INFFOREST Variable Importance on Random Forests

A Thesis
Presented to
The Division of Mathematics and Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Aurora Owens

May 2017

Approved for the Division
(Mathematics)

Andrew Bray

Acknowledgements

.

Table of Contents

List of Tables

List of Figures

Abstract

Random forests are powerful predictive models but there is not a built-in method for statistical inference within these models. This paper compares several of the most common methods of performing inference on random forests while presenting a new method, INFFOREST variable importance. On simulated data with multicolinearity, the INFFOREST method is able to show significance for four out of five predictors used to generate the response. Existing methods are tested and performed similarly on the simulated data set. INFFOREST variable importance allows claims to be made about the relationship between a predictor and the response, in the context of the rest of the variables in the model. This sets it apart from the exiting methods and creates some interesting implications.

Dedication

.

Chapter 1

Introduction

1.1 Trees and Random Forests

To begin our discussion of trees and random forests, we will first consider the following example using data from a dendrologic study of five orange trees. This study measured two things for each tree: the age of the tree (recorded in days) and the circumference of the trunk (in cm). These are called, in general terms, the variables recorded by the study. In the dataset below, each column represents a variable and each row represents one set of measurements. There are 2 columns (age and circumference) and 35 rows. The first six rows are displayed in table 1.1.

Let's pretend we are interested in the following question: knowing only the circumference of the orange tree, can we predict the age of the tree? This question is called a formula, or a guess at how the relation between the two variables functions. We often refer to formulas using the following notation:

$$Age \sim Circumference$$

In this formula, circumference is the predictor and age is the response. Suppose we expanded the study to include the height of the orange trees at various stages of development. Now we can consider both the circumference and the height of the tree when we make our predictions of the age. When we have multiple predictors, we add

Table 1.1: The first six rows of the Orange data set

age	circumference
118	30
484	58
664	87
1004	115
1231	120
1372	142

them to the notation in the following way:[1]

$$Age \sim Circumference, Height$$

Returning to the original orange tree data set, we can begin our investigation into the validity of the formula by plotting the data and observing the relationship between the variables in figure 1.1.

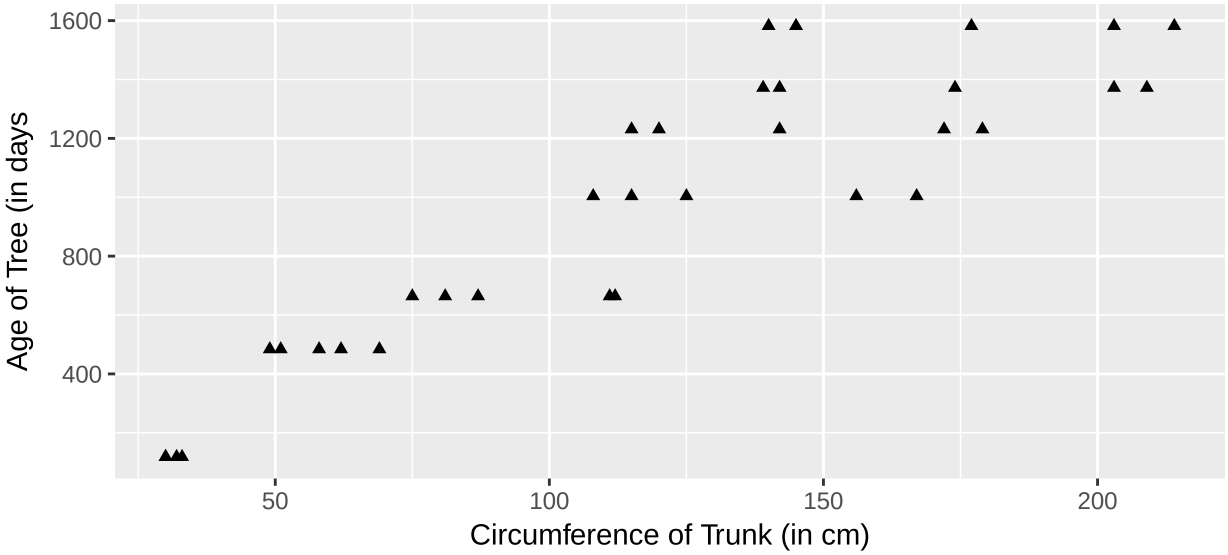


Figure 1.1: The relationship between age and circumference of the trunk of orange trees.

As can be seen in figure 1.1, generally older trees have thicker trunks, and it seems like we are not wrong to suspect that circumference is a good predictor of age. As the data could be reasonably represented by a straight line, we can say that the relationship between trunk circumference and tree age is roughly linear. To create our predictions of age, we fit the formula $Age \sim Circumference$ to a model. A predictive model is, put simply, a systematic way to make our predictions. The most common type of model is the linear model, which creates predictions from a line through the data.

As can be guessed from figure 1.2, the linear model works better on certain data than others. The linear model necessitates several assumptions that may not always be appropriate. We'll return to a brief discussion of the assumptions of the linear model later in this chapter.

This paper discusses at length tree-based models. A tree for the formula $age \sim circumference$ is similar to the linear model in that it presents a systematic way to make predictions, but the two differ in that the tree is not linear in any fashion. In fact, we can compare the differences between the two models by comparing figures 1.2 and 1.3.

[1] Often the notation for multiple predictors is written $Y \sim X + W$ but this assumes an additive, linear relationship between the predictors and the response. This assumption is unnecessary for tree-based models so the notation, $Y \sim X, W$ is used.

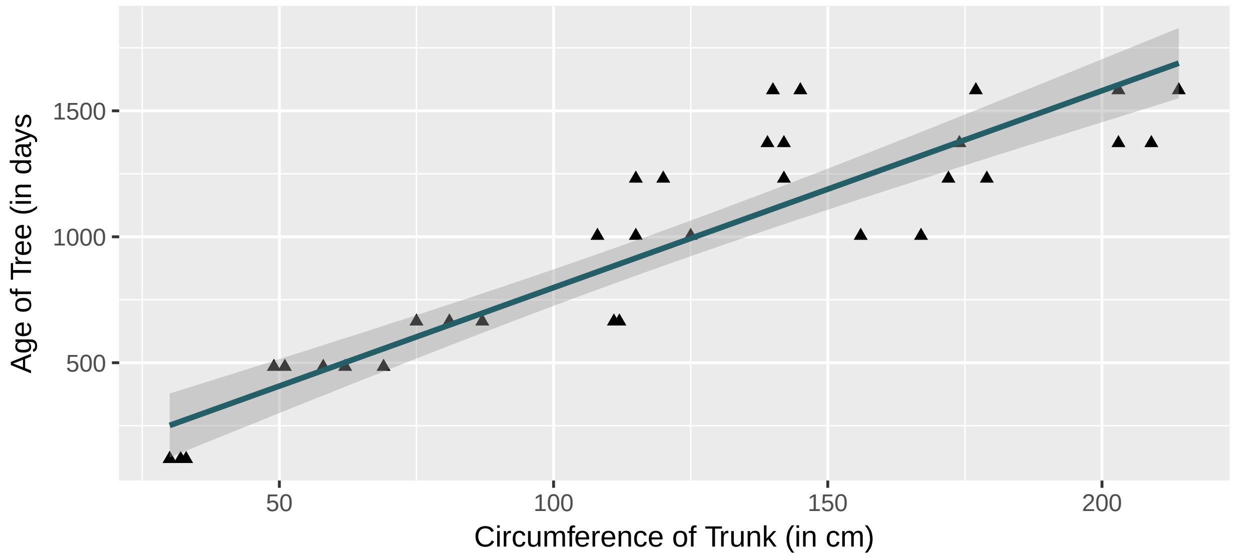


Figure 1.2: A linear model representing age ~ trunk circumference in orange trees. The shaded area represents a 95% confidence interval around this line.

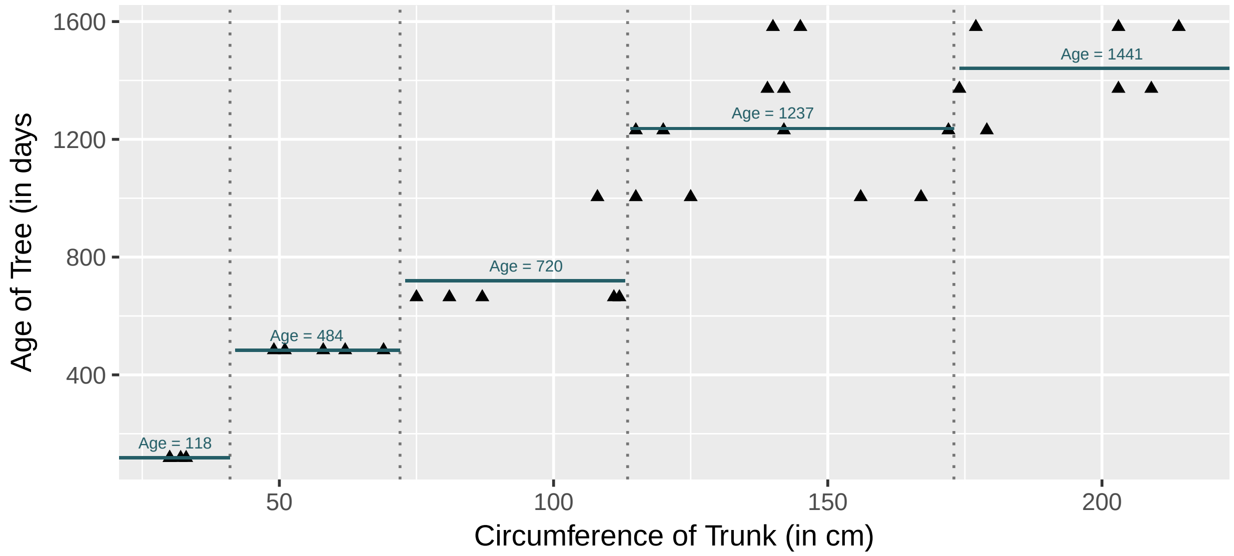


Figure 1.3: A tree modeling the formula age trunk circumference first creates partitions on the predictor, seen as vertical lines, and then predicts the value of the response within that partition, seen as text.

When using the linear model, we make predictions in the following way: given a value of X (circumference) the corresponding value of Y (age) on the line is our prediction. The predictions from the tree are gathered similarly: given a value of X (circumference), our prediction is the average value of Y (age) within the partition that X falls into.[2]

[2] While any curve can be approximated in a step-wise manner when the number of steps approaches

Tree methods get their name from a common way of representing them in higher dimensions, when there is more than one predictor. Figure 1.4 shows this method. In this case, given a new value for circumference, one would start their predictions at the top of the tree and, depending on the value of circumference and the instructions at each intersection or split, one would fall down branch by branch before landing on a prediction for age.

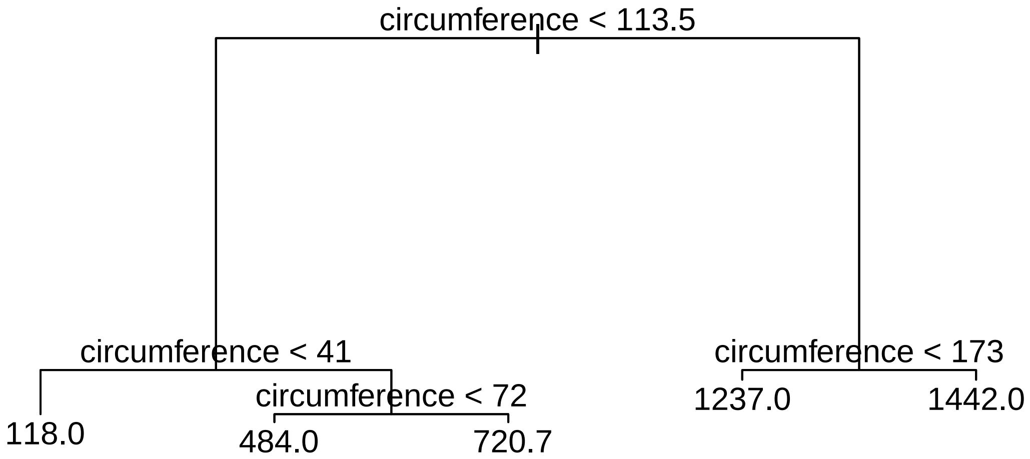


Figure 1.4: A tree representing age trunk circumference in orange trees.

1.1.1 Trees: Background

Decision trees are a convenient way to represent data and assist in decision making. Morgan and Sonquist (1963) derived a way for constructing trees motivated by the specific characteristics of data collected from interviews and surveys. The first difficulty in analyzing this data was that data collected from surveys is mostly categorical, where the observation is that the participant is a member of some discrete group. Some common categorical variables are gender, ethnicity, and education level. Numeric variables, like age, height, and weight are, in general, much easier to work with. On top of this, the data sets Morgan and Sonquist dealt with had few participants (rows) and many variables (columns). To add to their difficulties, there was reason to believe that there were lurking errors in the data that would be hard to identify and quantify. Lastly, many of the predictors were correlated. Morgan and Sonquist doubted that the additive assumptions of many models would be appropriate for this data. They noted that while many statistical methods would have difficulty accurately parsing this data, a clever researcher with quite a lot of time could create a suitable model simply by grouping values of the predictors and predicting that the response corresponding to these values would be an average of the observed responses given the grouped conditions. Their formalization of this procedure in terms of "decision rules" laid the

∞, a tree model does not converge to the linear model as the number of splits approaches ∞, even when the data is linear.

Table 1.2: Estimated linear coefficients, error, and p-values from the model fit in section 1.1 on the orange tree dataset

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	16.603609	78.1406182	0.2124837	0.8330368
circumference	7.815998	0.6058806	12.9002281	0.0000000

groundwork for future research on decision trees. See figure 1.3 for a visualization of this process.

Later researchers proposed new methods for creating trees that improved upon the Morgan and Sonquist model. Leo Breiman et al (1984) proposed an algorithm called CART (Classification And Regression Trees) to fit trees on various types of data. Torsten Hothorn, Kurt Hornik and Achim Zeileis argue in their 2006 paper *Unbiased Recursive Partitioning: A Conditional Inference Framework* CART has a selection bias toward variables with either missing values or a great number of possible splits. This bias can affect the interpretability of all tree models fit using this method. As an alternative to CART and other algorithms, Hothorn et al. propose a new method: conditional inference trees.

There is a limit to the predictive capabilities of a single tree as they suffer from high variance. To alleviate this, aggregate methods called forests are often used instead. They function by enlisting the help of many trees, and then by aggregating the responses over all of them. The two most common types of forests are bagged and random forests. This is further explored in chapter 2.

1.1.2 Inferential vs Descriptive Statistics

In the earlier sections, we focused on building predictive models, but this paper hopes to use tree-based methods beyond this context. The linear model is a mainstay in social science because it allows for easy and interpret-able statistical inference. Return to the orange tree example from section 1.1. The linear model gives us a line with which we can make predictions, but it also gives estimated coefficients and conducts hypothesis tests on the values of the coefficients.

This table provides evidence that not only is trunk circumference a good predictor of age, the relationship between them is the equation of the line:

$$Age = 7.81 \cdot Circumference + 16.6$$

Roughly, for every 1 cm of trunk growth, we would expect the tree to be 7.81 days older. Not only are we provided with a way to describe the relationship between age and circumference, we have conducted statistical tests to make reasonably sure that our estimates are not just due to chance. Inferential claims about the nature of tree age and trunk growth are possible here. It is important to note the difference between inferential and descriptive statistics. Descriptive statistics describe the data at hand without making any reference to a larger data generating system that they come from. It follows that inferential statistics then make claims about the data generating system given the data. The model in figure 1.4 could be used to make descriptive claims about the orange tree data. For example, given the data we have, we expect a sapling with a trunk circumference less than 41 cm to be 118 days old. However, trees are variable; they are very sensitive to changes in the data set. It's entirely possible that if we fit this tree on a sample of the data, the predictions would change. See chapter 2 for more discussion on the variability of trees. Our claims about the relation between circumference and age in young orange trees can only be descriptive as we have not taken into account the variability in gathering them. This paper's aim is to describe a process of making inferential claims using trees and random forests that employs permutation tests.

As stated in the introduction of the *Chronicle of Permutations Statistical Methods* by KJ Berry et al. 2014, there are two models of statistical inference. One is the population model, where we assume that the data was randomly sampled from one (or more) populations. Under this model, we assume that the data generated follows some known distribution. "Under the population model, the level of statistical significance that results from applying a statistical test to the results of an experiment or a survey corresponds to the frequency with which the null hypothesis would be rejected in repeated random samplings from the same specified population(s)" (Berry et al, 2014).

The permutation family of methods, on the other hand, only assumes that the observed result was caused by experimental variability. The test statistics are calculated for the observed data, then the data is permuted a number of times. The statistic is calculated after each permutation to derive a distribution of possible values. Then the original test statistic is tested against this distribution. If the observed value is exceptionally rare, then there is evidence that our observation was not simply experimental variability.

1.2 A Step Back

A random forest R_f is the set of functions $T_1, ..., T_N$ where each T_j is a piece-wise function from the sample space Ω into itself. In general, Ω is defined by an n x p +1 matrix where each column is a random variable.

$$when\ n = 3, p = 2$$

$$D = \begin{bmatrix} y_1 & x_{11} & x_{1p} \\ y_2 & x_{21} & x_{2p} \\ y_3 & x_{31} & x_{3p} \end{bmatrix}$$

Each tree in a random forest, $T_j \in R_f$, is generated on a subset of Ω called the training set. This training set is a bootstrapped sample of the original dataset and is noted as B^t. [3]

[3] In the bootstrapped sample, there will be repeated rows from the original data. This allows B^t to be a subset of Ω without sacrificing the number of observations.

$$B^t = \begin{bmatrix} y_1 & x_{11} & x_{1p} \\ y_1 & x_{11} & x_{1p} \\ y_3 & x_{31} & x_{3p} \end{bmatrix}$$

It is then tested on a disjoint subset of Ω called the test set, $\bar{B}^t$, where $\bar{B}^t = \Omega \backslash B^t$. The image of T_j is called the predictions of T_j.

$$\bar{B}^t = \begin{bmatrix} y_2 & x_{21} & x_{2p} \end{bmatrix}$$

As outlined in the 1984 textbook *Classification and Regression Trees*, Breiman, Friedman, Olshen, and Stone described their method for creating, pruning, and testing regression trees. There are essentially three steps: one, decide on a variable to split over, two, partition that variable space in two distinct partitions, and three, set our initial predictions for each partition to be mean value of the response according to the observed responses corresponding to the values in the partitions. Recursively, this process is repeated for each new partition until some stopping condition is reached. This is a top down, greedy algorithm that functions by creating as large a tree as possible ([@bibCART]).

Random Forests are generated by fitting a large number of trees, each on a boosted sample of the data. The crucial difference, however, between the trees in CART and the trees in a random forest, is that at each node in a random forest, only a subset of the predictors are considered as candidates for possible splits. This decorrelates each tree from its neighbors, and limits variability of the whole forest. ([@bibISL])

Predictor columns from the bootstrapped sample are themselves sampled to select the columns available for a tree to split over.

$$B^t = \begin{bmatrix} y_1 & x_{11} \\ y_1 & x_{11} \\ y_3 & x_{31} \end{bmatrix}$$

1.3 Inference on Random Forests

1.3.1 The Problem

Random forests create models with great predictive, but poor inferential capabilities. After Morgan and Sonquist's initial development of decision trees, trees quickly moved to the domain of machine learning and away from statistics. Researchers focused on bettering predictions and improving run times and less on the statistics behind them. In a single tree, descriptive claims may be simple to make, but it is much more difficult to describe the behavior of the whole forest. Inferential statistics with random forests generally falls behind the predictions in importance. This has limited the applications of random forests in certain fields, as to many the question of "why" the data is the way it is is more important than building predictions. There are several means of performing descriptive statistics with random forests that could be interpreted incorrectly as attempting to answer this but without a statistically backed method for performing inference, the use of random forest is limited to prediction-only settings.

1.3.2 Proposed solutions to this problem

Variable importance could be the tree-based analogue to the coefficients of the linear model, in that the variable importance for the predictor X_i in the model for $Y \sim X_1, ..., X_p$ is the amount of variance in model accuracy due to X_i. Breiman proposed a method of permuted variable importance in his paper *Statistical Modeling: The Two Cultures* to answer this problem. Their method compares the variable importance for each variable in a tree-wise manner. For each tree, the permuted variable importance of the variable X_j is:

$$VI^t(x_j) = \frac{\sum_{i \in |\bar{B}^t|}(y - \hat{y})^2}{|\bar{B}^t|} - \frac{\sum_{i \in |\bar{B}_p^t|}(y - \hat{y_p})^2}{|\bar{B_p}^t|}$$

Where $\bar{B}^t$ is the out of bag sample for tree t, $|B|$ is the number of observations in that sample, $\bar{B}_p^t$ is with X_j permuted, $\hat{y}$ is the predicted outcome, and $*\hat{y}^t$ is the predicted outcomes after variable X_j has been permuted. This value is averaged over all the trees. It is important to note that if the variable X_j is not split on in the tree t, the tree-wise variable importance will be 0.

Strobl et al from the University of Munich criticize this method in their 2008 technical report *Danger: High Power! – Exploring the Statistical Properties of a Test for Random Forest Variable Importance.* First, this method has the downside of increasing power with increasing numbers of trees in the forest. This is a more or less arbitrary parameter which we would hope would not affect our importance estimates. Second, the null hypothesis under Breiman and Cutler's strategy is that the variable importance VI for any variable X_j is not equal to zero given Y, the response. Because random forests are most often used in situations with multicolinearity that would make other methods like the linear model difficult, Strobl argues that any variable importance measure worth its salt should not be misled by correlation within the predictors.

The researchers at the University of Munich published a fully fleshed response to the Breiman and Cutler method in 2008, titled *Conditional Variable Importance for Random Forests* that addresses these issues. Strobl et al propose restructuring the Breiman and Cutler algorithm to account for conditional dependence among the predictors. The null hypothesis is that $VI_\beta(X_j) = 0$ given the predictor Y *and all other predictors* $X_1, ..X_n$. This accounts for interactions between X_j and the other predictors, while preserving the relationship between Y and the remaining predictors.

This paper aims to provide a response to this method. The partitions are made from the random forest corresponding to the formula of Y $X_1, ..., X_n$ instead of a model of X_j $X_1, ..., X_n$. This ignores the common situation where if the predictors are correlated enough, then they act as stand ins for each other, so that if one variable is heavily influential in a certain tree at predicting Y, the other variable will be forgotten altogether.

Chapter 2

Simulations and Comparisons

Our goal for this chapter is to compare trees, random forests, and linear models. In this chapter, we will use simulated data instead of the orange data set. One reason for this is theoretical consistency. One hopes that one's results will not be rendered null and void by any misstep in the data collection that comes to light. This also ensures that these simulations can be repeated by later researchers, but, granted, it does not make for the most exciting analysis. For now, consider Y to be our response variable. In the first simulation, V will be the set of our predictors, and V_j to be a predictor in V. The formula will be the same for each model: $Y \sim V$. In our second simulation, our set of predictors will be denoted as X and X_j will be a member of X. The formula in this case is $Y \sim X$.

2.0.1 Simulated Data

Random forests excel in predicting outcomes with correlated predictors, although these situations can make it difficult to perform intelligible inference. In a situation in which the correlated predictors are X_1 and X_2 and the formula we're estimating is $Y \sim X_1 + X_2$, it can be difficult to say if X_1 or X_2 is truly the better predictor. To illustrate this idea, compare a few existing methods, and explore methods of inference on tree based models, we will simulate two data sets with different correlation structures. We will focus more on the correlation structure between the predictors than on their relationships with the response and this will be reflected in the simulations.

The first simulated dataset is generated under the same scheme as in [@bib-strobl2008]. Under this method, the 13 x 1000 data set, D_1, has 12 predictors, $V_1, .., V_{12}$, where $V_j \sim N(0,1)$. The first four are block correlated to each other with $\rho = .9$. They are related to Y by the linear equation:

$$Y = 5 \cdot V_1 + 5 \cdot V_2 + 2 \cdot V_3 + 0 \cdot V_4 + -5 \cdot V_5 + -5 \cdot V_6 + 0 \cdot V_7 + 0 \cdot ..... + E, E \sim N(0, \frac{1}{2})$$

Note in table 2.1, the coefficients for $V_7, ..., V_{12}$ are all zero.

In the last column of table 2.1, the coefficient "beta", refers to the function used to generate the Y values. Although $V4$ was not included in the model $Y \sim V1, ..V_{12}$, its strong correlation with more influential predictors $V_1, ..., V_3$ ensures that it still

Table 2.1: Empirical correlations and coefficients of the variables in the first simulated data set

	V1	V2	V3	V4	V5	V6	V7	y	beta
V1	1.000	0.915	0.908	0.907	-0.034	0.006	0.012	0.839	5
V2	0.915	1.000	0.914	0.914	-0.020	-0.001	-0.001	0.838	5
V3	0.908	0.914	1.000	0.903	-0.017	-0.007	0.007	0.818	2
V4	0.907	0.914	0.903	1.000	-0.002	-0.015	0.023	0.800	0
V5	-0.034	-0.020	-0.017	-0.002	1.000	0.044	0.005	-0.392	-5
V6	0.006	-0.001	-0.007	-0.015	0.044	1.000	-0.005	-0.368	-5
V7	0.012	-0.001	0.007	0.023	0.005	-0.005	1.000	0.004	0

shows a strong, empirical linear correlation with Y. A linear model would likely *overstate* the effect of V_4 on Y. [1] [2]

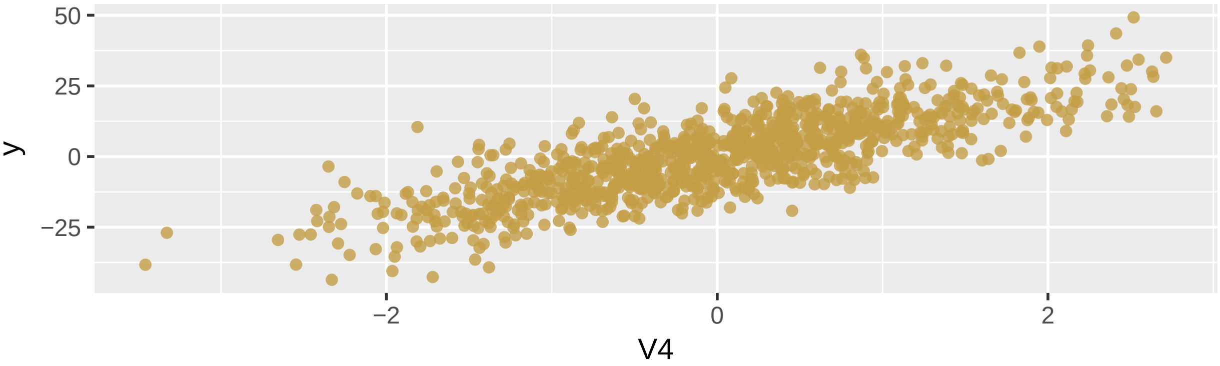


Figure 2.1: Relation between V4 and Y. This relation has empirical linear correlation = .789

The densities of $V_1, ..., V_4$ in figure 2.2 are all very similar due to the way they were generated. D_1 represents the case where some of the predictors are linearly correlated with each other, but that is not the only possible correlation structure. The data set D_2 is simulated similarly to D_2 in that D_2 contains twelve predictors and one response variable. The first four variables are generated in the following way:

$$X_1 \sim N(1, 0)$$

[1] A brief note on uncertainty is needed here. It's true that in this setting we can say that V_4 is actually unimportant to understanding Y, but in situations with real data this is profoundly more difficult to parse. Often like in the social science situations that Morgan and Sonquist encountered, the real relationship between correlated predictors is complicated and often there is some theoretical backing or other insight that is gained to include variables that may not be important to the model.

[2] Another point that could be said is that, no V_4 is not unimportant, V_1, V_2, and V_3 are just stand ins for the real star, V_4, as they are nearly the same ($\rho \sim 1$). Then the real relationship represented here is $Y \sim (5 + 5 + 2) \cdot V_4 + -5 \cdot V_5 + -5 \cdot V_6 + -2 \cdot V_7$. This model is not unsuccessful in capturing the structure of the data, and this is typically the practice used to model data with highly correlated predictors. If this seems philosophically satisfying to you, the rest of this thesis may seem a bit inconsequential.

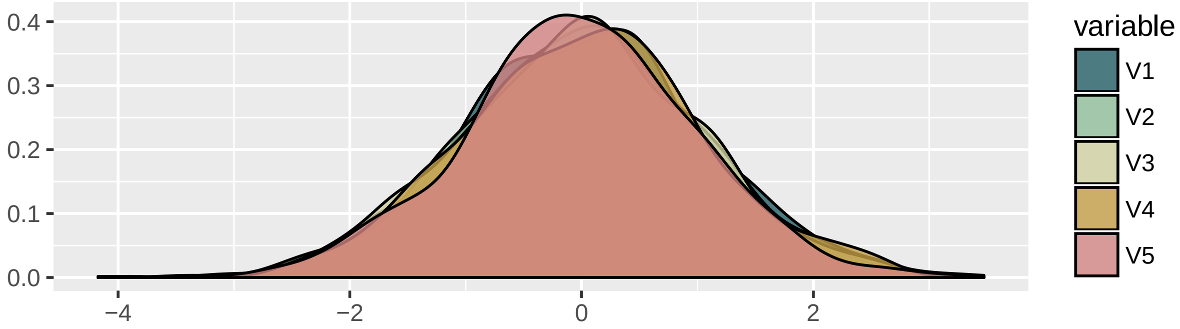


Figure 2.2: Empirical densities for V1 through V4

Table 2.2: Empirical correlations and coefficients of the first seven predictors and the response using the second simulated dataset

	X1	X2	X3	X4	X5	X6	X7	y	beta
X1	1.000	0.013	0.044	0.002	-0.012	-0.012	-0.044	-0.003	5
X2	0.013	1.000	-0.520	-0.272	0.073	0.004	0.006	-0.697	5
X3	0.044	-0.520	1.000	-0.004	0.030	-0.013	0.023	0.271	2
X4	0.002	-0.272	-0.004	1.000	-0.031	0.022	-0.002	0.352	0
X5	-0.012	0.073	0.030	-0.031	1.000	0.010	-0.030	-0.099	-5
X6	-0.012	0.004	-0.013	0.022	0.010	1.000	-0.088	-0.022	-5
X7	-0.044	0.006	0.023	-0.002	-0.030	-0.088	1.000	0.013	0

$$X_2 = log(X_1) + E, E \sim N(1,0)$$

$$X_3 = log(X_2) + E, E \sim N(1,0)$$

$$X_4 = log(X_4) + E, E \sim N(1,0)$$

This simulation scheme leads to the first four variables having an obvious relationship between each other, but relatively low linear correlations, as seen in figure 2.3. Predictors are sampled by $X_5, ..., X_{12} \sim N(0,1)$. The Y values are generated according to the following formula:

$$Y = 5 \cdot (X_1)^2 + 5 \cdot (X_2)^2 + 2 \cdot (X_3)^2 + 0 \cdot X_4 + -5 \cdot X_5 + -5 \cdot X_6 + 0 \cdot X_7 + 0 \cdot ..... + E, E \sim N(0, \frac{1}{2})$$

The correlation structure in D_2 is much more difficult to capture with a single line. The relationships between the first four predictors form striking, symmetrical scatter plots in figure 2.3. This information is considered again in table 2.2, where the empirical correlations of the first seven variables are presented along with their observed correlations with Y and their simulation coefficients.

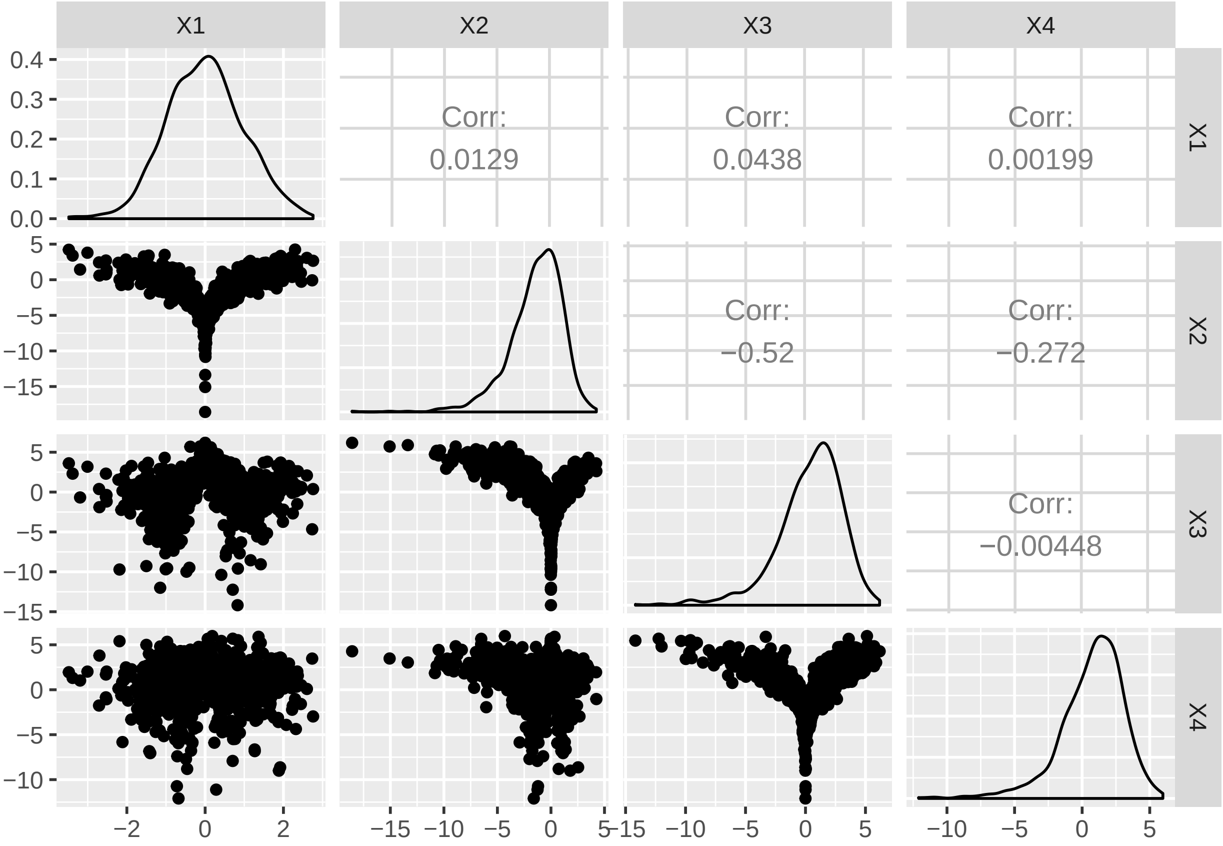


Figure 2.3: Correlation structure of the first four variables in D2

2.1 Models and Comparisons

CART: Regression Trees

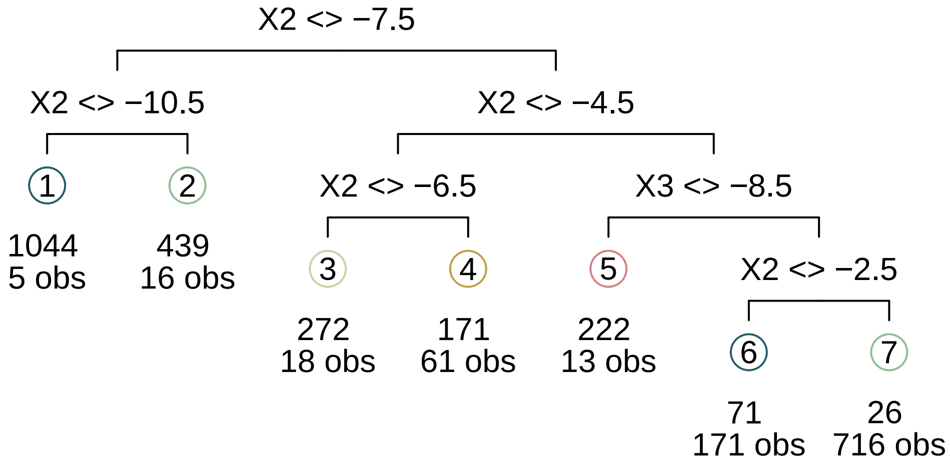


Figure 2.4: CART representing Y~ X, from D2

The CART tree representing the model $Y \sim X$ in figure 2.4 is easy enough to understand. Starting at the very top of the tree, predictions can be made based on the values of the leaves (or ending nodes) given the requirements of the path to get there. Trees can be quite variable, so to get a better idea of the differences between the methods let's run a simulation. This simulation scheme will take advantage of the non linearity present in D_2.

Algorithm 1 Simulation Scheme 2.1

```
for $i \leq 1000$ do
    Randomly sample $\frac{2}{3}$ of the observations in $D_1$ to a training set, $D^i_{1,train}$. The other observations, $x \in D_2, x \notin D^i_{2,train}$ form the testing set $D^i_{2,test}$
    Fit a tree, $T^i$, to the data under the model $Y \sim X_1, ..., X_2$ using the observations in $D^i_2$
    Calculate the $MSE_{test}$ of the model using the equation: $MSE_{test} = \frac{1}{n}\sum(y_j - \hat{y}_j)^2$
end for
```

Note that n is the number of observations in $D^i_{1,test}$, $y_j \in D^i_{2,test}, \hat{y}_j \in T^i(D^i_{2,test})$ for $1 \leq j \leq n$ This produces one distribution of MSE_{test} for CART. This simulation scheme will be repeated for the linear model and the random forest and the MSE_{test} distributions are compared in figure 2.5.

The linear model is characteristically less flexible and less prone to over-fitting than either of the tree-based methods, CART and random forests, and has a MSE_{test} distribution that is quite peaked. CART is flexible and suffers from high variance. The random forest models perform much better on average than either the CART or the linear model, due to both the non-linear relationships between Y and the predictors

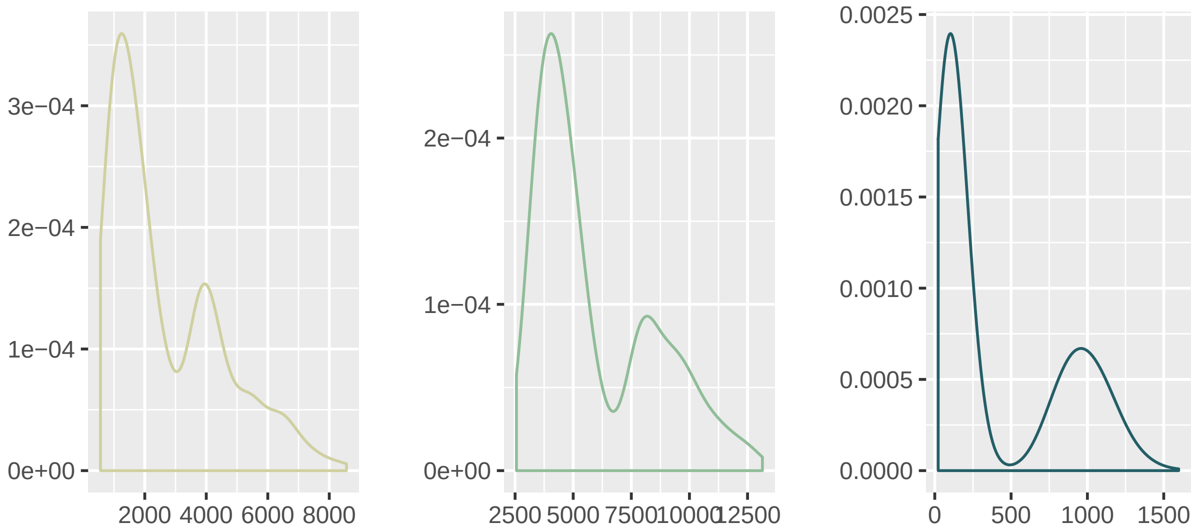


Figure 2.5: The simulated MSE distributions of CART, linear model, and the random forest on D2

and the random forest's ability to decorrelate each of the trees by restricting the variables available on each split. See chapter 3 for more discussion on the enforced heterogeneity of trees in the random forest. As, MSE_{test}, is a test of *predictive* accuracy it is not surprising that the random forest performed admirably. On a certain level, that is what they are designed to do. The linear model is not popular in predictive situations but it is ubiquitous in inferential ones.

Chapter 3

Random Forest Variable Importance

To implement the various variable importance measures discussed in this chapter and in chapter 4, functions for creating trees, random forests, and their importance measures were created. The trees were fit using the standard two-part CART-like algorithm. The function chooses a variable to split on with linear correlation with respect to Y, but instead of looking for correlations above a certain threshold which is common, it chooses the variable with the highest correlation when compared to its peers. This alleviates the situation where a variable with a non-linear relationship would be passed over again and again. The splitting is then done via minimization of the following function with respect to i:

$$RSS_{node}(i, X, Y) = RSS_{leaf}(Y|X < i) + RSS_{leaf}(Y|X \geq i)$$

$$RSS_{leaf} = \sum(y - \hat{y})^2$$

This function considers the regression case only, and only numeric predictors. Leaves are created when the resultant split would be unsatisfactory, i.e. at least one of these cases applies: one daughter node would have five members or less, the split on the chosen variable would not result in a decrease in RSS, or the data contained in the node is already suitably homogeneous. This generates very large trees: a quality that is not an issue in random forests but may be problematic in a stand-alone setting.

There are several ways to display a tree, but when it is displayed as a table it is read in the following way: each row corresponds to a node of the tree which contains a certain number, `n` observations. This number of observations, or rows in the data set is naturally a subset of both the original data set and the subsets above the node on the tree. Here our predictions, `ypred`, are the mean of the Y values included in the node. If there is an optimal and allowable split, [1] then the chosen variable, `var`, and the RSS_{node}, `dev`, are recorded.[2] The value of the variable in question that acts

[1] Recall that we only allow splits to take place that split the data into two groups, each with more than five members.

[2] It's the convention to call the RSS_{node} the deviance at a node N, but, of course, this only makes sense when the node is a leaf.

Table 3.1: T, a home-grown tree grown on the first four columns and the first 140 rows of D2

var	n	dev	ypred	split.cutleft
X2	140	307025.374636735	60.9097319888865	-6.96100384656169
X2	135	83169.2792831922	43.0483530491652	-3.7891602649377
X4	120	50705.5649592679	31.9334396084353	3.2448237665193
leaf	9	6741.84547650294	78.1476720868562	0
X3	111	29377.3267469149	28.1863396777525	1.7396083892122
X3	22	6346.55869350374	51.2429853154712	2.20767162100899
leaf	15	4912.82259054594	60.1969690603788	0
leaf	7	1433.7361029578	32.0558772906692	0
X4	89	18301.6870406292	22.4869441268558	-0.836982585945597
leaf	56	11890.2428451579	24.9942412092431	0
X3	33	5898.67340773121	18.232136956744	0.0529476269400149
leaf	21	4741.70649830849	21.2119268553199	0
leaf	12	1156.96690942271	13.0175046342361	0
leaf	15	11683.3926459505	131.967660575004	0
leaf	5	90431.4904950311	543.166963361362	0

as the split point is recorded as `split.cutleft`. If there is no split on the node in question, then `var` will be recorded as `<leaf>` and the `dev` value will be the value of RSS_{leaf} at this node.

The tree output is read roughly from top to bottom, with a coda in the middle. The first row corresponds to the first node, or the node that includes the entire data set. The second row is the beginning of the right subtree or the right daughter of the first node. This pattern continues, favoring the right daughter, until a leaf is reached. The left daughter of the first node is found after all of the splits off of the right daughter have finished but is easily identified as the row with a value of `n` that is exactly the difference between the `n` values of the first two rows. In the case where the right daughter contained many more observations of the original data set, there may be a node within the right subtree that contains the same number of observations as the left daughter of the first node. In this case, the left daughter is simply the second row with this property. The pattern of following the right daughter until a leaf is reached continues with the left subtree.

3.1 Breiman et al. Introduce Permuted Variable Importance (1984)

3.1.1 Variable Importance on a Single Tree

Breiman et al. in *Classification and Regression Trees* (1984) propose a method for variable importance for individual trees that stems from their definition of $\tilde{s}$,

Table 3.2: The number of splits on each variable in the tree T.

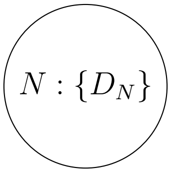
variable	appearances.in.tree
X1	0
X2	2
X3	3
X4	2

a surrogate split. Surrogate splits help Breiman et al. deal with several common problems: missing data, masking, and variable importance. They are defined using logic that resembles that behind random forests.

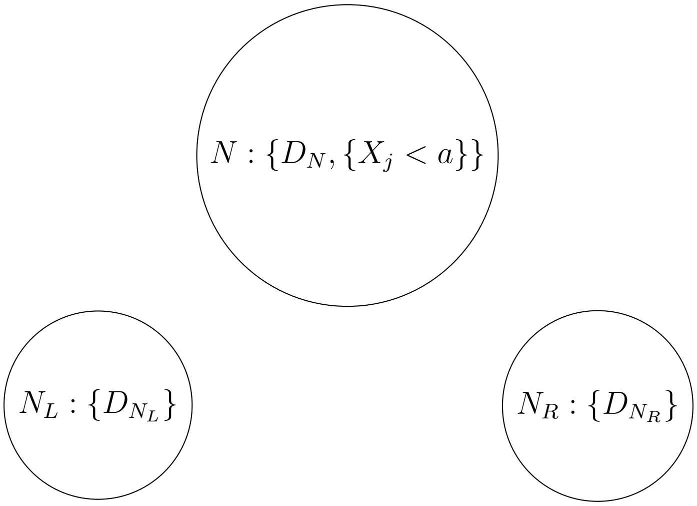
Before we discuss surrogate splits, lets cover an obvious definition of variable importance for a single tree. In the tree represented by table 3.1, define variable importance as the number of splits on each variable. This would allow us to answer the question: how useful (important) was variable X_i in constructing our model for Y? Just by counting the splits on that variable, we would arrive at the following ranking:

There are several downfalls to this method. One, trees are variable. If we were to resample this data and fit another tree, it's likely that this ranking would change. Two, in the case where two variables are close enough to each other that they could act as stand-ins for one another, these rankings are much less interesting. We are lucky in this case to know without doubt that X_1 has a rich relationship with X_2 and the other predictors included in this model (see chapter 2, section 1). This leads us to believe that while X_1 is left out of these rankings, it just as easily could have been included instead of X_2, or one of the other predictors. X_1 had bad luck by not being in this model and it wouldn't make sense to say that the X_1 is the least important predictor of Y when it is very nearly identical to X_2. However, it's possible that the tree algorithm would only pick one of the correlated predictors to be included in a model at a time. Is it possible that we can grasp this relationship by only fitting one tree?

This dilemma is solved by surrogate splits. To set the stage for surrogate splitting, imagine a CART tree, T, fit on some data set D according to the formula $Y \sim X$ where $X_i \in X, \ if i \in 1 : p$. Now say that we're only considering a single node, N, in T. The node N contains the subset of the rows in the original data set D, D_N. D_N is determined by the previous nodes and splits in the tree.



On that node, we have the split on X_j where $X_j < a$. This gives us two daughter nodes to N, N_L and N_R.



The data sets D_{N_L} and D_{N_R} are subsets of D_N and when combined, they equal D_N. They are determined by the rule: if a row of observations has a value of $X_j < a$ then it is a member of D_{N_L}, if the value of X_j in that row is greater than or equal to a then it belongs to D_{N_R}. X_j was chosen to split on in node N because the correlation between the subsets of X_j and Y in D_N was stronger than the correlations between Y and any of the other predictors in that subset of the original data. Imagine, however, that a split on X_i would lead to very similar [3] left and right daughter nodes, even though X_i and Y had a lower correlation than Y and X_j. This would be considered a surrogate split for our original split on N. Now define variable importance for a predictor X_j across the tree T as the decrease in RSS_{node} according to the split on X_j, whether *surrogate* or not. This allows X_j and X_i to share the importance measure, if both X_j and X_i would have provided a similar, valuable split on node N. In *Classification and Regression Trees*, Breiman et al, outline several potential problems with this method that they do not attempt to solve. First, that this is only one of a number of reasonable ways to define variable importance. Second, the variable importances for variables $X_1, .., X_p$ can be affected by outliers or random fluctuations within the data. (Ch 5.3). The second problem is mitigated when we move from single trees to a random forest, but the first is a problem with variable importance in general.

3.1.2 Variable Importance for a Random Forest

One way to define variable importance for a random forest follows directly from Breiman et al's definition for a single tree. Recall that each tree in a random forest is fit to a bootstrapped sample of the original observations. To estimate the test error, therefor, no cross validation is needed - each tree is simply tested against the test set of observations that were not in that tree's initial training set. Additionally, we may be interested in defining variable importance for a predictor X_j by considering the predictive capabilities of the other $p-1$ predictors. Recall: a random forest is a set of

[3] This is intentionally vague. The level of similarity considered "similar enough" depends on the properties of the data set and there's no guarantee that suitable surrogate splits exist. (@bibCART)

trees that are de-correlated with each other because at each split on each tree, less than half of the predictors are not even considered as possible candidates for splitting. To estimate the importance of X_j given the other variables X_{-j} and their relationship with Y, we can consider the "test" RSS of the set of trees that did not ever split on X_j. These values are then averaged over the subset forest that did not include X_j. A large value would imply that in trees that included X_j, the predictive capabilities were increased.

To formalize that idea, let's refer to the set of trees that did not consider X_j, $T^c_{x_j}$. Now, $T^c_{x_j} \subset R_f$, the random forest. The subset of the original data that will be tested on each tree, t, is $\bar{B}^t$. The dimensions of $\bar{B}^t$ are ν_t x p, where p is the number of predictors and $\nu \leq n$. The number of trees in $T^c_{x_j}$ is μ where $\mu \leq ntree$

Now, base variable importance is:

$$VI_\alpha(X_j, R) = \sum_{t \in T^c_{x_j}} \frac{1}{\nu_t} RSS(t, \bar{B}_t)$$

However, this method poses some problems. Namely, while variable importance for random forests is more stable than for the variable importance values for CART (this is because the model is less variable in general), it is lacking the traditional inferential capabilities of other regression models. In an effort to derive a p-value for variable importance values, Breiman (2001b) describes a *permuted variable importance* or VI_β that does not utilize $T^c_{x_j}$.

Algorithm 2 Permuted Variable Importance for Random Forests, VI_β

```
Fit a random forest, $R_f$ on the data set $D$ estimating the model $Y \sim X_1, ..., X_p$.
for each $X_i \in X_1, ..., X_p$ do
    for each $T \in R$ do
        Calculate: $\Phi_o = \frac{1}{\nu_t} RSS(T, \bar{B}^t)$
        Permute $X_i$. Now find $\Phi^* = \frac{1}{\nu_t} RSS(T, \bar{B}^*_t)$
        The difference between these values, $\Phi^* - \Phi_o$, is the variable importance for $X_j$ on $t$,
    end for
end for
```

In other words, we start with one tree in the random forest, T_u, and one variable, X_j, where $1 \leq u \leq ntree$ and $1 \leq j \leq p$. There are two subsets of the original data associated with T_u, one is the subset used to generate the tree B^t and the other is the rest of the original data set, notated as $\bar{B}^t$. We calculate the residual sum of squares for T_u on the new (to T_u) data, $\bar{B}^t$. Then we alter $\bar{B}^t$ by randomly shuffling X_j. This means that in one row of $\bar{B}^t$, the values are the same as they were before, except the values for X_j may be interchanged with the values in other rows. Then RSS is calculated again and compared with the RSS before the shuffling took place. As each tree in the random forest is fit to a bootstrapped sample of the original data set and splits on a fraction of the possible predictors, the tree-wise computation gives an estimate of the distribution of $VI_\beta(X_j)$.

3.2 Strobl et al Respond (2008)

Strobl et al respond to Breiman's method with one main argument: the null hypothesis implied by the permutation distribution utilized in permuted variable importance is that X_i is independent of Y **and** $X_j \notin X_1, ..., X_p$ so the null hypothesis will be rejected in the case where X_j is independent of Y but not some subset of the other predictors. As correlation among the predictors is very common in data sets that are used for random forests, this is a large problem for Breiman's method. To alleviate this difficulty, Strobl et al propose a permutation scheme under the null hypothesis that X_j given its relationship with the other predictors is independent of Y.

Algorithm 3 Conditional Variable Importance for Random Forests, VI_γ

```
Fit a random forest, $R$ on the data set $D$ fitting the model $Y \sim X_1, ..., X_p$.
for each $t \in R$ do
    Calculate: $\Psi_o = \frac{1}{\nu_t} RSS(t, \bar{B}^t)$
    for each $X_i \in X_1, ..., X_p$ do
        Select $Z \in X_1, ..., X_{i-1}, X_{i+1}, ..., X_p$ to condition on when permuting $X_j$
        Use the cutpoints on each variable in $Z$ to create a grid on $X_j$
        Permute $X_j$ with respect to this grid
        Now find $\Psi^* = \frac{1}{\nu_t} RSS(t, \bar{B}_t^*)$
        The difference between these values, $\Psi^* - \Psi_o$, is the variable importance for $X_j$ on $t$,
    end for
end for
```

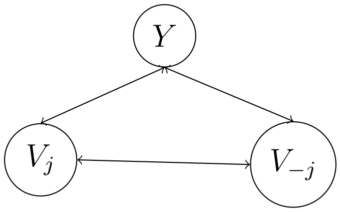
This method is fairly similar to permuted variable importance, but there are a few key differences. Given a tree T_u and a variable X_j, first we find the out of bag RSS, then we permute. In this case, however, our permutations or shuffling of X_j is not always done blindly. If X_j has no (or low) empirical correlation with each of its fellow predictors, then X_j is shuffled exactly as in permuted variable importance. If that is not the case, then we select the set, Z, of the predictors with the strongest empirical correlation [4] to X_j. Recall that our tree T_u contains many nodes, and each node contains a subset of the data along with a split that determines the subsets of the daughter nodes. We feed the out of bag sample for T_u into T_u and look at all the subsets of data in nodes that split on a predictor in Z. This time when we shuffle X_j it will only be in these subsets. The union of these subsets is called $\bar{B}_t^*$ and is used to calculate the permuted RSS.

[4] @bibstrobl2008 recommends constructing the set Z from prior information about the data or as the set of predictors where each one has empirical correlation greater than or equal to .2 with X_j.

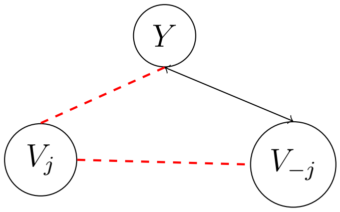
Chapter 4

INFFOREST Variable Importance

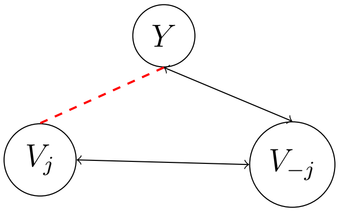
Variable importance measures must contend with the following relationships in the data, for each V_j in V: first, there is the relationship between Y and V_j, then the relationship between V_j and the other predictors, V_{-j}, and finally the relationship between Y and the other predictors V_{-j}.



In permuted variable importance, the null hypothesis is that Y is independent of V_j, regardless of the relationship between V_j and V_{-j}. By permuting V_j blindly and then calculating the RSS, the relationship between V_j and Y and the relationship between V_j and V_{-j} are broken. The other variables, V_{-j} are not permuted and since the RSS is calculated using the original model fitting $Y \sim V$, the relationship between Y and the V_{-j} is maintained.

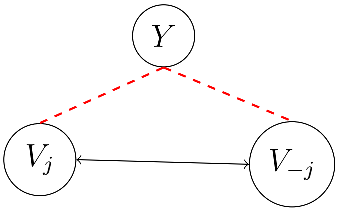


In conditional variable importance, however, the null hypothesis that is tested is that V_j is independent of Y given the relationship between V_j and V_{-j} and the relationship between V_{-j} and Y. Therefore, the permutations on V_j are done in such a way that the relationship between V_j and Y are broken, while approximately maintaining the relationships between V_j and V_{-j} and V_{-j} and Y.



This permutation structure has the following implications: 1. If V_j has a weak relationship with Y but a strong relationship with V_{-j}, the conditional variable importance value will be high. 2. If V_j is approximately independent of V_{-j}, but a good predictor of Y, then the conditional variable importance of V_j will be high.

Conditional variable importance provides a method of statistical inference on random forests, but it does not answer the same question as statistical inference on linear models. Namely, what is the relationship between V_j and Y given the other V_{-j} variables in the model? The INFFOREST method of variable importance permutes under the null hypothesis that V_j given V_{-j} is independent of Y. This leads us to break the relationships between Y and the V_j en mass, according to the respective structure of V_j and V_{-j}.



This has the following conclusions: 1. If V_j is a good predictor of Y but independent of the rest of the predictors, the INFFOREST variable importance will be high. 2. If V_j is a good predictor of Y but is heavily correlated with at least one of the other predictors, the INFFOREST variable importance will be low. It is assumed that the information gained from adding V_j to the model could be gained from one of the other predictors.

4.1 Algorithm and Implementation

The INFFOREST variable importance is a method of permuted variable importance not unlike that of conditionally permuted variable importance. INFFOREST values are calculated at the tree level, using the partitions on V_j from a tree created to predict the model $V_j \sim V_1, ..., V_{j-1}, V_{j+1}, ..., V_p$. This auxiliary tree is fit by considering all $p-1$ predictors at each split and so may be quite large or quite small depending on the richness of the correlation structure around V_j. The auxiliary tree is also fit using the OOB sample, $\hat{B}_t$, for the tree at question. If the auxiliary tree results in a single leaf (i.e. there are no splits), then $\hat{B}_t$ is permuted blindly, without partitions. If the auxiliary tree results in two leaves, there will be two partitions on $\hat{B}_t$ to permute $\hat{B}_t$ within, and so on. After permuting $\hat{B}_t$ within these partitions, the RSS is calculated for that tree using the permuted dataset. The absolute difference of the RSS after permutation and the RSS before permuting the sample is INFFOREST variable importance for that tree.

Note that for this reason, the INFFOREST variable importance is always greater than or equal to zero, and is standardized by the max INFFOREST variable importance value given by that tree. As the variable importance values are calculated for each tree for each variable, once the method is completed there is a distribution of potential variable importance values for V_j, one for each tree. These distributions may or may not be normal, depending on the multicolinearity of the predictors. The INFFOREST variable importance algorithm works as follows:

Algorithm 4 INFForests, $VI_{inf}(R)$

```
Fit a random forest, $R$ on the dataset $D$ fitting the model $Y \sim V_1, ..., V_p$.
for each $V_i \in V_1, ..., V_p$ do
    for each $t \in R$ do
        Calculate: $\Xi_o = \frac{1}{\nu_t} RSS(t, \bar{B}_t)$
        Calculate a tree $T_i$ that predicts $V_i \sim V_1, ..., V_{i-1}, V_{i+1}, ...V_p$ using the subset of the observations used to fit $t$
        Permute $\bar{B}_t$ with respect to the set of partitions $P_{xi}$ from $T_i$.
        if thenT_i is a leaf
            Permute the $V_i$ values blindly with respect to no partitions. Set $\bar{B}^*_t$ to be equal to $\bar{B}_t$ except $V_i$ is permuted.
        end if
        Now find $\Xi^* = \frac{1}{\nu_t} RSS(t, \bar{B}^*_t)$
        The difference between these values, $\Xi^* - \Xi_o$, is the variable importance for $V_i$ on $t$
    end for
    Test the null hypothesis that 0 is the likely value of $\frac{1}{\nu_t} RSS(t, \bar{B}^*_t)$ using the distribution of values of $\Xi^*$ gathered from each tree in $R$
end for
```

INFFOREST variable importance operates under the null hypothesis that Y is independent of V_J given the correlation structure of V_j and the other V_{-j} predictors, or that the true INFFOREST variable importance for V_j is 0. The alternative hypothesis is that Y and V_j are not independent given the correlation structure of V_j and the other predictors or that the INFFOREST variable importance for V_j is greater than zero. After INFFOREST values have been computed for the entire forest, they are treated as samples from the population of possible INFFOREST values for V_j given the random forest R_f, a significance test can be run under the null hypothesis stated above.

Recall the data sets D_1 and D_2, introduced in chapter 2. Both datasets contain 12 predictors and one response, where there is some type of correlation structure between the first four variables. In D_1, this structure is linear, and in D_2 it is not. The median INFFOREST values for a random forest fit to a subset of each data set are presented in the following tables. The p-values listed in table 4.1 are the observed proportion of INFFOREST values for that variable that are above zero.

Table 4.1: Median INFFOREST variable importance values from random forests of 100 trees fit on the first simulated data set

	median	Coefficient	pval
V1	0.3252743	5	0.0000000
V2	0.3325859	5	0.0000000
V3	0.3044358	2	0.0000000
V4	0.2967535	0	0.0000000
V5	0.4687643	-5	0.0505051
V6	0.5529826	-5	0.0202020
V7	0.4586385	0	0.1414141
V8	0.5320091	0	0.0505051
V9	0.5216065	0	0.0303030
V10	0.5001833	0	0.1717172
V11	0.4885839	0	0.1111111
V12	0.4824380	0	0.1414141

Table 4.2: Out of bag RSS values for random forests on data set D1 with mtry equal to 4, 7, or 12

mtry	RSS
4	21770.58
7	21485.52
12	24640.66

At significance level $\alpha = .05$, we would reject the null hypothesis that the true INFFOREST value for these variables is zero for variables $V_1, ..., V_4$ and V_6. We have found that in the context of the other predictors these predictors have a significant relationship with Y.

The parameters of a random forest are the data, the formula, the number of trees (*ntree*), and the number of variables to consider as possible candidates at each split (*mtry*). We'll investigate how INFFOREST variable importance on D_1 changes as the last two parameters are altered. First, random forests will be created for the following values of *mtry*: $mtry = 4, 7, 12$. The random forest behind table 4.1 was created using $mtry = 7$. These forests will all have the same number of trees, $ntree = 50$.

In figure 4.1 the shading represents the most common values of INFFOREST for that variable. The shaded area represents a 95% confidence interval around the average value. As *mtry* approaches the full number of predictors for the data set, the distributions become less variable. The parameter *mtry* is generally taken to be between one third and just over one half of the predictors in the data set, but should be optimized.

While the significance of the variables changed slightly for different values of *mtry*, in applications this may not be anything more than a practical inconvenience. The

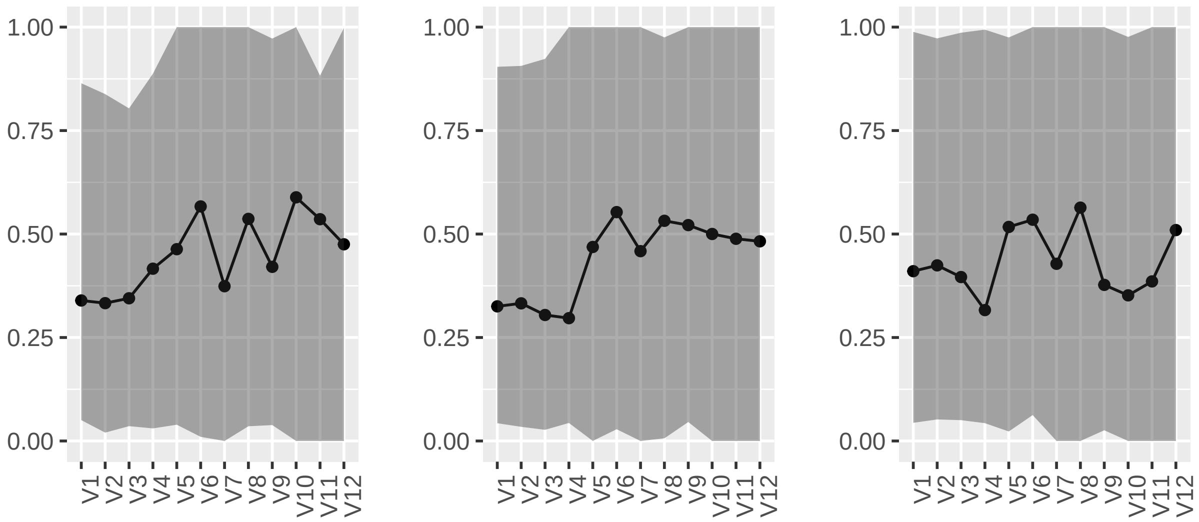


Figure 4.1: Distribution of INFFOREST variable importance values for data set D1 in random forests with mtry = 4,7,12.

value of *mtry* that optimizes the tree is $mtry = 7$ as seen in table 4.2, and this is the model that would be used both for prediction and for inference.

Unlike *mtry* which must be optimized for each data set manually, traditionally the number of trees to fit in a forest follows a simpler rule: more trees are better. There is some risk that, given a high enough value of *mtry*, after a certain number of trees are fit, they will begin to be more correlated with each other than they would have been otherwise. To demonstrate the consistency in the INFFOREST variable importance significance testing, two new random forests will be constructed. Each will follow the same formula as the random forest from table 4.1, $Y \sim V$, and will have $mtry = 7$, and $ntree = 50, 200$ (the random forest where $ntree = 100$ was fit in the previous example).

```
Warning in noder(y = yhatl, xs = xsl, mtry, min): NAs introduced by
coercion
```

The p-values for each random forest with $50, 100, 200$ trees vary slightly, but ultimately the same variables are considered significant. INFForest variable importance allows for consistent statistical inference with random forests.

Table 4.3: INFFOREST Variable Importance for random forests with 50, 100, and 200 trees

	median	pval	median	pval	median	pval
V1	0.39	0.00	0.33	0.00	0.32	0.01
V2	0.36	0.00	0.33	0.00	0.35	0.00
V3	0.30	0.00	0.30	0.00	0.29	0.00
V4	0.29	0.00	0.30	0.00	0.36	0.00
V5	0.52	0.06	0.47	0.05	0.51	0.03
V6	0.52	0.00	0.55	0.02	0.55	0.03
V7	0.35	0.18	0.46	0.14	0.49	0.14
V8	0.55	0.02	0.53	0.05	0.51	0.06
V9	0.51	0.06	0.52	0.03	0.50	0.07
V10	0.48	0.16	0.50	0.17	0.48	0.16
V11	0.52	0.10	0.49	0.11	0.50	0.08
V12	0.53	0.14	0.48	0.14	0.50	0.16

Conclusion

Chapter 5

INFFOREST Comparisons With Other Methods

As discussed in the beginning of chapter 4, each type of permuted variable importance, permuted, conditionally permuted, and INFFOREST, operates on a slightly different null hypothesis. This explains the differences in the results when each method is run on the same random forest.

To compare the results, a random forest was generated on the first 200 rows of the data set D_1, following the formula $Y \sim V$. This random forest considered 7 of the 12 predictors at each split and contained 200 trees. Then the INFFOREST, conditional permuted, and permuted variable importance distributions were calculated for each variable. These distributions are represented below in figure 5.1.

Mimicking the construction of the plot in chapter 4, the ribbon surrounding the average values is the 95% confidence interval constructed around the average importance values. The first main difference visible between INFFOREST and the other methods in figure 5.1 is that the median INFFOREST values are above zero, even for the variables that are not considered significant. The random forest is conducted in such a way that even predictors that may not be important in the overall model are important in a few trees. These are three different methods, following three different permutation schemes, and they are based on three different null hypotheses. Table 5.1 demonstrates the implications of each permutation scheme.

Variable Importance Method	Ties between X_j and Y	Ties between X_j and X_{-j}	Ties betwee
Permuted	Broken	Broken	Main
Conditional Permuted	Broken	Maintained	Main
INFFOREST	Broken	Maintained	Bro

Table 5.1: The permutation structure in each variable importance method functions to break one or more ties between the predictors and the response.

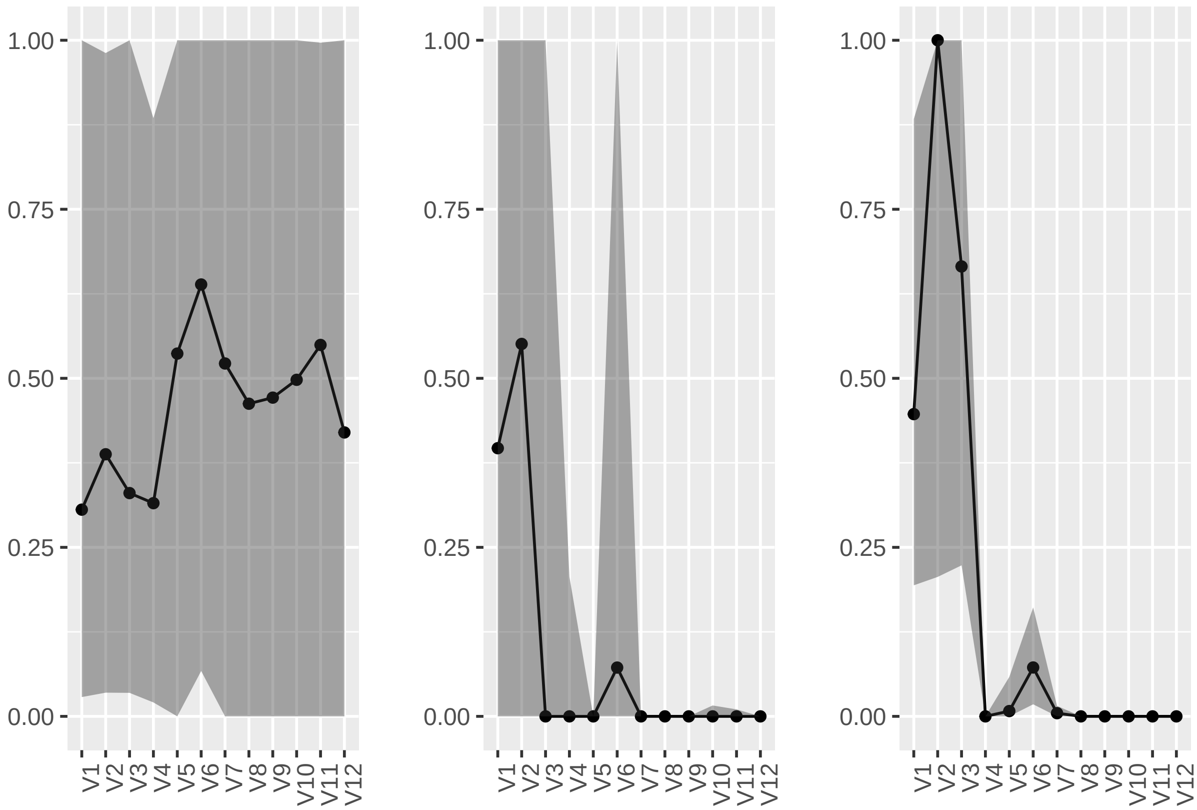


Figure 5.1: Median Values of INFFOREST, Conditionally Permuted, and Permuted Variable Importance

References