Improving Variable Importance Measures*

Zenobia Chan† Marc Ratkovic‡

July 14, 2020

Abstract

Boosting and random forests are among the best off-the-shelf prediction tools. These methods offer a variable importance measure (VIM), which is a cumulative measure of the improvement in accuracy over the algorithm. We show existing variable importance measures, as implemented, are biased, returning positive scores on irrelevant variables. Intuitively, if a variable is irrelevant but correlates with a relevant variable, this correlation may lead to an improvement in performance may be misattributed to the irrelevant variable. We introduce a method that removes this bias. The method works by separating each predictor into a component explained by other predictors (a "predicted variable"), and a component not (a "partialed out variable"). We assess variable importance only through any improvement attributable to the latter. We prove the method returns a valid VIM, meaning it is mean-zero and asymptotically normal for irrelevant variables. Simulation evidence and applications to UCI data suggest the method also performs favorably relative to several existing machine learning methods in terms of predictive accuracy.

Key Words: variable importance measures, boosting, machine learning

Preliminary Draft: Please do not cite or circulate without permission of the authors.

*Prepared for the Annual Meating of the Society of Political Methodology, 2020.

†Graduate Student, Department of Politics, Princeton University, Princeton NJ 08544. Email: zchan@princeton.edu

‡Assistant Professor, Department of Politics, Princeton University, Princeton NJ 08544. Phone: 608-658-9665, Email: ratkovic@princeton.edu, URL: http://scholar.princeton.edu/ratkovic

1 Introduction

Random forests and boosting are now commonly recognized as two of the best off-the-shelf predictive methods, and they have found widespread use. At the same time, it is notoriously difficult to discern which predictors, or combinations thereof, are driving the model's performance. The most common variable importance measure (VIM) is a measure of how much a split taken on a particular value increases fit, where at each step the model's performance is assessed with and without the selected covariate. Any gain in predictive performance is attributed to that predictor. These measures are now included and reported in popular existing software.

We refer to a VIM that is asymptotically mean-zero when the variable is irrelevant as valid. Existing VIMs, though, are not valid as they can lead to positive scores on irrelevant variables. To see why, every predictor can be split into a component that can be predicted by the other covariates (which we call the *predicted variable*) and a component that cannot (which we call the *partialed out variable*). Adding a predictor improves importance through both its predicted and its partialed out components, but if the increase in performance is actually attributable to another predictor, the observed improvement is actually a mirage.

We introduce a boosting method that returns a valid variable importance measure, while performing comparably to existing methods. Our method involves pre-processing the data by estimating a partialed out and predicted version for each variable, entering both as variables into an algorithm, and then only measuring the improvement attributable to the partialed out component. Importantly, we are not developing a new, more complex variant of a boosting or forest algorithm. We are proposing instead a framework for thinking about variable

importance in a manner that can leave any machine learning method more transparent.

We implement a boosting algorithm that takes as inputs the partialed out and predicted variables, permuting only the partialed out variables to assess variable importance. We then implement a boosting method using these two sets of covariates, which generates a variable importance measure from only permuting the partialed out variables. We present theoretical and simulation evidence that this leads to a valid variable importance measure, in that the limiting distribution of the change between using the permuted and the raw variables is mean-zero. We then present theoretical and applied evidence that the method can achieve a competitive MSE relative to existing random forest and boosting algorithms.

2 Previous Literature

The idea of pre-processing covariates to allow for valid inference has several antecedents in the literature. In terms of existing work, we have found many of our ideas foreshadowed by Freedman and Lane (1983) in the context of multivariate linear regression, who generate what we refer to as "partialed out" covariates for sequential testing. Barber and Candès (2015, 2019) extend these ideas, including what we term "predicted" covariates and they term "knock-offs" as additional controls in a high-dimensional regression, and then comparing the difference in estimated coefficients on the original and knockoff covariates.

With the explosion of interest in machine learning methods in recent decades, introducing valid variable importance measures has remained an important field of study. Most recently, Scornet (2020) shows that VIMs for classification forests are valid if the predictors are all independent. Ishwaran and Lu (2019) provide bootstrap variance estimates for permutation-based VIMs, while Strobl et al. (2008) provide similar results but are limited to categorical

predictors, and Janitza, Celik and Boulesteix (2018-12) use only negative estimates of VIMs to infer their null distribution.

For important foundational work on boosting as a functional gradient descent method, see Friedman, Hastie and Tibshirani (2000); Friedman (2001); Buhlmann and Yu (2003); Mason (2000); Breiman (1998); Zhang and Yu (2005); Mason (2000).

3 The Method

We next introduce our notation and present our algorithm.

3.1 Notation and Setup

We consider the situation with outcome variable y, feature vector $\mathbf{x} \sim F_{\mathbf{x}}$, and the researcher is interested in a target function $g(\mathbf{x})$ for predicting y. This function is a known minimizer of the risk,

$$R_g(y, \mathbf{x}) = \mathbb{E}\left(m_g(y, \mathbf{x})\right) = \int_{\mathcal{X}} m_g(y, \mathbf{x})\, dF_{\mathbf{x}}.$$

with $m_g(y, \mathbf{x})$ a loss function. In this current work, we focus on the regression setting, In this version of our work, we consider the regression setting where $y \in \Re$ and $m_g(y, \mathbf{x}) = \frac{1}{2}(y - g(\mathbf{x}))^2$ and $g(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x})$, the population minimizer.

We denote the observed data $\{y_i, \mathbf{x}_i\}_{i=1}^n$, $\mathbf{y}$ the $n \times 1$ vector of observed outcomes, and $\mathbf{X}$ is the $n \times p$ matrix of predictors. $\widehat{g}$ is an estimate that minimizes the risk, which we will subscript with n when necessary to signal that it is converging in sample size. $R_{g,n}(y_i, \mathbf{x}_i)$ and $R_{\widehat{g},n}(y_i, \mathbf{x}_i)$ will denote the empirical risk evaluated at the population and estimated g.

We decompose each predictor, x_j into two components: one that is explained by other predictors and one that is not,

$$x_j = x_j^{\Pi} + x_j^{\perp}$$
$$x_j^{\Pi} = \mathbb{E}(x_j \mid \mathbf{x}_{-j}); \;\; x_j^{\perp} = x_j - x_j^{\Pi}.$$

We will refer to x_j^{Π} as the "projected value" for x_j and $x_j^{\perp}$ as the "partialed-out value" of x_j. In order to return a valid VIM, we are going to permute only $x_j^{\perp}$, rather than the full x_j.

In order to avoid overfitting, we are going to split the data into two subsamples, $\mathcal{S}_0$ and $\mathcal{S}_1$ of size $n_0 + n_1 = n$, each with half the data. Including $\mathcal{S}_0$ or $\mathcal{S}_1$ to a subscript will denote that the data come from that subsample.

3.2 Variable Importance Measures

We will denote as x_j the j^{th} variable and $\mathbf{x}_{i,-j}$ the vector $\mathbf{x}_i$ less its j^{th} element. The population variable importance measure (VIM) of variable j is the decrease in risk attributable to adding x_j to the predictor set,

$$\text{VIM}_j^o = R_g(y, \mathbf{x}_{-j}) - R_g(y, \mathbf{x}) \geq 0.$$

We will say that a variable is irrelevant if its addition or deletion from the control set does not reduce the risk, $\text{VIM}_j^o = 0$. We will denote the sample version of this variable important

measure as

$$\widehat{\mathrm{VIM}}^{o}_{j,n} = R_{\widehat{g}_{-j},n}(y_i, \mathbf{x}_{i,-j}) - R_{\widehat{g},n}(y_i, \mathbf{x}_i).$$

which is the algorithm fit twice, once with and without variable j, with $\widehat{g}_{-j}$ the estimate using all variables but x_j.

We compare this to another measure, where the decrease in error is evaluated successively at each step of the algorithm. This is the variable importance measure returned at default by random forests, where the decrease in mean squared error attributable to splitting on a variable in a tree is aggregated over predictors.

We denote as $R_{\widehat{g},j,n}(y_i, \mathbf{x}_i)$ the estimated empirical loss, except at every step x_j is used, it is instead not. This gives us a variable importance measure

$$\widehat{\mathrm{VIM}}_{j,n} = R_{\widehat{g},j,n}(y_i, \mathbf{x}_i) - R_{\widehat{g},n}(y_i, \mathbf{x}_i)$$

Our third variable importance measure comes from fitting the model to $(\mathbf{x}^{\perp}, \mathbf{x}^{\Pi})$ and only evaluating the contribution attributable to the partialed out variable,

$$\widehat{\mathrm{VIM}}^{\perp}_{j,n} = R_{\widehat{g},j^{\perp},n}(y_i, \mathbf{x}^{\perp}_{i,j}, \mathbf{x}^{\Pi}_{i,j}) - R_{\widehat{g},n}(y_i, \mathbf{x}_i)$$

This is the statistic we show below is valid.

4 Theoretical Results

We first present our assumptions, then our main results.

4.1 Class of Algorithms

We consider here iterative methods which can be written recursively as an update along predictor j at iteration t as

$$\widehat{g}^{(t+1)}(\mathbf{x}) = \widehat{g}^{(t)}(\mathbf{x}) + \widehat{\Delta}^{(t)}_{j^{(t)}}(\mathbf{x}) \tag{1}$$

or, equivalently, at some step T,

$$\widehat{g}^{(T)}(\mathbf{x}) = \widehat{g}^{(0)}(\mathbf{x}) + \sum_{t=1}^{T} \widehat{\Delta}^{(t)}_{j^{(t)}}(\mathbf{x}) \tag{2}$$

This method clearly includes boosting methods that implement additive functional gradient descent.[1]

They also include any tree-based methods, where a split is considered constructed from variable $j^{(t)}$ on a leaf given by $\mathbf{x}$. As such, it also includes random forests. It also includes least squares or maximum likelihood methods, with updates done coordinatewise via Newton-Raphson or iteratively weighted least squares steps, as well as any estimate that can be expressd as some version of functional coordinate descent (e.g., LASSO).

4.2 Assumptions and Discussion

We make the following assumptions.

[1] Our current theoretical work is in working to characterize the continuous-time process,

$$\widehat{g}^{(T)} = \widehat{g}^{(0)} + \int_0^T \widehat{\Delta}^{(t)} dt, \tag{3}$$

but we restrict our attention to the additive sum at the moment.

Assumption 1 *Assumptions on the Model.*

1. *The data* $\{y_i, \mathbf{x}_i\}_{i=1}^n$ *are a representative sample.*

2. *The loss function is the squared loss* $R_g = \frac{1}{2}\mathbb{E}\left\{(y_i - g(\mathbf{x}_i))^2\right\}$.

3. g *is Hadamard differentiable.*

4. *The predicted values are generated by a fixed sequence of linear maps constructed from* $\mathbf{x}_i$ *such that* $\Pi_{i,n} : y_i \mapsto \widehat{g}(\mathbf{x}_i)$*, with dimensionality* $\dim(\Pi_n) = k_n$ *and* $k_n = o_p\left(1/n\right)$.

5. *The error terms converge as* $\mathbb{E}\left\{(y_i - g(\mathbf{x}_i))^4\right\} \leq \infty$

6. $\widehat{g}$ *converges uniformly to* g.

7. *In the limit,* $x_j^{\perp} \perp\!\!\!\perp x_j^{\Pi}$ *for all predictors.*

8. $\widehat{\Delta}_{j^{(t)}}^{(t)}(\mathbf{x})$ *is constructed from variable* j *and selected to maximize the correlation with the current residuals.*

9. *Denote as* $h = \sup_{j,\mathbf{x},t}\left|\widehat{\Delta}_{j^{(t)}}^{(t)}(\mathbf{x})\right|$. *Then,*$\sqrt{n}T_n h^2 = o_p(1)$.

The first assumption guarantees that our sample is not truncated or censored, such that moments on the sample are unbiased for population moments of interest. The second assumption will allow us a Taylor expansion of m_g in g.[2]

Our boosting algorithm involves aggregating over a set of gradient descent steps. In order to guarantee consistent, asymptotic normality at our estimate, we require that each step converge to a normal random variable. This is guaranteed by the third assumption, as

[2]We are currently working to generalize this beyond squared loss, but do not report this at the moment.

Hadamard differentiability ensures the gradients exist and are uniformly bounded, and by the next assumption, so that the sampling distribution of each estimate concentrates in a sufficiently small space. The fourth and fifth assumption guarantees asymptotic normality of our boosting estimate, and the sixth that it converges on the true population value. The next assumptions requires that our predicted and partialed-out covariates be independent, in the limit.

The final two assumptions deal with the boosting algorithm. The first requires that steps are taken to decrease the risk, and the second serves to control the model complexity.

4.3 Statement of Results

We next move onto our theoretical results. All proofs are in Appendix A, though we summarize the intuition after each statement.

Our first result drives the remainder, that our fitted values are consistent and converge on normal distribution. We use a linear operator such that $\Pi_i(\mathbf{X})\mathbf{y} = \widehat{g}(\mathbf{x}_i)$. Under the assumptions above, we have pointwise convergence,

PROPOSITION 1 *The fitted values converge as*

$$\sqrt{n}(\widehat{g}(\mathbf{x}_i) - g(\mathbf{x}_i)) \rightsquigarrow \mathcal{N}\left(0, \Pi_i(\mathbf{X})\text{Var}(\mathbf{y} \mid \mathbf{X})\Pi_i(\mathbf{X})^\top\right)$$

The result follows from $\widehat{g} \xrightarrow{\text{u}} g$ and the fixed projection matrix with dimensionality tending to zero at rate $1/n$. Note below that our split-sample approach will allow us to treat the projection matrices as fixed, since we learn them on different data than on which we conduct inference.

Our next proposition establishes that fitting the model with and without a variable, and comparing the loss under the two, leads to a valid variable importance measure:

PROPOSITION 2 *Adding an irrelevant variable x_j affects the risk as*

$$\sqrt{n}\,\widehat{VIM}^{o}_{j,n} \xrightarrow{\text{p}} 0$$

The reason for this is reasonably straightforward, as the difference between the two loss functions is $o_p(1/n)$, as a scaled χ^2 distribution on the increase of dimensionality of the model by adding variable x_j.

Our third proposition is that an irrelevant variable may appear relevant, if the statistic is found by aggregating over the course of the algorithm:

PROPOSITION 3 *For an irrelevant variable, x_j,*

$$\sqrt{n}\,\widehat{VIM}_{j,n} \rightsquigarrow \mathcal{N}(\delta, \sigma^2)$$

with some $\delta \geq 0$.

This bias, stated most simply, arises because the an irrelevant variable may correlate with a relevant one, and the improvement due to the latter will be attributed to the former.

Our fourth proposition is that conducting this process using the partialed out variable will return a valid VIM.

PROPOSITION 4 *For a partialed out variable,* $x_j^{\perp}$,

$$\sqrt{n}\,\widehat{VIM}_{j,n}^{\perp} \rightsquigarrow \mathcal{N}(0, \sigma^2).$$

Lastly, we offer some simulation evidence that constructing the fit from $(\mathbf{x}^{\perp}, \mathbf{x}^{\Pi})$ instead of $\mathbf{x}$ can help generate a lower mean-squared error. As this is still ongoing work, we do not yet call it a proposition, but note it as a point for discussion.

5 The Algorithm

We implement a gradient boosting method. Gradient boosting is a form of functional gradient descent, where preliminary fitted values are updated with an estimate in a direction that will decrease the risk.

The method successively updates the estimate $\widehat{g}$ in a negative gradient direction of the empirical risk. The next estimate $\widehat{g}^{(t+1)}(\mathbf{x})$ is constructed from the current one, $\widehat{g}^{(t)}(\mathbf{x})$ but updated by step-size $\lambda^{(t)}$ in the direction of path $h^{(t)}$, a direction that will lower the empirical risk, gradient $\nabla_g^{(t)}$

$$\widehat{g}^{(t+1)}(\mathbf{x}) = \widehat{g}^{(t)}(\mathbf{x}) + \lambda^{(t)} h^{(t)}(\mathbf{x})$$

where we fix the step-size $\lambda^{(t)}$ at some small value, $0.001 \times \sqrt{\text{Var}(y_i)}$.

In this structure, we see the major questions we address here. First, how do we construct a direction to move in, $h^{(t)}$? Second, how do we select T, the total number of steps, and $\lambda_{(t)}$, the step size? Third, how can we estimate which variables are driving the model?

In answering these questions, we rely on two strategies. First, we implement a cross-fitting strategy, such that we select a direction $h^{(t)}$ by fitting a tree to half the training data, and then regress the current residuals on this tree structure on the other half of the training data. We then switch the roles of the two halves, generating two sets of predicted values, which are averaged to get the final predicted values. As we show below, doing so allows us to establish a normal limiting distribution for the fitted values and to generate a stopping rule.

The second strategy we employ is new. We decompose each predictor, x_j into two components: one that is explained by other predictors and one that is not,

$$x_j = x_j^{\Pi} + x_j^{\perp}$$
$$x_j^{\Pi} = \mathbb{E}(x_j \mid \mathbf{x}_{-j}); \;\; x_j^{\perp} = x_j - x_j^{\Pi}.$$

We will refer to x_j^{Π} as the "projected value" for x_j and $x_j^{\perp}$ as the "partialed-out value" of x_j.

Doing so allows us two advantages. The first comes in terms of the risk itself. Rather than construct a single tree and move the fitted values in that direction, we fit two trees, one from x_j^{Π} and $x_j^{\perp}$. This generates an estimate with a lower variance than splitting simply on x_j, as we discuss below. Second, splitting x_j in this way allows us to estimate the importance of variable j by permuting only the part of the variable independent of other covariates ($x_j^{\perp}$). We prove below that not splitting the variables in this fashion can lead to a biased variable importance measure, while constructing a VIM from the partialed out predictors removes this bias.

We move on next to a high-level discussion of our algorithm, then give our theoretical

results.

5.1 Step 1: Estimating a Path

We split the data into two subsamples: an auxiliary split, $\mathcal{S}_0$, with which we construct a tree, and then we turn to our estimation split, $\mathcal{S}_1$, in order to project the current residuals onto the structure learned in the auxiliary sample. Splitting the process in this way leaves the tree structures fixed with respect to the outcome, greatly easing the problem of characterizing the estimates limiting distributions (van der Vaart, 1998).

In our auxiliary sample, we learn two trees. In the first, we use only partialed out values, with the number of splits selected via cross-validation. We denote this tree structure as a set of indicator variables for each terminal node, $Z^{(t),\perp}$.

In order to choose efficient gradient descent steps, we want the steps to be in the direction of two independent trees. In order to approximate this, we regress each covariate x_j^{Π} on $Z^{(t),\perp}$, and construct a second tree from these partialed out variables $x_j^{\Pi,(t)}$. This gives us a tree that can be represented as a matrix of indicators, $Z^{(t),\Pi}$.

5.2 Step 2: The Update

We then evaluate the two matrices, $Z^{(t),\perp}$ and $Z^{(t),\Pi}$, on the estimation split. Then, using the data on this split, we regress the current residuals on these matrices, and update our estimate $\widehat{g}^{(t+1)}$.

Formally, denote as

$$Z^{(t),\mathcal{S}_1} = [Z^{(t),\perp};\ Z^{(t),\Pi}]$$

the design matrices of the trees evaluated on split $\mathcal{S}_1$.

We then construct the hat matrix

$$H^{(t)} = Z^{(t),\mathcal{S}_1} \left\{ Z^{(t),\mathcal{S}_1,\top} Z^{(t),\mathcal{S}_1)} \right\}^{-} Z^{(t),\mathcal{S}_1,\top}$$

and we can update as

$$\widehat{g}^{(t+1)}(\mathbf{X}_{i\in\mathcal{S}_1}) = \widehat{g}^{(t)}(\mathbf{X}_{i\in\mathcal{S}_1}) + \lambda H^{(t)} \nabla_{\widehat{g}}(\mathbf{X}_{i\in\mathcal{S}_1})$$

5.3 Step 3: The Stopping Rule

Given preliminary estimate $\widehat{g}_{\mathcal{S}_1}$ evaluate on our estimation split, we can write our fitted values at step T as a progression of successive, small parametric updates,

$$\begin{aligned} \widehat{g}^{(T)}_{\mathcal{S}_1}(\mathbf{X}_{i\in\mathcal{S}_1}) &= \left\{ I_{n_1} - \prod_{t=1}^{T} \left(I_{n_1} - \lambda H^{(t)} \right) \right\} y_{\mathcal{S}_1} \\ &\doteq H^T_{\mathcal{S}_1} y_{\mathcal{S}_1}. \end{aligned}$$

Importantly, the particular covariates selected in each step (t) are conditionally independent of our estimate $\widehat{g}_{\mathcal{S}_1}$, since the tree structure was learned on split $\mathcal{S}_0$. This allows us to treat our model on split $\mathcal{S}_1$ as a set of small regressions with a set of fixed covariates.

This gives us several advantages. First, we can estimate the model degrees of freedom,

$$\widehat{df}^{T}_{\mathcal{S}_1} = \mathbf{tr}(H^T_{\mathcal{S}_1})$$

which can be used as a stopping rule. We take a generalized cross validation (GCV) statistic,

$$\widehat{T}^{GCV}_{\mathcal{S}_1} = \underset{t}{\text{argmin}} \frac{1}{n_1} \frac{\left\| y_{\mathcal{S}_1} - H^t_{\mathcal{S}_1} y_{\mathcal{S}_1} \right\|^2}{\left(1 - \frac{\widehat{df}^t_{\mathcal{S}_1}}{n_1}\right)^2}.$$

The GCV criterion allows for consistent estimation of g (e.g. Shao, 1997). To construct our final estimate, we "cross-fit," selecting the minimal GCV estimates in $\mathcal{S}_0$ and in $\mathcal{S}_1$, and average the two sets of predictions. We provide evidence of the efficacy of this stopping rule below in several datasets.

5.4 A Valid Variable Importance Measure

To return a variable importance measure, we keep track at each step of the change in squared error attributable to predictor $x_j^{\perp}$ in set $\mathcal{S}_1$, when the model at each step has been selected on set $\mathcal{S}_0$. We return this as our importance measure.

6 Applications

Our first step is to establish that our method has prediction performance comparable to existing methods. Instead of simulated data, we chose *real-world* datasets to assess the method's performance in prediction. These datasets were chosen because they consist of real, continuous outcome variables suitable for regression tasks and have a relatively small number of observations. The characteristics of the datasets are summarized in Table 1.

We compare the proposed method to four other commonly used models, namely BART, generalized random forest (GRF), randomForest, and mboost to four different datasets, three from the UCI Machine Learning Repository and one from Foster and Chan (2019).

Source	Name	Data Type	Default Task	# Obs	# Features
Brooks, Pope, and Marcolini 2014*	Airfoil Self-Noise	Multivariate	Regression	1,503	6
Cortez and Morais 2007*	Forest Fires	Multivariate	Regression	517	13
Foster and Chan 2019	Estonian Ideology	Multivariate	Regression	409	51
Yeh 2007*	Concrete Compressive Strength	Multivariate	Regression	1,030	9

*UCI Machine Learning Repository dataset

Table 1: Evaluation Datasets.

6.1 Performance on UCI Datasets

For each of the four datasets, we randomly split them into halves as the training and testing sets. We then use the same training set to estimates five different models. The out-of-bag (OOB) prediction mean squared-error (MSE) was estimated using the held-out testing set. We normalized the OOB MSE using the variance of the outcome variable in the testing set. Figure 4 presents the results and shows that our method outperforms the other four in two of the four datasets, with a comparable performance in the remainder. The proposed method performs well in terms of predictive accuracy.

We also illustrate the accuracy of our GCV statistic in finding the best number of boosting steps. Figure 2 compares the GCV statistic to predictive performance on the held-out sample. Note that there are two GCV curves, one for each subsample, and we average the two to get our final prediction. If the GCV minimizes at near the OOB MSE minimizer, it has done a good job finding the minimum. We find it does a reasonable job of estimating a stopping

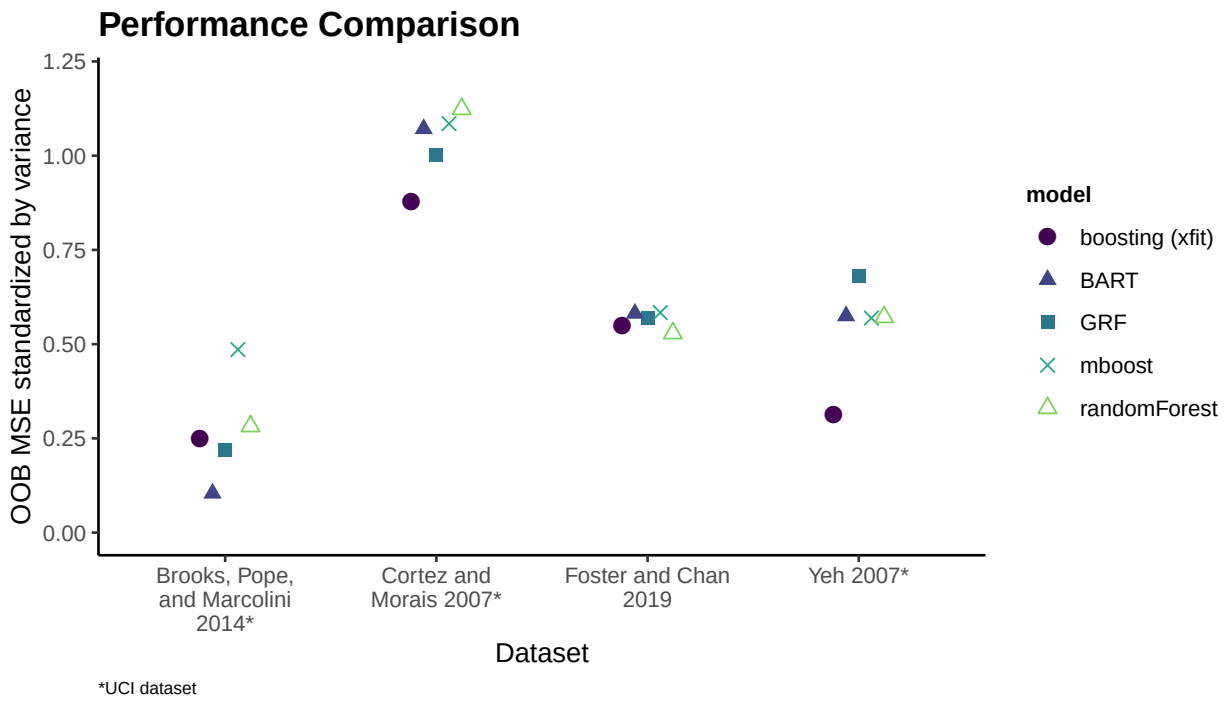

Figure 1: Performance Comparison by Model.

rule for our algorithm in all save the top-left dataset.

6.2 Simulation Evidence on Variable Importance Measure

Assessing a variable importance measure requires knowing which variables generated the outcome, so we turn to a simulation setting. We generate five covariates $[X_{i1}, X_{i2}, \ldots, X_{i5}]$, standard multivariate normal with all columns equicorrelated at 0.5.

The outcome y_i is generated as

$$y_i = X_{i1} \times X_{i2} + \epsilon_i$$

with $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. Only predictors X_1 and X_2 are relevant; the rest are irrelevant. If a VIM is valid, it should return a zero or negative value for X_{i3}, X_{i4}, X_{i5}.

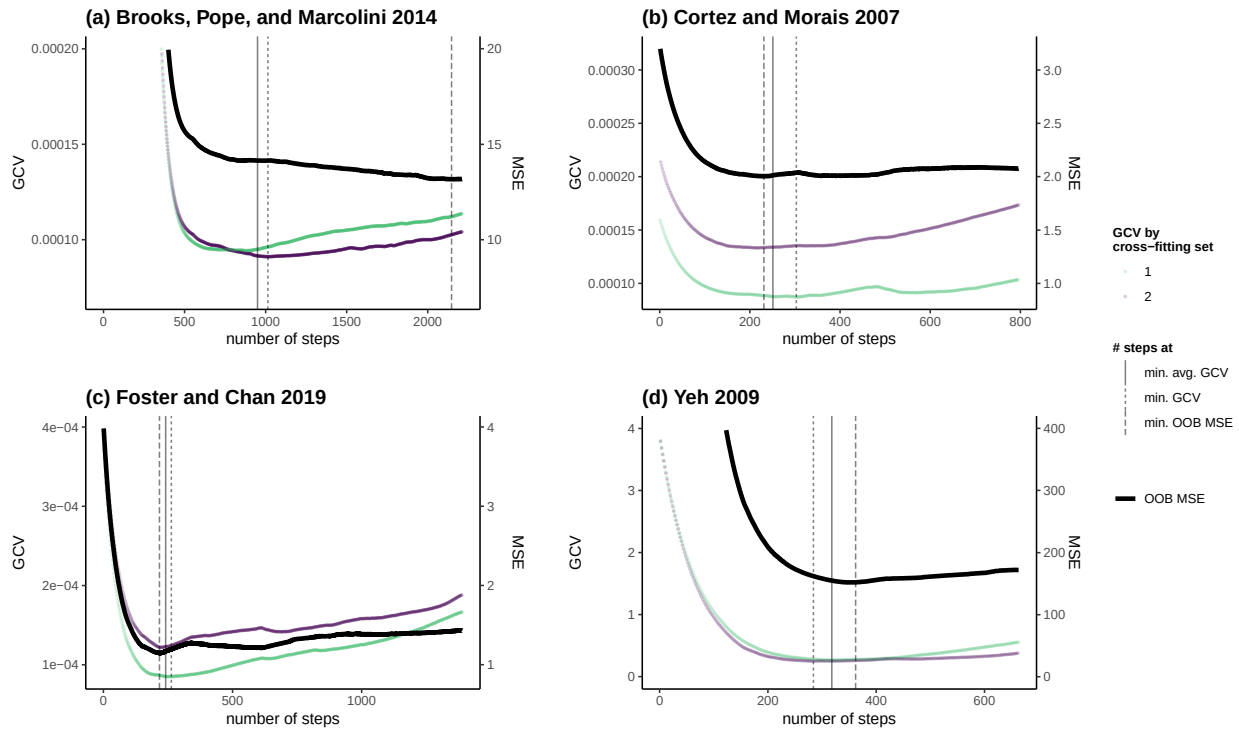

Figure 2: Efficacy of the GCV Statistic.

For comparison, we normalized the VIM for each variable in each model by dividing each VIM by the largest VIM value of the same variable in the same model. The simulation results are presented in Figure 3.[3]

In these simulations, our method returned a positive VIM for both X_1 and X_2 and zero or negative VIM for X_3, X_4, and X_5 20% of the time. For 84% of the time, it assigned positive VIM for at least one of X_1 and X_2. In only one simulation did our method incorrectly estimate a positive VIM for any one of X_3, X_4, and X_5. On the other hand, randomForest estimated positive VIM for *all* five variables in 48 of the 50 simulations, while mboost did in 44 out of 50 simulations and GRF always assigned a positive VIM for all variables in the

[3]We do not include BART in this evaluation, as it does not have a VIM.

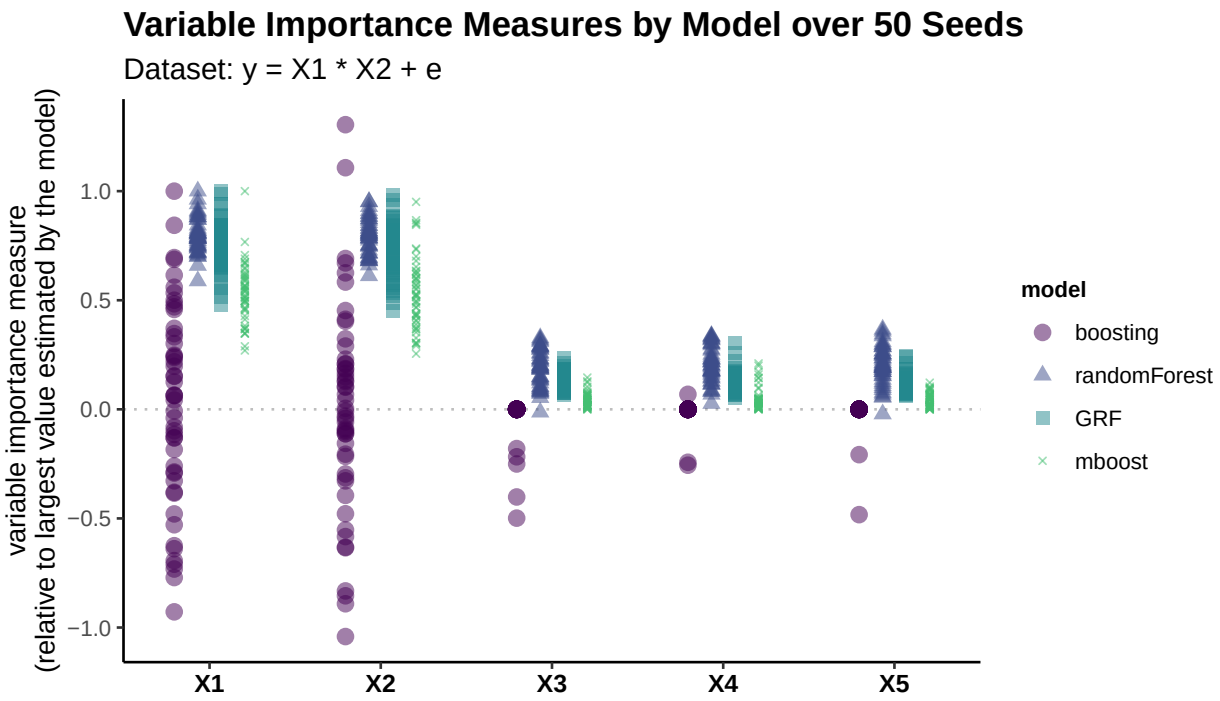

Figure 3: VIM by Model and Seed.

simulations. These results suggest that our VIM is less prone to false positives than the other three methods.

Similar to our findings in Section 6.1 using UCI datasets, our method either outperforms or is on par with existing models in terms of OOB prediction MSE. In Figure 4, we present the distribution of the OOB prediction MSE of three commonly used models as a ratio to the OOB prediction MSE of our method in the 50 simulations. A ratio above 1 means that the model in a given simulation has a higher OOB prediction MSE than our method. Our method gave the lowest OOB prediction MSE in 39 of the 50 simulations.

7 Future Developments

This project is still ongoing, so we conclude with a discussion of planned future developments. We plan to extend the project in three directions.

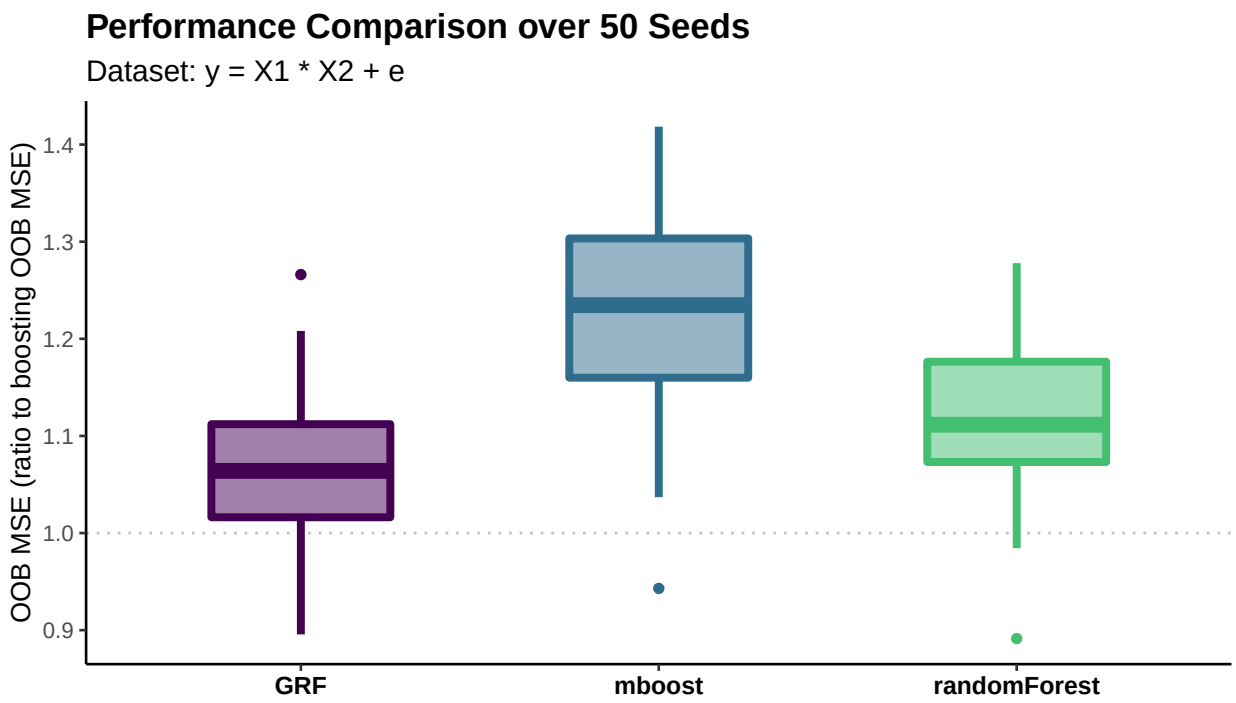

Figure 4: Distribution of OOB prediction MSE as a ratio to OOB MSE from our model in 50 simulations.

The first two are analytic. We plan to extend our results to more general loss functions. This will include likelihood methods, as well as several non-likelihood methods of interest like SVMs. Second, we are working to replacing the analysis of boosting steps with its limit as the step-size goes to zero and the number of steps to infinity, namely a continous time process.

Our second contribution will be to extend the project to data with structured variance. In this case, we can replace the gradient, $\widehat{\delta}^{(t)}$, with a term $\Omega^{-1}\delta^{(t)}$, where Ω^{-1} will allow for known clustering in the data.

Our goal is to develop a set of methods that achieve cutting-edge predictive performance, yet are transparent, and ultimately will apply to a broad array of loss functions as well as data with known clustering.

References

Barber, Rina Foygel and Emmanuel J. Candès. 2015. "Controlling the False Discovery Rate via Knockoffs.".

Barber, Rina Foygel and Emmanuel J. Candès. 2019. "A Knockoff Filter for High-Dimensional Selective Inference." 47(5):2504–2537.

Breiman. 1998. "Arcing Classifiers." *Annals of Statistics* 26(3):801–849.

Buhlmann, Peter and Bin Yu. 2003. "Boosting With the L2 Loss." *Journal of the American Statistical Association* 98(462):324–339.

Foster, Noel and Zenobia T. Chan. 2019. "Polarization for Paralysis: How Authoritarian States Weaponize Heuristic Biases to Shift Foreign Political Behaviors." *Paper presented at the American Political Science Association Political Communication Pre-Conference* .

Freedman, David and David Lane. 1983. "A Nonstochastic Interpretation of Reported Significance Levels." 1(4):292–298.

Friedman, Jerome H. 2001. "Greedy function approximation: A gradient boosting machine." *Annals of Statistics* 29(5):1189–1232.

Friedman, Jerome, Trevor Hastie and Robert Tibshirani. 2000. "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)." *Annals of Statistics* 28(2):337–407.

Ishwaran, Hemant and Min Lu. 2019. "Standard Errors and Confidence Intervals for Variable Importance in Random Forest Regression, Classification, and Survival." 38:558–582.

Janitza, Silke, Ender Celik and Anne-Laure Boulesteix. 2018-12. "A Computationally Fast Variable Importance Test for Random Forests for High-Dimensional Data." 12(4):885–915.

Mason, Baxter, Bartlett Frean. 2000. Boosting Algorithms as Gradient Descent. In *Advances in Neural Information Processing Systems*, ed. Müller Solla, Leen. Vol. 12 NeurIPS pp. 512–518.

Scornet, Erwan. 2020. "Trees, Forests, and Impurity-Based Variable Importance.".
URL: *http://arxiv.org/abs/2001.04295*

Shao, Jun. 1997. "An asymptotic theory for linear model selection." *Statistica Sinica* 7(2):221–264.

Strobl, Carolin, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin and Achim Zeileis. 2008. "Conditional Variable Importance for Random Forests." 9(1).

van der Vaart, Aad. 1998. *Asymptotic Statistics*. Vol. 3 of *Cambridge Series in Statistical and Probabilistic Mathematics* Cambridge University Press.

Zhang, Tong and Bin Yu. 2005. "Boosting with early stopping: Convergence and consistency." *Annals of Statistics* 33(4):1538–1579.

A Proofs

PROOF OF PROPOSITION 1 *Assume a fixed $n \times n$ projection matrix such that $H_n \mathbf{y} = \widehat{\mathbf{y}}$ with diagonal element i as $H_{n,i,i} \doteq k_{n,i}$, and $\widehat{\epsilon}_i \doteq y_i - \widehat{y}_i = y_i - \widehat{g}(\mathbf{x}_i)$*

Then

$$\mathbb{E}\left\{ \left(\frac{\widehat{\epsilon}_i}{1 - \frac{k_{n,i}}{n}} \right)^2 \middle| \mathbf{X} \right\} = \mathrm{Var}(y_i \mid \mathbf{x}_i) \tag{4}$$

Tightness is established by characterizing the limiting distribution of

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{\widehat{\epsilon}_i}{1 - \frac{k_{n,i}}{n}} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \widehat{\epsilon}_i \left(1 + \frac{k_{n,i}}{n} + o_p\left(\frac{1}{n}\right) \right).$$

Squaring gives

$$\begin{aligned} \frac{1}{n} \left\{ \sum_{i=1}^{n} \widehat{\epsilon}_i \left(1 + \frac{k_{n,i}}{n} + o_p\left(\frac{1}{n}\right) \right) \right\}^2 &\xrightarrow{\mathrm{u}} \mathbb{E}\left\{ \epsilon_i^2 \left(1 + \frac{k_{n,i}}{n} + o_p\left(\frac{1}{n}\right) \right)^2 \right\} \\ &\leq \sqrt{\mathbb{E}\left\{\epsilon_i^4\right\} \mathbb{E}\left\{ \left(1 + \frac{k_{n,i}}{n} + o_p\left(\frac{1}{n}\right) \right)^4 \right\}} \end{aligned}$$

which is finite so long as $\mathbb{E}(\epsilon_i^2) < \infty$ and the term $k_{n,i}/n = o_p(1/n)$.

Then, we have $\widehat{g}(\mathbf{x}_i) \xrightarrow{\mathrm{p}} g(\mathbf{x}_i)$ since uniform convergence implies pointwise convergence, and since the projection matrix is fixed,

$$\begin{aligned} \mathrm{Var}(\widehat{y}_i \mid \mathbf{X}) &= \mathrm{Var}(\Pi_i(\mathbf{X})\mathbf{y} \mid \mathbf{X}) \\ &= \Pi_i(\mathbf{X}) \mathrm{Var}(\mathbf{y} \mid \mathbf{x}_i) \Pi_i(\mathbf{X})^\top \end{aligned}$$

PROOF OF PROPOSITION 2 *This statistic is calculated from nested models, comparing the empirical loss with and without* x_j*. Calculating directly, we get*

$$\widehat{VIM}^{o}_{j,n} = R_{\widehat{g}_{-j},n}(y_i, \mathbf{x}_{i,-j}) - R_{\widehat{g},n}(y_i, \mathbf{x}_i) \tag{5}$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\left\{\left(y_i - \widehat{\mathbb{E}}\left(y_i \mid \mathbf{x}_{i,-j}\right)\right)^2 - \left(y_i - \widehat{\mathbb{E}}\left(y_i \mid \mathbf{x}_i\right)\right)^2\right\} \tag{6}$$

In the case with $\mathrm{Var}(y_i \mid \mathbf{x}_i) = 1$ *for all* i*, this statistic is* $\chi^2_{k_j}$*, with* k_j *the increase in model dimensionality attributable to variable* j*. In general, the statistic is a variance weighted average of scaled* χ^2 *statistics, which converges at rate* n*. Therefore* $\sqrt{n}\,\widehat{VIM}^{o}_{j,n} \to 0$*.*

PROOF OF PROPOSITION 3

By our construction,

$$\widehat{VIM}_{j,n} = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j^{(t)}=j}\left\{\widehat{y}_i^{(t)2} - \left(\widehat{y}_i^{(t)} + \Delta^{(t)}_{j^{(t)}}(\mathbf{x}_i)\right)^2\right\}$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\sum_{j^{(t)}=j}\left\{-2\widehat{y}_i^{(t)}\Delta^{(t)}_{j^{(t)}}(\mathbf{x}_i) - \Delta^{(t)}_{j^{(t)}}(\mathbf{x}_i)^2\right\}$$

By our assumption on model complexity, the square term goes to zero.

Next, assume $x_{j^{(t)}} = x_j$ *and* x_j *is irrelevant. Denote as* $h_j^{\perp}, h_j^{\Pi}$ *the projections of* $\Delta^{(t)}_{j^{(t)}}(\mathbf{x}_i)$ *onto* $x_j^{\perp}, x_j^{\Pi}$*, respectively. Then,*

$$-\widehat{y}_i^{(t)}\Delta^{(t)}_{j^{(t)}}(\mathbf{x}_i) = -\widehat{y}_i^{(t)}h_j^{\perp} - \widehat{y}_i^{(t)}h_j^{\Pi}$$

which has a finite variance since the first term in the product in the lefthand side is asymp-

totically normal and the second is fixed given $x_{j^{(t)}}$*. Then,*

$$\begin{aligned}\mathbb{E}\left(-\widehat{y}_i^{(t)}h_{\tilde{j}}^{\perp}-\widehat{y}_i^{(t)}h_j^{\Pi}\right) &= \mathbb{E}\left(-\widehat{y}_i^{(t)}h_j^{\Pi}\right)\\ &= \mathbb{E}\left(\widehat{\epsilon}_i^{(t)}h_j^{\Pi}\right)-\mathbb{E}\left(yh_j^{\Pi}\right)\end{aligned}$$

On the righthand side, the first term is zero, since x_j *is irrelevant. The second, though, is not.*

We can say $\mathbb{E}\left(\widehat{\epsilon}_i^{(t)}h_j^{\Pi}\right)\geq 0$ *since we are selecting a direction to decrease the residual sum of squares. We can also say* $\mathbb{E}\left(\widehat{\epsilon}_i^{(t)}h_j^{\Pi}\right)\geq\mathbb{E}\left(yh_j^{\Pi}\right)$ *since we are taking a step that will maximally decrease it. Therefore,* $\sqrt{n}\,\widehat{VIM}_{j,n}$ *will converge to a normal with nonnegative mean.*

Proof of Proposition 4 *Since the bias in the previous proof is driven by a correlation with a function of* x_j^{Π}*, removing this term will remove the bias.*