BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
http://dx.doi.org/10.1090/bull/1611
Article electronically published on January 25, 2018

*Computer age statistical inference: Algorithms, evidence, and data science*, by Bradley Efron and Trevor Hastie, Institute of Mathematical Statistics (IMS) Monographs, Vol. 5, Cambridge University Press, New York, NY, 2016, xix+475 pp., ISBN 978-1-107-14989-2, US$74.99

That computers are changing the nature of scientific inquiry and the way we learn about ourselves, the world around us, and our day-to-day lives is obvious. Computing is ubiquitous. Focus has shifted from "if" to "how" when contemplating the extent to which computers are augmenting, automating, and enhancing a particular enterprise. Some experiences are touched more than others, and not always in a good way. Nowhere is this more true than in data science. Computers have revolutionized the degree of automation in learning from data and have expanded the fidelity of statistical inquiry to an astonishing degree in recent decades. At the same time, computers are responsible for a similarly dramatic expansion in the amount of data recorded, banking measurements on more things and at resolutions vastly higher than ever imagined, except perhaps in science fiction. To cope, statistical methodology has had to pare back in some areas, as much as ratchet up.

What was for a century, or more, a fairly stable and plug-and-play toolkit for testing hypotheses and making predictions is today rapidly evolving in a dynamic landscape. Computational machinery has broken us out of the point-estimation, central limit theorem-style uncertainty quantification that practitioners seldom fully internalize. Now we have more intuitive tools, such as cross-validation (CV), bootstrap, and Bayesian posterior inference via Markov chain Monte Carlo (MCMC). Computation offers robustness via ensembles and long run averages or (when all eggs must be in one basket) model selection from alternatives, which, if fully enumerated, would be of astronomic proportion. Methods are scored not just on the old-school trifecta of theoretical justification, empirical accuracy, and interpretive capability but increasingly on their algorithmics, implementations, speed, potential for parallelization, distribution, execution on specialized hardware, automation, and so on. Development has been so feverish that it can be hard for practitioners—experts in other areas of science—to keep up. Yet at the same time, it has never been more essential to utilize statistical machinery: to incorporate data and to make decisions, often in real time, in the face of uncertainty. Many are desperate for an atlas to help navigate the modern data science landscape. More data and greater automation may have paradoxically led to greater uncertainty.

**Essential tools.** Despite the dizzying array of acronyms, experts largely agree on a relatively compact set of modern fundamentals: bias-variance trade-off and regularization, control for false discovery, randomization and Monte Carlo, latent variables, divide-and-conquer, basis expansion, and kernels. Take Monte Carlo as a first example, a class of methods that are meaningless without the advent of cheap computing. Monte Carlo, or MC for short, is named for the games of chance played in the eponymous city on the Mediterranean. From a data science perspective, one applies MC by interjecting randomness into an otherwise deterministic procedure. At first glance this seems to be of dubious benefit: how could adding random noise help, and doesn't that represent a lot of effort? Computers don't mind repetitive tasks, potentially accommodating a substantial degree of randomization without

breaking a sweat. As for why randomization is useful, well that's more subtle and depends on the task at hand.

The simplest and perhaps most widely applied MC method is the bootstrap. The bootstrap draws its power from the empirical distribution of the (training) data. Recall that, in the statistics literature, observations are regarded as a random sample from an underlying population, and the goal is to learn about that population from the sample. Most statistical methods posit a model offering a mathematically convenient caricature of the data-generating mechanism. Models have parameters that are somehow optimized, say via a measure of fit to the data like squared-error loss or the likelihood. Since the data are a random sample, the optimized parameters may be regarded as random variables whose distribution depends on the relative frequencies of occurrences in the underlying population. Those estimated parameters are, in the jargon, a *statistic*. The trouble is, only with very special models, special parameters, and populations can the distribution of statistics be derived, or even asymptotically approximated and, therefore, properly understood. The literature of old is peppered with mathematical acrobatics toward closed form (approximate) so-called *sampling distributions*. Even when they work out, the results can be inscrutable, at least from a practitioner's perspective, and therefore they rarely furnish forecasts with meaningful summaries of uncertainty. Along comes the bootstrap, which says you can usually get the same thing, at least empirically, for a wide class of models and parameters with a simple loop: randomly resample your data, estimate parameters, repeat. The collection of optimized parameters constitute an empirical sampling distribution. The implementation is trivial, may be highly parallelized (because each data resample is handled in a statistically and algorithmically independent way), and may even be distributed, meaning that you can even bootstrap over partitioned data whose elements are, due to storage, legislative, or communication bottlenecks, effectively quarantined apart from one another. Examples include data collected and stored locally by ISPs or e-commerce giants like Amazon or Google.

Methods like the bootstrap, offering the potential to understand uncertainty in almost any estimate, open up the potential to explore a vast array of alternative explanations of data. But that uncertainty, or variance, is but one side of the accuracy coin, i.e., how far our inferences are from the "truth", or at least something useful. The other side is bias. Statisticians learned long ago that it is relatively easy to reduce uncertainty in forecasts with stronger (or simpler) modeling assumptions, more data, and usually both. But often that did not lead to better forecasts. Exploring the bias-variance trade-off was difficult before computers got fast. These days it is easy to enumerate thousands of alternative models and evaluate forecasts out-of-sample with MC validation schemes such as CV. The most common CV setup partitions the data into equal-sized chunks and then iterates, alternately holding each out for testing. Candidate predictors are fit to training data comprising of the chunk's complement in the partition, and are evaluated on the hold-out testing set. By performing a double loop, over predictors/models and partition elements, one can access the predictive accuracy, or any other score out-of-sample, and select the best alternative.

Now for some classes of models it is possible to leverage a degree of analytic tractability while exploring the bias-variance trade-off, and thereby explore (at least implicitly) a dizzying array of alternatives. The best example is the *lasso* for linear models relating a response, or output variable, to a potentially enormous

set of explanatory, or input variables. The lasso is part of a wider family of regularized regression methods pairing a loss function (usually squared error) with a constraint that the estimated coefficients are not too large. It turns out to be easier, and equivalent, to work with an additive penalty instead. The lasso uses an L1 penalty, $\lambda \sum_j |\beta_j|$, which maps out a space wherein optimal solutions are on "corners" where some coefficients are set identically to zero, effectively deselecting input coordinates—de facto model selection, in a limited sense. Clever coordinatewise algorithms make the search for optimal coefficients blazingly fast and even enable a continuum of penalty parameters $\lambda$ to be entertained in one fell swoop, so that all that remains is to pick one according to some meta-criteria. CV is the most popular option, for which automations are readily available in software. Information criteria (IC), which essentially contemplate out-of-sample accuracy without actually measuring it empirically, offer further computational savings in many settings. Some formulations, i.e., of penalty and information criteria, have links to Bayesian posterior inference under certain priors on the unknown coefficients. It can be shown that the lasso estimator represents the maximum a posteriori (MAP) estimator under an independent Laplace prior for the $\beta_j$. In a fully Bayesian framework one has the option of Markov chain Monte Carlo (MCMC) inference if sufficient computational resources are available. The advantage could be more accurate prediction via model averaging, i.e., entertaining many models of high posterior probability rather than simply the most probable (MAP) selection.

Researchers have discovered that this "trick" (regularizing in such a way as to automatically detect useful explanatory variables) has analogies in settings well beyond ordinary linear regression: from linear-logistic (and other generalized linear model families) to nonlinear and nonparametric settings. Basis expansion, which generates so-called *features* from explanatory variables by transformation and interaction (multiplying pairs of features together), allows linear models on those features to span rich *function spaces*, mapping inputs to outputs. This all works splendidly as long as judicious regularization is applied. Again, CV, other MC validation methods, and MCMC can play a viral role. MC and regularization are even important when entertaining inherently nonlinear models, such as those based on trees, kernels, and artificial neural networks.

Tree-based regression provides a divide-and-conquer approach to large-scale nonlinear modeling, and it is especially attractive when nonlinear interactions are present. Trees—merely special graphs to mathematicians—are a fundamental data structure to computer scientists with many efficient libraries available for convenient abstraction and fast implementation. Statisticians have simply ported tree-based data structures over to learning. The idea is to let the data decide how to "divvy" up the input space recursively via binary splits on individual input coordinates (e.g., $x_j \leq 5$) placed at internal *nodes* of the tree, so that simple models can be fit at the regions of the input space demarcated by the partitions, or the so-called *leaves* of the tree. Appropriate leaf models are dictated by the nature of the response, with constant (unknown mean) and linear models being appropriate for regression, and multinomial models for classification. Splitting locations can be chosen according to any of several optimization heuristics, again depending on the leaf model, but the key is preventing over-fit by growing trees too deep, i.e., by having too many elements in the partition. One solution to this again involves MC

validation (e.g., CV). These form the basis of older CART[1] methods. However, two newer schemes have become popular as computing power has increased several-fold: Bayesian MCMC exploration of tree-posteriors, and boosting and the bootstrap. The former boasts organic regularization through a prior over tree space, and it has been extended to sums of trees with Bayesian additive regression trees (BART). In BART, the prior encourages shallow trees when many are being summed. A non-Bayesian analogue of sums of trees, both actually predating BART, can be obtained via boosting or the bootstrap. Boosting targets a best (weighted) sum of shallow trees, or decision stumps, whereas a bootstrap can yield many trees, each fit to randomly subsampled data, which can be averaged, leading to a *random forest* predictor—an example of so-called bootstrap aggregating, or bagging. Tree based predictors, especially in ensembles, are hard to beat when inputs interact in their functional relationship to outputs, and when both sets of variables lack a degree of regularity, usually manifested in the form of smoothness.

Take either of those challenges away, and kernels are king. To understand how kernels work, it helps to think about how distance in the input space relates to correlation (i.e., linear dependence) and other forms of probabilistic or functional dependence in outputs. Testing locations with inputs far from training data inputs should have outputs which are less highly correlated, or dependent, and vice versa. There are some nice computational tricks in play when distances are measured in a certain way. And you can think of the choice of distance as a mapping from the original input space into a feature space, wherein calculations are relatively straightforward (linear). The best example is so-called Gaussian process regression, where pairwise (often inverse exponential Euclidean) distances in the input space define a multivariate normal (MVN) covariance structure. Then, simple MVN conditioning rules provide the predictive distribution. One can interpret the entire enterprise as Bayesian, with priors over function spaces leading to posteriors. However, in the opinion of many practitioners, that endows things with the sort of technical scaffolding that obfuscates, unless you are already of the opinion that all things Bayesian are good.

**A road-map.** One criticism may be that modern statistical learning feels a bit like shooting first and asking questions later: algorithms targeting a particular behavior, and if they seem to work somebody might try to prove a little theory about it—to explain why it works and how similar tactics may (or may not) port to other learning tasks. That makes for a landscape that can be hard to navigate, particularly for "newbies". Although there are prospectors who know their particular canyons well, there are few who know how we got here or where "here" is. Building a map from the old to the new requires cartographers to span both worlds. Such folks are in short supply. Brad Efron and Trevor Hastie are vanguards in the statistics world, having managed to be many things to many people and having been claimed by (modern) classical statisticians, Bayesians, and machine learning researchers alike.

Their new book, *Computer age statistical inference* is the (mathematically sophisticated) data scientist's road-map to modern statistical thinking and computation. It covers nearly all of the topics outlined above, in a succinct and elegant way, and with carefully crafted illustrative examples, emphasizing a methodology's evolution as well as its implementation, aspects which often speak more loudly about their popularity than technical detail. However this is not a cookbook. Although

[1]Classification and regression trees

pointers are made, usually to R packages, the book has almost no code. This is probably by design. If they had provided, say, R code, few in the machine learning community would buy it, as they prefer Python. And vice-versa with statisticians.

The book is in three parts. The first part, on classic statistical inference, offers some context. This is a great read for someone who knows the material already—an abridged summary of the landscape "a hundred years ago". The middle of the book, Part II, is even better. It covers a transitional period. The researchers who developed this methodology were prescient, or lucky, in that they anticipated computational developments on the horizon. They developed algorithms for the machines of the 1970s–1990s, but if computing never matured, none of them would have been household names (James–Stein, ridge regression, expectation maximization, bootstrap, CV, etc.). Although the topics in Part III comprise of essential tools in the modern arsenal, those in Part II are in many ways more important to the reader. These chapters teach the foundational statistical and computational concepts, out of which the Part III topics grew, and they will be key to understanding the next big thing.

Part III offers a sampling of the most important and most recent advances. It starts with large-scale (*big-p*) model selection, and the multiple testing issues which ensue, perhaps as a means to transition into variable selection via the lasso. Then comes tree-based methods, and ensembles thereof, via the bootstrap and boosting. These chapters, though short and sweet, are surprisingly complete. The latter chapters, however, leave much to the imagination. In part that is because research on some topics (e.g., deep neural networks (DNNs)) is perhaps in its infancy. In other places, full monographs offer greater insight, and references are provided. Or it may be a matter of the author's taste and expertise. Whereas support vector machines are rather more mature and, thus, well-presented in the text, they are paired with kernels and local regression, and the development here barely scratches the surface. Gaussian processes (GPs), which often out-perform neural networks (even deep ones) in lower signal-to-noise regimes (in part because they require less training data), do not even get a mention. The final two chapters feel a little misplaced. Post-selection inference, or how to correctly quantify uncertainty in estimated quantities (like standard errors) after model-selection via optimization (e.g., lasso), might be better placed closer to the start of the chapter.

Experts in the literature would pick up on the following three omissions. The first is that treatment of Bayesian inference is awkward. Of the four chapters covering Bayes, two have essentially the same title, "Empirical Bayes". Just about every method—from lasso/linear regression to generalized linear models, trees, forests, neural networks, and kernels—has highly impactful Bayesian analogues which do not get a mention in the text. Emphasis in the text is on Bayesian-lite methodology, including objective and Empirical Bayes, yet these are the variations which have benefited least from the advent of modern computation. The second issue regards methods tailored to ubiquitous yet specialized computing architecture such as graphical processing units (GPSs), symmetric multi-core, and distributed computing. Much work has been done to leverage these modern paradigms, to achieve astounding computational efficiency gains, yet sometimes at the expense of statistical efficiency. Divide and conquer, the technique exploited by trees, has been applied to kernels and GPs for vastly parallelized prediction. GPU implementations explain much of the excitement in DNNs, and algorithmic tricks derived thereof (not limited to stochastic gradient descent) are bleeding into other modeling paradigms.

Third and finally, there is nothing on fast/sparse and distributed linear algebra, which is dramatically expanding the sizes of problems that can be tackled with otherwise established ("old") methodology.

To wrap up with such shortcomings is definitely unfair as overall this is an excellent text, and to do justice to the hottest methods in statistics/machine learning could take thousands of pages. Computing advances have revolutionized so many aspects of our lives, and data—either its collection or the science or making sense of it—has come to dominate almost all of those aspects. The pace of innovation is feverish and is poaching talent from every corner of the quantitative sciences. Efron and Hastie's *Computer age statistical inference* offers an excellent handbook for new recruits: either post-doctoral scholars from non–data science backgrounds, or graduate students in statistics, machine learning, and computer science. It is a great primer that helps readers appreciate how we got to where we are, and how challenges going forward are linked to computation, implementation, automation, and a pragmatic yet disciplined approach to inference.

Robert B. Gramacy
Department of Statistics
Virginia Polytechnic Institute and State University
*Email address*: rbg@vt.edu