# Big Data: New Tricks for Econometrics

Hal R. Varian*

June 2013
Revised: December 26, 2013

**Abstract**

Nowadays computers are in the middle of almost every economic transaction. These "computer-mediated transactions" generate huge amounts of data, and new tools are necessary to manipulate and analyze this data. This essay offers a brief introduction to some of these tools and methods.

Computers are now involved in many economic transactions and can capture data associated with these transactions, which can then be manipulated and analyzed using a variety of techniques. Conventional statistical and econometric techniques such as regression often work well but there are some problems unique to big data sets that require different tools. There are several reasons for this.

First, the sheer size of the data involved may require more sophisticated data manipulation tools. Second, we may have more potential predictors than appropriate for estimation, so we need to do some kind of variable selection. Third, large data sets may allow for more flexible relationships than simple

*Thanks to Jeffrey Oldham, Tom Zhang, Rob On, Pierre Grinspan, Jerry Friedman, Art Owen, Steve Scott, Bo Cowgill, Brock Noland, Daniel Stonehill, and Gary King for comments on earlier versions of this paper.

linear models. Machine learning techniques such as decision trees, support vector machines, neural nets, deep learning and so on may allow for more effective ways to model complex relationships.

In this essay I will describe a few of these tools for manipulating and analyzing big data. I believe that these methods have a lot to offer and should be more widely known and used by economists. In fact, my standard advice to graduate students these days is "go to the computer science department and take a class in machine learning." There have been very fruitful collaborations between computer scientists and statisticians in the last decade or so, and I expect collaborations between computer scientists and econometricians will also be productive in the future.

# 1 Tools for data manipulation

Economists have historically dealt with data that fits in a spreadsheet, but that is changing as new more detailed data becomes available; see Einav and Levin [2013] for several examples and discussion. If you have more than a million or so rows in a spreadsheet, you probably want to store it in a relational database, such as MySQL. Relational databases offer a simple way to store, manipulate and retrieve data using a Structured Query Language (SQL) which is easy to learn and very useful for dealing with medium-sized data sets.

However, if you have several gigabytes of data or several million observations, standard relational databases become unwieldy. Databases to manage data of this size are generically known as "NoSQL" databases. The term is used rather loosely, but is sometimes interpreted as meaning "not only SQL." NoSQL databases are more primitive than SQL databases in terms of data manipulation capabilities but can handle larger amounts of data.

Due to the rise of computer mediated transactions, many companies have found it necessary to develop systems to process billions of transactions per day. For example, according to Sullivan [2012], Google has seen 30 trillion URLs, crawls over 20 billion of those a day, and answers 100 billion search queries a month. Analyzing even one day's worth of data of this size is virtually impossible with conventional databases. The challenge of dealing with data sets of this size led to the development of several tools to manage and analyze big data.

These tools are proprietary to Google, but have been described in academic publications in sufficient detail that open-source implementations have been developed. The list below has both the Google name and the name of related open source tools. Further details can be found in the Wikipedia entries associated with the tool names.

**Google File System** [Hadoop Distributed File System] This system supports files of to be distributed across hundreds or even thousands of computers.

**Bigtable** [Cassandra] This is a table of data that lives in the Google File System. It too can stretch over many computers.

**MapReduce** [Hadoop] This is a system for accessing manipulating data in large data structures such as Bigtables. MapReduce allows you to access the data in parallel, using hundreds or thousands of machines to do the particular data extraction you are interested in. The query is "mapped" to the machines and is then applied in parallel to different shards of the data. The partial calculations are then combined ("reduced") to create the summary table you are interested in.

**Go** [Pig] Go is an open-source general-purpose computer language that makes it easier to do parallel data processing.

**Dremel** [Hive, Drill,Impala] This is a tool that allows data queries to be written in a simplified form of SQL. With Dremel it is possible to run an SQL query on a petabtye of data (1000 terabytes) in a few seconds.

Though these tools can be run on a single computer for learning purposes, real applications use large clusters of computers such as those provided by Amazon, Google, Microsoft and other cloud computing providers. The ability to rent rather than buy data storage and processing has turned what was previously a fixed cost into a variable cost and has lowered the barriers to entry for working with big data.

## 2 Tools for data analysis

The outcome of the big data processing described above is often a "small" table of data that may be directly human readable or can be loaded into an SQL database, a statistics package, or a spreadsheet.

If the extracted data is still inconveniently large, it is often possible to select a subsample for statistical analysis. At Google, for example, I have found that random samples on the order of 0.1 percent work fine for analysis of economic data.

Once a dataset has been extracted it is often necessary to do some exploratory data analysis along with consistency and data-cleaning tasks. This is something of an art which can be learned only by practice, but there are data cleaning software tools such as OpenRefine and DataWrangler that can be used to assist in this task.

Data analysis in statistics and econometrics can be broken down into four categories: 1) prediction, 2) summarization, 3) estimation, and 4) hypothesis testing. Machine learning is concerned primarily with prediction; the closely related field of data mining is also concerned with summarization. Econometricians, statisticians, and data mining specialists are generally looking for insights that can be extracted from the data. Machine learning specialists are often primarily concerned with developing computers systems that can provide useful predictions and perform well in the presence of challenging computational constraints. Data science, a somewhat newer term, is concerned with both prediction and summarization, but also with data manipulation, visualization, and other similar tasks. The terminology is not standardized in these areas, so these statements reflect general usage, not hard-and-fast definitions. Other terms used computer assisted data analysis include knowledge extraction, information discovery, information harvesting, data archeology, data pattern processing, and exploratory data analysis.

Much of applied econometrics is concerned with detecting and summarizing relationships in the data. The most common tool used to for summarization is (linear) regression analysis. As we shall see, machine learning offers a set of tools that can usefully summarize more complex relationships in the data. We will focus on these regression-like tools since those are the most natural for economic applications.

In the most general formulation of a statistical prediction problem, we are interested in understanding the conditional distribution of some variable $y$ given some other variables $x = (x_1, \ldots, x_P)$. If we want a point prediction we could use the mean or median of the conditional distribution.

In machine learning, the x-variables are usually called "predictors" or "features." The focus of machine learning is to find some function that provides a good prediction of $y$ as a function of $x$. Historically, most work in machine learning has involved cross-section data where it is natural to think of the data being IID or at least independently distributed. The data may be "fat," which means lots of predictors relative to the number of observations, or "tall" which means lots of observations relative to the number of predictors.

We typically have some observed data on $y$ and $x$ and we want to compute a "good" prediction of $y$ given new values of $x$. Usually "good" means it minimizes some loss function such as the sum of squared residuals, mean of absolute value of residuals, and so on. Of course, the relevant loss is that associated with *new* observations of $x$, not the observations used to fit the model.

When confronted with a prediction problem of this sort an economist would think immediately of a linear or logistic regression. However, there may be better choices, particularly if a lot of data is available. These include nonlinear methods such as 1) neural nets, 2) support vector machines, 3) classification and regression trees, 4) random forests, and 5) penalized regression such as lasso, lars, and elastic nets.

I will focus on the last three methods in the list above, since they seem to work well on the type of data economists generally use. Neural nets and support vector machines work well for many sorts of prediction problems, but they are something of a black box. By contrast it is easy to understand the relationships that trees and penalized regressions describe. Much more detail about these methods can be found in machine learning texts; an excellent treatment is available in Hastie et al. [2009], which can be freely downloaded. Other suggestions for further reading are given at the end of this article.

# 3 General considerations for prediction

Our goal with prediction is typically to get good *out-of-sample predictions.* Most of us know from experience that it is all too easy to construct a predictor that works well in-sample, but fails miserably out-of-sample. To take a trivial example, $n$ linearly independent regressors will fit $n$ observations perfectly but will usually have poor out-of-sample performance. Machine learning specialists refer to this phenomenon as the "overfitting problem."

There are three major techniques for dealing with the overfitting problem which are commonly used in machine learning.

First, since simpler models tend to work better for out of sample forecasts, machine learning experts have come up with various ways penalize models for excessive complexity. In the machine learning world, this is known as "regularization" and we will encounter a some examples later one. Economists tend to prefer simpler models for the same reason, but have not been as explicit about quantifying complexity costs.

Second, it is conventional to divide the data into separate sets for the purpose of training, testing and validation. You use the training data to estimate a model, the validation data to choose your model, and the testing data to evaluate how well your chosen model performs. (Often validation and testing sets are combined.)

Third, in the training stage, it may be necessary to estimate some "tuning parameters" of the model. The conventional way to do this in machine learning is to use *k-fold cross validation*.

1. Divide the data into $k$ roughly equal subsets and label them by $s = 1, \ldots, k$. Start with subset $s = 1$.

2. Pick a value for the tuning parameter.

3. Fit your model using the $k - 1$ subsets other than subset $s$.

4. Predict for subset $s$ and measure the associated loss.

5. Stop if $s = k$, otherwise increment $s$ by 1 and go to step 2.

Common choices for $k$ are 10, 5, and the sample size minus 1 ("leave one out"). After cross validation, you end up with $k$ values of the tuning parameter and the associated loss which you can then examine to choose an appropriate value for the tuning parameter. Even if there is no tuning parameter, it is useful to use cross validation to report goodness-of-fit measures since it measures out-of-sample performance which is what is typically of interest.

Test-train and cross validation, are very commonly used in machine learning and, in my view, should be used much more in economics, particularly when working with large datasets. For many years, economists have reported in-sample goodness-of-fit measures using the excuse that we had small datasets. But now that larger datasets have become available, there is no reason not to use separate training and testing sets. Cross-validation also turns out to be a very useful technique, particularly when working with reasonably large data. It is also a much more realistic measure of prediction performance than measures commonly used in economics.

# 4 Classification and regression trees

Let us start by considering a discrete variable regression where our goal is to predict a 0-1 outcome based on some set of features (what economists would call explanatory variables or predictors.) In machine learning this is known as a *classification problem*. Economists would typically use a generalized linear model like a logit or probit for a classification problem.

A quite different way to build a classifier is to use a decision tree. Most economists are familiar with decision trees that describe a sequence of decisions that results in some outcome. A tree classifier has the same general form, but the decision at the end of the process is a choice about how to classify the observation. The goal is to construct (or "grow") a decision tree that leads to good out-of-sample predictions.

Ironically, one of the earliest papers on the automatic construction of decision trees was co-authored by an economist (Morgan and Sonquist [1963]). However, the technique did not really gain much traction until 20 years later in the work of Breiman et al. [1984] and his colleagues. Nowadays this prediction technique is known as "classification and regression trees", or "CART."

Consider the simple example shown in Figure 1, where we are trying to predict survivors of the Titanic using just two variables, age and which class of travel the passenger purchased.

Here is a set of rules that can be read off of this tree (more of a bush, really):

- class 3: predict died (370 out of 501)
- class 1 or 2 and younger than 16: predict lived (34 out of 36)

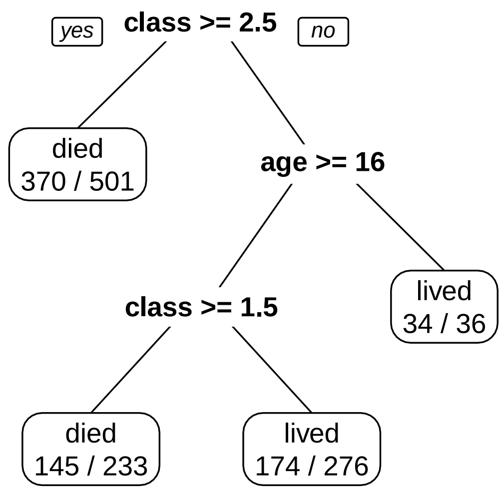


Figure 1: A classification tree for survivors of the Titanic. See text for interpretation.

- class 2 or 3 and older than 16: predict died (145 out of 233)
- class 1, older than 16: predict lived: (174 out of 276)

The rules fit the data reasonably well, misclassifying about 30% of the observations in the testing set.

This classification can also be depicted in the "partition plot" shown in Figure 2 which shows how the tree divides up the space of (age, class) pairs. Of course, the partition plot can only be used for 2 variables while a tree representation can handle an arbitrarily large number.

It turns out that there are computationally efficient ways to construct classification trees of this sort. These methods generally are restricted to binary trees (two branches at each node). They can be used for classification with multiple outcomes ("classification trees") , or with continuous dependent variables ("regression trees.")

Trees tend to work well for problems where there are important nonlin-

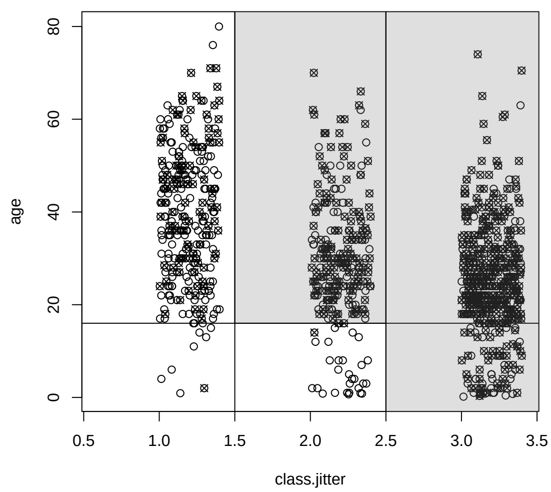


Figure 2: The simple tree model predicts death in shaded region. White circles indicate survival, black crosses indicate death.

earities and interactions. As an example, let us continue with the Titanic data and create a tree that relates survival to age. In this case, the rule generated by the tree is very simple: predict "survive" if age $< 8.5$ years. We can examine the same data with a logistic regression to estimate the probability of survival as a function of age:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.464813   0.034973  13.291   <2e-16 ***
age         -0.001894   0.001054  -1.796   0.0727 .
```

The tree model suggests that age is an important predictor of survival important, while the logistic model says it is barely important. This discrepancy is explained in Figure 3 where we plot survival rates by bins. Here we see that survival rates for those under 10 years old were elevated compared to older passengers, except for the very oldest group. So what mattered for survival is not so much age, but whether the passenger was a child or a senior. It

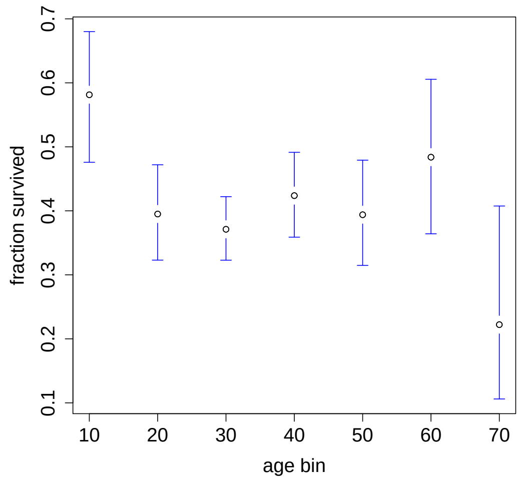


Figure 3: The figure shows the fraction of the population that survived for different age groups (0-10,10-20, and so on). The error bars are computed using the Wilson method.

would be difficult to discover this fact from a logistic regression alone.[1]

Trees also handle missing data well. Perlich et al. [2003] examined several standard data sets and found that "logistic regression is better for smaller data sets and tree induction for larger data sets." Interestingly enough, trees tend *not* to work very well if the underlying relationship really is linear, but there are hybrid models such as RuleFit (Friedman and Popescu [2005]) which can incorporate both tree and linear relationships among variables.

However, even if trees may not improve on predictive accuracy compared to linear models, the age example shows that they may reveal aspects of the data that are not apparent from a traditional linear modeling approach.

[1]It is true that if you knew that there was a nonlinearity in age, you use age dummies in the logit model to capture this effect. However the tree formulation made this nonlinearity quite apparent.

## 4.1 Pruning trees

One problem with trees is that they tend to overfit the data. The most widely-used solution to this problem is to "prune" the tree by imposing some complexity cost for having too many branches. This penalty for complexity is a form of regularization, which was mentioned earlier.

So, a typical tree estimation session might involve dividing your data into 10 folds, using 9 of the folds to grow a tree with a particular complexity, and then predict on the excluded fold. Repeat the estimation with different values of the complexity parameter using other folds and choose the value of the complexity parameter that minimizes the out-of-sample classification error. (Some researchers recommend being a bit more aggressive than that and choosing the complexity parameter that is one standard deviation lower than the loss-minimizing value.)

Of course, in practice, the computer program handles most of these details for you. In the examples in this paper I mostly use default choices, but in practices these default will often be tuned. As with any other statistical procedure, skill, experience and intuition are helpful in coming up with a good answer and diagnostics, exploration, and experimentation are just as useful with these methods as with regression techniques.

There are many other approaches to creating trees, including some that are explicitly statistical in nature. For example, a "conditional inference tree," or ctree for short, chooses the structure of the tree using a sequence of hypothesis tests. The resulting trees tend to need very little pruning. (Hothorn et al. [2006]) An example for the Titanic data is shown in Figure 4.

One might summarize this tree by the following principle: "women and children first ... particularly if they were traveling first class." This simple example again illustrates that classification trees can be helpful in summarizing relationships in data, as well as predicting outcomes.

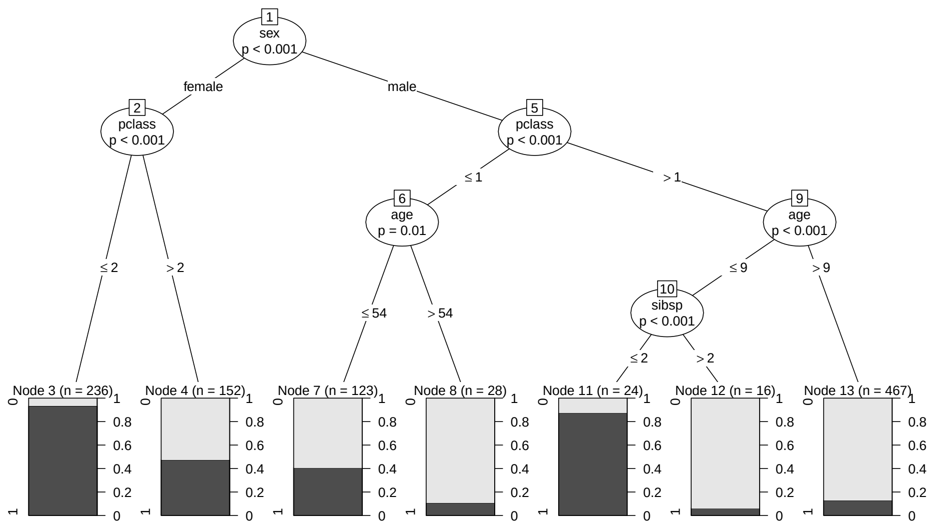


Figure 4: A ctree for survivors of the Titanic. The black bars indicate fraction of the group that survival.

## 4.2 Economic example: HMDA data

Munnell et al. [1996] examined mortgage lending in Boston to see if race played a significant role in determining who was approved for a mortgage. The primary econometric technique was a logistic regression where race was included as one of the predictors. The race effect indicated a statistically significant negative impact on probability of getting a mortgage for black applicants. This finding prompted lively subsequent debate and discussion, with 725 citations on Google Scholar as of July 2013.

Here I examine this question using the tree-based estimators described in the previous section. The data consists of 2380 observations of 12 predictors, one of which was race. Figure 5 shows a conditional tree estimated using the `R` package `party`. (For reasons of space, I have omitted variable descriptions which are readily available on the web site.)

The tree fits pretty well, misclassifying 228 of the 2380 observations for an error rate of 9.6%. By comparison, a simple logistic regression does slightly better, misclassifing 225 of the 2380 observations, leading to an error rate of 9.5%. As you can see in Figure 5, the most important variable is `dmi` = "denied mortgage insurance". This variable alone explains much of the variation in the data. The race variable (`black`) shows up far down the tree and seems to be relatively unimportant.

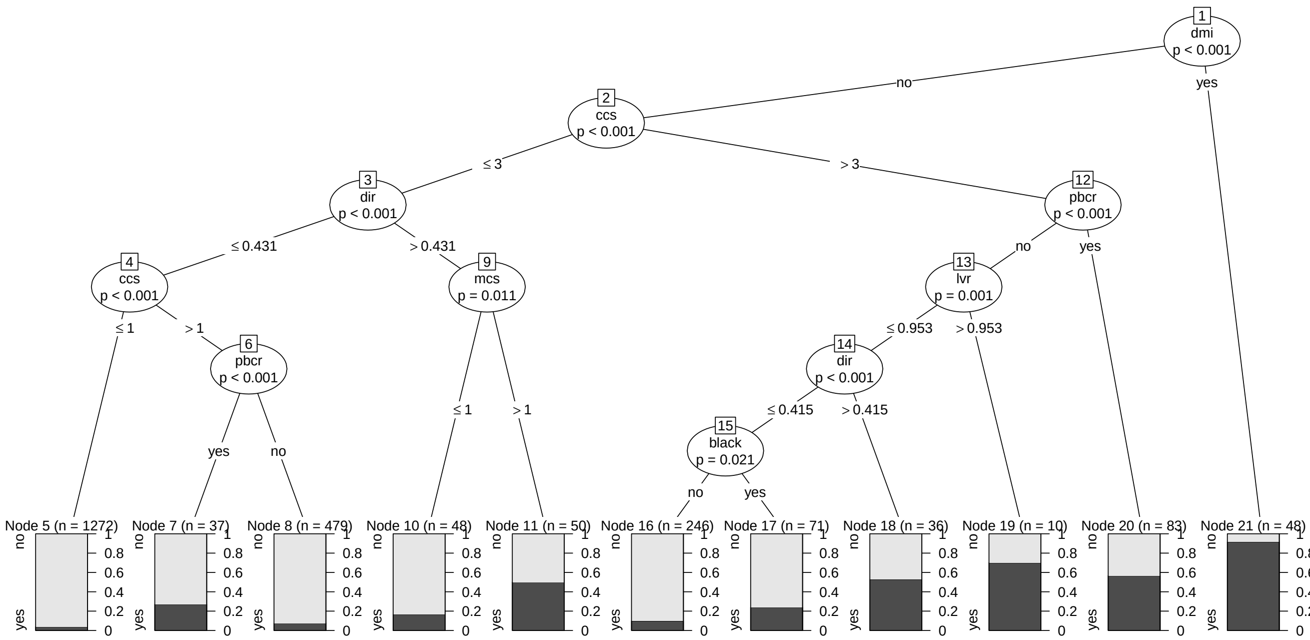


Figure 5: HMDA tree. The black bars indicate the fraction of each group that were denied mortgages. The most important determinant of this is the variable `dmi`, "denied mortgage insurance."

One way to gauge whether a variable is important is to exclude it from the prediction and see what happens. When this is done, it turns out that the accuracy of the tree based model doesn't change at all: exactly the same cases are misclassified. So there is a plausible decision tree model that ignores race that fits the observed data just as well as a model that includes race.

# 5 Boosting, bagging and bootstrap

There are several useful ways to improve classifier performance. Interestingly enough, the some of these methods work by *adding* randomness to the data. This seems paradoxical at first, but adding randomness turns out to be a helpful way of dealing with the overfitting problem.

**Bootstrap** involves choosing (with replacement) a sample of size $n$ from a data set of size $n$ to estimate the sampling distribution of some statistic. A variation is the "$m$ out of $n$ bootstrap" which draws a sample of size $m$ from a dataset of size $n > m$.

**Bagging** involves averaging across models estimated with several different bootstrap samples in order to improve the performance of an estimator.

**Boosting** involves repeated estimation where misclassified observations are given increasing weight in each repetition. The final estimate is then a vote or an average across the repeated estimates.

Econometricians are well-acquainted with the bootstrap rarely use the other two methods. Bagging is primarily useful for nonlinear models such as trees. (Friedman and Hall [2005].) Boosting tend to improve predictive performance of an estimator significantly and can be used for pretty much any kind of classifier or regression model, including logits, probits, trees, and so on.

It is also possible to combine these techniques and create a "forest" of trees that can often significantly improve on single-tree methods. Here is a rough description of how such "random forests" work.

**Random forests** refers to a technique that uses multiple trees. A typical procedure uses the following steps.

1. Choose a bootstrap sample of the observations and start to grow a tree.

2. At each node of the tree, choose a random sample of the predictors to make the next decision. Do not prune the trees.
3. Repeat this process many times to grow a forest of trees
4. The final classification is then determined by majority vote among all the trees in the forest

This method produces surprisingly good out-of-sample fits, particularly with highly nonlinear data. In fact, Howard [2013] claims "ensembles of decision trees (often known as Random Forests) have been the most successful general-purpose algorithm in modern times." He goes on to indicate that "the algorithm is very simple to understand, and is fast and easy to apply." See also Caruana and Niculescu-Mizil [2006] who compare several different machine learning algorithms and find that ensembles of trees perform quite well. There are a number variations and extensions of the basic "ensemble of trees" model such as Friedman's "Stochastic Gradient Boosting" (Friedman [1999]).

One defect of random forests is that they are a bit of a black box—they don't offer simple summaries of the data. However, they can be used to determine which variables are "important" in predictions in the sense of contributing the biggest improvements in prediction accuracy.

Note that random forests involves quite a bit of randomization; if you want to try them out on some data, I strongly suggest choosing a particular seed for the random number generator so that your results can be reproduced.

I ran the random forest method on the HMDA data and found that it misclassified 223 of the 2380 cases, a small improvement over the logit and the ctree. I also used the importance option in random forests to see how the predictors compared. It turned out that `dmi` was the most important predictor and race was second from the bottom which is consistent with the ctree analysis.

# 6 Variable selection

Let us return to the familiar world of linear regression and consider the problem of variable selection. There are many such methods available, including stepwise regression, principal component regression, partial least squares, AIC and BIC complexity measures and so on. Castle et al. [2009] describes and compares 21 different methods.

## 6.1 Lasso and friends

Here we consider a class of estimators that involves penalized regression. Consider a standard multivariate regression model where we predict $y_t$ as a linear function of a constant, $b_0$, and $P$ predictor variables. We suppose that we have standardized all the (non-constant) predictors so they have mean zero and variance one.

Consider choosing the coefficients $(b_1, \ldots, b_P)$ for these predictor variables by minimizing the sum of squared residuals plus a penalty term of the form

$$\lambda \sum_{p=1}^{P} [(1-\alpha)|b_p| + \alpha |b_p|^2]$$

This estimation method is called *elastic net regression*; it contains three other methods as special cases. If there is no penalty term ($\lambda = 0$), this is *ordinary least squares.* If $\alpha = 1$ so that there is only the quadratic constraint, this is *ridge regression.* If $\alpha = 0$ this is called the *lasso*, an acronym for "least absolute shrinkage and selection operator."

These penalized regressions are classic examples of regularization. In this case, the complexity is the number and size of predictors in the model. All of these methods tend to shrink the least squares regression coefficients towards zero. The lasso and elastic net typically produces regressions where some of the variables are set to be exactly zero. Hence this is a relatively straightforward way to do variable selection.

It turns out that these estimators can be computed quite efficiently, so doing variable selection on reasonably large problems is computationally feasible. They also seem to provide good predictions in practice.

## 6.2 Spike and slab regression

Another approach to variable selection that is novel to most economists is spike-and-slab regression, a Bayesian technique. Suppose that you have $P$ possible predictors in some linear model. Let $\gamma$ be a vector of length $P$ composed of zeros and ones that indicate whether or not a particular variable is included in the regression.

We start with a Bernoulli prior distribution on $\gamma$; for example, initially we might think that all variables have an equally likely chance of being in the regression. Conditional on a variable being in the regression, we specify a prior distribution for the regression coefficient associated with that variable. For example, we might use a Normal prior with mean 0 and a large variance. These two priors are the source of the method's name: the "spike" is the probability of a coefficient being non-zero; the "slab" is the (diffuse) prior describing the values that the coefficient can take on.

Now we take a draw of $\gamma$ from its prior distribution, which will just be a list of variables in the regression. Conditional on this list of included variables, we take a draw from the prior distribution for the coefficients. We combine these two draws with the likelihood in the usual way which gives us a draw from posterior distribution on both $\gamma$ and the coefficients. We repeat this process thousands of times using a Markov Chain Monte Carlo (MCMC) tehnique which give us a table summarizing the posterior distribution for $\gamma$ and the coefficients and the associated prediction of $y$.

We end up with a table of thousands of draws from the posterior distributions of $\gamma$, $\beta$, and $y$ which we can summarize in a variety of ways. For example, we can compute the average value of $\gamma_p$ which shows the posterior probability variable $p$ is included in the regressions.

| predictor | BMA | CDF(0) | lasso | spike-slab |
|---|---|---|---|---|
| GDP level 1960 | 1.000 | 1.000 | - | 0.9992 |
| Fraction Confucian | 0.995 | 1.000 | 6 | 0.9730 |
| Life expectancy | 0.946 | 0.942 | 5 | 0.9610 |
| Equipment investment | 0.757 | 0.997 | 1 | 0.9532 |
| Sub-Saharan dummy | 0.656 | 1.000 | - | 0.5834 |
| Fraction Muslim | 0.656 | 1.000 | - | 0.6590 |
| Rule of law | 0.516 | 1.000 | - | 0.4532 |
| Open economy | 0.502 | 1.000 | 3 | 0.5736 |
| Degree of Capitalism | 0.471 | 0.987 | - | 0.4230 |
| Fraction Protestant | 0.461 | 0.966 | - | 0.3798 |

Table 1: Comparing variable selection algorithms. See text for discussion.

## 6.3 Economic example: growth regressions

We illustrate the lasso and spike and slab regression with an example from Sala-i-Martín [1997]. This involves examining a multi-country set of predictors of economic growth in order to see which variables appeared to be the most important. Sala-i-Martín [1997] looked at all possible subsets of regressors of manageable size. Ley and Steel [2009] investigated the same question using Bayesian techniques related to, but not identical with, spike-and-slab, while Hendry and Krolzig [2004] examined an iterative significance test selection method.

Table 1 shows 10 predictors that were chosen by Sala-i-Martín [1997], Ley and Steel [2009], `lasso`, and `spike-and-slab`. The table is based on that in Ley and Steel [2009] but metrics used are not strictly comparable across models. The "BMA" and "spike-slab" columns are posterior probabilities of inclusion; the "lasso" column is just the ordinal importance of the variable with a dash indicating that it was not included in the chosen model; and the CDF(0) measure is defined in Sala-i-Martín [1997].

The `lasso` and the Bayesian techniques are very computationally efficient and on this ground would likely be preferred to exhaustive search. All 4 of these variable selection methods give similar results for the first 4 or 5 variables, after which they diverge. In this particular case, the data set appears to be too small to resolve the question of what is "important" for economic growth.

# 7 Time series

The machine learning techniques described up until now are generally applied to cross-sectional data where independently distributed data is a plausible assumption. However, there are also techniques that work with time series. Here we describe an estimation method which we call Bayesian Structural Time Series (BSTS) that seems to work well for variable selection problems in time series applications.

Our research in this area was motivated by Google Trends data which provides an index of the volume of Google queries on specific terms. One might expect that queries on [file for unemployment] might be predictive of the actual rate of filings for initial claims, or that queries on [Orlando vacation] might be predictive of actual visits to Orlando. Indeed, Choi and Varian [2009, 2012], Goel et al. [2010], Carrière-Swallow and Labbé [2011], McLaren and Shanbhoge [2011], Arola and Galan [2012], Hellerstein and Middeldorp [2012] and many others have shown that Google queries do have significant short-term predictive power for various economic metrics.

The challenge is that there are billions of queries so it is hard to determine exactly which queries are the most predictive for a particular purpose. Google Trends classifies the queries into categories, which helps a little, but even then we have hundreds of categories as possible predictors so that overfitting and spurious correlation are a serious concern. BSTS is designed to address these issues. We offer a very brief description here; more details are available in Scott and Varian [2012a,b].

Consider a classic time series model with *constant* level, linear time trend, and regressor components:

- $y_t = \mu + bt + \beta x_t + e_t$.

The "local linear trend" is a stochastic generalization of this model where the level and time trend can vary through time.

- Observation: $y_t = \mu_t + z_t + e_{1t}$ = level + regression
- State 1: $\mu_t = \mu_{t-1} + b_{t-1} + e_{2t}$ = random walk + trend
- State 2: $z_t = \beta x_t$ = regression
- State 3: $b_t = b_{t-1} + e_{3t}$ = random walk for trend

It is easy to add an additional state variable for seasonality if that is appropriate. The parameters to estimate are the regression coefficients $\beta$ and the variances of $(e_{it})$ for $i = 1, \ldots, 3$. We can then use these estimates to construct the optimal Kalman forecast.

For the regression we use the spike-and-slab variable choice mechanism described above. A draw from the posterior distribution now involves a draw of variances of $(e_{1t}, e_{2t})$, a draw of the vector $\gamma$ that indicates which variables are in the regression, and a draw of the regression coefficients $\beta$ for the included variables. The draws of $\mu_t$, $b_t$, and $\beta$ can be used to construct estimates of $y_t$ and forecasts for $y_{t+1}$. We end up with an (estimated) posterior distribution for the metric of interest. If we seek a point prediction, we could average over these draws, which is essentially a form of Bayesian model averaging.

As an example, consider the non-seasonally adjusted data for new homes sold in the U.S. (HSN1FNSA) from the St. Louis Federal Reserve Economic Data. This time series can be submitted to Google Correlate, which then returns the 100 queries that are the most highly correlated with the series. We feed that data into the BSTS system which identifies the predictors with the largest posterior probabilities of appearing in the housing regression are shown in Figure 6. Two predictors, [oldies lyrics] and [www.mail2web] appear to be spurious so we remove them and re-estimate, yielding the results in Figure 7. The fit is shown in Figure 8 which shows the incremental contribution of the trend, seasonal, and individual regressors components.

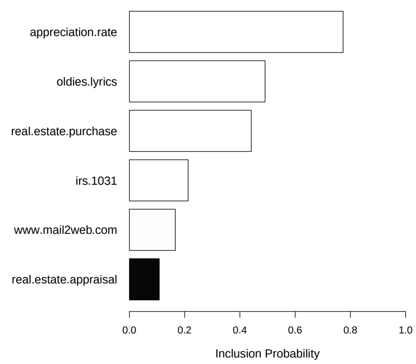


Figure 6: Initial predictors.

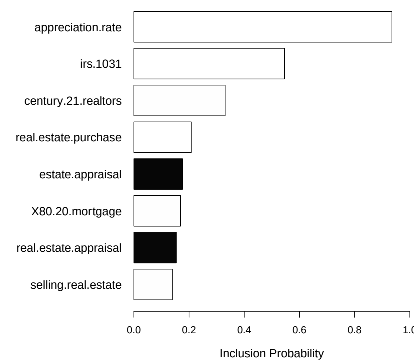


Figure 7: Final predictors.

# 8 Econometrics and machine learning

There are a number of areas where there would be opportunities for fruitful collaboration between econometrics and machine learning. I mentioned above that most machine learning uses IID data. However, the BSTS model shows that some of these techniques can be adopted for time series models. It is also be possible to use machine learning techniques to look at panel data and there has been some work in this direction.

Econometricians have developed several tools for causal modeling such as instrumental variables, regression discontinuity, and various forms of experiments. (Angrist and Krueger [2001].) Machine learning work has, for the most part, dealt with pure prediction. In a way this is ironic, since theoretical computer scientists, such as Pearl [2009a,b] have made significant contributions to causal modeling. However, it appears that these theoretical advances have not as yet been incorporated into machine learning practice to a significant degree.

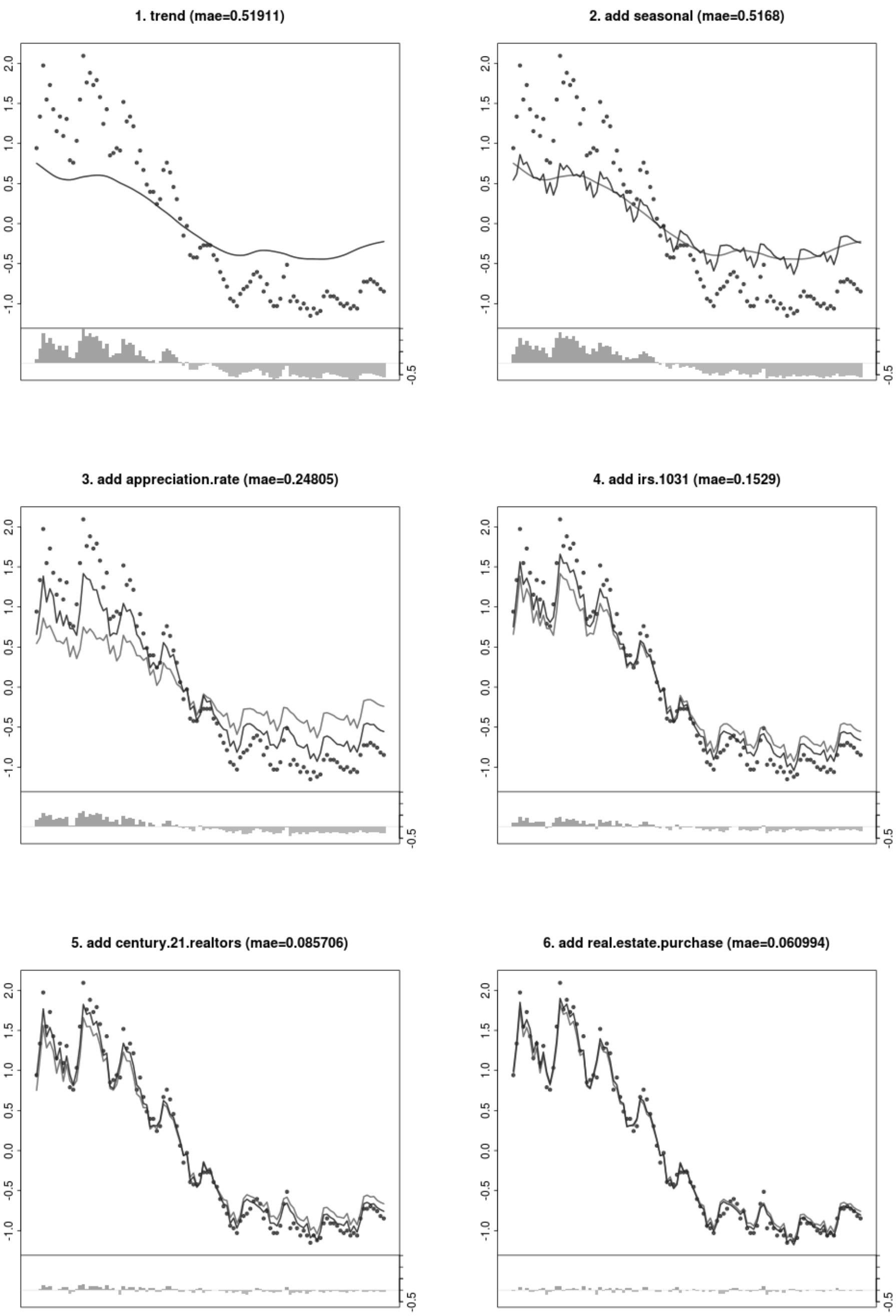


Figure 8: Incremental plots. The plots show the impact of the trend, seasonal, and a few individual regressors. The residuals are shown on the bottom.

## 8.1 Causality and prediction

As economists know well there is a big difference between correlation and causation. A classic example: there are often more police in precincts with high crime, but that does not imply that increasing the number of police in a precinct would increase crime.

The machine learning models we have described so far have been entirely about prediction. If our data was generated by policymakers who assigned police to areas with high crime, then the observed relationship between police and crime rates could be highly predictive for the *historical* data, but not useful in predicting the causal impact of explicitly *assigning* additional police to a precinct.

To enlarge on this point, let us consider an experiment (natural or designed) that attempts to estimate the impact of some policy, such as adding police to precincts. There are two critical questions.

- Which precincts will receive additional police in the experiment and policy implementation and how will this be determined? Possible assignment rules could be 1) random, 2) based on perceived need, 3) based on cost of providing service, 4) based on resident requests, 5) based on a formula or set of rules, 6) based on asking for volunteers, and so on. Ideally the assignment procedure in the experiment will be similar to that used in the policy. A good model for predicting which precincts will receive additional police under the proposed policy can clearly be helpful in estimating the impact of the policy.

- What will be the impact of these additional police in both the experiment and the policy? As Rubin [1974] and many subsequent authors have emphasized, when we consider the causal impact of some treatment we need to compare the outcome with the intervention to what *would have happened* without the intervention. But this counterfactual cannot be observed, so it must be predicted by some model. The better

> predictive model you have for the counterfactual, the better you will be able to estimate the causal effect, an observation that is true for both pure experiments and natural experiments.

So even though a predictive model will not necessarily allow one to conclude anything about causality by itself, such a model may help in estimating the causal impact of an intervention when it occurs.

To state this in a slightly more formal way, consider the identity from Angrist and Pischke [2008], page 11:

$$\begin{aligned}\text{observed difference in outcome} &= \text{average treatment effect on the treated}\\ &+ \text{selection bias}\end{aligned}$$

If you want to model the average treatment effect as a function of other variables, you will usually need to model both the observed difference and the selection bias. The better your predictive model for those components, the better predictions you can make about the average treatment effect. Of course, if you have a true randomized treatment-control experiment, selection bias goes away and those treated are an unbiased random sample of the population.

To illustrate these points, let us consider the thorny problem of estimating the causal effect of advertising on sales. (Lewis and Rao [2013].) The difficulty is that there are many confounding variables, such as seasonality or weather, that cause both increased ad exposures and increased purchases by consumers. Consider the (probably apocryphal) story about an advertising manager who was asked why he thought his ads were effective. "Look at this chart," he said. "Every December I increase my ad spend and, sure enough, purchases go up." Of course, seasonality can be observed and included in the model. However, generally there will be other confounding variables that affect both exposure to ads and the propensity of purchase, which makes causal interpretations of relationships problematic.

The ideal way to estimate advertising effectiveness is, of course, to run a controlled experiment. In this case the control group provides an estimate of what would have happened without ad exposures. But this ideal approach can be quite expensive, so it is worth looking for alternative ways to predict the counterfactual. One way to do this is to use the Bayesian Structural Time Series method described earlier. In this case, a model based on historical time series data can, in some cases, be used to estimate what *would have happened* in the absence of the advertising intervention. See Brodersen et al. [2013] for an example of this approach.

# 9 Model uncertainty

An important insight from machine learning is that averaging over many small models tends to give better out-of-sample prediction than choosing a single model.

In 2006, Netflix offered a million dollar prize to researchers who could provide the largest improvement to their existing movie recommendation system. The winning submission involved a "complex blending of no fewer than 800 models" though they also point out that "predictions of good quality can usually be obtained by combining a small number of judiciously chosen methods." (Feuerverger et al. [2012].) It also turned out that a blend of the best and second-best model outperformed both of them.

Ironically, it was recognized many years ago that averages of macroeconomic model forecasts outperformed individual models, but somehow this idea was rarely exploited in traditional econometrics. The exception is the literature on Bayesian model averaging which has seen a steady flow of work; see Steel [2011] for a survey.

However, I think that model uncertainty has crept in to applied econometrics through the back door. Many papers in applied econometrics present regression results in a table with several different specifications: which variables are included in the controls, which variables are used as instruments, and so on. The goal is usually to show that the estimate of some interesting parameter is not very sensitive to the exact specification used.

One way to think about it is that these tables illustrate a simple form of model uncertainty: how an estimated parameter varies as different models are used. In these papers the authors tend to examine only a few representative specifications, but there is no reason why they couldn't examine many more if the data were available.

In this period of "big data" it seems strange to focus on *sampling uncertainty,* which tends to be small with large data sets, while completely ignoring *model uncertainty* which may be quite large. One way to address this is to be explicit about examining how parameter estimates vary with respect to choices of control variables and instruments.

# 10 Summary and further reading

Since computers are now involved in many economic transactions, big data will only get bigger. Data manipulation tools and techniques developed for small datasets will become increasingly inadequate to deal with new problems. Researchers in machine learning have developed ways to deal with large data sets and economists interested in dealing with such data would be well advised to invest in learning these techniques.

I have already mentioned Hastie et al. [2009] which has detailed descriptions of all the methods discussed here but at a relatively advanced level. James et al. [2013] describes many of the same topics at an undergraduate-level, along with `R` code and many examples.[2]

Venables and Ripley [2002] contains good discussions of these topics with emphasis on applied examples. Leek [2013] presents a number of YouTube

[2] There are several economic examples in the book where the tension between predictive modeling and causal modeling is apparent.

videos with gentle and accessible introductions to several tools of data analysis. Howe [2013] provides a somewhat more advanced introduction to data science that also includes discussions of SQL and NoSQL databases. Wu and Kumar [2009] gives detailed descriptions and examples of the major algorithms in data mining, while Williams [2011] provides a unified toolkit. Domingos [2012] summarizes some important lessons which include "pitfalls to avoid, important issues to focus on and answers to common questions."

# References


Joshua D. Angrist and Alan B. Krueger. Instrumental variables and the search for identification: From supply and demand to natural experiments. *Journal of Economic Perspectives*, 15(4):69–85, 2001. URL `http://www.aeaweb.org/articles.php?doi=10.1257/jep.15.4.69`.

Joshua D. Angrist and Jörn-Steffen Pischke. *Mostly Harmless Econometrics*. Princeton University Press, 2008.

Concha Arola and Enrique Galan. Tracking the future on the web: Construction of leading indicators using internet searches. Technical report, Bank of Spain, 2012. URL `http://www.bde.es/webbde/SES/Secciones/Publicaciones/PublicacionesSeriadas/DocumentosOcasionales/12/Fich/do1203e.pdf`.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks/Cole, Monterey, 1984.

Kay H. Brodersen, Nicolas Remy, Fabian Gallusser, Steven L. Scott, Jim Koehler, and Penny Chu. Inferring causal impact using Bayesian structural time series models. Technical report, Google, Inc., 2013. URL `http://research.google.com/pubs/pub41854.html`.

Yan Carrière-Swallow and Felipe Labbé. Nowcasting with Google Trends in an emerging market. *Journal of Forecasting*, 2011. doi: 10.1002/for.1252. URL http://ideas.repec.org/p/chb/bcchwp/588.html. Working Papers Central Bank of Chile 588.

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

Jennifer L. Castle, Xiaochuan Qin, and W. Robert Reed. How to pick the best regression equation: A review and comparison of model selection algorithms. Technical Report 13/2009, Department of Economics, University of Canterbury, 2009. URL http://www.econ.canterbury.ac.nz/RePEc/cbt/econwp/0913.pdf.

Hyunyoung Choi and Hal Varian. Predicting the present with Google Trends. Technical report, Google, 2009. URL http://google.com/googleblogs/pdfs/google_predicting_the_present.pdf.

Hyunyoung Choi and Hal Varian. Predicting the present with Google Trends. *Economic Record*, 2012. URL http://people.ischool.berkeley.edu/~hal/Papers/2011/ptp.pdf.

Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10), October 2012. URL http://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf.

Liran Einav and Jonathan Levin. The data revolution and economic analysis. Technical report, NBER Innovation Policy and the Economy Conference, 2013.

Andrey Feuerverger, Yu He, and Shashi Khatri. Statistical significance of the Netflix challenge. *Statistical Science*, 27(2):202–231, 2012. URL http://arxiv.org/abs/1207.5649.

Jerome Friedman. Stochastic gradient boosting. Technical report, Stanford University, 1999. URL http://www-stat.stanford.edu/~jhf/ftp/stobst.pdf.

Jerome Friedman and Peter Hall. On bagging and nonlinear estimation. Technical report, Stanford University, 2005. URL http://www-stat.stanford.edu/~jhf/ftp/bag.pdf.

Jerome Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. Technical report, Stanford University, 2005. URL http://www-stat.stanford.edu/~jhf/R-RuleFit.html.

Sharad Goel, Jake M. Hofman, Sbastien Lahaie, David M. Pennock, and Duncan J. Watts. Predicting consumer behavior with web search. *Proceedings of the National Academy of Sciences*, 2010. URL http://www.pnas.org/content/107/41/17486.full.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2 edition, 2009. URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/download.html.

Rebecca Hellerstein and Menno Middeldorp. Forecasting with internet search data. *Liberty Street Economics Blog of the Federal Reserve Bank of New York*, January 2012. URL http://libertystreeteconomics.newyorkfed.org/2012/01/forecasting-with-internet-search-data.html.

David F. Hendry and Hans-Martin Krolzig. We ran one regression. *Oxford Bulletin of Economics and Statistics*, 66(5):799–810, 2004.

Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006.

Jeremy Howard. The two most important algorithms in predictive modeling today. Conference presentation, February 2013. URL http://strataconf.com/strata2012/public/schedule/detail/22658.

Bill Howe. Introduction to data science. Technical report, University of Washington, 2013. URL https://class.coursera.org/datasci-001/lecture/index.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, New York, 2013.

Jeff Leek. Data analysis, 2013. URL http://blog.revolutionanalytics.com/2013/04/coursera-data-analysis-course-videos.html.

Randall A. Lewis and Justin M. Rao. On the near impossibility of measuring the returns to advertising. Technical report, Google, Inc. and Microsoft Research, 2013. URL http://justinmrao.com/lewis_rao_nearimpossibility.pdf.

Eduardo Ley and Mark F. J. Steel. On the effect of prior assumptions in Bayesian model averaging with applications to growth regression. *Journal of Applied Econometrics*, 24(4):651–674, 2009. URL http://ideas.repec.org/a/jae/japmet/v24y2009i4p651-674.html.

Nick McLaren and Rachana Shanbhoge. Using internet search data as economic indicators. *Bank of England Quarterly Bulletin*, June 2011. URL http://www.bankofengland.co.uk/publications/quarterlybulletin/qb110206.pdf.

James N. Morgan and John A. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58 (302):415–434, 1963. URL http://www.jstor.org/stable/2283276.

Alicia H. Munnell, Geoffrey M. B. Tootell, Lynne E. Browne, and James McEneaney. Mortgage lending in Boston: Interpreting HDMA data. *American Economic Review*, pages 25–53, 1996.

Judea Pearl. *Causality*. Cambridge University Press, 2009a.

Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 4:96–146, 2009b.

Claudia Perlich, Foster Provost, and Jeffrey S. Simonoff. Tree induction vs. logistic regression: A learning-curve analysis. *Jounral of Machine Learning Research*, 4:211–255, 2003. URL `http://machinelearning.wustl.edu/mlpapers/paper_files/PerlichPS03.pdf`.

Donald Rubin. Estimating causal effects of treatment in randomized and non-randomized studies. *Journal of Educational Psychology*, 66(5):689, 1974.

Xavier Sala-i-Martín. I just ran two million regressions. *American Economic Review*, 87(2):178–83, 1997.

Steve Scott and Hal Varian. Bayesian variable selection for nowcasting economic time series. Technical report, Google, 2012a. URL `http://www.ischool.berkeley.edu/~hal/Papers/2012/fat.pdf`. Presented at JSM, San Diego.

Steve Scott and Hal Varian. Predicting the present with Bayesian structural time series. Technical report, Google, 2012b. URL `http://www.ischool.berkeley.edu/~hal/Papers/2013/pred-present-with-bsts.pdf`.

Mark F. J. Steel. Bayesian model averaging and forecasting. *Bulletin of E.U. and U.S. Inflation and Macroeconomic Analysis*, 200:30–41, 2011. URL `http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic-research/steel/steel_homepage/publ/bma_forecast.pdf`.

Danny Sullivan. Google: 100 billion searches per month, search to integrate gmail, launching enhanced search app for iOS. *Search Engine Land*, 2012. URL `http://searchengineland.com/google-search-press-129925`.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer-Verlag, New York, 4 edition, 2002.

Graham Williams. *Data Mining with Rattle and R*. Springer, New York, 2011.

Xindong Wu and Vipin Kumar, editors. *The Top Ten Algorithms in Data Mining*. CRC Press, 2009. URL `http://www.cs.uvm.edu/~icdm/algorithms/index.shtml`.