A New Model-Based Notion of Variable Importance

Author 1
Author 1 address

Author 2
Author 2 address

Abstract

Keep to 200 words or less and don't use references or mathematical markup.

Features of this approach:

- can be used with any supervised learning algorithm, including those that do not have a natural way of defining variable importance;
- model-based;
- takes other predictors into account;
- can define interaction importance;
- embarrassingly parallel and flexible.

1 Introduction

In the age of "big data", we are often confronted with the task of extracting knowledge from large databases. For this task we turn to various statistical learning algorithms which, when tuned correctly, have state-of-the-art predictive performance. However, having a model that predicts well is only solving part of the problem. It is still necessary to extract information about the relationships uncovered by the learning algorithm. For instance, we often want to know which predictors, if any, are important by assigning some type of variable importance score to each feature. Once a set of "important" features has been identified, the next step would be to summarize the functional relationship between each feature, or subset thereof, and the outcome of interest. However, since most statistical learning algorithms are "black box" models, extracting this information is not always straightforward. Fortunately, some learning algorithms have a natural way of defining variable importance.

In a binary decision tree, at each node t, a single predictor is used to partition the data into two homogenous groups. The chosen predictor is the one that maximizes some measure of improvement $\widehat{i}_t$. The relative importance of predictor x is just the sum of the squared improvements over all internal nodes of the tree for which x was chosen as the partitioning variable:

$$\operatorname{imp}(x) = \sum_{t=1}^{J-1} \widehat{i}_t^2 I\left(v\left(t\right) = l\right), \tag{1}$$

where J is the number of terminal nodes in the tree; see **??** for details.

1.1 Variable importance metrics

1.1.1 Model-based approaches

- tree-based methods;
- linear and generalized linear models;
- MARS algorithm;
- Neural networks (specifically, the two approaches used later).

1.1.2 Filter-based approaches

For classification problems, the area under the ROC curve (AUC) is used as a measure of variable importance. For two class problems, a series a split points are chosen (not unlike a decision tree) for each predictor to predict each class and generate an ROC curve from which the AUC statistic is taken. For classification problems involving more than two classes, the problem is broken down into every possible one-vs-one problem, and the same procedure is applied. The maximum AUC for each variable is used as a measure of variable importance.

For regression, a parametric or nonparametric model is fit between each predictor and the outcome. For parametric models, some measure depending on the absolute value of the associated t-value is used as a measure of importance. Otherwise, the R^2 statistic is used.

1.2 Partial dependence plots

Harrison and Rubinfeld [1978] were among the first to analyze the well-known Boston housing data. One of their goals was to find a housing value equation using data on median home values from $n = 506$ census tracts in the suburbs of Boston from the 1970 census; see Harrison and Rubinfeld [1978, Table IV] for a description of each variable. The data violate many classical assumptions like linearity, normality, and constant variance. Nonetheless, Harrison and Rubinfeld—using a combination of transformations, significance testing, and grid searches—were able to find a reasonable fitting model ($R^2 = 0.81$). Part of the payoff for there time and efforts was an interpretable prediction equation which is reproduced in Equation (2).

$$\begin{aligned}\widehat{\log(MV)} = {} & 9.76 + 0.0063RM^2 + 8.98 \times 10^{-5} AGE - 0.19 \log(DIS) + 0.096 \log(RAD) \\ & - 4.20 \times 10^{-4} TAX - 0.031 PTRATIO + 0.36 (B - 0.63)^2 - 0.37 \log(LSTAT) \\ & - 0.012 CRIM + 8.03 \times 10^{-5} ZN + 2.41 \times 10^{-4} INDUS + 0.088 CHAS \\ & - 0.0064 NOX^2.\end{aligned} \tag{2}$$

Nowadays, many supervised learning algorithms can fit the data automatically in seconds—typically with higher accuracy. The downfall, however, is some loss of interpretation since these algorithms typically do not produce simple prediction formulas like Equation (2). These models can still provide insight into the data, but it is not in the form of simple equations. For example, quantifying predictor importance has become an essential task in the analysis of "big data", and many supervised learning algorithms, like tree-based methods, can naturally assign variable importance scores to all of the predictors in the training data.

While determining predictor importance is a crucial task in any supervised learning problem, ranking variables is only part of the story and once a subset of "important" features is identified it is often necessary to assess the relationship between them (or subset thereof) and the response. This can be done in many ways, but in machine learning it is often accomplished by constructing *partial dependence plots* (PDPs); see Friedman [2001] for details. PDPs help visualize the relationship between a subset of the features (typically 1-3) and the response while accounting for the average effect of the other predictors in the model. They are particularly effective with black box models like random forests and support vector machines.

Let $\boldsymbol{x} = \{x_1, x_2, \ldots, x_p\}$ represent the predictors in a model whose prediction function is $\widehat{f}(\boldsymbol{x})$. If we partition $\boldsymbol{x}$ into an interest set, $\boldsymbol{z}_s$, and its compliment, $\boldsymbol{z}_c = \boldsymbol{x} \setminus \boldsymbol{z}_s$, then the "partial dependence" of the response on $\boldsymbol{z}_s$ is defined as

$$f_s(\boldsymbol{z}_s) = E_{\boldsymbol{z}_c}\left[\widehat{f}(\boldsymbol{z}_s, \boldsymbol{z}_c)\right] = \int \widehat{f}(\boldsymbol{z}_s, \boldsymbol{z}_c)\, p_c(\boldsymbol{z}_c)\, d\boldsymbol{z}_c, \tag{3}$$

where $p_c(\boldsymbol{z}_c)$ is the marginal probability density of $\boldsymbol{z}_c$: $p_c(\boldsymbol{z}_c) = \int p(\boldsymbol{x})\, d\boldsymbol{z}_s$. Equation (3) can be estimated from a set of training data by

$$\bar{f}_s(\boldsymbol{z}_s) = \frac{1}{n} \sum_{i=1}^{n} \widehat{f}(\boldsymbol{z}_s, \boldsymbol{z}_{i,c}), \tag{4}$$

where $\boldsymbol{z}_{i,c}$ $(i = 1, 2, \ldots, n)$ are the values of $\boldsymbol{z}_c$ that occur in the training sample; that is, we average out the effects of all the other predictors in the model.

Constructing a PDP (4) in practice is rather straightforward. To simplify, let $\boldsymbol{z}_s = x_1$ be the predictor variable of interest with unique values $\{x_{11}, x_{12}, \ldots, x_{1k}\}$. The partial dependence of the response on x_1 can be constructed as follows:

1. For $i \in \{1, 2, \ldots, k\}$:
   (a) Copy the training data and replace the original values of x_1 with the constant x_{1i}.
   (b) Compute the vector of predicted values from the modified copy of the training data.
   (c) Compute the average prediction to obtain $\bar{f}_1(x_{1i})$.
2. Plot the pairs $\{x_{1i}, \bar{f}_1(x_{1i})\}$ for $i = 1, 2, \ldots, k$.

Algorithm 1: A simple algorithm for constructing the partial dependence of the response on a single predictor x_1.

Algorithm 1 can be quite computationally intensive since it involves k passes over the training records. Fortunately, the algorithm can be performed in parallel quite easily (more on this in Section **??**). It can also be easily extended to larger subsets of two or more features as well.

For illustration, we will use a corrected version of the Boston housing data analyzed in Harrison and Rubinfeld [1978]; the data are available from Statlib at `http://lib.stat.cmu.edu/datasets/boston_corrected.txt`. Using the R package `randomForest` [REFERENCE], we fit a random forest with tuning parameter $m_{try} = 6$ (chosen using 5-fold cross-validation) and 1000 trees. The model fit is reasonable, with an *out-of-bag* (pseudo) R^2 of 0.89. The variable importance scores are displayed in Figure **??**. Both plots indicate that the percentage of lower status of the population (`lstat`) and the average number of rooms per dwelling (`rm`) are highly associated with the median value of owner-occupied homes (`cmedv`). They also indicate that the proportion of residential land zoned for lots over 25,000 sq.ft (`zn`) has little association with `cmedv`.

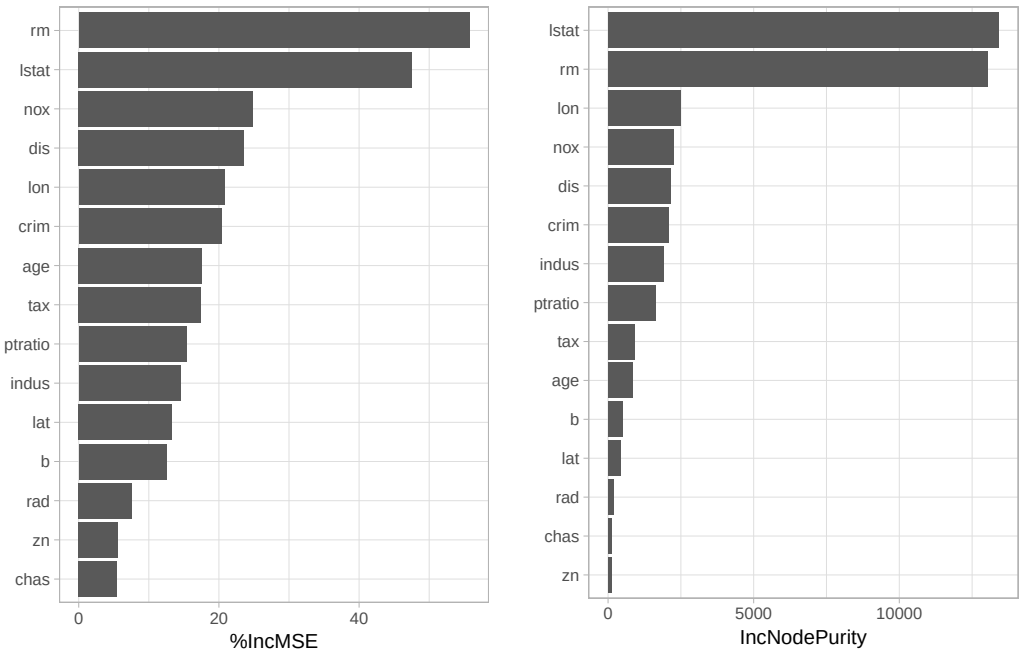

Figure 1: TBD.

The partial dependence functions for these three variables are displayed in Figure **??**. Notice how the PDP for `zn` is essentially flat. It is this notion of "flatness" which we will use to define our variable importance measure.

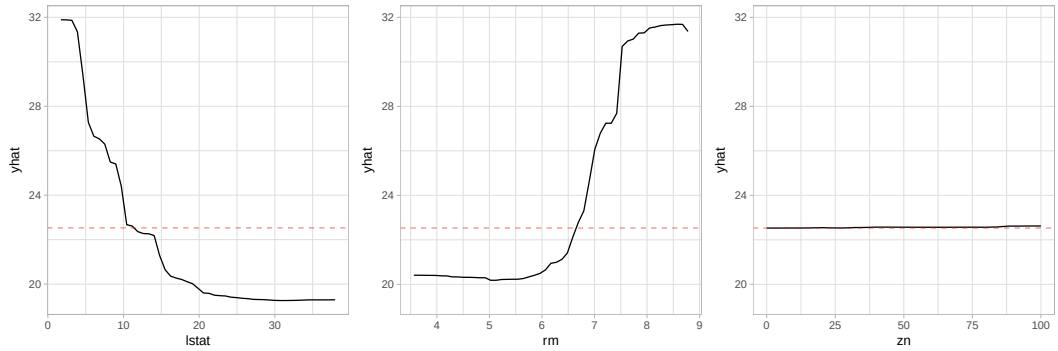

Figure 2: TBD.

2 Prediction-based variable importance

The PDP for `zn` in Figure **??** is relatively flat indicating that `zn` does not have much influence on the predicted median home value. . One might conclude that any variable for which the PDP is "flat" is likely to be less important than those predictors whose PDP varies across a wider range of the response.

Our notion of variable importance is based on any measure of the "flatness" of the partial dependence function. In general, we define

$$\text{imp}\left(x\right) = \text{flatness}\left(\bar{f}_s\left(\boldsymbol{z}_s\right)\right), \tag{5}$$

where flatness is any measure of "flatness" of the curve. A simple and effective measure to use is the sample standard deviation. Based on Algorithm 1, our importance metric for predictor x_1 is simply

$$\text{imp}\left(x_1\right) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left[\bar{f}_1\left(x_{1i}\right) - \frac{1}{k}\sum_{j=1}^{k}\bar{f}_1\left(x_{1i}\right)\right]^2}. \tag{6}$$

2.1 Detecting interaction effects

The same idea can be applied to finding interactions. Essentially a strong interaction effect of x_1 and x_2 on $\mathcal{Y}$ would imply that $\text{imp}\left(x_1, x_2\right)$ be roughly constant when either x_1 or x_2 is held constant while the other varies.

3 Examples: a simple function of ten variables

For illustration, we use one of the regression problems described in Friedman (1991) and Breiman (1996). Inputs are 10 independent variables uniformly distributed on the interval $[0, 1]$; only 5 out of these 10 are actually used. Outputs are created according to the formula

$$\mathcal{Y} = 10\sin\left(\pi x_1 x_2\right) + 20\left(x_3 - 0.5\right)^2 + 10x_4 + 5x_5 + \epsilon, \tag{7}$$

where $\epsilon \sim \mathcal{N}\left(0, \sigma\right)$.

We fit a simple neural network with eight hidden units and a weight decay of 0.01. These parameters were chosen using 5-fold cross-validation to maximize the root mean squared error (RMSE). The network diagram is displayed in Figure **??** below.

Variable importance plots are displayed in Figure **??**. Notice how the Garson and Olden algorithms incorrectly label some of the features not in the true model as "important".

For comparison, we fit a random forest [REFERENCE] with 1000 trees to the same data. Random forests (and other tree-based methods) have a natural way of defining variable importance. For details, see [REF]. Figure **??** displays two types of importance measures obtained from the fitted random forest.

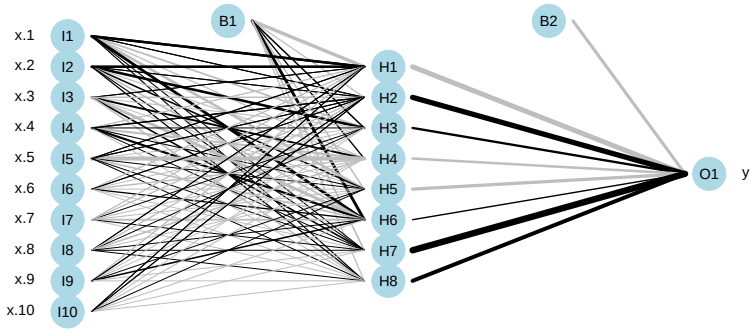

Figure 3: Diagram of the neural network fitted to the Friedman 1 data set.

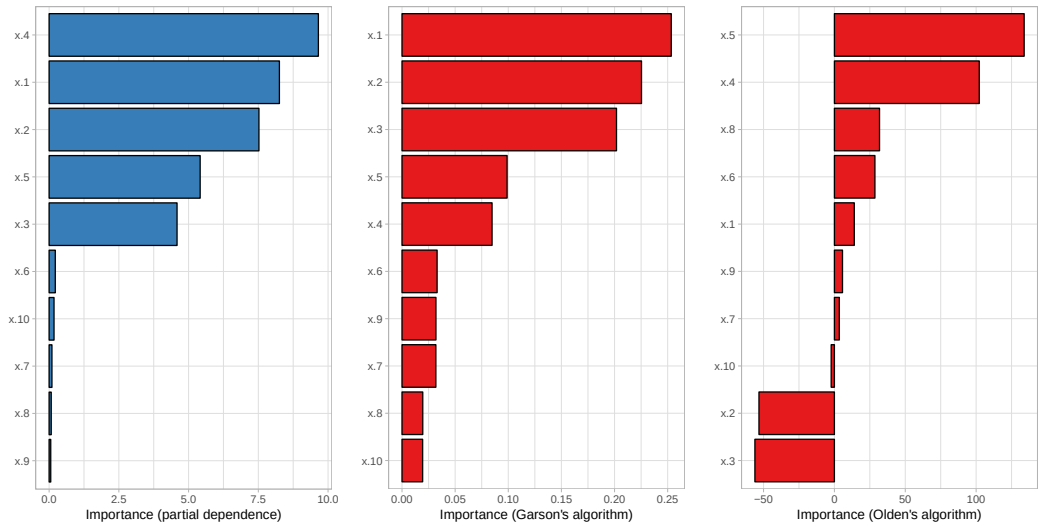

Figure 4: Variable importance plots for the neural network in Figure **??**. *Left*: partial dependence-based algorithm. *Middle*: Garson's algorithm. *Right*: Olden's algorithm.

4 Discussion

References

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29:1189–1232, 2001. URL https://doi.org/10.1214/aos/1013203451.

David Harrison and Daniel L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978. URL https://doi.org/10.1016/0095-0696(78)90006-2.

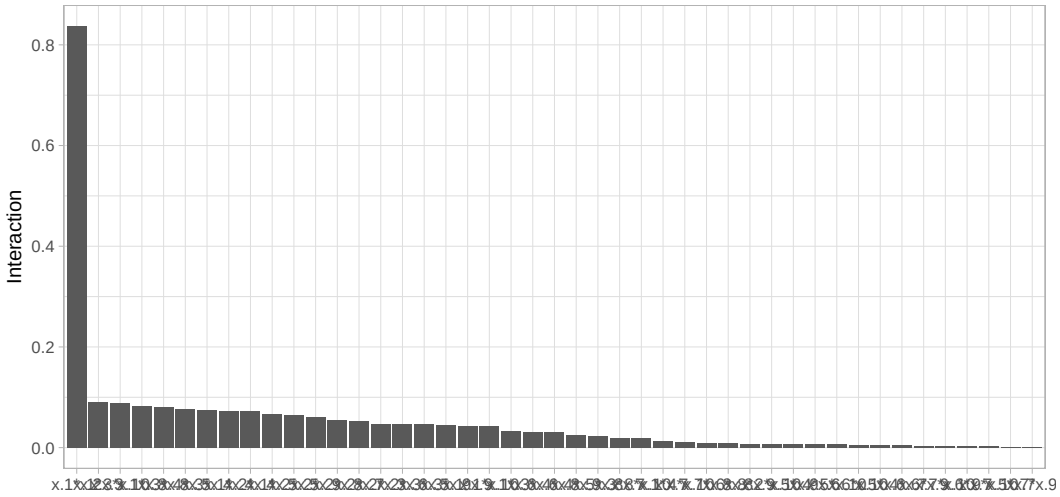

Figure 5: TBD.