Algorithmic Criminology

Richard Berk
Department of Statistics
Department of Criminology
University of Pennsylvania

11/5/2012

Abstract

Computational criminology has been seen primarily as computer-intensive simulations of criminal wrongdoing. But there is a growing menu of computer-intensive applications in criminology that one might call "computational," which employ different methods and have different goals. This paper provides an introduction to computer-intensive, tree-based, machine learning as the method of choice, with the goal of *forecasting* criminal behavior. The approach is "black box," for which no apologies are made. There are now in the criminology literature several such applications that have been favorably evaluated with proper hold-out samples. Peeks into the black box indicate that conventional, causal modeling in criminology is missing significant features of crime etiology.

Keywords: machine learning; forecasting; criminal behavior; classification; random forests; stochastic gradient boosting; Bayesian additive regression trees

1 Introduction

Computational Criminology is a hybrid of computer science, applied mathematics, and criminology. Procedures from computer science and applied mathematics are used to animate theories about crime and law enforcement [1,2,3]. The primary goal is to learn how underlying mechanisms work; computational criminology is primarily about explanation. Data play a secondary

role either to help tune the simulations or, at least ideally, to evaluate how well the simulations perform [4].

There are other computer-intensive application in criminology that one might also call "computational." Procedures from statistics, computer science, and applied mathematics can be used to develop powerful visualization tools that are as engaging as they are effective [5,6]. These tools have been recently used in a criminology application [7], and with the growing popularity of electronic postings, will eventually become important components of circulated papers and books.

There are also a wide variety of computer-intensive methods used in law enforcement to assemble datasets, provide forensic information, or more broadly to inform administrative activities such as COMPSTAT [8]. Although these methods can be very useful for criminal justice practice, their role in academic criminology has yet to be clearly articulated.

In this paper, yet another form of computational criminology is discussed. Very much in the tradition of exploratory data analysis developed by John Tukey, Frederick Mosteller and others three decade ago [9], powerful computational tools are being developed to inductively characterize important but elusive structures in a dataset. The computational muscle has grown so rapidly over the past decade that the new applications have the look and feel of dramatic, qualitative advances. Machine learning is probably the poster-child for these approaches [10,11,12,13].

There are many new journals specializing in machine learning (e.g., *Journal of Machine Learning*) and many older journals that are now routinely sprinkled with machine learning papers (e.g., *Journal of the American Statistical Association*). So far, however, applications in criminology are hard to find. Part of the reason is time; it takes a while for new technology to diffuse. Part of the reason is software; the popular statistical packages can be at least five years behind recent developments. Yet another part of the reason is the need for a dramatic attitude adjustment among criminologists. Empirical research in criminology is thoroughly dominated by a culture of causal modeling in which the intent is to explain in detail the mechanisms by which nature generated the values of a response variable as a particular function of designated predictors and stochastic disturbances.[1] Machine learning comes from a different culture characterized by an "algorithmic" perspective.

> "The approach is that nature produces data in a black box whose insides are complex, mysterious, and, at least, partly unknowable.

> What is observed is a set of $\mathbf{x}'s$ that go in and a subsequent set of $\mathbf{y}'s$ that come out. The problem is to find an algorithm $f(\mathbf{x})$ such that for future $\mathbf{x}$ in a test set, $f(\mathbf{x})$ will be a good predictor of $\mathbf{y}$." [15]

As I discuss at some length elsewhere [16], the most common applications of machine learning in criminology have been to inform decisions about whom to place on probation, the granting of parole, and parole supervision practices. These are basically classification problems that build directly on parole risk assessments dating back to the 1920s. There are related applications informing police decisions in domestic violence incidents, the placement of inmates in different security levels, and the supervision of juveniles already in custody. These can all be seen successful *forecasting* exercises, at least in practical terms. Current decisions are informed by projections of subsequent risk. Such criminal justice applications guide the discussion of machine learning undertaken here. We will focus on *tree-based*, machine learning procedures as an instructive special case.

Four broad points will be made. First, machine learning is computational not just because it is computer-intensive, but because it relies algorithmic procedures rather than causal models.[2] Second, the key activity is data exploration in ways that can be surprisingly thorough. Patterns in the data commonly overlooked by conventional methods can be effectively exploited. Third, the forecasting procedures can be hand-tailored so that the consequences of different kinds of forecasting errors can be properly anticipated. In particular, false positives can be given more or less weight than false negatives. Finally, the forecasting skill can be impressive, at least relative to past efforts.

2 Conceptual Foundations

It all starts with what some call "meta-issues." These represent the conceptual foundation on which any statistical procedure rests. Without a solid conceptual foundation, all that follows will be *ad hoc*. Moreover, the conceptual foundation provides whatever links there may be between the empirical analyses undertaken and subject-matter theory or policy applications.

2.1 Conventional Regression Models

Conventional causal models are based on a quantitative theory of how the data were generated. Although there can be important difference in detail, the canonical account takes the form of a linear regression model such as

$$y_i = \mathbf{X}_i\boldsymbol{\beta} + \varepsilon_i, \tag{1}$$

where for each case i, the response y_i is a linear function of fixed predictors $\mathbf{X}_i$ (ususally including a column if 1's for the intercept), with regression coefficients $\boldsymbol{\beta}$, and a disturbance term $\varepsilon_i \sim NIID(0, \sigma^2)$. For a given case, nature (1) sets the value of each predictor, (2) combines them in a linear fashion using the regression coefficients as weights, (3) adds the value of the intercept, and (4) adds a random disturbance from a normal distribution with a mean of zero and a given variance. The result is the value of y_i. Nature can repeat these operations a limitless number of times for a given case with the random disturbances drawn independently of one another. The same formulation applies to all cases.

When the response is categorical or a count, there are some differences in how nature generates the data. For example, if the response variable $\mathbf{Y}$ is binary,

$$p_i = \frac{1}{1 + e^{-(\mathbf{X}_i\boldsymbol{\beta})}}, \tag{2}$$

where p_i is the probability of some event defined by $\mathbf{Y}$. Suppose that $\mathbf{Y}$ is coded "1" if a particular event occurs and "0" otherwise. (e.g., A parolee is arrested or not.) Nature combines the predictors as before, but now applies a logistic transformation to arrive at a value for p_i. (e.g., The cumulative normal is also sometimes used.) That probability leads to the equivalent of a coin flip with the probability that the coin comes up "1" equal to p_i. The side on which that "coin" lands determines for case i if the response is a "1" or a "0." As before, the process can be repeated independently a limitless number of times for each case.

The links to linear regression become more clear when Equation 2 is rewritten as

$$\ln\left(\frac{p_i}{1 - p_i}\right) = \mathbf{X}_i\boldsymbol{\beta}, \tag{3}$$

where p_i is, again, the probability of the some binary response whose "logit" depends linearly on the predictors.[3]

For either Equation 1, 2 or 3, a causal account can be overlaid by claiming that nature can manipulate the value of any given predictor independently of all other predictors. Conventional statistical inference can also be introduced because the sources of random variation are clearly specified and statistically tractable.

Forecasting would seem to naturally follow. With an estimate $\mathbf{X}_i\hat{\boldsymbol{\beta}}$ in hand, new values for $\mathbf{X}$ can be inserted to arrive at values for $\hat{\mathbf{Y}}$ that may be used as forecasts. Conventional tests and confidence interval can then be applied. There are, however, potential conceptual complications. If $\mathbf{X}$ is fixed, how does one explain the appearance of new predictor values $\mathbf{X}^*$ whose outcomes one wants to forecast? For better or worse, such matters are typically ignored in practice.

Powerful critiques of conventional regression have appeared since the 1970s. They are easily summarized: the causal models popular in criminology, and in the social sciences more generally, are laced with far too many untestable assumptions of convenience. The modeling has gotten far out ahead of existing subject-matter knowledge.[4] Interested readers should consult the writings of economists such as Leamer, LaLonde, Manski, Imbens and Angrist, and statisticians such as Rubin, Holland, Breiman, and Freedman. I have written on this too [19].

2.2 The Machine Learning Model

Machine Learning can rest on a rather different model that demands far less of nature and of subject matter knowledge. For given case, nature generates data as a random realization from a joint probability distribution for some collection of variables. The variables may be quantitative or categorical. A limitless number of realizations can be independently produced from that joint distribution. The same applies to every case. That's it.

From nature's perspective, there are no predictors or response variables. It follows that there is no such thing as omitted variables or disturbances. Often, however, researchers will use subject-matter considerations to designate one variable as a response $\mathbf{Y}$ and other variables as predictors $\mathbf{X}$. It is then sometimes handy to denote the joint probability distribution as $\Pr(\mathbf{Y}, \mathbf{X})$. One must be clear that the distinction between $\mathbf{Y}$ and $\mathbf{X}$ has absolutely nothing to do with how the data were generated. It has everything to do the what interests the researcher.

For a quantitative response variable in $\Pr(\mathbf{Y}, \mathbf{X})$, researchers often want

to characterize how the means of the response variable, denoted here by $\boldsymbol{\mu}$, may be related to $\mathbf{X}$. That is, researchers are interested in the conditional distribution $\boldsymbol{\mu}|\mathbf{X}$. They may even write down a regression-like expression

$$y_i = f(\mathbf{X}_i) + \xi_i, \tag{4}$$

where $f(\mathbf{X}_i)$ is the unknown relationship in nature's joint probability distribution for which

$$\mu_i = f(\mathbf{X}_i). \tag{5}$$

It follows that the mean of ξ_i in the joint distribution equals zero.[5] Equations 4 and 5 constitute a theory of how the response is related to the predictors in $\Pr(\mathbf{Y}, \mathbf{X})$. But any relationships between the response and the predictors are "merely" associations. There is no causal overlay. Equation 4 is not a causal model. Nor is it a representation of how the data were generated — we already have a model for that.

Generalizations to categorical response variables and their conditional distributions can be relatively straightforward. We denote a given outcome class by $\mathbf{G}_k$, with classes $k = 1 \ldots K$ (e.g., for $K = 3$, released on bail, released on recognizance, not released). For nature's joint probability distribution, there can be for any case i interest in the conditional probability of any outcome class: $p_{ki} = f(\mathbf{X}_i)$. There also can be interest in the conditional outcome class itself: $g_{ki} = f(\mathbf{X}_i)$.[6]

One can get from the conditional probability to the conditional class using the Bayes classifier. The class with the largest probability is the class assigned to a case. For example, if for a given individual under supervision the probability of failing on parole is .35, and the probability of succeeding on parole is .65, the assigned class for that individual is success. It is also possible with some estimation procedures to proceed directly to the outcome class. There is no need to estimate intervening probabilities.

When the response variable is quantitative, forecasting can be undertaken with the conditional means for the response. If $f(\mathbf{X})$ is known, predictor values are simply inserted. Then $\boldsymbol{\mu} = f(\mathbf{X}^*)$, where as before, $\mathbf{X}^*$ represents the predictor values for the cases whose response values are to be forecasted. The same basic rationale applies when outcome is categorical, either through the predicted probability or directly. That is, $\mathbf{p}_k = f(\mathbf{X}^*)$ and $\mathbf{G}_k = f(\mathbf{X}^*)$.

The $f(\mathbf{X})$ is usually unknown. An estimate, $\hat{f}(\mathbf{X})$, then replaces $f(\mathbf{X})$ when forecasts are made. The forecasts become estimates too. (e.g., $\boldsymbol{\mu}$ becomes $\hat{\boldsymbol{\mu}}$.) In a machine learning context, there can be difficult complications

for which satisfying solutions may not exist. Estimation is considered in more depth shortly.

Just like the conventional regression model, the joint probability distribution model can be wrong too. In particular, the assumption of independent realizations can be problematic for spatial or temporal data, although in principle, adjustments for such difficulties sometimes can be made. A natural question, therefore, is why have any model at all? Why not just treat the data as a population and describe its important features?

Under many circumstances treating the data as all there is can be a fine approach. But if an important goal of the analysis is to apply the findings beyond the data on hand, the destination for those inferences needs to be clearly defined, and a mathematical road map to the destination provided. A proper model promises both. If there is no model, it is very difficult to generalize any findings in a credible manner.[7]

A credible model is critical for forecasting applications. Training data used to build a forecasting procedure and subsequent forecasting data for which projections into the future are desired, should be realizations of the same data generation process. If they are not, formal justification for any forecasts breaks down and at an intuitive level, the enterprise seems misguided. Why would one employ a realization from one data generation process to make forecasts about another data generation process?[8]

In summary, the joint probability distribution model is simple by conventional regression modeling standards. But it provides nevertheless an instructive way for thinking about the data on hand. It is also less restrictive and far more appropriate for an inductive approach to data analysis.

3 Estimation

Even if one fully accepts the joint probably distribution model, its use in practice depends on estimating some of its key parameters. There are then at least three major complications.

1. In most cases, $f(\mathbf{X})$ is unknown. In conventional linear regression, one assumes that $f(\mathbf{X})$ is linear. The only unknowns are the values of the regression coefficients and the variance of the disturbances. With the joint probability distribution model, any assumed functional form is typically a matter descriptive convenience [14]. It is not informed by the model. Moreover, the functional form is often arrived at in

an inductive manner. As result, the functional form as well as its parameters usually needs to be estimated.

2. Any analogy to covariance adjustments is also far more demanding. To adjust the fitted values for associations among predictors, one must know the functional forms. But one cannot know those functional forms unless the adjustments are properly in place.

3. $\mathbf{X}$ is now a random variable. One key consequence is that estimates of $f(\mathbf{X})$ can depend *systematically* on which values of the predictors happen to be in the realized data. There is the likely prospect of bias. Because $f(\mathbf{X})$ is allowed to be nonlinear, which parts of the function one can "see" depends upon which values of the predictors are realized in the data. For example, a key turning point may be systematically missed in some realizations. When $f(\mathbf{X})$ is linear, it will materialize as linear no matter what predictor values are realized.

This is where the computational issues first surface. There are useful responses all three problems if one has the right algorithms, enough computer memory, and one or more fast CPUs. Large samples are also important.

3.1 Data Partitions as a Key Idea

We will focus on categorical outcomes because they are far more common than quantitative outcomes in the criminal justice applications emphasized here. Examples of categorical outcomes include whether or not an individual on probation or parole is arrested for a homicide [20], whether there is a repeat incident of domestic violence in a household [21], and different rule infractions for which a prison inmate may have been reported [22].

Consider a 3-dimensional scatter plot of sorts. The response is three color-coded kinds of parole outcomes: an arrest for a violent crime (red), an arrest for a crime that is not violent (yellow), and no arrest at all (green). There are two predictors in this cartoon example: age in years and the number of prior arrests.

The rectangle is a two-dimension predictor space. In that space, there are concentrations of outcomes by color. For example, there is a concentration of red circles toward the left hand side of the rectangle, and a concentration of green circles toward the lower right. The clustering of certain colors means that there is structure in the data, and because the predictor space is defined

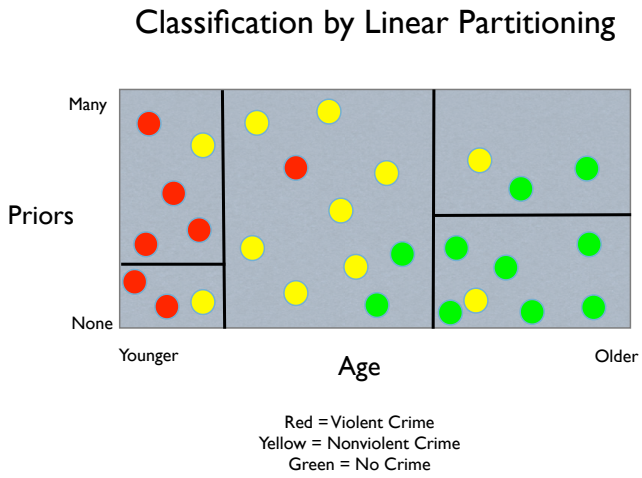

Figure 1: Data Partitions for Thee Categorical Outcomes By Age and the Number of Priors

by age and priors, the structure can be given substantive meaning. Younger individuals and individuals with a greater number of priors, for instance, are more likely to be arrested for violent crimes.

To make use of the structure in the data, a researcher must locate that structure in the predictor space. One way to locate the structure is to partition the predictor space in a manner that tends to isolate important patterns. There will be, for example, regions in which violent offenders are disproportionately found, or regions where nonviolent offenders are disproportionately found.

Suppose the space is partitioned as in Figure 1, where the partitions are defined by the horizontal and vertical lines cutting through the predictor space. Now what? The partitions can be used to assign classes to observations. Applying the Bayes classifier, the partition at the upper right, for example, would be assigned the class of no crime — the vote is 2 to 1. The partition at the upper left would be assigned the class of violent crime — the vote is 4 to 1. The large middle partition would be assigned the class of nonviolent crime — the vote is 7 to 2 to 1. Classes would be assigned to each partition by the same reasoning. The class with the largest estimated

probability wins.

The assigned classes can be used for forecasting. Cases with unknown outcomes but predictor values for age and priors can be located in the predictor space. Then, the class of the partition in which the case falls can serve as a forecast. For example, a case falling in the large middle partition would be forecasted to fail through an arrest for a nonviolent crime.

The two predictors function much like longitude and latitude. They locate a case in the predictor space. The partition in which a case falls determines its assigned class. That class can be the forecasted class. But there need be nothing about longitude and latitude beyond their role as map coordinates. One does not need to know that one is a measure of age, and one is a measure of the number of priors. We will see soon, somewhat counter-intuitively, that separating predictors from what they are supposed to measure can improve forecasting accuracy enormously. If the primary goal is to search for structure, how well one searchers drives everything else. This is a key feature of the algorithmic approach.

Nevertheless, in some circumstances, the partitions can be used to described how the predictors and the response are related in subject matter terms. In this cartoon illustration, younger individuals are much more likely to commit a violent crime, individuals with more priors are much more likely to commit a violent crime, and there looks to be a strong statistical interaction effect between the two. By taking into account the meaning of the predictors that locate the partition lines (e.g., priors more than 2 and age less than 25) the meaning of any associations sometimes can be made more clear. We can learn something about $f(\mathbf{X})$.

How are the partitions determined? The intent is to carve up the space so that overall the partitions are as homogeneous as possible with respect to the outcome. The lower right partition, for instance, has six individuals who were not arrested and one individual arrested for a nonviolent crime. Intuitively, that partition is quite homogeneous. In contrast, the large middle partition has two individuals who were not arrested, seven individuals who were arrested for a crime that was not violent, and one individual arrested for a violent crime. Intuitively, that partition is less homogenous. One might further intuit that any partition with equal numbers of individuals for each outcome class is the least homogenous it can be, and that any partition with all cases in a single outcome class is the most homogeneous it can be.

These ideas can be made more rigorous by noting that with greater homogeneity partition by partition, there are fewer classification errors overall.

For example, the lower left partition has an assigned class of violent crime. There is, therefore, one classification error in that partition. The large middle category has an assigned class of nonviolent crime, and there are three classification errors in that partition. One can imagine trying to partition the predictor space so that the total number of classification errors is as small as possible. Although for technical reasons this is rarely the criterion used in practice, it provides a good sense of the intent. More details are provided shortly.

At this point, we need computational muscle to get the job done. One option is to try all possible partitions of the data (except the trivial one in which partitions can contain a single observation). However, this approach is impractical, especially as the number of predictors grows, even with very powerful computers.

A far more practical and surprisingly effective approach is to employ a "greedy algorithm." For example, beginning with no partitions, a single partition is constructed that minimizes the sum of the classification errors in the two partitions that result. All possible splits for each predictor are evaluated and the best split for single predictor is chosen. The same approach is applied separately to each of the two new partitions. There are now four, and the same approach is applied once again separately to each. This recursive partitioning continues until the number of classification errors cannot be further reduced. The algorithm is called "greedy" because it takes the best result at each step and never looks back; early splits are never revisited.

Actual practice is somewhat more sophisticated. Beginning with the first subsetting of the data, it is common to evaluate a function of the proportion of cases in each outcome class (e.g., .10 for violent crime, .35 for nonviolent crime, and .55 for no crime). Two popular functions of proportions are

$$\text{Gini Index}: \qquad \sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} \tag{6}$$

and

$$\text{Cross Entropy or Deviance}: \qquad -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}. \tag{7}$$

The notation denotes different estimates of proportions $\hat{p}_{mk}$ over different outcome categories indexed by k, and different partitions of the data indexed by m. The Gini Index and the Cross-Entropy take advantage of the arithmetic fact that when the proportions over classes are more alike, their

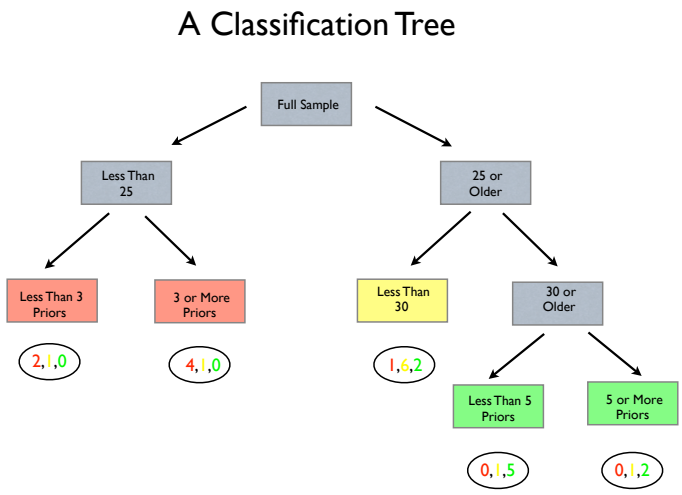

Figure 2: A Classication Tree For The Three Parole Outcomes and Preditors Age and Prior Record

product is larger (e.g., $[.5 \times .5] > [.6 \times .4]$). Intuitively, when the proportions are more alike, there is less homogeneity. For technical reasons we cannot consider here, the Gini Index is probably the preferred measure.

Some readers have may have already figured out that this form of recursive partitioning is the approach used for classification trees [23]. Indeed, the classification tree shown in Figure 2 is consistent with the partitioning shown in Figure 1.[9] The final partitions are color-coded for the class assigned by vote, and the number of cases in each final partition are color coded for their actual outcome class. In classification tree parlance, the full dataset at the top before any partitioning is called the "root node," and the final partitions at the bottom are called "terminal nodes."

There are a number of ways this relatively simple approach can be extended. For example, the partition boundaries do not have to be linear. There are also procedures called "pruning" that can be used to remove lower nodes having too few cases or that do not sufficiently improve the Gini Index.

Classification trees are rarely used these days as stand-alone procedures. They are well known to be very unstable over realizations of the data, especially if one wants to use the tree structure for explanatory purposes. In

addition, implicit in the binary partitions are step functions — a classification tree can be written as a form of regression in which the right hand side is a linear combination of step functions. However, it will be unusual if step functions provide a good approximation of $f(\mathbf{X})$. Smoother functions are likely to be more appropriate. Nevertheless, machine learning procedures often make use of classification trees as a component of much more effective and computer-intensive algorithms. Refinements that might be used for classification trees themselves are not needed; classification trees are means to an estimation end, not the estimation end itself.

4 Random Forests

Machine learning methods that build on classification trees have proved very effective in criminal justice classification and forecasting applications. Of those, random forecasts has been by far the most popular. We consider now random forests, but will provide a brief discussion of two other tree-based methods later. They are both worthy competitors to random forests.

A good place to start is with the basic random forests algorithm that combines the results from a large ensemble of classification trees [24].

1. From a training dataset with N observations, a random sample of size N is drawn with replacement. A classification tree is grown for the chosen observations. Observations that are not selected are stored as the "out-of-bag" (OOB) data. They serve as test data for that tree and will on average be about a third the size of the original training data.

2. A small sample of predictors is drawn at random (e.g., 3 predictors).

3. After selecting the best split from among the random subset of predictors, the first partition is determined. There are then two subsets of the data that together maximize the improvement in the Gini index.

4. Steps 2 and 3 are repeated for all later partitions until the model's fit does not improve or the observations are spread too thinly over terminal nodes.

5. The Bayes classifier is applied to each terminal node to assign a class.

6. The OOB data are "dropped" down the tree. Each observation is assigned the class associated with the terminal node in which it lands. The result is the predicted class for each observation in the OOB data for that tree.

7. Steps 1 through 6 are repeated a large number of times to produce a large number of classification trees. There are usually several hundred trees or more.

8. For each observation, the final classification is by vote over all trees when that observation is OOB. The class with the most votes is chosen. That class can be used for forecasting when the predictor values are known but the outcome class is not.

Because random forests is an ensemble of classification trees, many of the benefits from recursive partitioning remain. In particular, nonlinear functions and high order interaction effects can found inductively. There is no need to specify them in advance.

But, random forests brings its own benefits as well. The sampling of training data for each tree facilitates finding structures that would ordinarily be overlooked. Signals that might be weak in one sample might be strong in another. Each random sample provides look at the predictor space from different vantage point.

Sampling predictors at each split results in a wide variety of classification trees. Predictors that might dominate in one tree are excluded at random from others. As a result, predictors that might otherwise be masked can surface. Sampling predictors also means that the number of predictors in the training data can be *greater* than the number of observations. This is forbidden in conventional regression models. Researchers do not have to be selective in the predictors used. "Kitchen sink" specifications are fine.

The consequences of different forecasting errors are rarely the same, and it follows that their costs can differ too, often dramatically. Random forests accommodates in several ways different forecasting-error costs. Perhaps the best way is to use stratified sampling each time the training data are sampled. Oversampling the less frequent outcomes changes the prior distribution of the response and gives such cases more weight. In effect, one is altering the loss function. Asymmetric loss functions can be built into the algorithm right from the start. An illustration is provided below.

	Forecasted Not Fail	Forecasted Fail	Accuracy
Not Fail	9972	987	.91
Fail	45	153	.77

Table 1: Confusion Table for Forecasts of Perpetrators and Victims. True negatives are identified with 91% accuracy. True positives are identified with 77% accuracy. (Fail = perpetrator or victim. No Fail = not a perpetrator or victim.)

Random forests does not overfit [24] even if thousands are trees are grown. The OOB data serve as a test sample to keep the procedure honest. For other popular machine learning procedures, overfitting can be a problem. One important consequence is that random forests can provide consistent estimates of generalization error in nature's joint probability distribution for the particular response and predictors employed.[10]

4.1 Confusion Tables

The random forests algorithm can provide several different kinds of output. Most important is the "confusion table." Using the OOB data, actual outcome classes are cross-tabulated against forecasted outcome classes. There is a lot of information in such tables. In this paper, we only hit the highlights.

Illustrative data come from a homicide prevention project for individuals on probation or parole [25]. A "failure" was defined as (1) being arrested for homicide, (2) being arrest for an attempted homicide, (3) being a homicide victim, or (4) being a victim of a non-fatal shooting. Because for this population, perpetrators and victims often had the same profiles, no empirical distinction was made between the two. If a homicide was prevented, it did not matter if the intervention was with a prospective perpetrator or prospective victim.

However, prospective perpetrators or victims had first to be identified. This was done by applying random forests with the usual kinds predictor variables routinely available (e.g., age, prior record, age at first arrest, history of drug use, and so on). The number of classification trees was set at 500.[11]

Table 1 shows a confusion table from that project. The results are broadly representative of recent forecasting performance using random forests [10]. Of those who failed, about 77% were correctly identified by the random forests

algorithm. Of those who did not fail, 91% were correctly identified by the algorithm. Because the table is constructed from OOB data, these figures capture true forecasting accuracy.

But was this good enough for stakeholders? The failure base rate was about 2%. Failure was, thankfully, a rare event. Yet, random forests was able to search through a high dimensional predictor space containing over 10,000 observations and correctly forecast failures about 3 times out of 4 among those who then failed. Stakeholders correctly thought this was impressive.

How well the procedure would perform *in practice* is better revealed by the proportion of times when a forecast is made, the forecast is correct. This, in turn, depends on stakeholders' costs of false positives (i.e., individuals incorrectly forecasted to be perpetrators or victims) relative to the costs of false negatives (i.e., individuals incorrectly forecasted to neither be perpetrators nor victims). Because the relative costs associated with failing to correctly identify prospective perpetrators or victims were taken to be very high, a substantial number of false positives were to be tolerated. The cost ratio of false negatives to false positives was set at 20 to 1 *a priori* and built into the algorithm through the disproportional stratified sampling described earlier. This meant that relatively weak evidence of failure would be sufficient to forecast a failure. The price was necessarily an increase in the number of individuals incorrectly forecasted to be failures.

The results reflect this policy choice; there are in the confusion table about 6.5 false positives for every true positive (i.e., 987/153). As a result, when a failure is the forecasted, is it correct only about 15% of the time. When a success is the forecasted, it is correct 99.6% of the time. This also results from the tolerance for false positives. When a success is forecasted, the evidence is very strong. Stakeholders were satisfied with these figures, and the procedures were adopted.

4.2 Variable Importance For Forecasting

Although forecasting is the main goal, information on the predictive importance of each predictor also can be of interest. Figure 3 is an example from another application [16]. The policy question was whether to release an individual on parole. Each inmate's projected threat to public safety had to be a consideration in the release decision.

The response variable defined three outcome categories measured over 2 years on parole: being arrested for a violent crime ("Level 2"), being arrested

for a crime but not a violent one ("Level 1"), and not being arrested at all ("Level 0"). The goal was to assign one such outcome class to each inmate when a parole was being considered. The set of predictors included nothing unusual except that several were derived from behavior while in prison: "Charge Record Count," "Recent Report Count," and "Recent Category 1 Count" refer to misconduct in prison. Category1 incidents were considered serious.

Figure 3 shows the predictive importance for each predictor. The baseline is the proportion of times the true outcome is correctly identified (as shown in the rows of a confusion table). Importance is measured by the drop in accuracy when each predictor in turn is precluded from contributing. This is accomplished by randomly shuffling one predictor at a time when forecasts are being made. The set of trees constituting the random forests is not changed. All that changes is the information each predictor brings when a forecast is made.[12]

Because there are three outcome classes, there are three such figures. Figure 3 shows the results when an arrest for a violent crime is forecasted. Forecasting importance for each predictor is shown. For example, when the number of prison misconduct charges is shuffled, accuracy declines approximately 4 percentage points (e.g., from 60% accurate to 56% accurate).

This may seem small for the most important predictor, but because of associations between predictors, there is substantial forecasting power that cannot be cleanly attributed to single predictors. Recall that the use of classification trees in random forests means that a large number of interaction terms can be introduced. These are product variables that can be highly correlated with their constituent predictors. In short, the goal of maximizing forecasting accuracy can comprise subject-matter explanation.

Still, many of the usual predictors surface with perhaps a few surprises. For example, age and gender matter just as one would expect. But behavior in prison is at least as important. Parole risk instruments have in the past largely neglected such measures perhaps because they may be "only" indicators, not "real" causes. Yet for forecasting purposes, behavior in prison looks to be more important by itself than prior record. And the widely used LSIR adds nothing to forecasting accuracy beyond what the other predictors bring.

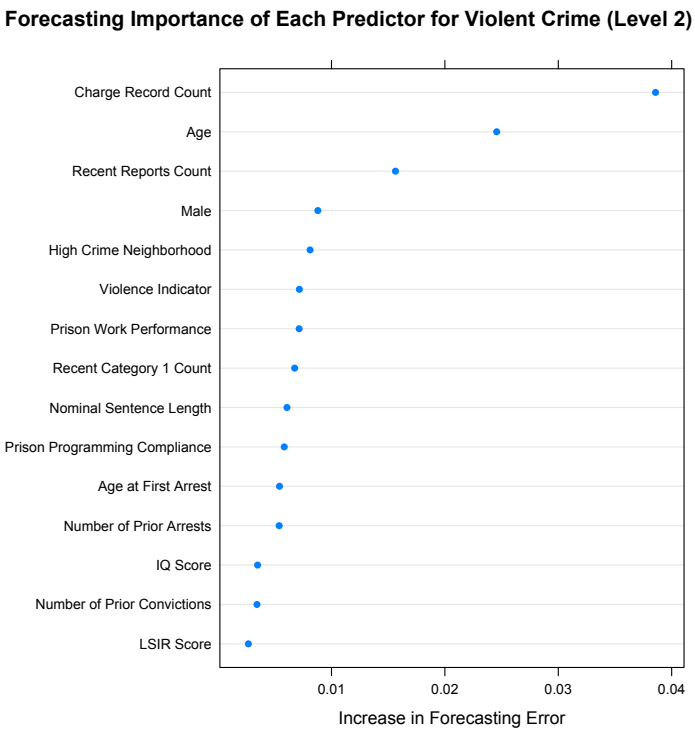

Figure 3: Predictor importance measured by proportional reductions in forecasting accuracy for violent crimes committed within 2 years of release on parole.

4.3 Partial Response Plots

The partial response plots that one can get from random forecasts and other machine learning procedures can also be descriptively helpful. The plots shows how a given predictor is related to the response with all other predictors held constant. An outline of the algorithm is as follows.

1. A predictor and a response class are chosen. Suppose the predictor is IQ and the response class is an arrest for a violent crime.
2. For each case, the value of IQ is set to one of the IQ values in the dataset. All other predictors are fixed at their existing values.
3. The fitted values of are computed for each case, and their mean calculated.
4. Steps 2 and 3 are repeated for all other IQ values.
5. The means are plotted against IQ.
6. Steps 2 through 5 are repeated for all other response variable classes.
7. Steps 1 through 6 are repeated for all other predictors.

Figure 4 shows how IQ is related to commission of a violent crime while on parole, all other predictors held constant. The vertical axis is in centered logits. Logits are used just as in logistic regression. The centering is employed so that when the outcome has more than two classes, no single class need be designated as the baseline.[13] A larger value indicates a greater probability of failure.

IQ as measured in prison has a nonlinear relationship with the log odds are being arrested for a violent crime. There is a strong, negative relationship for IQ scores from about 50 to 100. For higher IQ scores, there no apparent association. Some might have expected a negative association in general, but there seems to be no research anticipating a nonlinear relationship of the kind shown in Figure 4.

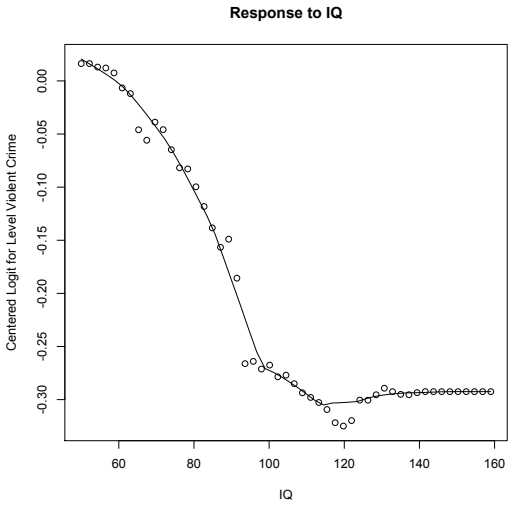

Figure 4: How inmate IQ is related to whether a violent crime is committed while on parole. The relationship is negative for below average IQ scores and flat thereafter.

5 Other Tree-Based Algorithms

For a variety of reasons, random forests is a particularly effective machine learning procedure for criminal justice forecasting [16]. But there are at least two other tree-based methods than can also perform well: stochastic gradient boosting [26, 27] and Bayesian additive regression trees [28]. Both are computer intensive and algorithmic in conception.

5.1 Stochastic Gradient Boosting

The core idea in stochastic gradient boosting is that one applies a "weak learner" over and over to the data. After each pass, the data are reweighted so that observations that are more difficult to accurately classify are given more weight. The fitted values from each pass through the data are used to update earlier fitted values so that the weak learner is "boosted" to become a strong learner. Here is an outline of the algorithm.

Imagine a training dataset in which the response is binary. Suppose "fail" is coded as "1" and "succeed" is coded as "0."

1. The algorithm is initialized with fitted values for the response. The overall proportion of cases that fail is a popular choice.

2. A simple random sample of the training data is drawn with a sample size about half the sample size of the training data.[14]

3. The negative gradient, also called "pseudo residuals," is computed. Just like with conventional residuals, each fitted value is subtracted from its corresponding observed value of 1 or 0. The residual will be *quantitive* not categorical: $(1 - p)$ or $-p$, where p is the overall proportion coded as "1."

4. Using the randomly-selected observations, a *regression* tree is fit to the pseudo residuals.[15]

5. The conditional mean in each terminal node serves as an estimate of the probability of failure.

6. The fitted values are updated by adding to the existing fitted values the new fitted values weighted to get the best fit.

7. Steps 2 through 6 are repeated until the fitted values no longer improve a meaningful amount. The number of passes can in practice be quite large (e.g., 10,000), but unlike random forests, stochastic gradient boosting can overfit [27]. Some care is needed because there is formally no convergence.

8. The fitted probability estimates can be used as is, or with the Bayes classifier transformed into assigned classes.

Stochastic gradient boosting handles wide range of response variable types in the spirit of the generalized linear model and more. Its forecasting performance is comparable to the forecasting performance of random forests. A major current liability is the requirement of symmetric loss functions for categorical response variables.

5.2 Bayesian Additive Regression Trees

Bayesian additive regression trees [28] is a procedure that capitalizes on an ensemble of classification (or regression) trees is a clever manner. Random

forests generates an ensemble of trees by treating the tree parameters as fixed but the data as random — data and predictors are sampled. Stochastic gradient boosting proceeds in an analogous fashion. Bayesian additive trees turns this upside-down. Consistent with Bayesian methods more generally, the data are treated as fixed once they are realized, and tree parameters are treated as random — the parameters are sampled. Uncertainty comes from the parameters, not from the data. Another difference is that one needs a model well beyond the joint probability distribution model. That model is essentially a form of linear regression. The outcome is regressed in a special way on a linear additive function of the fitted values from each tree combined with an additive disturbance term [29].

The parameter sampling takes four forms:

1. Whether or not to consider any partition of a node is determined by chance in a fashion that discourages larger trees;

2. if there is to be a split, the particular partitioning is determined by chance;

3. the proportion for each terminal node is selected at random from a distribution of proportions; and

4. the overall probability for each class is selected at random from a distribution of proportions.

The result can be a large ensemble classification trees (e.g., 300) that one might call a Bayesian forest.[16] The forest is intended to be a representative sample of classification trees constrained so that trees more consistent with prior information are more heavily represented.

The growing of Bayesian trees is embedded in the algorithm by which the fitted values from the trees are additively combined. The algorithm, a form of "backfitting" [30], starts out with each tree in its simplest possible form. The algorithm cycles though each tree in turn making it more complicated as needed to improve the fit while holding all other trees fixed at their current structure. Each tree may be revisited and revised many times. The process continues until there is no meaningful improvement in the fitted values. One can think of the result as a form of nonparametric regression in which a linear combination of fitted values is constructed, one set from each tree. In that sense, it is in the spirit of boosting.[17]

If one takes the model and the Bayesian apparatus seriously, the approach is not longer algorithmic. If one treats the model as a procedure, an algorithmic perspective is maintained. The perspective one takes can make a difference in practice. For example, if the model parameters are treated as tuning parameters, they are of little substantive interest and can be directly manipulated to improve performance. They are a means to an end, not an end in themselves.

Forecasting performance for Bayesian trees seems to be comparable to forecasting performance for random forests and stochastic gradient boosting. However, a significant weakness is that currently, categorical outcomes are limited to two classes. There is work in progress to handle the multinomial case. Another weakness is an inability to incorporate asymmetric loss, but here too there may soon be solutions.

6 Statistical Inference for Tree-Based Machine Learning

Even when the training data are treated as a random realization from nature's joint probability distribution, conventional statistical tests and confidence intervals are problematic for random forests and stochastic gradient boosting. Bayesian additive regression trees raises different issues to be briefly addressed shortly.

Consider statistical inference for forecasts, which figure so centrally in our discussion. In conventional practice, forecasting confidence intervals can be very useful. There is a model representing how outcome probabilities and/or classes are generated. That model specifies the correct functional form and disturbance distribution, and typically treats the predictors as fixed. A 95% confidence interval will cover each true probability 95% of the time over random realizations of the response variable. There can be similar reasoning for the outcome classes themselves.

The conventional formulation does not apply under the joint probability distribution model. There can be a "true" probability for every outcome class that one would like to estimate. There can be a "true" class, also a potential estimation target. But, there are no credible claims that estimates of either have their usual convenient properties. In particular, they not assumed to be an unbiased or even consistent estimates.

For reasons given at the beginning of Section 3, $\hat{f}(\mathbf{X})$ is taken to be some approximation of $f(\mathbf{X})$ that can contain both systematic and random error. Biased estimated are essentially guaranteed. When there is bias, confidence intervals do not have their stated coverage and test statistics computed under the null hypothesis do not have their stated probabilities.

In a forecasting setting, there is nevertheless the prospect of appropriate 95% "error bands." One takes the machine learning results as fixed, as they would be in a forecasting exercise, and considers bands around the fitted values that would contain 95% of the forecasts. Work is under way on how to construct such bands, and there is no doubt useful information in the residuals or a bootstrap using those residuals [31]. There are also useful, though less complete, approaches that can be applied to random forests in particular. The votes over trees provide some purchase on uncertainty associated with a forecasted class [16].

Bayesian additive regression trees can generate a predictive distribution of the fitted probabilities for either outcome class. These probabilities are treated like much like another set of parameters whose values are unknown but can be characterized by a particular distribution. Bayesian forecasting intervals then can be constructed [32]. One can determine, for instance, the range in which the middle 95% of the forecasted probabilities fall. And by placing a threshold through these probabilities at an appropriate location (e.g., .50), probabilities can be transformed into classes. However, one must not forget that the uncertainty being represented comes from uncertainty in the parameters that influence how the trees are grown. These depend on priors that can seem to some as fictions of convenience. In addition, some may not favor the Bayesian approach to begin with. Nevertheless, an algorithmic interpretation can still be appropriate and then forecasts and forecast uncertainty can be addressed in much the same fashion as for other kinds of tree-based machine learning.

7 Conclusions

Random forests, stochastic gradient boosting, and Bayesian additive trees are very different in conception and implementation. Yet in practice, they all can forecast well and typically much better than conventional regression models. Is there something important these tree-based method share beyond the use of large number of classification trees?

The use of tree ensembles can be viewed more abstractly as a way to effectively search a large predictor space for structure. With a large number of trees, the predictor space is sliced up in many different ways. Some sets of partitions will have stronger associations with the response variable than others and in the end, will have more weight in the forecasts that result. From this point of view, the subject-matter meaning of the predictors is a secondary concern. The predictors serve as little more than very high-dimensional coordinates for the predictor space.

Ensembles of classification trees are effective search engines for that space because of the following features that tree-based methods can share.

1. Using nature's joint probability distribution, compared to a regression causal model, as an account of how the data were generated removes a range of complications that are irrelevant for forecasting and otherwise put unnecessary constraints on the predictor-space search.

2. The use of step functions as each tree is grown can produce a very large number of new predictors. A single predictor such as age, might ultimately be represented by many indicator variables for different break points and many indicators for interaction effects. A search using, for example, 20 identified predictors such as gender and prior record, may be implemented with several hundred indicator variables. As a result, information in the initial 20 predictors can be more effectively exploited.

3. The use of indicator variables means that the search can arrive inductively at highly nonlinear relationships and very high order interactions, neither of which have to be specified in advance. Moreover, because any of the original predictors or sets of predictor can define splits differently over different trees, nonlinear relationships that are not step functions can be smoothed out as needed when the trees are aggregated to more accurately represent any associations.

4. When some form of random sampling is part of the algorithm — whether sampling of the data or sampling of the parameters — the content of the predictor space or the predictor space itself will vary [33]. Structure that might be masked for one tree might not be masked for another.

5. Aggregating fitted values over trees can add stability to forecasts. In effect, noise tends to cancel out.

6. Each of these assets are most evident in forecasting procedures built from large data sets. The high-dimensional predictor space needs lots of observations to be properly explored, especially because much of the search is for associations that one by one can be small. Their importance for forecasting materializes when the many small associations are allowed to contribute as a group. Consequently, training data sample sizes in the hundreds creates no formal problems, but the power of machine learning may not be fully exploited. Ideally, samples sizes should be at least in the 10s of thousands. Sample sizes of 100,000 or more are still better. It is sometimes surprising how much more accurate forecasts can be when the forecasting procedure is developed with massive datasets.

In summary, when subject-matter theory is well-developed and the training data set contains the key predictors, conventional regression methods can forecast well. When existing theory is weak or available data are incomplete, conventional regression will likely perform poorly, but tree-based forecasting methods can shine.[18]

There is growing evidence of another benefit from tree-based forecasting methods. Overall forecasting accuracy is usually substantially more than the sum of the accuracies that can be attributed to particular predictors. Tree-based methods are finding structure beyond what the usual predictors can explain.

Part of the reason is the black-box manner in a very large number of new predictors are generated as part of the search process. A new linear basis can be defined for each predictor and various sets of predictors. Another part of the reason is that tree-based methods capitalize on regions in which associations with the response are weak. One by one, such regions do matter much, and conventional regression approaches bent on explanation might properly choose to ignore them. But when a large number of such regions is taken seriously, forecasting accuracy can dramatically improve. There is important predictive information in the collection of regions, not in each region by itself.

An important implication is that there is structure for a wide variety of criminal justice outcomes that current social science does not see. In conventional regression, these factors are swept into the disturbance term

that, in turn, is assumed to be noise. Some will argue that this is a necessary simplification for causal modeling and explanation, but it is at least wasteful for forecasting and means that researchers are neglecting large chunks of the criminal justice phenomena. There are things to be learned from the "dark structure" that tree-based, machine learning shows to be real, but whose nature is unknown.

Notes

[1] A recent critique of this approach and a discussion of more promising, model-based alternatives can be found in [14].

[2] Usual criminology practice begins with a statistical model of some criminal justice process assumed to be have generated the data. The statistical model has parameters whose values need to be estimated. Estimates are produced by conventional numerical methods. At the other extreme are algorithmic approaches found in machine learning and emphasized in this paper. But, there can be hybrids. One may have a statistical model that motivates a computer-intensive search of a dataset, but there need be no direct connection between the parameters of the model and the algorithm used in that search. Porter and Brown [17] use this approach to detect simulated terrorist "hot spots" and actual concentrations of breaking and entering in Richmond, Virginia.

[3] Normal regression, Poisson regression, and logistic regression are all special cases of the generalized linear model. There are other special cases and close cousins such as multinomial logistic regression. And there are relatively straightforward extensions to multiple equation models, including hierarchical models. But in broad brush strokes, the models are motivated in a similar fashion.

[4] A very instructive illustration is research claiming to show that the death penalty deters crime. A recent National Research Council report on that research [18] is devastating.

[5] Recall that in a sample, the sum of the deviation scores around a mean or proportion (coded 1/0) is zero.

[6] The notation $f(\mathbf{X})$ is meant to represent *some* function of the predictors that will vary depending on the contect.

[7] Sometimes, the training data used to build the model and the forecasting data for which projections are sought are probabilitiy samples from the same finite population. There is still a model of how the data were generated, but now that model can be demostrably correct. The data were generated by a particular (known) form of probability sampling.

[8] One might argue that the two are sufficiently alike. But then one is saying that the two data generations processes are similar enough to be treated as the same.

[9] In the interest of space we do not consider the order in which the partitions shown in Figure 1 were constructed. One particular order would lead precisely to the classification tree in Figure 2.

[10] Roughly speaking, Breiman's generalization error is the probability that a case will be classified incorrectly in limitless number of independent realizations of the data. Breiman provides an accessible formal treatment [24] in his classic paper on random forests. One must be clear that this is not generalization error for the "right" response and predictors. There is no such thing. The generalization error is for the particular response and predictors analyzed.

[11] Although the number of trees is a tuning parameter, the precise number of trees does not usually matter as long as there are at least several hundred. Because random forests does not overfit, having more trees than necessary is not a serious problem. 500 trees typically is an appropriate number.

[12] In more conventional language, the "model" is fixed. It is not reconstructed as each predictor in turn is excluded from the forecasting exercise. This is very different from dropping each predictor in turn and regrowing the forest each time. Then, both the forest and the effective set of predictors would change. The two would be confounded. The goal here is to characterize the importance of each predictor for a *given* random forest.

[13] For any fitted value, the vertical axis units can be expressed as $\mathrm{f}_k(m) = \log p_k(m) - \frac{1}{K}\sum_{l=1}^{K} \log p_l(m)$. The function is the difference in log units between the proportion for outcome class k computed at the value m of a given predictor and the average proportion at that value over the K classes for that predictor.

[14] This serves much the same purpose as the sampling with replacement used in random forests. A smaller sample is adequate because when sampling without replacement, no case is selected more than once; there are no "duplicates."

[15] The procedure is much the same as for classification trees, but the fitting criterion is the error sum of squares or a closely related measure of fit.

[16] As before, the number of trees is a tuning parameter, but several hundred seems to be a good number.

[17] The procedure is very computationally intensive because each time fitted values are required in the backfitting process, a posterior distribution must be approximated. This leads to the repeated use of an MCMC algorithm that by itself is computer intensive.

[18] Another very good machine learning candidate is support vector machines. There is no ensemble of trees. Other means are employed to explore the predictor space effectively. Hastie and his colleagues [12] provide an excellent overview.

References

1. Groff E, Mazerolle L (eds.): **Special issue: simulated experiments in criminology and criminology justice.** *Journal of Experimental Criminology* 2008, 4(3).

2. Liu L, Eck J (eds.): *Artificial crime analysis: using computer simulations and geographical information systems* IGI Global; 2008.

3. Brantingham PL: **Computational criminology.** *Intelligence and Security Infomatics Conference* 2011, European.

4. Berk RA: **How you can tell if the simulations in computational criminology are any good.** *Journal of Experimental Criminology* 2008, 3: 289–308.

5. Cook D, Swayne DF: *Interactive and dynamic graphics for data analysis* Springer; 2007.

6. Wickham H: *ggplot: elegant graphics for data analysis* Springer; 2009.

7. Berk RA, MacDonald J: (2009) **The dynamics of crime regimes.** *Criminology* 2009, 47(3): 971–1008.

8. Brown DE, Hagan S: **Data association methods with applications to law enforcement.** *Decision Support Systems* 2002, 34: 369–378.

9. Hoaglin DC, Mosteller F, Tukey, J (eds.): *Exploring data tables, trends, and shapes.* John Wiley; 1985.

10. Berk RA: *Statistical learning from a regression perspective.* Springer; 2008.

11. Bishop C: *Pattern recognition and machine learning* Springer; 2006.

12. Hastie T, Tibshirani R, Friedman J: *The elements of statistical learning* Second Edition. Springer; 2009.

13. Marsland S: *Machine learning: an algorithmic perspective* CRC Press; 2009.

14. Berk RA, Brown L, George E, Pitkin E, Traskin M, Zhang K, Zhao L: **What you can learn from wrong causal models.** *Handbook of causal analysis for social research*, S. Morgan (ed.) Springer; 2012.

15. Breiman L: **Statistical modeling: two cultures.** (with discussion) *Statistical Science* 2001, 16: 199–231.

16. Berk RA: *Criminal justice forecasts of risk: a machine learning approach* Springer; 2012.

17. Porter MD, Brown DE: **Detecting local regions of change in high-dimensional criminal or terrorist point Processes.** *Computational Statistics & Data Analysis* 2007, 51: 2753–2768.

18. Nagin DS, Pepper JV: *Deterrence and the death penalty.* Committee on Law and Justice; Division on Behavioral and Social Sciences and Education; National Research Council; 2012.

19. Berk RA: *Regression analysis: a constructive critique.* Sage publications; 2003.

20. Berk RA, Sherman L, Barnes G, Kurtz E, Ahlman L: **Forecasting murder within a population of probationers and parolees: a high stakes application of statistical learning.** *Journal of the Royal Statistics Society — Series A* 2009, 172 (part I): 191–211.

21. Berk RA, Sorenson SB, He Y: **Developing a practical forecasting screener for domestic violence incidents.** *Evaluation Review* 2005, 29(4): 358–382.

22. Berk RA, Kriegler B, Baek J-H: **Forecasting dangerous inmate misconduct: an application of ensemble statistical procedures.** *Journal of Quantitative Criminology* 2006, 22(2) 135–145.

23. Breiman L, Friedman JH, Olshen RA, Stone CJ: *Classification and regression trees* Wadsworth Press; 1984.

24. Breiman L: **Random forests.** *Machine Learning* 2001, 45: 5–32.

25. Berk RA: **The role of race in forecasts of violent crime.** *Race and Social Problems* 2009, 1: 231–242.

26. Friedman JH: **Stochastic gradient boosting.** *Computational Statistics and Data Analysis* 2002, 38: 367–378.

27. Mease D, Wyner AJ, Buja A: **Boosted classification trees and class probability/quantile estimation.** *Journal of Machine Learning Resaerch* 2007, 8: 409–439.

28. Chipman HA, George EI, McCulloch RE: **BART: bayesian additive regression trees.** *Annals of Applied Statistics* 2010, 4(1): 266–298.

29. Chipman, HA, George, EI, McCulloch, RE (2007). **Bayesian Ensemble Learning.** *Advances in Neural Information Processing Systems* 19 2007: 265-272 .

30. Hastie TJ, Tibshirani RJ: *Generalized additive models* Chapman & Hall; 1990.

31. Efron BJ, Tibshirani RJ: *An introduction to the bootstrap* Chapman & Hall; 1993.

32. Gweke, J, Whiteman, C **Bayesian Forecasting** *Handbook of Forecasting, Volume 1*, G Elliot, CWJ Granger, A. Timmermann (eds.) Elsevier; 2006.

33. Ho TK: **The random subspace method for constructing decision trees.** *IEEE Transactions on Pattern Recognizition and Machine Intelligence* 1998, 20 (8) 832–844.