

Visualizing Variable Importance and Variable Interaction Effects in Machine Learning Models

Alan Inglis*

Hamilton Institute, Maynooth University

and

Andrew Parnell*

Hamilton Institute, Insight Centre for Data Analytics, Maynooth University

and

Catherine B. Hurley

Department of Mathematics and Statistics, Maynooth University

October 12, 2021

Abstract

Variable importance, interaction measures, and partial dependence plots are important summaries in the interpretation of statistical and machine learning models. In this paper we describe new visualization techniques for exploring these model summaries. We construct heatmap and graph-based displays showing variable importance and interaction jointly, which are carefully designed to highlight important aspects of the fit. We describe a new matrix-type layout showing all single and bivariate partial dependence plots, and an alternative layout based on graph Eulerians focusing on key subsets. Our new visualizations are model-agnostic and are applicable to regression and classification supervised learning settings. They enhance interpretation even in situations where the number of variables is large. Our R package `vivid` (variable importance and variable interaction displays) provides an implementation.

Keywords: Model visualization; Model explanation; Black-box

*Alan Inglis and Andrew Parnell's work was supported by a Science Foundation Ireland Career Development Award grant 17/CDA/4695. In addition Andrew Parnell's work was supported by: an investigator award (16/IA/4520); a Marine Research Programme funded by the Irish Government, co-financed by the European Regional Development Fund (Grant-Aid Agreement No. PBA/CC/18/01); European Union's Horizon 2020 research and innovation programme InnoVar under grant agreement No 818144; SFI Centre for Research Training in Foundations of Data Science 18CRT/6049, and SFI Research Centre awards I-Form 16/RC/3872 and Insight 12/RC/2289.P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

1 Introduction

Visualization is a key tool in understanding statistical and machine learning models. In this paper we present new visualizations to serve two main goals, namely improved model understanding and interpretation. Our new visualizations are based on variable¹ importance and interaction measures, and partial dependence plots. A variable importance value is used to express (in a scalar quantity) the degree to which a variable affects the response value through the chosen model. A variable interaction is a scalar quantity that measures the degree to which two (or more) variables combine to affect the response variable. Variable importance and variable interaction (henceforth VImp and VInt; together VIVI) are widely used in many fields to understand and explain the behaviour of a model. In biology they are used to examine gene-gene interactions (e.g. Wang et al., 2012). In high-energy physics VImp can be an important tool in high dimensional feature selection processes (e.g. Gleyzer and Prosper, 2008). In econometrics they are common tools to evaluate interaction behaviour (e.g. Balli and Sorensen, 2010).

Traditional methods of displaying VImp or VInt use variants of line or bar plots, see for example Molnar (2019). However, in variable importance plots there is relatively little emphasis on displaying how pairs of interacting variables may be important in a model. This can be a hindrance to model interpretation, especially if a variable has low importance but a high interaction strength. The inclusion of interacting terms in a model has been shown to affect the prediction performance (Oh, 2019). However, as shown in Wei et al. (2015a), for high-dimensional models that are governed mainly by interaction effects, the performance of certain types of permutation-based variable importance measures will decrease and thereby produce low values of importance. Consequently, viewing the VInt and VImp together provides a more complete picture of the behaviour of a model fit.

Our new displays present VInt and VImp jointly in a single plot. We allow for seriation so that variables are reordered with those exhibiting high VIVI grouped together. This

¹We use the term ‘variable’ throughout to denote the input to a statistical and machine learning model as this seems to be the most common parlance. Other terms commonly used include: feature, predictor, explanatory variable, independent variable, etc

assists in interpretation and is particularly useful as the number of variables becomes large. Furthermore, we make use of filtering, so less influential variables can be removed. For our network displays we use graph clustering to group together interacting variables.

Partial dependence plots (PDP) were introduced by Friedman (2000) to show how the model’s predictions are affected by one or two predictors. In addition to the above we propose a new display which shows all pairwise partial dependence plots in a matrix-type layout, with a univariate partial dependence plot on the diagonal, similar to a scatterplot matrix. With this display the analyst can explore, at a glance, how important pairs of variables impact the fit. Once again, careful reordering of the variables facilitates interpretation.

Our final display takes the filtering of all pairwise partial dependence plots a step further. We select only those pairwise partial dependence plots with high VInt, and display an Eulerian path visiting these plots by extending the zigzag display algorithm of Hofert and Oldford (2020). We call this a zen-partial dependence plot (ZPDP).

These new visualizations can be used to explore machine learning models more thoroughly in an easily interpretable way, providing useful insights into variable impact on the fit. This is demonstrated by practical examples. In each plot careful consideration is given to various aspects of the design, including color choices, optimising layouts via seriation, graph clustering, and Euler paths for the ZPDP. Filtering options limit the plots to variables deemed relevant from VImp or VInt scores. Our new displays are appropriate for supervised regression and classification fits, and are model and metric agnostic in that no particular model fit nor importance method is prescribed. The methods described here are implemented in our R package `vivid` (Inglis et al., 2021).

The organisation of the paper is as follows. In Section 2 we discuss the concepts of VImp and VInt. Then we describe our new heatmap and network displays of joint variable importance and interaction and demonstrate these on an example. In Section 3 we discuss our new layouts for collections of partial dependence plots, either in a matrix format or zig-zag layout and show their application. In Section 4 we use our new methodology to explore a machine learning fit from a larger dataset. Finally in Section 5, we offer some

concluding discussion.

2 Visualizing variable importance and interaction

We begin with a non-exhaustive review of the concepts of VImp and VInt. Though the visualizations we present are agnostic to the measures used to determine these scalar quantities, some degree of understanding is helpful in interpreting the later plots. We then describe our new visualizations and their design principles and provide illustrations.

2.1 Measuring variable importance

A VImp is a scalar measure of a variable’s influence on the response. Many techniques have been proposed to calculate variable importance, depending on the type of model. The term ‘influence’ here may encompass changes in the mean response or that of higher order uncertainty. In our work we focus exclusively on changes in the mean. For a wider review of variable importance techniques, and the different goals that a variety of approaches may achieve see Wei et al. (2015b).

Much of the initial work in VImp focused on estimating the partial derivative of the response with respect to one or two input variables (Frey and Patil, 2002). This is a global VImp measure when the model is linear, but perhaps less useful (though still potentially interesting) in non-linear models where it is often defined as a local importance measure. In high dimensional settings these methods can be discretized across a hyper-cube to allow for the identification of, e.g., linearity in a non-linear model (Helton and Davis, 2002). Due to their local behaviour, we do not incorporate them into our visualizations below.

Some VImp measures arise naturally out of a model structure. The most familiar would be those based on summary statistics created from regression models, such as standardized coefficient values, (partial) correlation coefficients, and R^2 . Many of these can be extended to non-linear models such as generalized additive models (Wood, 2000), or projection pursuit regression (Friedman and Stuetzle, 1981). R^2 in particular seems useful as a VImp measure, as it can be defined for a wide variety of statistical models and can be decom-

posed into main and potentially high order interaction effects, yielding a VInt measure in addition.

Similarly, other model-structure based methods arise out of now standard machine learning techniques. Random forests, for example, involves the use of the Gini coefficient, and the reduction in mean square error, to catalog a variable’s influence on the ‘purity’ of a model output (Breiman, 2001). This can naturally be seen as a VImp measure. Others have extended these approaches to introduce conditional and permutation VImp statistics which aim to reduce the bias that may occur due to variable collinearity (for example, see Hothorn et al., 2006).

Conditional variants of permutation variable importance were proposed by Strobl et al. (2008) for a random forest. This method examines splits of the trees in a random forest and permutes the variables within these subgroups (see Section 2.2 for more details). Whereas Strobl et al. (2008) relied on the splitting of trees to determine the subgroups, a model-agnostic approach was introduced by Molnar et al. (2020) that builds the subgroups explicitly from the conditional distribution of the variables. In tree-based models such as CART and random forests, Ishwaran et al. (2010) proposed a VImp called minimal depth, which is the proximity of a variable to the root node, averaged across all trees.

Permutation importance was introduced by Breiman (2001) and is measured by calculating the change in the model’s predictive performance after a variable has been permuted. The algorithm works by initially recording the model’s predictive performance, then, for each variable, randomly permuting a variable and re-calculating the predictive performance on the new dataset. The variable importance score is taken to be the difference between the baseline model’s performance and the permuted model’s performance when a single feature value is randomly shuffled. A similar agnostic permutation concept was developed by Fisher et al. (2019). This method permutes inputs to the overall model instead of permuting the inputs to each individual ensemble member. In situations where no embedded variable importance is available, a model-agnostic approach such as permutation importance is a useful tool.

In theory any of the above global importance measures could be used in our visual-

izations. However, providing code for each would be a daunting task. Instead we take a pragmatic approach and use the associated VImp measure with the model that we are fitting. In cases where there is no such obvious method, we use the Fisher et al. (2019) agnostic permutation approach discussed above to measure VImp.²

2.2 Measuring variable interaction

Measuring variable interaction in a machine learning model can be considerably harder than estimating marginal importance. Even the definition of the term ‘interaction’ is disputed (Boulesteix et al., 2015). We focus here on bivariate interaction only, though higher order interactions may certainly be present in many situations. Friedman and Popescu (2008) state that a function $f(\mathbf{x})$ exhibits an interaction between two of its variables x_k and x_l if the difference in the value of a function $f(\mathbf{x})$ as a result of changing the value of x_k depends on the value of x_l . That is, the effect of one independent variable on the response depends on the values of a second independent variable. Often, an interaction is taken to mean a simple multiplication of two (continuous) variables (e.g. Berrington de González and Cox, 2007), though in machine learning models much more complex relationships can exist. We follow the definition of Friedman and Popescu (2008) by considering an interaction to be estimated from the difference between joint and marginal partial dependence; a full mathematical definition is given below. Even this definition should not be used without care, as in the case of highly correlated or potentially confounding variables.

In tree-based models such as CART and random forests, much focus has been on measuring interactions via the structure of trees (e.g. Ishwaran et al., 2010; Deng, 2019). If two variables are used as splits on the same branch, this might initially appear like a measure of interaction. However, this does not separate out the interaction from potential marginal effects. The problem is partially overcome by permuting the variables (individually for a VImp, jointly for VInt), to assess the effect on prediction performance. The resulting VInt measure is known as pairwise prediction permutation importance (Wright et al., 2016).

For models that are not tree-based, or when a model-agnostic measure is required, a

²In our implementation, any available VImp may be used.

variety of other methods can be used. Many of these are based on the idea of partial dependence (Friedman, 2000). The partial dependence measures the change in the average predicted value as specified feature(s) vary over their marginal distribution. The partial dependence of the model fit function g on predictor variables S (where S is a subset of the p predictor variables) is estimated as:

$$f_S(\mathbf{x}_S) = \frac{1}{n} \sum_{i=1}^n g(\mathbf{x}_S, \mathbf{x}_{C_i}) \quad (1)$$

where C denotes predictors other than those in S , $\{\mathbf{x}_{C_1}, \mathbf{x}_{C_2}, \dots, \mathbf{x}_{C_n}\}$ are the values of \mathbf{x}_C occurring in the training set of n observations, and $g()$ gives the predictions from the machine learning model. For one or two variables, the partial dependence functions $f_S(\mathbf{x}_S)$ are plotted (a so-called PDP) to display the marginal fits.

Friedman's H -statistic or H -index (Friedman and Popescu, 2008) is a VInt measure created from the partial dependence by comparing the partial dependence for a pair of variables to their marginal effects. Squaring and scaling gives a value in the range $(0, 1)$:

$$H_{jk}^2 = \frac{\sum_{i=1}^n [f_{jk}(x_{ij}, x_{ik}) - f_j(x_{ij}) - f_k(x_{ik})]^2}{\sum_{i=1}^n f_{jk}^2(x_{ij}, x_{ik})} \quad (2)$$

where $f_j(x_j)$ and $f_k(x_k)$ are the partial dependence functions of the single variables and $f_{jk}(x_j, x_k)$ is the two-way partial dependence function of both variables, where all partial dependence functions are mean-centered.

The H -statistic requires $O(n^2)$ predictions for each pair of variables, and so can be slow to evaluate. Sampling from the training set will reduce the time, though at a cost of increasing the variance of the partial dependence estimates and the H -statistic.

When the denominator in Equation 2 is small, the partial dependence function for variables j and k is flat, and small fluctuations in the numerator can yield spuriously high H -values. Biased partial dependence curves will also lead to inflated H . This occurs in some machine learning approaches which exhibit regression to the mean in their one-way partial dependencies. Furthermore biased partial dependence curves are a particular problem in the presence of correlated predictors. These issues with the H -statistic seem to be not widely known by practitioners (though see Apley and Zhu, 2020), and we provide a short illustration of these problems in the appendix.

In our visualizations throughout this paper, we use the square-root of the average un-normalized (numerator only) version of Friedman’s H^2 for calculating pairwise interactions:

$$H_{jk} = \sqrt{\frac{1}{n} \sum_{i=1}^n [f_{jk}(x_{ij}, x_{ik}) - f_j(x_{ij}) - f_k(x_{ik})]^2} \quad (3)$$

This reduces the identification of spurious interactions and provides results that are on the same scale as the response (for regression). It does not, however, remove the possibility that some large H -values arise from correlated predictor variables.

We follow the convention of Hastie et al. (2009) by using the logit scale for both the partial dependence and in calculation of the H -statistic when fitting a classification model with a binary response. If the response is multi-categorical a near-logit is used, defined as:

$$g_k(x) = \log[p_k(x)] - \frac{1}{K} \sum_{k=1}^K \log[p_k(x)] \quad (4)$$

where $k = 1, 2, \dots, K$ and $p_k(x)$ is the predicted probability of the k -th class. PDPs of $g_k(x)$ from Equation 4 can reveal the dependence of the log-odds for the k -th class on different subsets of the input variables.

Alternatives to the H -statistic have been suggested, which could be used in place of the the H -statistic in our visualizations. Hooker (2004) uses a functional ANOVA construction to decompose the prediction function into variable interactions and main effects. Greenwell et al. (2018) suggested a partial dependence-based feature interaction which uses the variance of the partial dependence function as a measure of importance of one variable conditional on different fixed points of another.

2.3 Heatmap visualization with seriation

Traditionally, variable importance and interaction are displayed separately, with variable interaction itself spread over multiple plots, one for each variable. We direct the reader to Chapter 8 of Molnar (2019) for examples. We propose a new heatmap display showing VImp on the diagonal and VInt on the upper and lower diagonals. The benefit of such a display is that one can see which variables are important as individual predictors and at the same time see which pairs of variables jointly impact on the response. It also facilitates

easy comparison of multiple model fits, which is far less straightforward with separate VImp and VInt displays.

We illustrate the heatmap using a random forest fit to a college applications data set (American Statistical Association, 1995), with Enroll (i.e., the number of new students enrolled) as the response. The data was gathered from 777 colleges across the U.S. and contains 18 variables ranging from economic factors (such as room and board and book costs) to the number of applications received and accepted. As some of the variables are skewed they are log-transformed prior to building the model. The data was split 70-30 into training and test sets. A value of $R^2 = 0.96$ was obtained for the test set. All plots were made from the training set. See the supplementary materials for a description of the data and transformations.

Figure 1 shows our heatmap with two different orderings. Figure 1(a) has the vari-

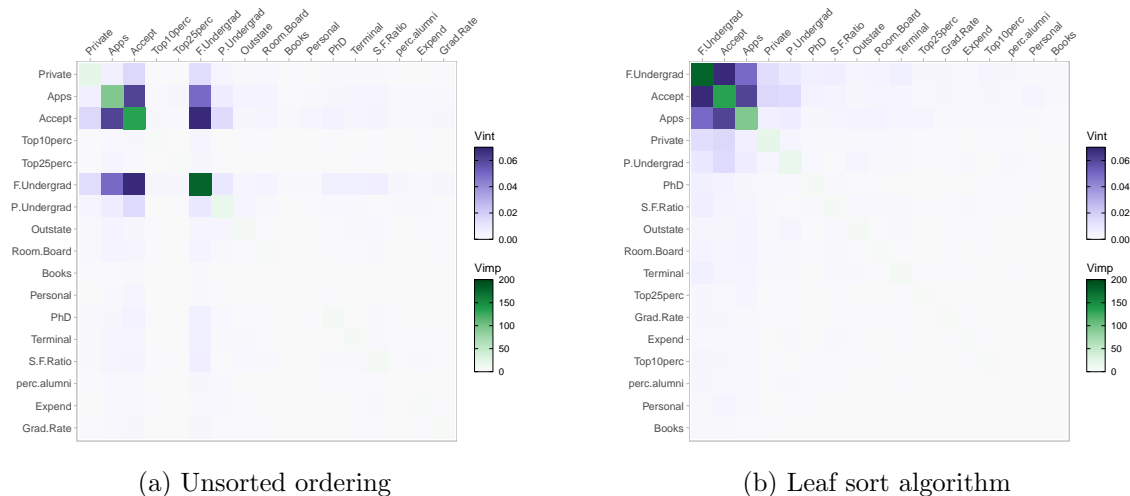


Figure 1: Heatmap from random forest of college application data. In (a) variables are in original order. In (b), the heatmap is re-ordered using leaf sort. In (b) we can see three important and mutually interacting variables, F.Undergrad, Accept and Apps.

ables in their original order, while Figure 1(b) uses the leaf-sorting algorithm (described below). The purple color scale used on the off-diagonal shows the Friedman’s H -statistic values (un-normalized) with deeper purple indicating a higher VInt. Similarly the green color scale on the diagonal represents the level of VImp, here measured using an embedded

approach supplied by the random forest (in this case, the increase in node purity). We use colorblind-friendly, single-hued sequential color palettes from Zeileis et al. (2020) going from low to high luminance in both cases, designed to draw attention to high VInt/VImp variables. From the improved ordering in Figure 1(b), there are three clearly important and potentially interacting variables, F.Undergrad (the number of full-time undergraduate students), Accept (the number of applicants accepted), and Apps (the number of applications received), with F.Undergrad having the largest VImp when predicting Enroll.

Many authors have investigated the benefits of re-ordering (also known as seriation) for graphical displays, see for example Hurley (2004), Hahsler et al. (2008) and Earle and Hurley (2015). The benefits of reordering the variables in Figure 1(b) are clear. The right-hand plot lends itself to easy interpretation whereas the left-hand plot does not.

Most seriation algorithms start with a matrix of dissimilarities or similarities between objects and produce an ordering where similar objects are nearby in the sequence. Our goal here is a little different. As well as placing mutually interacting variables nearby in the sequence, we would like to bring important variables or pairs of variables to the start of the sequence so that the most relevant portion of the heatmap will be in the top-left corner.

We use the leaf sort seriation algorithm from Earle and Hurley (2015). This uses hierarchical clustering followed by a sorting step. Let v_i be a measure of variable importance and s_{ij} be the interaction measure between variables i and j . Treating the matrix of interactions as a similarity matrix, we first construct a hierarchical clustering. This produces a dendrogram, resulting in a variable ordering where high-interacting variables are nearby. Using this ordering in a heatmap generally brings high interactions close to the diagonal, but ignores our goal of placing important variables early in the sequence. For the sorting step we calculate for each variable a combined measure of its importance and contribution to the interactions, defining these scores as:

$$w_i = \lambda_1 v_i + \lambda_2 \max_{j \neq i} s_{ij}.$$

Here λ_1 and λ_2 are scaling parameters to account for the fact that variable importance and interaction are not measured in the same units. Reasonable choices of λ_1 and λ_2 rescale

dendrogram 系统树图

leaf sort seriation

v_i 是 feature importance

s_{ij} 是 features interaction.

将 s_{ij} 的矩阵当作相似度矩阵

① 先构建层级簇，得到 dendrogram，以及一个顺序

② 按这个顺序画热度图，将高交互的 features 放在一角

③ 将 v_i 和 s_{ij} 正规化，使它们在同一比重相加，得出 w_i

④ 用 w_i 重排 dendrogram 叶子

importance and interaction to, say, unit range or unit standard deviation. We use unit range by default. As there are many possible dendrogram orderings consistent with a hierarchical clustering of the matrix of interactions, the sorting step re-orders the dendrogram leaves so that the weights w_i are generally decreasing.

Sorting the variables in this way will achieve our goals of placing high-interacting pairs of variables nearby in the sequence, while simultaneously pulling predictors with high importance and interaction to the top-left of the heatmap, leaving less relevant predictors to the bottom-right. Setting $\lambda_2 = 0$ or $\lambda_1 = 0$ produces plots which sort by descending VImp or max VInt respectively. For all future heatmap plots, we use the sorting strategy discussed above to optimize the arrangement of variables. After using seriation to re-order the heatmap variables, filtering can be applied to limit the display to the most important or interacting variables; this strategy is especially useful when there are large numbers of predictors.

The heatmap display can be further used to compare different model fits. In Figure 2 we compare the random forest to a k-nearest neighbours (kNN) fit. In the left panel

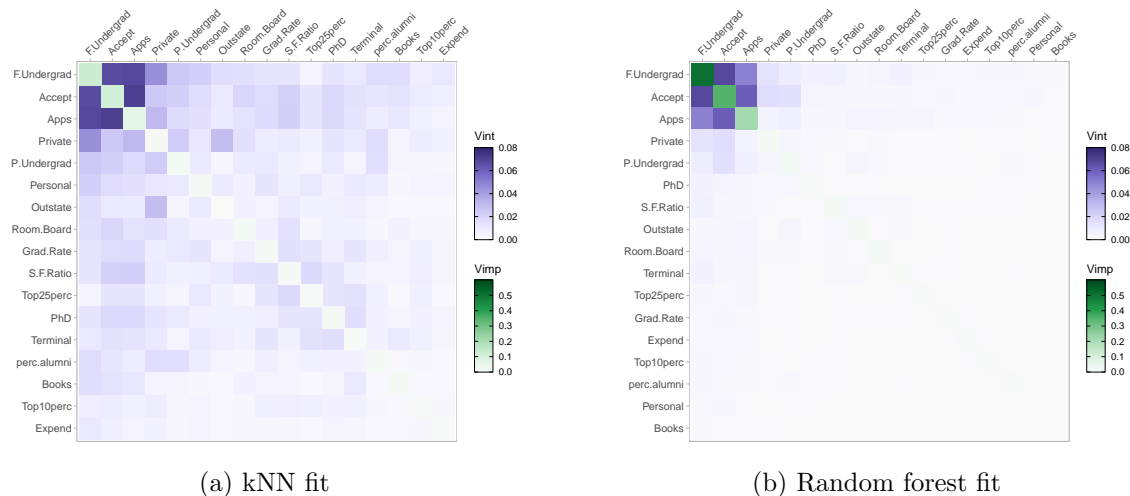


Figure 2: A comparison of a kNN and random forest fit on the college application data. Both fits identify F.Undergrad as the most important variable as well as having similar mutual interactions between F.Undergrad, Accept and Apps. The kNN fit identifies many more moderate interactions between variables, especially concerning the variable Private

of Figure 2 we have a heatmap of a kNN fit (with $k = 7$ neighbours considered), while the right panel shows the random forest heatmap. To make a direct comparison of the heatmaps, we swap the embedded VImp measures that are available from a random forest fit and instead measure importance with an agnostic permutation approach that allows direct comparison of both the kNN and random forest models. Furthermore, we set both heatmaps to use the same color scale for the VImp and VInt values.

We see in Figure 2 that both the random forest and kNN fit identify F.Undergrad as the most important variable for predicting the number of students enrolled. The top three variables are identical in both models, though the VImp values are much smaller in general across the kNN fit (e.g. the measured VImp for F.Undergrad for the kNN and random forest fits are 0.16 and 0.6 respectively). Both fits show mutual interactions between F.Undergrad, Accept and Apps. However, the kNN fit also suggests a moderate interaction between Private (i.e., whether the university was public or private) and F.Undergrad, which appears somewhat lower in the random forest fit. As Private has a relatively low VImp in both model fits, a simple VImp screening could miss its relevance to the fit. We note though, that this kNN-random forest comparison is for the sake of illustration only, as in this instance the kNN fits poorly by comparison with the random forest, having a test mean square error (MSE) over three times bigger.

2.4 Network visualization

As our second offering for displaying VIVI, we propose a network plot that shares similar benefits to the heatmap display but differs from it by giving a visual representation of the magnitude of the importance and interaction values not only via color but also by the size of the nodes and edges in a graph. In this plot, each variable is represented by a node and each pairwise interaction is represented by a connecting edge. See Figure 3(a) for an example. The color scales were chosen to match that used in the heatmap, with node size and color luminance increasing with variable importance. Similarly, edge width and color reflects the strength of the VInt. By default we choose a radial layout to display the variables (although this can be changed according to preference) and use the same seriation