

Interpretable Statistical Learning Methods

Zachary M. Jones*

Abstract

Statistical learning methods, which are a flexible class of methods capable of estimating complex data generating functions, offer an attractive alternative to conventional methods, which often rely on strong, frequently inappropriate, assumptions about the functional form of the data generating process. A key impediment to the use of statistical learning methods is that they often output a "black box" which makes predictions but cannot be directly interpreted. I show that this need not be the case by describing and implementing methods based on Monte-Carlo integration that are capable of making any method that generates predictions interpretable. This allows researchers to learn about relationships in the data without having to prespecify their functional form. I illustrate this approach using a simulated example and an application to the prediction of burglary reports in Chicago.

Introduction

The social world is often complex, messy, and difficult to describe with a simple set of rules or laws. In contrast, conventional methods *presume* data are generated by relatively simple processes (King 1989; R. A. Berk 2008). This hampers our ability to discover relationships in our data that we did not expect and reduces the predictive validity of our models (Miller and Page 2009; Beck, King, and Zeng 2000; Beck and Jackman 1998; Schrodt 2014; Fariss and Jones 2017; R. A. Berk 2004). With conventional methods the data cannot tell us directly about relationships which were not prespecified in the model. Furthermore, the manner in which the data can inform our understanding of relationships which *were* prespecified in the model is constrained by the often strong assumptions made about their functional form, e.g., it is difficult to learn about nonlinear and interactive relationships from a linear additive model. A key benefit of making strong functional form assumptions is that models can be made to be directly interpretable, by, e.g., assuming linearity, simple forms of nonlinearity (e.g., polynomial terms), and/or the absence of interactions involving more than, say, two variables.¹. This comes at a substantial price, however. Conventional methods can miss interesting structure in data when that structure cannot be captured by the assumed model, which means that this information is not extracted from the data, damaging our ability to improve our understanding and degrading the quality of our predictions.

Advances in computing have allowed the development of statistical learning methods which allow us to make fewer assumptions about the structure in our data and consequentially to

*Email: zmj@zmjones.com.

¹Interpretability can be thought of in a number of ways (See, e.g., Doshi-Velez and Kim (2017)), but here I use to to mean models which have additive components which can be visualized, e.g., for continuous inputs, functions of two or fewer covariates.

discover unexpected relationships in our data (J. Friedman, Hastie, and Tibshirani 2001). Statistical learning methods differ from conventional statistical methods in that the number of parameters estimated is a function of the data, rather than specified by the researcher. This gives statistical learning methods the ability to find relationships in data that were not prespecified. This ability to find the unexpected, however, frequently makes the outputs from these methods uninterpretable, which, I argue, has prevented their wide use in political science, where the focus is on substantive insights in addition to prediction (Fariss and Jones 2017; Ward, Greenhill, and Bakke 2010; Hill and Jones 2014). This is not generally a problem with conventional statistical methods because we *assume* that the function that generated the data has a relatively simple functional form which is usually easily interpretable.

I describe and implement general tools which can make any statistical learning method interpretable in ways that allow political scientists to answer questions about the importance of covariates, the nature of their estimated relationship with an outcome, and the presence of interactions amongst said covariates (and their shape) (J. H. Friedman 2001; J. H. Friedman and Popescu 2008; Hooker 2004; Goldstein et al. 2015). I have made these methods available in an accessible manner via three R packages: the **Machine Learning in R** (**mlr**) package, which also provides infrastructure for fitting, tuning, and evaluating a wide variety statistical learning methods, the **Exploratory Data Analysis using Random Forests** (**edarf**) package, which also contains interpretation methods specific to random forests, a particularly attractive statistical learning method for political scientists, and **Monte-Carlo Methods for Prediction Functions** (**mmpf**) which allows prediction functions to be marginalized to depend on a subset of the covariates (Jones and Linder 2016; Bischl et al. 2016; Jones 2017).

The reason that statistical learning methods do not generally give interpretable outputs is that the manner in which they estimate a model, which is necessitated by their ability to learn complex relationships which were not presupposed, results in an output that, even if the true data generating process or true model *is* human interpretable, would hide that fact: a “black box.” Throughout this paper I will use the following simulated example

$$\begin{aligned} f(v, w, x, z) &= x + v^2 + xw + \sin(z) \\ y &= f + \epsilon \end{aligned} \tag{1}$$

where w , v , and z are covariates uniformly distributed between -2 and 2 , x is a covariate equal to $-1, 0, 1$ with equal probability, and ϵ is drawn from a standard normal distribution. Implicitly the “weight” or β for each term is 1 .

A simple example of a statistical learning method is a regression tree (Breiman et al. 1984). The way a regression tree estimates a model (i.e., how it arrives at predictions), is by subsetting (i.e., partitioning or grouping) y using the covariates repeatedly so that the variability of y within these subsets is minimized. This amounts to finding observations which are similar in y using the covariates, i.e., estimating a smoothed relationship between the covariates and the outcome, which is the same task that conventional regression methods solve. The result of this partitioning process is a nested set of subsets, a hierarchical structure (a tree): rules which describe the final subset an observation ends up in. As can be seen in fig. 1 this output is substantially more difficult to interpret than a simple linear model. This problem is exacerbated when the estimated tree is larger (more complex), or when it is combined with other trees as in ensemble methods like boosted trees or random forests, which are generally preferable for reasons described in later sections. While eq. 1 *could* be

estimated with conventional methods, it would require the inclusion of the product of x and w , a third degree polynomial expansion of z to approximate the sine function between -2 and 2 , a squared term to estimate the effect of v and a separate term to estimate the linear effect of x . Statistical learning methods can estimate this function without prespecification, and the interpretation methods I will describe herein can extract these terms.

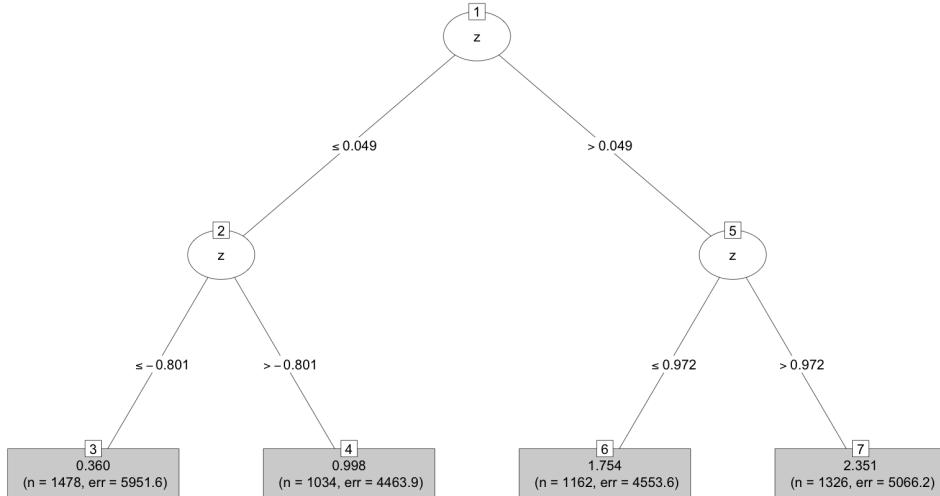


Figure 1: A model estimated via a regression tree from 5000 samples from the data generating function eq. 1. Each circle indicates a subset or partition defined on the labeled covariate, with the rule for creating the subset listed on the line below the circle. The final subsets, or terminal nodes, shown in grey, give them predictions of the tree, the number of observations that fall into that subset, and the mean square error of the prediction on those observations. This tree was constrained to be simple for this visualization.

While there have been some applications of statistical learning in political science, their use has been largely limited to instances in which forecasting (e.g., Goldstone et al. (2010)) is the primary goal, with a few notable exceptions (Beck, King, and Zeng 2000; Hainmueller and Hazlett 2013; Hill and Jones 2014). While the feed-forward artificial neural networks and kernel regularized least squares used in Beck, King, and Zeng (2000) and Hainmueller and Hazlett (2013) respectively are powerful methods which could be appropriately applied to many problems, they are not appropriate for *every* application, and thus it is desirable to be able to use, and interpret, the results of the best method available for a given problem. The methods described herein do precisely that, and, additionally, can be used with conventional methods. As previously mentioned I will illustrate these methods using data simulated from eq. 1 as well as burglary reports from the Chicago Police Department from 2001 to the present. In both cases I use random forests, as implemented by Wright and Ziegler (2015).

The prediction of crime is a high-stakes application because crime prediction systems are

used in many cases to make decisions about where to allocate limited policing resources (Lum and Isaac 2016). Using a “black box” prediction method may suffice for some purposes, but understanding how the covariates used in these models relate to the predictions is important in developing explanations for reported patterns of crime as well as diagnosing any differential effects these predictions may have on economic and demographic groups (Adebayo and Kaggle 2016; R. Berk 2009). These data are taken from the City of Chicago’s online data portal provided by the Chicago Police Department, and list reported burglaries from 2001 to the present. I aggregate these data to the police-beat level (a geographic unit assigned to 8-9 police officers who are assigned to that area for at least a year) for the morning and evening of each day, giving X -many observations over this period, with Y -many reported burglaries. I include “seasonality” covariates such as binary variables for AM/PM, the day of the week, the month of the year, and the year, the police beat, which is a geographic variable, and variables which measure temporal dynamics: the total number of burglaries reported in the previous day, the last week, and the last month, for each beat. Lastly, I include similar lagged totals of other reported crimes. This is a relatively small set of covariates but it is similar to that used by other crime forecasting systems and serves to illustrate the usefulness of interpreting models, even in cases where their primary use is forecasting (based on entries to the National Institute of Justice’s real-time crime forecasting challenge).

Before proceeding to the details of these methods for making statistical learning methods interpretable, I will review why statistical learning methods can outperform conventional methods, why this makes their output uninterpretable, and how they can be viewed in the same statistical framework.

Statistical Learning

Statistical learning methods like artificial neural networks, decision trees, random forests, boosting, and others have seen use in political science, but are far from common (Beck, King, and Zeng 2000; Grimmer and Stewart 2013; Hill and Jones 2014; Montgomery et al. 2015; Green and Kern 2012; Montgomery and Olivella 2015; Hainmueller and Hazlett 2013). Statistical learning methods can be represented in the same framework as more conventional statistical methods. Here I review these equivalences and highlight both the difficulties and advantages of using statistical learning methods as compared to conventional methods.

Assume that the (\mathbf{X}, Y) are random variables drawn from some probability distribution \mathbb{P} over the sample space $\mathcal{X} \times \mathcal{Y}$, which represents all combinations of \mathbf{X} and Y that could be obtained, wherein Y represents an outcome and \mathbf{X} covariates.

We typically estimate the function which maps \mathbf{X} to Y by using sample data presumed drawn from \mathbb{P} . Often we are interested in the conditional expectation of $Y|\mathbf{X}$, and we assume that this function depends on some parameters β .

$$\mathbb{E}[Y|\mathbf{X} = \mathbf{x}] = f(\mathbf{x}, \beta)$$

Samples of Y are denoted \mathbf{y} and of \mathbf{X} , \mathbf{x} . To estimate f we need to define a set of functions which can be searched for the best approximation to f , e.g., for a linear regression model the space of all possible values of the regression coefficients for the covariates included. We also

need a way to decide which values of these parameters are “best.” Finally, for computational reasons, we need a way to efficiently search through the set of possible models.

Defining the Space of Possible Models

A *hypothesis space*, denoted \mathcal{H} , is a class which defines all possible functions that could be used to approximate the true data generating function f . In the case of a linear regression the hypothesis space is the set of functions which are linear and additive in the parameters. Specifically, for a two parameter simple linear regression of the form $y_i = \beta_0 + \beta_1 x_i + \epsilon$, the hypothesis space contains all the models where β_0 and β_1 are each real numbers between negative and positive infinity. So \mathcal{H} contains an infinite number of models. We can estimate an f with nonlinear relationships between \mathbf{X} and Y if we expand the basis (i.e., the set of functions which represent \mathbf{X}) by, for example, including polynomials of \mathbf{x} , but this must be specified in advance rather than being automatically learned from the data. In the case of this two parameter linear model the dimension of the space of possible parameters is fixed even though \mathcal{H} contains an infinite number of models.

$$h(\mathbf{x}, \boldsymbol{\beta}) = \boldsymbol{\beta}^T \mathbf{x} \quad (2)$$

Here each $h \in \mathcal{H}$, that is, eq. 2, is one of the possible models contained in the hypothesis space. Estimating, or learning, a model from data is searching over the hypothesis space to find the “best” candidate model.

In contrast to conventional methods like linear regression, statistical learning methods like boosting, decision trees, random forests, and artificial neural networks allow the number of parameters to be determined by the data. For example a regression tree, as in fig. 1 can be written as

$$h(\mathbf{x}) = \sum_{m=1}^M w_m \mathbb{I}(\mathbf{x} \in R_m) \quad (3)$$

where $\mathbb{I}(\cdot)$ is the indicator function which equals 1 if $\mathbf{x} \in R_m$ and 0 if not, R_m is the m ’th of the final, disjoint, and exhaustive partitions of the data (the terminal nodes as in fig. 1), and w_m is the mean value of \mathbf{y} in this region. Each region R_m is defined by which covariates, and which values in those covariates, were used to create the final set of partitions in the tree: so their number is data-dependent, generally growing with the size and complexity of the sample data. Other statistical learning methods work in a similar fashion: increasing their complexity as needed to best estimate f .

Evaluating Candidate Models

How can we evaluate different candidate models in the hypothesis space? A loss function in this context \mathcal{L} describes how good a particular h is at predicting \mathbf{y} ; in particular, how much different sorts of errors “cost” in some sense. Specifically, the loss \mathcal{L} here is a function which maps predictions made by an approximation to f , that is, a member of \mathcal{H} , which we’ve called h but is conventionally referred to as \hat{f} , and observed realizations of Y , \mathbf{y} , to a

positive real number. Herein lies a problem though, as we are trying to approximate f , not perfectly reconstruct \mathbf{y} . Y is assumed herein to be random, and so is presumed to contain a random and a systematic component, i.e., $\mathbf{y} = f(\mathbf{x}) + \epsilon$, as in eq. 1. If we could compute the discrepancy between \hat{f} and f rather than \hat{f} and \mathbf{y} we would be able to distinguish between candidate hypotheses h which have over-adapted to the sample data \mathbf{y} , and those that have more accurately estimated f and hence will generalize better to new data (since ϵ will make any new data drawn from Y different than the sample data \mathbf{y}). However, since we do not know f we have to use \mathbf{y} , which conflates systematic error, when the function estimated from the sample data, called \hat{f} imperfectly represents f , and error due to the irreducible randomness in Y . Thus minimizing the loss over the observed data may result in models \hat{f} which are close approximations of \mathbf{y} rather than f : over-fitting. This is the opposite of the problem of picking a method where the closest member of the hypothesis space is far from f : under-fitting. Under or over-fitting leave us with models which can be rather useless in learning about f , that is, explanations about how covariates are related to the outcome, which also means that predictions made on new data will also be worse than they need be. This problem is illustrated graphically in fig. 2.

In the case of under-fitting the necessary solution is clear: chose a method with a hypothesis space \mathcal{H} which either contains f or is close to f . Since f is unknown, the practical implication of this is that avoiding under-fitting often necessitates the use of methods which have large hypothesis spaces: statistical learning methods, unless other information provides reliable information about the properties of f .

Like in the case of under-fitting, there are ways to avoid over-fitting such as minimizing the expected loss over the distribution of $Y, \mathbf{X}: \mathbb{P}$. The expected loss is usually called the risk, and minimizing it can eliminate optimism that comes from over-fitting the sample data. However, since \mathbb{P} is unknown this is not directly possible. While the sample-average loss is an estimator of the risk, it is generally biased downwards, that is, it is optimistic (i.e., it indicates the model fits better than it actually does), when applied to the same data on which the model was estimated (J. Friedman, Hastie, and Tibshirani 2001). A common solution to this issue is to estimate the risk by using Monte-Carlo methods like cross-validation or the bootstrap which randomly shuffle the data to make it more difficult for over-fit models to appear better than they truly are at generalizing beyond the sample data (J. Friedman, Hastie, and Tibshirani 2001).² Conventional penalized regression models like the least absolute shrinkage and selection operator (LASSO) or ridge regression make use of this idea to shrink the regression coefficients of a linear model (sometimes to 0), making it simpler, to avoid over-fitting.³ Statistical learning methods use this idea to make the model as simple or complex as is necessary to minimize risk.

Choosing a loss function is an application specific task that depends on the costs of making different sorts of prediction errors. With a binary outcome, it might be the case that positive cases are more costly when they are not predicted than are negative cases that are not predicted, or, with a continuous outcome, over-prediction might be more costly than under-prediction (R. Berk et al. 2009; Muchlinski et al. 2016). For example mispredicting cases where persons convicted of violent crimes reoffend after parole might be more costly

²It is also sometimes possible to derive an upper bound on the risk given the “size” (e.g., the cardinality or the complexity) of \mathcal{H} , the amount of data available, and the average loss on the sample data (V. Vapnik, Levin, and Le Cun 1994; V. N. Vapnik 1998).

³Another way of saying this is that regularization or shrinkage makes the model less sensitive to the sample data or that it makes the model smoother.

than predicting recitivism amongst those convicted of non-violent crimes, and it is possible that in some cases over-predicting recitivism is less consequential than under-predicting it. While squared-loss or misclassification-loss is often used because it makes the third step, the search or optimization strategy, easier, it is not necessarily the most appropriate choice, and what loss function is used has an enormous impact on what function from the hypothesis space is chosen as the best approximation to f since it is *the* criteria for that choice.

Searching for the Best Model

In many cases, even for a simple linear regression, the hypothesis space contains an infinite number of models, and so it is impossible to evaluate every possibility. A way to search through the hypothesis space must be devised. In some cases this can be done with relative ease, but for many hypothesis spaces even the best available optimization methods cannot guarantee that they will find the best approximation to the true function because of the shape of the loss function over the hypothesis space (i.e., due to non-convexity). While a hypothesis space which is easy to search over might be attractive, if the best candidate in said hypothesis space is far from the truth: f , this means little, since the best hypothesis in this space is still far from f . It may be the case that a candidate model taken from a larger hypothesis space, even if it is not guaranteed to be the best model in that space, is much closer to f . This is illustrated graphically in fig. 2 and discussed in more detail in Fariss and Jones (2017).

For illustrative purposes consider a linear additive model as in eq. 2 with a squared loss function. Then the loss function we want to minimize is the sum of the squared differences between the predictions \hat{f} and \mathbf{y} . In this simple case gradient descent suffices to find the parameters which minimize the loss function, though it is not the most efficient method available for this problem.

$$\begin{aligned} h(\beta, \mathbf{x}) &= \sum_j \beta_j x_j \\ \mathcal{L}(\mathbf{y}, h) &= \sum_{i=1}^N \frac{1}{2} (y_i - h)^2 \\ \nabla \mathcal{L} &= \frac{\partial \mathcal{L}}{\partial \beta} = \frac{1}{N} \sum_{i=1}^N -x_i (y_i - h) \end{aligned}$$

Since the gradient $\nabla \mathcal{L}$ gives the direction in which the loss function is changing as a function of β , iteratively picking candidate values of β based on the negative of the gradient converges to the set of values for β which minimize \mathcal{L} . From a starting value β^0 the sequential update for the next step is

$$\beta^{(k+1)} = \beta^{(k)} - \gamma_k \nabla \mathcal{L}(\beta^{(k)})$$

where γ controls the size of the steps along the gradient, which can itself be estimated by gradient descent. The procedure stops when $\nabla \mathcal{L}$ is sufficiently small, that is, when the loss function is no longer decreasing.

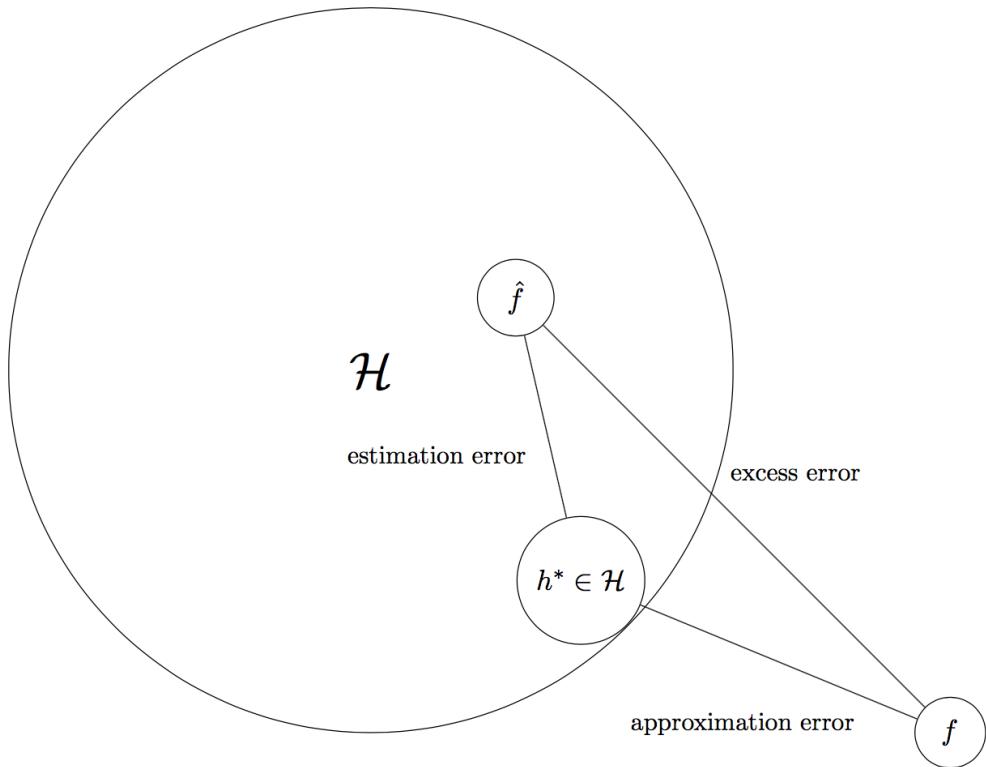


Figure 2: Here the circle represents the hypothesis space \mathcal{H} , f is the true model, which in this case is not contained in the hypothesis space. h^* is the point (the model) in the hypothesis space closest to f , i.e., what could be achieved with an infinite amount of data. The difference between this point and the true model is approximation error that is due to restricting the hypothesis space to \mathcal{H} . Decreasing the “size” of \mathcal{H} increases approximation error, but decreases estimation error, which is error that comes from having to search \mathcal{H} for the best model: h^* , which generally becomes more difficult as \mathcal{H} grows. I am arguing herein that because f is likely very complex in many social systems, approximation error is substantially larger than estimation error.

In contrast consider gradient boosting, a statistical learning method. Like with the aforementioned linear additive model we want to minimize the loss $\mathcal{L}(\mathbf{y}, h) = \frac{1}{2} \sum_{i=1}^N (y_i - h)^2$, however now each candidate h has more parameters. h must be estimated sequentially and is composed of a sequence of models which are combined in the last step. We denote the first m components of h as h_m .

$$h_m(\gamma, \theta, \mathbf{x}) = \sum_{m=1}^M \gamma_m \phi_m(\mathbf{x}, \theta_m) \quad (4)$$

Here ϕ_m is a model itself, such as a regression tree like is shown in fig. 1. For each m the gradient of \mathcal{L} with respect to h_m is

$$\nabla \mathcal{L}(h_m) = \frac{\partial \mathcal{L}}{\partial h_m} = \sum_{i=1}^N (y_i - h_m)$$

and so, like in the case of gradient descent for linear regression

$$h_{m+1} = h_m - \gamma_m \nabla \mathcal{L}(h_m)$$

The scaling parameters γ are estimated by a separate application of gradient descent.⁴

So at each step m boosting using this loss function amounts to fitting $\phi_m(\mathbf{x}, \theta_m)$ to $\mathbf{y} - h_{m-1}$: repeatedly refitting the residuals. This amounts to giving higher weight to observations where the sequence of models thus far has performed poorly, and by combining these models as in eq. 4 nonlinear relationships between \mathbf{x} and \mathbf{y} can be estimated without specifying ϕ s which can estimate nonlinearity (e.g., each could be a linear model, or, more commonly, a regression tree with only one split). It should also be clear that the resulting output h_M , which is a weighted sum of models ϕ_m , which are themselves estimated on residuals which depend on h_{m-1} is not directly interpretable, even if the data generating function it is approximating is. This property is a consequence of statistical learning methods' capacity for automatically finding patterns (e.g., nonlinearity and/or interaction amongst the covariates) that were not prespecified.

Interpreting Models

I argue that there are four main types of model interpretation which are commonly sought. Functional form describes the shape of the relationship between covariates and the model. Magnitude describes how important covariates are in determining the model's predictions. Interaction detection relates to the estimation of the functional form and magnitude of the relationship between groups of covariates and the model. Reliability relates to how different sources of variability could possibly change the estimated effect.

⁴Specifically, $\gamma_m = \arg \min_{\gamma} \mathcal{L}(\mathbf{y}, h_{m-1} + \gamma h_m)$.

Functional Form

Interpreting the functional form of the relationship between the covariates and the model is one of the most common tasks. For many conventional methods like linear models this is often described by a single number which is obtained by taking the partial derivative of the model with respect to said covariate; e.g., for a linear and additive model $\frac{\partial \hat{f}}{\partial \mathbf{x}_j} \sum_j \beta_j \mathbf{x}_j = \beta_j$. For models which are nonlinear in \mathbf{x}_j this derivative is non-constant but does not necessarily depend on \mathbf{x}_{-j} . For example if $\hat{f} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_1^2 + \beta_3 \mathbf{x}_2$ then $\frac{\partial \hat{f}}{\partial \mathbf{x}_1} = \beta_1 + \beta_2 \mathbf{x}_1$ and to interpret the relationship between \mathbf{x}_1 and \hat{f} it is necessary to plot \mathbf{x}_1 and this derivative at a grid of values on \mathbf{x}_1 . In cases where the partial derivative with respect to \mathbf{x}_j depends on \mathbf{x}_{-j} values of \mathbf{x}_j must be chosen as well, even when there is no interaction, as might be the case in a model like logistic regression. With statistical learning methods the estimated function \hat{f} is often a “black box” and so it is not known, necessarily, after estimating a model, whether there are interactions or nonlinearities that have been estimated, and hence any understanding of the relationship between \mathbf{x}_j and \hat{f} depends on the values of \mathbf{x}_j and \mathbf{x}_{-j} chosen to evaluate \hat{f} at.

Magnitude

Magnitude describes how important covariates are in determining a model's predictions. One way to measure importance in the case of a linear model is via the absolute size of the associated regression coefficients. For a nonlinear model this isn't so simple. If the derivative of the model with respect to a covariate is non-constant then there is no longer a single number summary of that covariate's influence on the model's predictions. Yet, this might be what is desired, as in Hill and Jones (2014) where the goal was to rank a large set of potentially relevant covariates in order to focus on ones that were identified as useful predictors by the model.

Interaction Detection

The identification and description of interactions is another common interpretative task (Brambor, Clark, and Golder 2006). In the case of a linear model, again, this is fairly straightforward, though the product terms commonly used to estimate interactions in these models are commonly misused in testing nonlinear hypotheses (Hainmueller, Mummolo, and Xu 2016; Pepinsky 2017). Using statistical learning methods avoids this problem by automatically detecting interactions while making few assumptions about their functional form. This allows researchers to ask these common questions, such as, how the effect of one variable on \hat{f} change as a possibly nonlinear function of another. Interaction detection can be cast as the functional form of the joint effects of covariates or their joint importance.

Reliability

Most of the above, when computed using conventional methods, are or can be accompanied by estimates of reliability designed to quantify variability from sampling from the data generating process. In some cases this can be a useful measure of reliability but in many

cases the assumptions necessary for these reliability estimates to be valid are violated. This is particularly problematic in social data, where dependence is both complex and ubiquitous, so much so that it is often of primary interest, as it is in analyses of social networks. It may also be the case that uncertainty from non-random measurement error is substantially more important than random measurement error (i.e., sampling error) in many cases, and instances where non-random measurement error is identified or where multiple measures of the same (or similar) concepts produce substantially different conclusions constitute anecdotal evidence that this is the case (e.g., Lupu 2017; Baum and Zhukov 2015; Jerven 2013; Hegre and Sambanis 2006). This suggests that many measures of uncertainty might not be as trustworthy as their theoretical best-case properties suggest. Thus I have not focused on discussing reliability measures herein. Despite this, all of the software referenced herein is capable of using estimates of the variability of a models' predictions to produce reliability bounds for the aforementioned types of interpretation tasks which correspond to the quantities that typically accompany the equivalent outputs from conventional methods.⁵

Approximating and Interpreting Functional Form

When the most complex functional form capable of learned from the data is simple, as in a linear additive model, the parameters of said model are often directly interpretable, which, to reiterate, is a consequence of assumed simplicity, which may or may not be appropriate in a given circumstance. Some statistical learning methods have similar properties (See, e.g., Hainmueller and Hazlett 2013; Beck, King, and Zeng 2000). However, many such methods cannot be directly interpreted but may be otherwise desirable for a given application: due to, e.g., their ability to deal with a large number of samples or many covariates, their predictive performance in this or similar applications, etc. One attractive method is the random forest, which is used here to model data from eq. 1 and to predict the reported incidence of burglary using data from the Chicago Police at the police beat-day level (Breiman 2001; Wright and Ziegler 2015)

One of the most common types of interpretation for conventional methods is functional form; i.e., for a linear model what is the sign and magnitude of the coefficient for a covariate, defining the slope of the line representing its relationship with the model's predictions. For conventional models where one or more of the covariates have a nonlinear relationship in the model, e.g., a logistic regression, the effect of a covariate on the model's predictions depends on the other covariates. To understand the effect of a covariate then, it is necessary to compute predictions at particular values of the other covariates. Choosing sample average values for these covariates is common practice, but with models which may have estimated interactions between the covariates, where the empirical distribution of the covariates is not unimodal, or when the distribution of the covariates is dependent, this approach makes it difficult to detect such interaction and may distort the marginal relationships as well. Further analysis of these problems are beyond the scope of this paper, but are elaborated in Jones (2017). One alternative is to compute an average marginal effect, which gives the effect of a particular covariate computed at, rather than fixed values for the other covariates, every observed combination of the other covariates (or a subsample thereof for computational reasons). This is a form of Monte-Carlo integration which also works with statistical learning

⁵See, e.g., Sexton and Laake (2009) or Wager, Hastie, and Efron (2014) for examples of methods for estimating the variance of predictions from random forests.

methods, where it is referred to as partial dependence. It allows the recovery of the partial relationship between particular covariates and \hat{f} regardless of how \hat{f} is estimated (J. H. Friedman 2001).

Partial Dependence

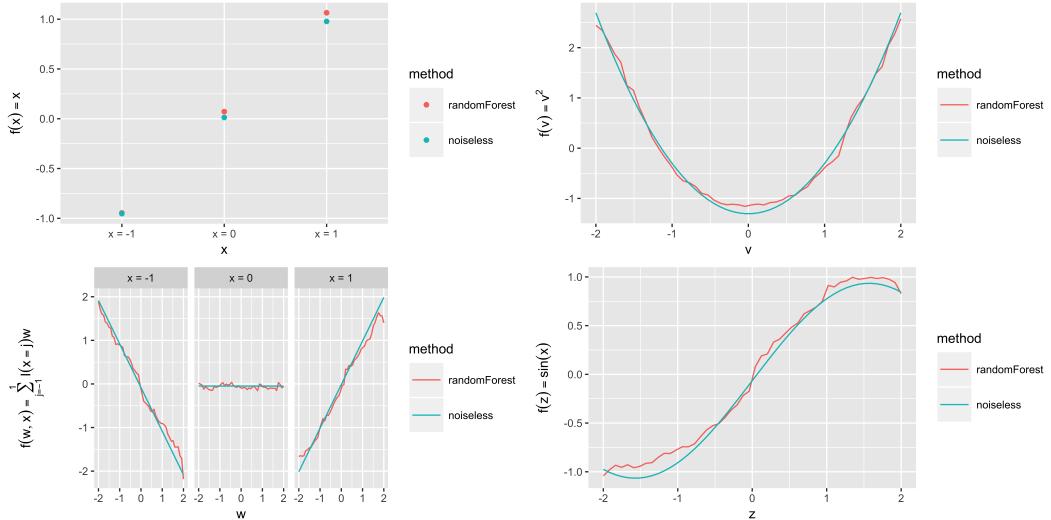


Figure 3: This figure shows the partial dependence: the marginalized prediction function, for each term in eq. 1, constructed from a random forest fit to 5,000 samples and the data generating function if it were known estimated exactly, but was still a “black box” (a “noiseless” estimate of eq. 1). In the top left panel is the partial dependence of the discrete covariate \mathbf{x} , and in the bottom left its interaction with the continuous covariate \mathbf{w} . In the upper and lower right panels the partial dependence of the covariates which have nonlinear effects, \mathbf{v} and \mathbf{w} are shown. While a conventional method like ordinary least squares could estimate eq. 1, the random forest estimated these relationships without prespecification, and with partial dependence we are able to recover the functional form of these relationships accurately.

The *partial dependence* of $\hat{f}(\mathbf{x})$ on \mathbf{x}_u , given in eq. 5 can be estimated from the sample data regardless of how \hat{f} was estimated (J. H. Friedman 2001 Jones (2017)).

$$\mathbb{E}[f(\mathbf{x})] = \int_{\mathbf{x}_{-u}} f(\mathbf{x}) d\mathbb{P}(\mathbf{x}_{-u}) \approx \frac{1}{N} \sum_{i=1}^N \hat{f}(\mathbf{x}_u, \mathbf{x}_{-u}^{(i)}) = \bar{f}_{\mathbf{x}_u}(\mathbf{x}_u) \quad (5)$$

This gives the relationship between the prediction made by $\hat{f}(\mathbf{x})$ and \mathbf{x}_u averaged across the observed values of \mathbf{x}_{-u} . A grid of points on \mathbf{x}_u must still be chosen, and this can be obtained by sampling from the observed values \mathbf{x}_u , by selecting values of interest, or by creating a uniform sequence of values.

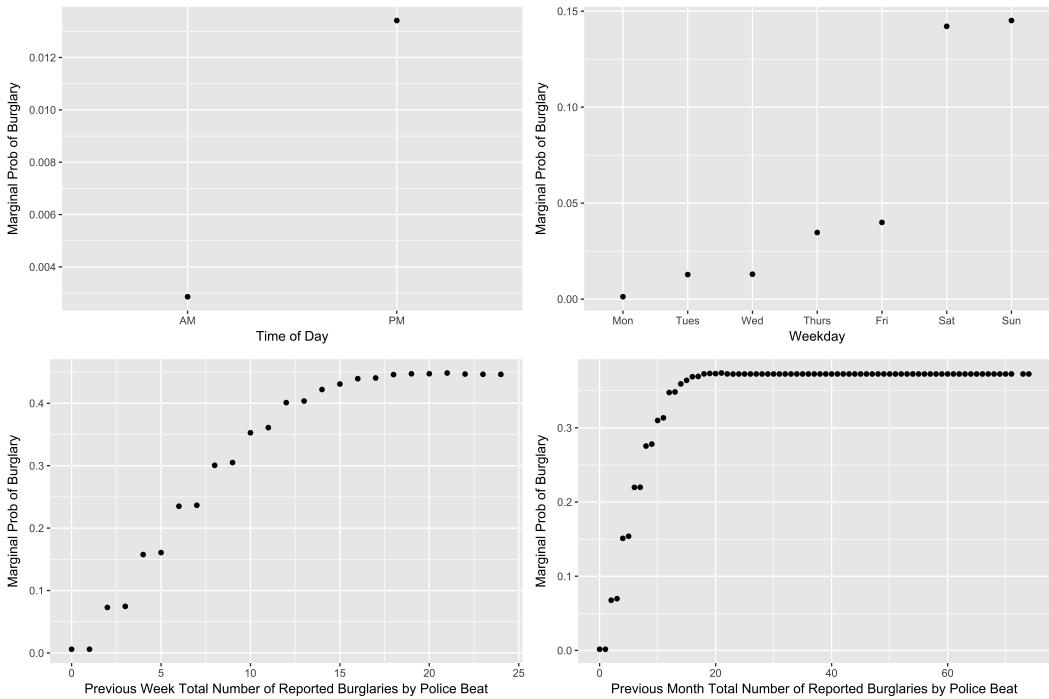


Figure 4: Partial dependence indicates that the marginal probability of a reported burglary in a police beat increases in the evening, on the weekend, and substantially for beats in which a large number of burglaries were reported in previous weeks and months.

How closely $\bar{f}_{\mathbf{x}_u}(\mathbf{x}_u)$ corresponds to $\hat{f}(\mathbf{x}_u)$ depends on the structure of \hat{f} . In the case where $\hat{f}(\mathbf{x})$ is additive or multiplicative in $\{\mathbf{x}_u, \mathbf{x}_{-u}\}$, that is, $\hat{f}(\mathbf{x}) = \bar{f}_{\mathbf{x}_u}(\mathbf{x}_u) + \bar{f}_{\mathbf{x}_{-u}}(\mathbf{x}_{-u})$ or $\hat{f}(\mathbf{x}) = \bar{f}_{\mathbf{x}_u}(\mathbf{x}_u)\bar{f}_{\mathbf{x}_{-u}}(\mathbf{x}_{-u})$ computing the partial dependence gives an approximation to $\hat{f}(\mathbf{x}_u)$ that is accurate to an additive or multiplicative constant, respectively (J. H. Friedman 2001; Jones 2017). Discovering when such (additive or multiplicative) factorizable structure is *not* present is a question about interaction, that is, non-separable dependence between \mathbf{x}_u and \mathbf{x}_{-u} in \hat{f} .

One way of detecting interaction amongst the covariates using partial dependence is to estimate the point-wise variability of eq. 5, as shown in fig. 5. This can be done again using Monte-Carlo methods by substituting the sample variance for the mean, i.e., the variance of \hat{f} for a fixed grid of values on \mathbf{x}_u across the sample values of \mathbf{x}_{-u} (or a subsample thereof). If there is interaction between \mathbf{x}_u and \mathbf{x}_{-u} which has been estimated in \hat{f} the the partial dependence of \mathbf{x}_{-u} will be non-constant, i.e., it will vary across the range of \mathbf{x}_u , and, its variability will in general increase as the strength of the interaction with \mathbf{x}_{-u} increases, which can be seen in fig. 5 with the two covariates x and w which interact in eq. 1.⁶

All of the above can be quickly and easily estimated for any prediction function \hat{f} , regardless of how it was estimated (i.e., both conventional and statistical learning methods) from the sample data using **Monte Carlo Marginalization of Prediction Functions** (**mmpf**) which is a publically available R package. The **Machine Learning in R** (**mlr**) package also makes all of this functionality available, and, additionally, makes available a wide variety of statistical learning methods available for use in a consistent manner, with a wide variety of tools for evaluating and tuning their performance (Bischl et al. 2016; Jones 2017). Lastly, **Exploratory Data Analysis using Random Forests** (**edarf**) again provides similar functionality to what is described above along with additional interpretation methods specific to random forests, which are beyond the scope of this paper (Jones and Linder 2016).

Variable Importance

The question of which and how important covariates are is a question that arises with predictive and exploratory tasks. In case of the former it is often advantageous to remove covariates which are not useful in predicting the outcome and in the latter case computing the importance of covariates is useful in finding strong predictive relationships for further study (Shmueli and others 2010). For example these methods were used by Hill and Jones (2014) to rank covariates according to their importance in predicting different measures of human rights abuses. As previously mentioned in the case where the derivative of the model with respect to a covariate is constant the importance of that covariate in that model can be summarized by this single number, but with models where this derivative is non-constant there is no straightforward way to summarize the importance of a covariate with a single number. Fortunately, there are methods which have generalized the notion of importance to models of an arbitrary form.

Permutation importance is a general method for estimating variable importance which is agnostic to the method used to estimate f . Specifically, it is the expected change in prediction

⁶Another possibility is to avoid Monte Carlo integration over the distribution of \mathbf{x}_{-u} and to instead visualize the expected value of the outcome at \mathbf{x}_u for each observation conditional on \mathbf{x}_{-u} . Goldstein et al. (2015) propose using this approach. **mmpf**, **edarf**, and **mlr** are capable of estimating this as well.

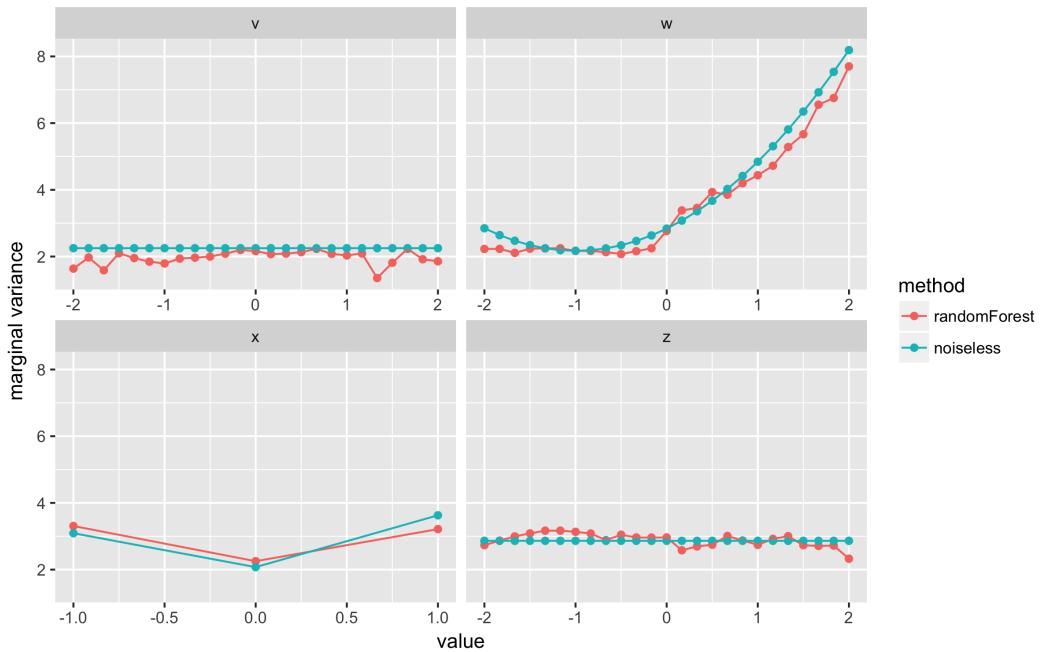


Figure 5: The marginal variance of the covariates indicated in the top of each panel. The covariates which do not interact in eq. 1 have a constant variance across their range, since in these cases changing the value of the indicated covariate does not modify the effects of any others: that is, there is no interaction, which the model has correctly estimated as can be seen in fig. 3. In the case of both w and x , which do interact, it is clear that the variance is non-constant, and that it increases when $x = -1$ or $x = 1$, which are the two areas in which x interacts with w (when $x = 0$ there is no effect).

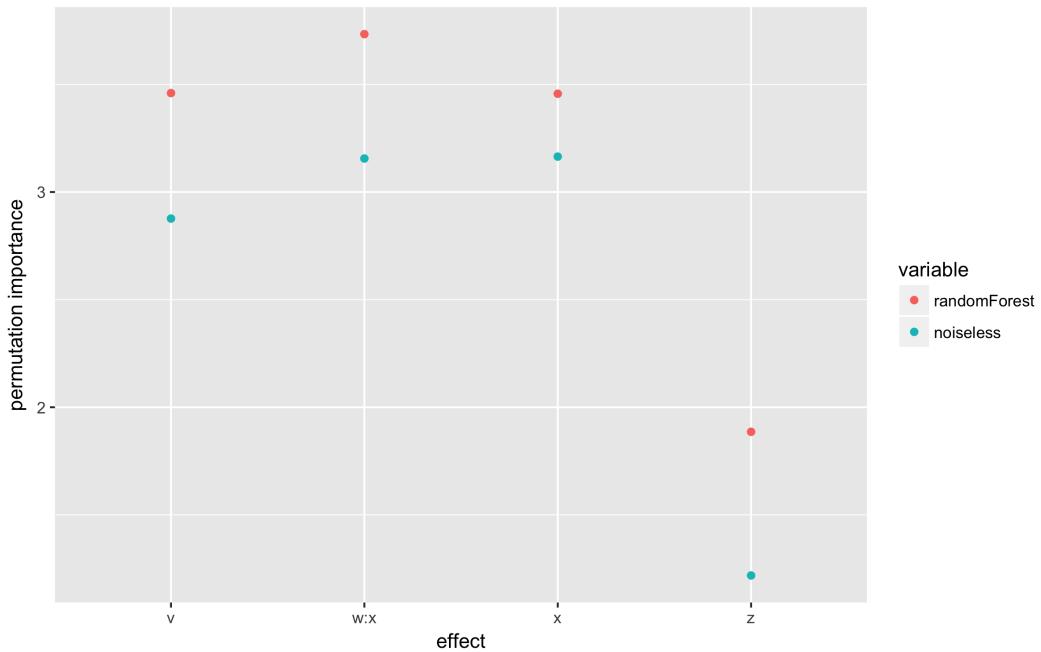


Figure 6: Permutation importance estimated from the data generating function in eq. 1. This indicates that the most important component of the model is the interaction of \mathbf{x} and \mathbf{w} . Since \mathbf{x} and \mathbf{w} interact in eq. 1 and the random forest estimated this interaction, as can be seen in fig. 3, permuting either or both covariates will increase prediction error substantially, as they introduce error from the individual components as well as their interaction. In this way permutation importance gives weight to covariates like \mathbf{w} which may only have a conditional effect. Components with less variability, like $\sin(\mathbf{z})$ (which has a sample variance of 0.592) have less importance than those with more variability (e.g., \mathbf{v}^2 , which has a sample variance of 1.418).

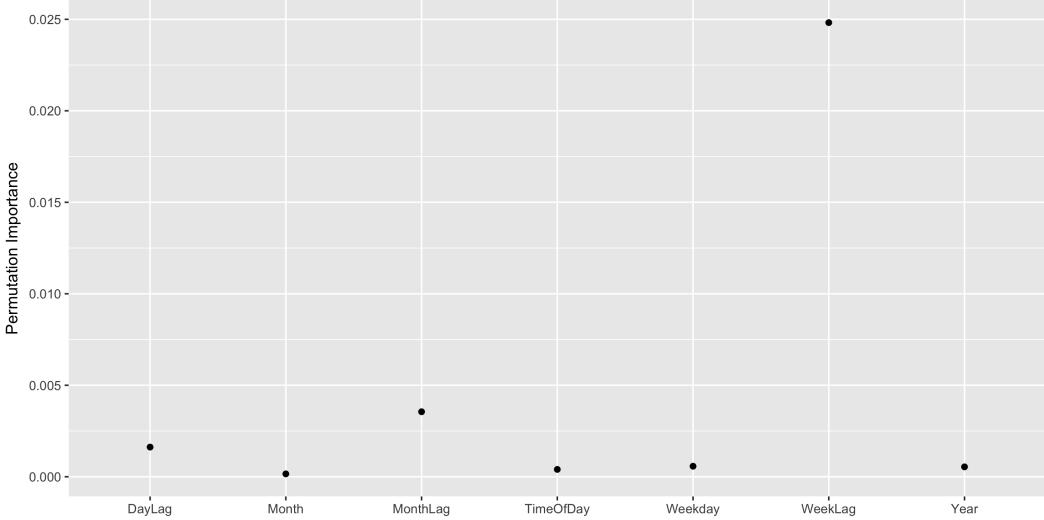


Figure 7: The permutation importance of covariates used to predict the reported incidence of Burglary between 2001 and present in Chicago at the police beat level. The covariates which are most important are those that measure temporal dependence: beats wherein there were a high number of reported burglaries in the previous weeks and months are more likely to have burglaries reported on any given day.

error that occurs when \mathbf{x}_u is permuted. If \mathbf{x}_u was not useful to $\hat{f}(\mathbf{x})$ in making predictions then we would expect no change or a decrease in prediction error, whereas if \mathbf{x}_u is important there should be an increase in prediction error when it is permuted. Due to Monte-Carlo error from the permutation, this is repeated M times to give the expectation of the change in the loss \mathcal{L} from permuting \mathbf{x}_u .

$$I_{\mathbf{x}_u} = \frac{1}{M} \sum_{j=1}^M \mathcal{L} \left(\hat{f}(\mathbf{x}_{u\pi^{(j)}}, \mathbf{x}_{-u}), \hat{f}(\mathbf{x}) \right) \quad (6)$$

Take for example a linear model of the form $\hat{f}(\mathbf{x}) = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2$. Suppose that β_1 is 0. Then if we randomly permuted \mathbf{x}_1 M times, and, for each one of those times we computed \hat{f} using this randomly shuffled version of \mathbf{x}_1 , denoted $\mathbf{x}_{1\pi}$ in eq. 6, we would find that the prediction error did not change at all, because \mathbf{x}_1 does not contribute to the prediction made by $\hat{f}(\mathbf{x})$. Conversely, if we permuted \mathbf{x}_2 , and β_2 was non-zero, perhaps relatively large, we would be multiplying a randomly shuffled covariates by a large number β_2 , and, unsurprisingly, this would make \hat{f} also random, and consequently its predictions would degrade, which would allow us to conclude that \mathbf{x}_2 is important in determining \hat{f} , and that it is more important than \mathbf{x}_1 . Clearly this is unnecessary in this simple case where we could simply inspect the parameters β_1 and β_2 , but with statistical learning methods this is not usually possible.

Comparing the the change in the loss that results from permuting certain covariates to the loss without permuting any covariates can be done in many ways. With categorical (ordered or unordered) outcome variables category-specific or overall (e.g., average agreement across

categories) comparisons can be made, which can be useful in cases where categories are imbalanced or the costs of making prediction errors differs across the categories, e.g., as may be the case when predicting violence, where an absence of violence is more common than violence but violence is often costly when not anticipated (R. Berk et al. 2009). With a continuous outcome the permutation importance can be computed for the mean, i.e., how permuting a covariate affects the conditional expectation function, or any other summary of the distribution of outcomes, such as a quantile. Lastly the permutation importance may not be aggregated at all, giving an observation specific measure of importance. This can, for example, be used to estimate the importance of covariates across the distribution of the outcome (a continuous analogue to category-specific importance).

Permutation importance is not the only method for computing predictive importance with statistical learning methods. There are other generic methods beyond the scope of this paper, as well as method specific measures, such as for Random Forests (Strobl et al. 2007; Strobl et al. 2008; Altmann et al. 2010; Nicodemus et al. 2010; Janitza, Strobl, and Boulesteix 2013; Grömping 2009; Louppe et al. 2013; J. H. Friedman 2001). Permutation importance as described above is implemented in `edarf`, `mmpf`, and `mlr`.

Summary and Future Work

Future empirical work can make use of statistical learning methods which allow us to avoid making unwarranted assumptions and learn about unexpected interactions and nonlinearities. Statistical learning methods will generally estimate models that generalize to new data better and extract more information about the relationships between the covariates and outcomes than do conventional methods as commonly used. Additionally, using statistical learning methods in cases where many conventional models would have been fit and compared on the sample data can help reduce researcher degrees of freedom (i.e., arbitrary but consequential data analysis decisions) that can have substantial effects on conclusions (Gelman and Loken 2013).

Statistical learning methods can be used on their own for data analysis and can also be used in conjunction with conventional methods (See, e.g., Muchlinski et al. 2016). For example the partial dependence of a covariate can be compared to the average marginal effect from a conventional model with a more restricted functional form, and the permutation importance of the covariates could be compared to the size of the coefficients from a conventional method (or their ranks). The expected prediction error could also be compared in this instance; if statistical learning methods do substantially better than conventional methods this would indicate that conventional methods under-fit the data, i.e., the models that they represent are under-theorized.

The methods for interpreting the “black box” functions which are commonly the output of statistical learning methods described herein have all relied on integration. Another class of interpretation methods relies on projecting the estimated function onto a more interpretable set of functions, which is the subject of ongoing research (Hooker 2007; Gaëlle Chastaing and Le Gratiet 2015; Gaelle Chastaing, Gamboa, and Prieur 2015).

References

- Adebayo, Julius, and Lalana Kagal. 2016. “Iterative Orthogonal Feature Projection for Diagnosing Bias in Black-Box Models.” *arXiv Preprint arXiv:1611.04967*.
- Altmann, André, Laura Tolosi, Oliver Sander, and Thomas Lengauer. 2010. “Permutation Importance: A Corrected Feature Importance Measure.” *Bioinformatics* 26 (10). Oxford Univ Press: 1340–7.
- Baum, Matthew A, and Yuri M Zhukov. 2015. “Filtering Revolution: Reporting Bias in International Newspaper Coverage of the Libyan Civil War.” *Journal of Peace Research* 52 (3). Sage Publications Sage UK: London, England: 384–400.
- Beck, Nathaniel, and Simon Jackman. 1998. “Beyond Linearity by Default: Generalized Additive Models.” *American Journal of Political Science* 42. University of Texas Press: 596–627.
- Beck, Nathaniel, Gary King, and Langche Zeng. 2000. “Improving Quantitative Studies of International Conflict: A Conjecture.” *American Political Science Review* 94 (01). Cambridge Univ Press: 21–35.
- Berk, Richard. 2009. “The Role of Race in Forecasts of Violent Crime.” *Race and Social Problems* 1 (4). Springer: 231.
- Berk, Richard A. 2004. *Regression Analysis: A Constructive Critique*. Vol. 11. Sage.
- . 2008. *Statistical Learning from a Regression Perspective*. Springer.
- Berk, Richard, Lawrence Sherman, Geoffrey Barnes, Ellen Kurtz, and Lindsay Ahlman. 2009. “Forecasting Murder Within a Population of Probationers and Parolees: A High Stakes Application of Statistical Learning.” *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 172 (1). Wiley Online Library: 191–211.
- Bischl, Bernd, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M. Jones. 2016. “Mlr: Machine Learning in R.” *Journal of Machine Learning Research* 17 (170): 1–5. <http://jmlr.org/papers/v17/15-066.html>.
- Brambor, Thomas, William Roberts Clark, and Matt Golder. 2006. “Understanding Interaction Models: Improving Empirical Analyses.” *Political Analysis* 14 (1). SPM-PMSAPSA: 63–82.
- Breiman, Leo. 2001. “Random Forests.” *Machine Learning* 45 (1). Springer: 5–32.
- Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and Regression Trees*. CRC press.
- Chastaing, Gaelle, Fabrice Gamboa, and Clémentine Prieur. 2015. “Generalized Sobol Sensitivity Indices for Dependent Variables: Numerical Methods.” *Journal of Statistical Computation and Simulation* 85 (7). Taylor & Francis: 1306–33.
- Chastaing, Gaëlle, and Loic Le Gratiet. 2015. “ANOVA Decomposition of Conditional Gaussian Processes for Sensitivity Analysis with Dependent Inputs.” *Journal of Statistical Computation and Simulation* 85 (11). Taylor & Francis: 2164–86.
- Doshi-Velez, Finale, and Been Kim. 2017. “Towards a Rigorous Science of Interpretable

- Machine Learning.” *arXiv Preprint arXiv:1702.08608*.
- Fariss, Christopher J., and Zachary M. Jones. 2017. “Enhancing Validity in Observational Settings When Replication Is Not Possible.” *Political Science Research and Methods*. Cambridge University Press, 1–16. doi:10.1017/psrm.2017.5.
- Friedman, Jerome H. 2001. “Greedy Function Approximation: A Gradient Boosting Machine.” *Annals of Statistics*. JSTOR, 1189–1232.
- Friedman, Jerome H, and Bogdan E Popescu. 2008. “Predictive Learning via Rule Ensembles.” *The Annals of Applied Statistics*. JSTOR, 916–54.
- Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. Vol. 1. Springer series in statistics Springer, Berlin.
- Gelman, Andrew, and Eric Loken. 2013. “The Garden of Forking Paths: Why Multiple Comparisons Can Be a Problem, Even When There Is No ‘Fishing Expedition’ or ‘P-Hacking’ and the Research Hypothesis Was Posited Ahead of Time.” *Department of Statistics, Columbia University*.
- Goldstein, Alex, Adam Kapelner, Justin Bleich, and Emil Pitkin. 2015. “Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation.” *Journal of Computational and Graphical Statistics* 24 (1). Taylor & Francis: 44–65.
- Goldstone, Jack A., Robert H. Bates, David L. Epstein, Ted Robert Gurr, Michael B. Lustik, Monty G. Marshall, Jay Ulfelder, and Mark Woodward. 2010. “A Global Model for Forecasting Political Instability.” *American Journal of Political Science* 54 (1). Wiley Online Library: 190–208.
- Green, Donald P, and Holger L Kern. 2012. “Modeling Heterogeneous Treatment Effects in Survey Experiments with Bayesian Additive Regression Trees.” *Public Opinion Quarterly*. AAPOR, nfs036.
- Grimmer, Justin, and Brandon M Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political Analysis*. JSTOR, 267–97.
- Grömping, Ulrike. 2009. “Variable Importance Assessment in Regression: Linear Regression Versus Random Forest.” *The American Statistician* 63 (4).
- Hainmueller, Jens, and Chad Hazlett. 2013. “Kernel Regularized Least Squares: Reducing Misspecification Bias with a Flexible and Interpretable Machine Learning Approach.” *Political Analysis*. SPM-PMSAPSA, mpt019.
- Hainmueller, Jens, Jonathan Mummolo, and Yiqing Xu. 2016. “How Much Should We Trust Estimates from Multiplicative Interaction Models? Simple Tools to Improve Empirical Practice.”
- Hegre, Håvard, and Nicholas Sambanis. 2006. “Sensitivity Analysis of Empirical Results on Civil War Onset.” *Journal of Conflict Resolution* 50 (4). Sage Publications: 508–35.
- Hill, Daniel W, and Zachary M Jones. 2014. “An Empirical Evaluation of Explanations for State Repression.” *American Political Science Review* 108 (03). Cambridge Univ Press:

661–87.

- Hooker, Giles. 2004. “Discovering Additive Structure in Black Box Functions.” In *Proceedings of the Tenth Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 575–80. ACM.
- . 2007. “Generalized Functional Anova Diagnostics for High-Dimensional Functions of Dependent Variables.” *Journal of Computational and Graphical Statistics* 16 (3).
- Janitza, Silke, Carolin Strobl, and Anne-Laure Boulesteix. 2013. “An Auc-Based Permutation Variable Importance Measure for Random Forests.” *BMC Bioinformatics* 14 (1). BioMed Central Ltd: 119.
- Jerven, Morten. 2013. *Poor Numbers: How We Are Misled by African Development Statistics and What to Do About It*. Cornell University Press.
- Jones, Zachary M. 2017. “Mmpf: Monte-Carlo Methods for Prediction Functions.”
- Jones, Zachary M., and Fridolin J. Linder. 2016. “Edarf: Exploratory Data Analysis Using Random Forests.” *The Journal of Open Source Software* 1 (6). The Open Journal. doi:10.21105/joss.00092.
- King, Gary. 1989. *Unifying Political Methodology: The Likelihood Theory of Statistical Inference*. University of Michigan Press.
- Louppe, Gilles, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. 2013. “Understanding Variable Importances in Forests of Randomized Trees.” In *Advances in Neural Information Processing Systems*, 431–39.
- Lum, Kristian, and William Isaac. 2016. “To Predict and Serve?” *Significance* 13 (5). Wiley Online Library: 14–19.
- Lupu, Jones, Yonatan. 2017. “Is There More Violence in the Middle?”
- Miller, John H, and Scott E Page. 2009. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton university press.
- Montgomery, Jacob M, and Santiago Olivella. 2015. “Tree-Based Models for Political Science Data.” Washington University working paper. Available at q-aps. princeton.edu/sites/default/files/q-aps/files/olivella-qaps.pdf.
- Montgomery, Jacob M, Santiago Olivella, Joshua D Potter, and Brian F Crisp. 2015. “An Informed Forensics Approach to Detecting Vote Irregularities.” *Political Analysis*. SPM-PMSAPSA, mpv023.
- Muchlinski, David, David Siroky, Jingrui He, and Matthew Kocher. 2016. “Comparing Random Forest with Logistic Regression for Predicting Class-Imbalanced Civil War Onset Data.” *Political Analysis* 24 (1). Cambridge University Press: 87–103.
- Nicodemus, Kristin K, James D Malley, Carolin Strobl, and Andreas Ziegler. 2010. “The Behaviour of Random Forest Permutation-Based Variable Importance Measures Under Predictor Correlation.” *BMC Bioinformatics* 11 (1). BioMed Central Ltd: 110.
- Pepinsky, Thomas B. 2017. “Visual Heuristics for Marginal Effects Plots.”
- Schrodt, Philip A. 2014. “Seven Deadly Sins of Contemporary Quantitative Political Analysis.”

- Journal of Peace Research* 51 (2). SAGE Publications: 287–300.
- Sexton, Joseph, and Petter Laake. 2009. “Standard Errors for Bagged and Random Forest Estimators.” *Computational Statistics & Data Analysis* 53 (3). Elsevier: 801–11.
- Shmueli, Galit, and others. 2010. “To Explain or to Predict?” *Statistical Science* 25 (3). Institute of Mathematical Statistics: 289–310.
- Strobl, Carolin, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. 2008. “Conditional Variable Importance for Random Forests.” *BMC Bioinformatics* 9 (1). BioMed Central Ltd: 307.
- Strobl, Carolin, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. 2007. “Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution.” *BMC Bioinformatics* 8 (1). BioMed Central Ltd: 25.
- Vapnik, Vladimir Naumovich. 1998. *Statistical Learning Theory*. 2nd ed. Wiley New York.
- Vapnik, Vladimir, Esther Levin, and Yann Le Cun. 1994. “Measuring the Vc-Dimension of a Learning Machine.” *Neural Computation* 6 (5): 851–76.
- Wager, Stefan, Trevor Hastie, and Bradley Efron. 2014. “Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife.” *Journal of Machine Learning Research* 15 (1): 1625–51.
- Ward, Michael D, Brian D Greenhill, and Kristin M Bakke. 2010. “The Perils of Policy by P-Value: Predicting Civil Conflicts.” *Journal of Peace Research* 47 (4). Sage Publications: 363–75.
- Wright, Marvin N, and Andreas Ziegler. 2015. “Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R.” *arXiv Preprint arXiv:1508.04409*.