ (2.1) may weaken the inference. Even modest amounts of concept drift can considerably increase the actual prediction error, as in the prostate data microarray example of Section 6. In some situations there are alternatives to random selection, for example, by selecting training and test according to early and late collection dates, as in Figure 8 and Figure 9. In the supernova data of Section 7, the goal is to apply a prediction rule to supernovas much farther from Earth, so choosing the more distant cases for the test set could be prudent.

In 1914 the noted astronomer Arthur Eddington,[25] an excellent statistician, suggested that mean absolute deviation rather than root mean square would be more efficient for estimating a standard error from normally distributed data. Fisher responded in 1920 by showing that not only was root mean square better than mean absolute deviation, it was better than *any* other possible estimator, this being an early example of his theory of sufficiency.

Traditional methods are founded on these kinds of parametric insights. The two sides of Table 5 are playing by different rules: the left side functions in a Switzerland of inference, comparatively well ordered and mapped out, while Wild West exuberance thrives on the right. Both sides have much to gain from commerce. Before the 1920s, statisticians didn't really understand estimation, and after Fisher's work we did. We are in the same situation now with the large-scale prediction algorithms: lots of good ideas and excitement, without principled understanding, but progress may be in the air.

[24] Bayes rule offers such a theory, but at a cost in assumptions far outside the limits of the current prediction environment.

[25] Later famous for his astronomical verification of Einstein's theory of relativity.

9 Traditional methods in the wide data era

The success of the pure prediction algorithms has had a stimulating effect on traditional theory and practice. The theory, forged in the first half of the Twentieth Century, was tall-data oriented: small values of n, but even smaller p, often just $p = 1$ or 2. Whether or not one likes prediction algorithms, parts of modern science have moved into the wide-data era. In response, traditional methods have been stretching to fit this new world. Three examples follow.

Big data is not the sole possession of prediction algorithms. Computational genetics can go very big, particularly in the form of a GWAS, genome-wide association study. An impressive example is given by Ikram et al. (2010), in a study concerning the narrowing of blood vessels in the eye.[26] The amount of narrowing was measured for $n = 15358$ individuals; each individual had their genome assessed for about $p = 10^6$ SNPs (single-nucleotide polymorphisms), a typical SNP having a certain choice of ATCG value that occurs in a majority of the population or a minor, less prevalent alternative value. The goal was to find SNPs associated with vascular narrowing.

With $\boldsymbol{x} = 15356 \times 10^6$ we are definitely in big data and wide data territory. Surface plus noise models seem out of the question here. Instead, each SNP was considered separately: a linear regression was carried out, with the predictor variable the number of minor polymorphisms in the chromosome pair at that location — 0, 1, or 2 for each individual — and response his or her narrowing measure. This gave a p-value p_i against the null hypothesis: *polymorphism at location i has no effect on narrowing*, $i = 1, 2, \ldots, 10^6$. The Bonferroni threshold for 0.05 significance is

$$p_i \le 0.05/10^6. \tag{9.1}$$

Ikram et al. displayed their results in a "manhattan plot" with $z_i = -\log_{10}(p_i)$ graphed against location on the genome. Threshold (9.1) corresponds to $z_i \ge 7.3$; 179 of the 10^6 SNPs had $z_i > 7.3$, rejecting the null hypothesis of no effect. These were bunched into five locations on the genome, one of which was borderline insignificant. The authors claimed

[26] Microvascular narrowing is thought to contribute to heart attacks, but it is difficult to observe in the heart; observation is much easier in the eye.

credit for discovering four novel loci. These might represent four genes implicated in vascular narrowing (though a spike in chromosome 12 is shown to spread over a few adjacent genes).

Instead of performing a traditional attribution analysis with $p = 10^6$ predictors, the GWAS procedure performed 10^6 analyses with $p = 1$ and then used a second layer of inference to interpret the results of the first layer. My next example concerns a more elaborate implementation of the two-layer strategy.

While not 10^6, the $p = 6033$ features of the prostate cancer microarray study in Section 4 are enough to discourage an overall surface plus noise model. Instead we begin with a separate $p = 1$ analysis for each of the genes, as in the GWAS example. The data (4.1) for the jth gene is

$$\boldsymbol{d}_j = \{x_{ij} : i = 1, 2, \ldots, 102\}, \tag{9.2}$$

with $i = 1, 2, \ldots, 50$ for the normal controls and $i = 51, 52, \ldots, 102$ for the cancer patients.

Under normality assumptions, we can compute statistics z_j comparing patients with controls which satisfy, to a good approximation,[27]

$$z_j \sim \mathcal{N}(\delta_j, 1), \tag{9.3}$$

where δ_j is the *effect size* for gene j: δ_j equals 0 for "null genes", genes that show the same genetic activity in patients and controls, while $|\delta_j|$ is large for the kinds of genes being sought, namely, those having much different responses for patients versus controls.

Inferences for the individual genes by themselves are immediate. For instance,

$$p_j = 2\Phi(-z_j) \tag{9.4}$$

is the two-sided p-value for testing $\delta_j = 0$. However, this ignores having 6033 p-values to interpret simultaneously. As with the GWAS, a second layer of inference is needed.

A Bayesian analysis would hypothesize a prior "density" $g(\delta)$ for the effect size, where g includes an atom of probability π_0 at $\delta = 0$ to account for the null genes. Probably, most of

[27] If t_j is the two-sample t-statistic comparing patients with controls, we take $z_j = \Phi^{-1}F_{100}(t_j)$, where F_{100} is the cdf of a t-statistic with 100 degrees of freedom and Φ is the standard normal cdf. Effect size δ_j is a monotone function of the difference in expectations between patients and controls; see Section 7.4 of Efron (2010).

the genes have nothing to do with prostate cancer so π_0 is assumed to be near 1. The *local false discovery rate* $\text{fdr}(z)$ — that is, the probability of a gene being null given z-value z — is, according to Bayes rule,

$$\text{fdr}(z) = \pi_0\phi(z-\delta)/f(z) \doteq \phi(z-\delta)/f(z), \tag{9.5}$$

where $\phi(z) = \exp\{-z^2/2\}/\sqrt{2\pi}$, and $f(z)$ is the marginal density of z,

$$f(z) = \int_{-\infty}^{\infty} \phi(z-\delta)g(\delta)\, d\delta. \tag{9.6}$$

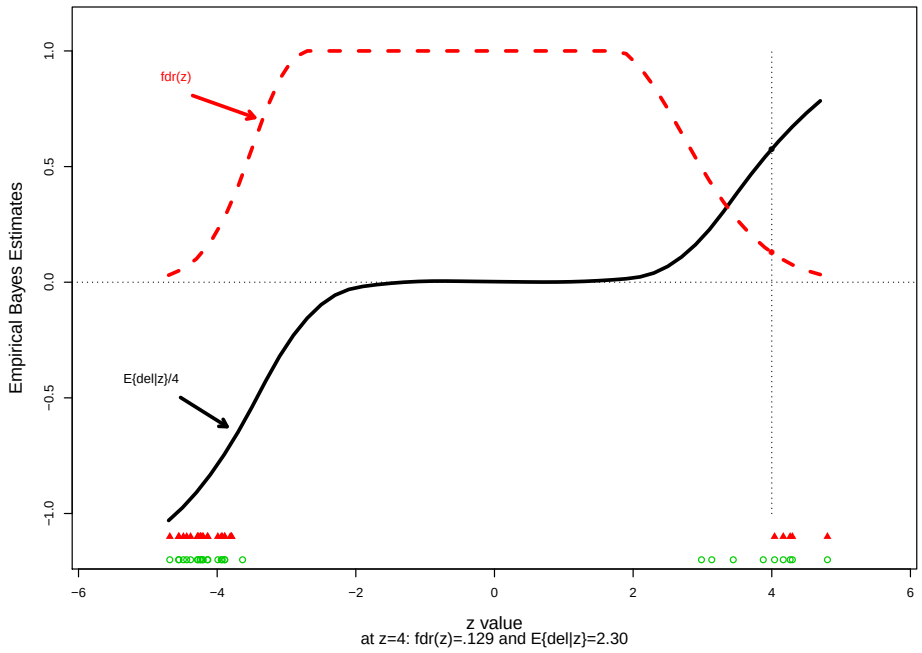

Figure 14: Estimated local false discovery curve $\widehat{\text{fdr}}(z)$ and posterior effect size estimate $\hat{E}\{\delta \mid z\}$ from empirical Bayes analysis of prostate cancer data (the latter divided by 4 for display purposes). Triangles indicate 29 genes having $\widehat{\text{fdr}}(z) \leq 0.20$; circles are 29 most significant genes from `glmnet` analysis.

The prior $g(\delta)$ is most often unknown. An *empirical Bayes* analysis supposes $f(z)$ to be in some parametric family $f_\beta(z)$; the MLE $\hat{\beta}$ is obtained by fitting $f_\beta(\cdot)$ to $\{z_1, z_2, \ldots, z_p\}$, the observed collection of all 6033 z-values, giving an estimated false discovery rate

$$\widehat{\text{fdr}}(z) = \phi(z-\delta)/f_{\hat{\beta}}(z). \tag{9.7}$$

The dashed curve in Figure 14 shows $\widehat{\text{fdr}}(z)$ based on a fifth-degree log-polynomial model for f_β,

$$\log\{f_\beta(z)\} = \beta_0 + \sum_{k=1}^{5} \beta_k z^k; \tag{9.8}$$

$\widehat{\text{fdr}}(z)$ is seen to be near 1.0 for $|z| \le 2$ (i.e., gene almost certainly null) and declines to zero as $|z|$ grows large, for instance equaling 0.129 at $z = 4$. The conventional threshold for attributing significance is $\widehat{\text{fdr}}(z) \le 0.20$; 29 genes achieved this, as indicated by the triangles in Figure 14.

We can also estimate the expected effect size. *Tweedie's formula* (Efron, 2011) gives a simple expression for the posterior expectation,

$$E\{\delta \mid z\} = z + \frac{d}{dz}\log f(z), \tag{9.9}$$

$f(z)$ the marginal density (9.6). Substituting $f_{\hat\beta}$ for f gave the estimate $E\{\delta \mid z\}$ in Figure 14. It is nearly zero for $|z| \le 2$, rising to 2.30 at $z = 4$.

By using a two-level hierarchical model, the empirical Bayes analysis reduces the situation from $p = 6033$ to $p = 5$. We are back in the comfort zone for traditional methods, where parametric modeling for estimation and attribution works well. Both are illustrated in Figure 14.

Sparsity offers another approach to wide-data estimation and attribution: we assume that most of the p predictor variables have no effect and concentrate effort on finding the few important ones. The *lasso* (Tibshirani, 1996) provides a key methodology. In an OLS type problem we estimate β, the p-vector of regression coefficients, by minimizing

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^t\beta)^2 + \lambda\|\beta\|_1, \tag{9.10}$$

where $\|\beta\|_1 = \sum_{j=1}^{p}|\beta_j|$.

Here λ is a fixed tuning parameter: $\lambda = 0$ corresponds to the ordinary least squares solution for β (if $p \le n$) while $\lambda = \infty$ makes $\hat\beta = 0$. For large values of λ, only a few of the coordinates $\hat\beta_j$ will be nonzero. The algorithm begins at $\lambda = \infty$ and decreases λ, admitting one new nonzero coordinate $\hat\beta_j$ at a time. This works even if $p > n$.

The lasso was applied to the supernova data of Section 7, where $\boldsymbol{x}$ has $n = 75$ and $p = 25$. Figure 15 shows the first six steps, tracking the nonzero coefficients $\hat\beta_j$ as new

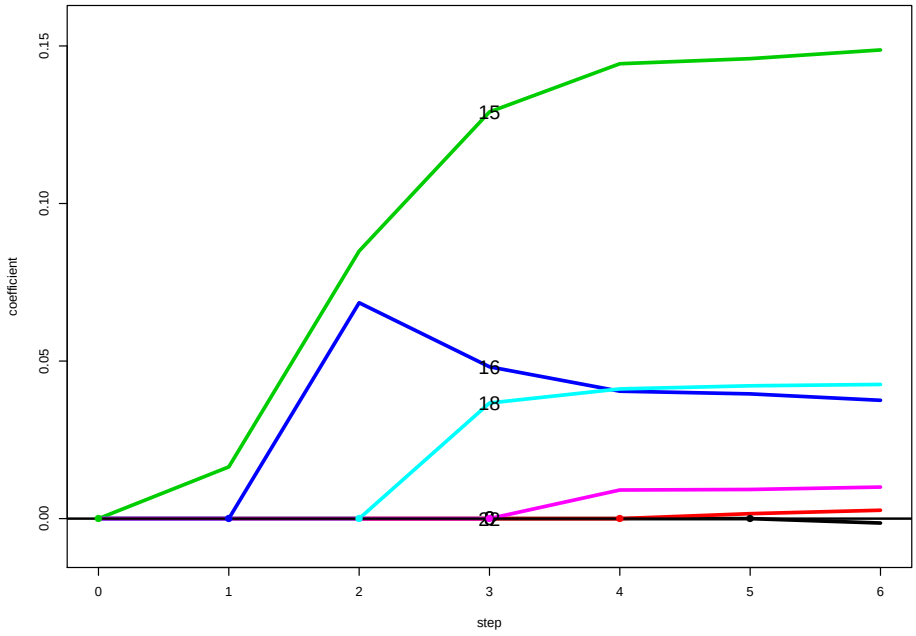

Figure 15: First 6 steps of lasso algorithm applied to supernova data; coefficients of various predictors are plotted as a function of step size. Predictor 15 was chosen first, followed by 16, 18, 22, 8, and 6. Stopping at step 4 gives the lowest Cp estimate of estimation error.

variables were added. Predictor 15 was selected first, then 16, 18, 22, 8, and 6, going on to the full OLS solution $\hat{\beta}$ at step 25. An accuracy formula suggested step 4, with the only nonzero coefficients 15, 16, 18, and 22, as giving the best fit. (These correspond to energy measurements in the iron portion of the spectrum.)

Sparsity and the lasso take us in a direction opposite to the pure prediction algorithms. Rather than combining a myriad of weak predictors, inference is based on a few of the strongest explanatory variables. This is well suited to attribution but less so for prediction.

An R program for the lasso, `glmnet`, was applied to the prostate cancer prediction problem of Section 4, using the same training/test split as that for Figure 5. It performed much worse than `randomForest`, making 13 errors on the test set. Applied to the entire data set of 102 men, however, `glmnet` gave useful indications of important genes: the circles in Figure 14 show z-values for the 29 genes it ranked as most influential. These have large values of $|z_i|$, even though the algorithm didn't "know" ahead of time to take t-statistics between the cancer and control groups.

The lasso produced *biased* estimates of β, with the coordinate values $\hat{\beta}_j$ shrunk toward

zero. The criticism leveled at prediction methods also applies here: biased estimation is not yet on a firm theoretical footing.

10 Two hopeful trends

This wasn't meant to be an "emperor has no clothes" kind of story, rather "the emperor has nice clothes but they're not suitable for every occasion". Where they *are* suitable, the pure prediction algorithms can be stunningly successful. When one reads an enthusiastic AI-related story in the press, there's usually one of these algorithms, operating in enormous scale, doing the heavy lifting. Regression methods have come a long and big way since the time of Gauss.

Much of this article has been concerned with what the prediction algorithms *can't* do, at least not in their present formulations. Their complex "black box" nature makes the algorithms difficult to critique. Here I've tried to use relatively small data sets (by prediction literature standards) to illustrate their differences from traditional methods of estimation and attribution. The criticisms, most of which will not come as a surprise to the prediction community, were summarized in the six criteria of Table 5 in Section 8.

Some time around the year 2000 a split opened up in the world of statistics.[28] For the discussion here we can call the two branches "pure prediction" and "GWAS": both accommodating huge data sets, but with the former having become fully algorithmic while the latter stayed on a more traditional math-modeling path. The "two hopeful trends" in the title of this section refer to attempts at reunification, admittedly not yet very far along.

Trend 1 aims to make the output of a prediction algorithm more interpretable, that is, more like the output of traditional statistical methods. Interpretable surfaces, particularly those of linear models, serve as the ideal for this achievement. Something like attribution is also desired, though usually not in the specific sense of statistical significance.

One tactic is to use traditional methods for the analysis of a prediction algorithm's output; see Hara and Hayashi (2016) and page 346 of Efron and Hastie (2016). Wager, Hastie, and Efron (2014) use bootstrap and jackknife ideas to develop standard error calculations

[28] See the triangle diagram in the epilogue of Efron and Hastie (2016).

for random forest predictions. Murdoch et al. (2019) and Vellido, Martín-Guerrero, and Lisboa (2012) provide overviews of interpretability, though neither focuses on pure prediction algorithms. Using information theoretic ideas, Achille and Soatto (2018) discuss statistical sufficiency measures for prediction algorithms.

Going in the other direction, Trend 2 moves from left to right in Table 5, hoping to achieve at least some of the advantages of prediction algorithms within a traditional framework. An obvious target is scalability. Qian et al. (2019) provide a `glmnet` example with $n = 500,000$ and $p = 800,000$. Hastie et al. (2009) successfully connected boosting to logistic regression. The traditional parametric model that has most currency in the prediction world is logistic regression, so it is reasonable to hope for reunification progress in that area.

"Aspirational" might be a more accurate word than "hopeful" for this section's title. The gulf seen in Table 5 is wide and the reunification project, if going at all, is just underway. In my opinion, the impediments are theoretical ones. Maximum likelihood theory provides a lower bound on the accuracy of an estimate, and a practical way of nearly achieving it. What can we say about prediction? The Common Task Framework often shows just small differences in error rates among the contestants, but with no way of knowing whether some other algorithm might do much better. In short, we don't have an optimality theory for prediction.

The talks I hear these days, both in statistics and biostatistics, bristle with energy and interest in prediction algorithms. Much of the current algorithmic development has come from outside the statistics discipline but I believe that future progress, especially in scientific applicability, will depend heavily on us.

A Appendix

A.1 Out-of-bag (oob) estimates, Section 3

A random forest application involves B nonparametric bootstrap samples from the original data set $\boldsymbol{d}$ (2.1), say $\boldsymbol{d}^{*1}, \boldsymbol{d}^{*2}, \ldots, \boldsymbol{d}^{*B}$. Let

$$\hat{y}_{ik} = f(x_i, \boldsymbol{d}^{*k}), \qquad i = 1, 2, \ldots, n \text{ and } k = 1, 2, \ldots, B, \tag{A.1}$$

be the estimate for the ith case obtained from the prediction rule based on the kth bootstrap sample. Also let

$$N_{ik} = \text{number of times case } (x_i, y_i) \text{ appears in } \boldsymbol{d}^{*k}. \tag{A.2}$$

The oob estimate of prediction errors is

$$\widehat{\text{Err}}_0 = \sum_{k=1}^{B} \sum_{i:N_{ik}=0} L(y_i, \hat{y}_{ik}) \Big/ \sum_{k=1}^{B} \sum_{i:N_{ik}=0} 1, \tag{A.3}$$

where $L(y_i, \hat{y}_{ik})$ is the loss function; that is, $\widehat{\text{Err}}_0$ is the average loss over all cases in all bootstrap samples where the sample did not include the case.

"Bagging" stands for "bootstrap aggregation", the averaging of the true predictors employed in random forests. The intuition behind (A.3) is that cases having $N_{ik} = 0$ are "out of the bag" of $\boldsymbol{d}^{*k}$, and so form a natural cross-validation set for error estimation.

A.2 Prediction is easier than attribution

We wish to motivate (5.6)–(5.7), beginning with model (5.4)–(5.5). Let $\bar{x}_{0j}$ be the average of the $n/2$ healthy control measurements for gene j, and likewise $\bar{x}_{1j}$ for the $n/2$ sick patient measurements. Then we can compute z values

$$z_j = c(\bar{x}_{1j} - \bar{x}_{0j}) \overset{\text{ind}}{\sim} \mathcal{N}(\delta_j, 1) \tag{A.4}$$

for $j = 1, 2, \ldots, N$. Letting $\pi_0 = N_0/N$ and $\pi_1 = N_1/N$ be the proportions of null and non-null genes, (5.5) says that the z_j take on two possible densities, in proportions π_0 and π_1,

$$\pi_0 : z \sim f_0(z) \quad \text{or} \quad \pi_1 : z \sim f_1(z), \tag{A.5}$$

where $f_0(z)$ is the standard normal density $f(z)$, and $f_1(z) = \phi(z - \Delta)$. With both N_0 and N_1 going to infinity, we can and will think of π_0 and π_1 as prior probabilities for null and non-null genes.

The posterior moments of δ_j given z_j, obtained by applying Bayes rule to model (A.4), have simple expressions in terms of the log derivatives of the marginal density

$$f(z) = \pi_0 f_0(z) + \pi_1 f_1(z), \tag{A.6}$$

$$d(z) \equiv E\{\delta \mid z\} = z + \frac{d}{dz}\log f(z),$$

and (A.7)

$$v(z) \equiv \mathrm{var}\{\delta \mid z\} = 1 + \frac{d^2}{dz^2}\log f(z).$$

See Efron (2011). That is,

$$\delta_j \mid z_j \sim (d_j, v_j), \tag{A.8}$$

where the parenthetical notation indicates the mean and variance of a random quantity and $d_j = d(z_j)$, $v_j = v(z_j)$. Then $X_j \sim \mathcal{N}(\delta_j/2c, 1)$ for the sick subjects in (5.4) gives

$$X_j \mid z_j \overset{\text{ind}}{\sim} \left(\frac{d_j}{2c}, \frac{v_j}{4c^2} + 1\right). \tag{A.9}$$

(The calculations which follow continue to focus on the "plus" case of (5.4).)

A linear combination

$$S = \sum_{j=1}^{N} w_j X_j \tag{A.10}$$

has posterior mean and variance

$$\begin{aligned} S &\sim \left(\sum_{j=1}^{N} w_0 A_0, \sum_{j=1}^{N} w_j^2 B_j\right), \\ \text{with} \qquad A_j &= \frac{d_j}{2c} \quad \text{and} \quad B_j = \frac{v_j}{4c^2} + 1. \end{aligned} \tag{A.11}$$

For the "minus" arm of (5.4) (the healthy controls), $S \sim (-\sum_1^N w_j A_j, \sum_1^N B_j^2)$. We can use S as a prediction statistic, predicting sick for $S > 0$ and healthy for $S < 0$.

The probability of a correct prediction depends on

$$R^2 = \mathrm{mean}(S)^2/\,\mathrm{var}(S) = \left(\sum_1^N w_j A_j\right)^2 \Big/ \left(\sum_1^N w_j^2 B_j\right). \tag{A.12}$$

This is maximized for $w_j = A_j/B_j$ (as with Fisher's linear discriminant function), yielding

$$S \sim \left(\sum_1^N A_j^2/B_j, \sum_1^N A_j^2/B_j\right) \quad \text{and} \quad R^2 = \sum_1^N A_j^2/B_j. \tag{A.13}$$

A normal approximation for the distribution of S gives

$$\Phi(-R) \tag{A.14}$$

as the approximate probability of a prediction error of either kind; see Efron (2009).

It remains to calculate R^2.

Lemma 1. *A continuous approximation to the sum* $R^2 = \sum_1^N A_j^2/B_j$ *is*

$$R^2 = \frac{N_1^2}{N_0}\int_{-\infty}^{\infty}\frac{\Delta^2}{4c^2}\;\frac{N_0/N_1}{1+\frac{N_0}{N_1}\,\frac{f_0(z)}{f_1(z)}}\;\frac{1}{1+\frac{v(z)}{4c^2}}\;f_1(z)\;dz. \tag{A.15}$$

Before verifying the lemma, we note that it implies (5.6): letting $N_0 \to \infty$ with $N_1 = O(N_0)$, and using (A.4), gives

$$R^2 = \frac{N_1^2}{N_0}\int_{-\infty}^{\infty}\frac{\Delta^2}{4c^2}\;\frac{\exp\{-(z^2/2)+2\Delta z-(\Delta^2/2)\}}{\sqrt{2\pi}\,(1+v(z)/4c^2)}\;dz. \tag{A.16}$$

The variance $v(z)$ is a bounded quantity under (A.4) — it equals 0.25 for $\Delta = 1$, for instance — so the integral is a finite positive number, say $I(\Delta)$. If $N_1 = \gamma N_0^{1/2}$, then the prediction error probabilities are approximately $\Phi(-\gamma I(\Delta)^{1/2})$ according to (A.14). However, $N_1 = o(N_0^{1/2})$ gives $R^2 \to 0$ and error probabilities $\to 1/2$.

It remains to verify the lemma. Since any δ_j equals either 0 or Δ (5.5),

$$\begin{aligned} d(z) &= E\{\delta \mid z\} = \Pr\{\delta \neq 0 \mid z\}\Delta \\ &= \mathrm{tdr}(z)\Delta, \end{aligned} \tag{A.17}$$

where $\mathrm{tdr}(z)$ is the *true discovery rate* $\Pr\{\delta \neq 0 \mid z\}$,

$$\begin{aligned} \mathrm{tdr}(z) &= \frac{\pi_1 f_1(z)}{\pi_0 f_0(z)+\pi_1 f_1(z)} = \frac{1}{1+\frac{\pi_0}{\pi_1}\,\frac{f_0(z)}{f_1(z)}} \\ &= \frac{1}{1+\frac{N_0}{N_1}\,\frac{f_0(z)}{f_1(z)}}. \end{aligned} \tag{A.18}$$

From (A.11) and (A.13) we get

$$\begin{aligned} R^2 &= \sum_1^N \frac{A_j^2}{B_j} = \sum_1^N \frac{d_j^2}{4c^2+v_j} \\ &= \sum_1^N \frac{\mathrm{tdr}_j^2}{4c^2+v_j} = N\int_{-\infty}^{\infty}\frac{\mathrm{tdr}(z)^2\Delta^2}{4c^2+v(z)}\;f(z)\;dz, \end{aligned} \tag{A.19}$$

the last equality being the asymptotic limit of the discrete sum.

Since $\mathrm{tdr}(z)f(z) = \pi_1 f_1(z) = (N_1/N_0)f_1(z)$, (A.18) becomes

$$R^2 = N_1\int_{-\infty}^{\infty}\frac{\mathrm{tdr}(z)\Delta^2}{4c^2+v(z)}\;f_1(z)\;dz. \tag{A.20}$$

Then (A.17) gives the lemma. Moreover, unless $N_1 = O(N_0)$, (A.17) shows that $\mathrm{tdr}(z) \to 0$ for all z, supporting (5.7).

References

Achille, A. and S. Soatto (2018). Emergence of invariance and disentanglement in deep representations. *J. Mach. Learn. Res. 19*(50), 1–34.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statist. Sci. 16*(3), 199–231.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software, Belmont, CA.

Brinker, T. J., A. Hekler, A. H. Enk, C. Berking, S. Haferkamp, A. Hauschild, M. Weichenthal, J. Klode, D. Schadendorf, T. Holland-Letz, C. von Kalle, S. Frhling, B. Schilling, and J. S. Utikal (2019). Deep neural networks are superior to dermatologists in melanoma image classification. *Europ. J. Cancer 119*, 11–17.

Efron, B. (2009). Empirical Bayes estimates for large-scale prediction problems. *J. Amer. Statist. Assoc. 104*, 1015–1028.

Efron, B. (2010). *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*, Volume 1 of *Institute of Mathematical Statistics Monographs*. Cambridge: Cambridge University Press.

Efron, B. (2011). Tweedie's formula and selection bias. *J. Amer. Statist. Assoc. 106*(496), 1602–1614.

Efron, B. and D. Feldman (1991). Compliance as an explanatory variable in clinical trials. *J. Amer. Statist. Assoc. 86*(413), 9–17.

Efron, B. and T. Hastie (2016). *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. Cambridge: Cambridge University Press. Institute of Mathematical Statistics Monographs (Book 5).

Hara, S. and K. Hayashi (2016). Making Tree Ensembles Interpretable. *arXiv e-prints*, arXiv:1606.05390.

Hastie, T., A. Montanari, S. Rosset, and R. J. Tibshirani (2019). Surprises in high-dimensional ridgeless least squares interpolation. *arXiv e-prints*, arXiv:1903.08560.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Second ed.). Springer Series in Statistics. New York: Springer.

Ikram, M. K., S. Xueling, R. A. Jensen, M. F. Cotch, A. W. Hewitt, and others (2010). Four novel loci (19q13, 6q24, 12q24, and 5q14) influence the microcirculation *in vivo*. *PLOS Genet. 6*(10), 1–12.

Johnstone, I. M. and B. W. Silverman (2004). Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences. *Ann. Statist. 32*(4), 1594–1649.

Mediratta, R., A. Tazebew, R. Behl, B. Efron, B. Narasimhan, A. Teklu, A. Shehibo, M. Ayalew, and S. Kache (2019). Derivation and validation of a prognostic score for neonatal mortality upon admission to a neonatal intensive care unit in Gondar, Ethiopia. Submitted.

Mosteller, F. and J. Tukey (1977). *Data Analysis and Regression: A Second Course in Statistics*. Addison-Wesley Series in Behavioral Science: Quantitative Methods. Reading, Mass.: Addison-Wesley.

Murdoch, W. J., C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu (2019). Interpretable machine learning: Definitions, methods, and applications. *arXiv e-prints*, arXiv:1901.04592.

Qian, J., W. Du, Y. Tanigawa, M. Aguirre, R. Tibshirani, M. A. Rivas, and T. Hastie (2019). A fast and flexible algorithm for solving the lasso in large-scale and ultrahigh-dimensional problems. *bioRxiv*, bioRxiv:630079.

Schmidt, C. (2019). Real-time flu tracking. *Nature (Outlook) 573*, S58–S59.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B 58*(1), 267–288.

Vellido, A., J. D. Martín-Guerrero, and P. J. G. Lisboa (2012). Making machine learning models interpretable. In *Proc. 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012), Bruges, Belgium, 25-27 April 2012*, pp. 163–172. i6doc.com.

Wager, S., T. Hastie, and B. Efron (2014). Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *J. Mach. Learn. Res. 15*(May), 1625–1651.

Yu, B. and K. Kumbier (2019). Three principles of data science: Predictability, computability, and stability (PCS). *arXiv e-prints*, arXiv:1901.08152.