JDSSV

Journal of Data Science, Statistics, and Visualisation

February 2024, Volume IV, Issue I. *doi: 10.52933/jdssv.v4i1.79*

Visualisations for Bayesian Additive Regression Trees

Alan Inglis
Maynooth University

Andrew Parnell
Maynooth University

Catherine Hurley
Maynooth University

Abstract

Tree-based regression and classification has become a standard tool in modern data science. Bayesian Additive Regression Trees (BART) has in particular gained wide popularity due its flexibility in dealing with interactions and non-linear effects. BART is a Bayesian tree-based machine learning method that can be applied to both regression and classification problems and yields competitive or superior results when compared to other predictive models. As a Bayesian model, BART allows the practitioner to explore the uncertainty around predictions through the posterior distribution. In this paper, we present new visualisation techniques for exploring BART models. Our work is designed as a visualisation add-on to some of the more popular BART R packages available, namely **BART**, **dbarts**, and **bartMachine**. We construct conventional plots to analyse a model's performance and stability as well as create new tree-based plots to analyse variable importance, interaction, and tree structure. We employ Value Suppressing Uncertainty Palettes (VSUP) to construct heatmaps that display variable importance and interactions jointly using colour scale to represent posterior uncertainty. Our approach is implemented in the R package **bartMan** (BART Model ANalysis).

Keywords: Model visualisation, Bayesian Additive Regression Trees, posterior uncertainty, variable importance, uncertainty visualisation.

1. Introduction

Bayesian Additive Regression Trees (BART) is a non-parametric sum-of-trees-based ensemble method developed by Chipman et al. (2010), who demonstrate its excellent predictive performance. BART has been applied in diverse areas such as risk management (Liu et al. 2015), proteomics (Hernández et al. 2015), and avalanche forecasting (Blattenberger and Fowles 2014). The BART method has also been extended into many areas, such as survival analysis (Sparapani et al. 2016) and causal inference (Hill 2011; Hahn et al. 2020). BART now enjoys widespread use due to its competitive performance against other tree-based predictive models, such as random forest (Breiman 2001) and gradient boosted trees (Friedman 2000). BART models are used for making predictions for both binary and continuous response variables and can be fit using a variety of R packages, such as; **BayesTree** Chipman and McCulloch (2016), **dbarts** (Dorie 2020), **bartMachine** (Kapelner and Bleich 2016), and **BART** (Sparapani et al. 2021). Implementations are also available in other programming languages, such as the `PyMC` (Salvatier et al. 2016) or `bartpy` (Coltman 2020) in Python. In our work we focus only on the **dbarts**, **bartMachine**, and **BART** packages as they are some of the more popular BART fitting packages in R.

Visualisation techniques are crucial for black-box regression tree ensemble models, such as random forests and gradient boosting machines, in addition to BART. These models, while powerful and flexible, can suffer from interpretability issues, making it challenging for practitioners to understand the model results. Visualisation techniques help address this limitation by providing a more in-depth look at the nature of predictions. The previously mentioned R-packages provide only a limited set of visualisations, and in certain cases, they require the user to create their own visualisations by extracting relevant information from the fitted model, such as the tree structure (which indicates tree stability and variability as the algorithm iteratively builds the posterior) and measures of variable importance/interaction or model performance (which can assist in comprehending the model fit). Our goal is to create visualisations and to streamline this process for the three aforementioned R-language BART packages by creating a suite of plots for and evaluating both the BART fit and the posterior distribution. In our work, we present three novel visualisations for examining different aspects of a BART fit, as well as a suite of visualisations for examining model performance. Various aspects of a BART model can be assessed (e.g., variable importance and variable interaction) by analysing the structure of the trees used in the model. However, our approach goes beyond many standard machine learning visualisation techniques by allowing for uncertainty in the posterior to propagate into the diagrams.

Examining the posterior can provide valuable insights into the uncertainty surrounding the estimated importance of each variable, and similarly for the interaction scores of variable pairs (for example, see Hahn et al. (2020) or Deng et al. (2022)). In linear regression, models the standard tool for measuring variable importance is the regression coefficient, which would always be presented with uncertainty to gauge the level of confidence. In our work, we transfer the concept of variable importance with associated uncertainty to complex machine learning models. In the Bayesian setting, the posterior distribution of the variable importance/interaction values provide an assessment of the associated uncertainty in that particular measure.

One of the more challenging aspects of model visualisation is the depiction of uncertainty. Due to the Bayesian inferential framework and full likelihood specification of BART, the uncertainty around predictions is easy to quantify through examination of the posterior distribution. This makes BART an ideal candidate for visualising uncertainty. However, how we choose to represent this uncertainty may have an impact on how the model is analysed and how our audience interprets the findings. This issue has been well studied in areas that regularly deal with uncertainties in data (e.g., Pang et al. 1997; Brodlie et al. 2012). For example, error bars, confidence intervals, or quantile intervals are common tools used to display uncertainty. However, these tools cannot be universally applied to all situations where displaying the uncertainty is necessary, such as in heatmaps or point clouds. When using point clouds to map data over many iterations, 95% confidence ellipses can be used to encircle points. An example of this can be seen in Section 3.3.

Methods for producing visualisations of importance and interaction for standard machine learning models can be found in Inglis et al. (2022). However, in Bayesian models, it is important to include the uncertainties that arise as part of the calculation of a full joint posterior distribution. Our first new display uses a method called Value Suppressing Uncertainty Palettes (Correll et al. 2018), which allows for both the value and the uncertainty to be displayed in a single plot. Traditional methods for displaying a value and uncertainty simultaneously require a 2D bivariate map, conventionally displayed as a square (for example, see Robertson and O'Callaghan 1986; Teuling et al. 2011). However, due to the large colour-space of 2D bivariate maps, the ability to distinguish between two different visual aspects can become challenging. VSUPs improve on this method by using an arc to assign colours and blend together data values with high uncertainty so that values become more distinguishable as the uncertainty decreases. This reduction of the visual colour-space helps to both distinguish between low and high uncertainty and promotes caution when the uncertainty is high (Correll et al. 2018). Hastie et al. (2009) note that when two or more variables interact, their individual effects on the response may be reduced by the presence of the other variable, potentially making some variables appear less important than they really are (see; Inglis et al. (2022) for an example). Our heatmap display provides the advantage of allowing users to identify both the important individual predictors and the pairs of variables that jointly affect the response whilst taking their associated uncertainty into account, thereby mitigating any potential interpretation issues.

By examining the trees in a BART model we can learn about the stability and variability of tree structures as the algorithm iterates to build the posterior. For our second offering, we display new tree-based plots that focus attention on certain aspects of the model fit in an intuitive way. We provide space-saving layouts as well as providing various sorting/filtering methods and colouring options. When combined with ordering techniques, we provide easy to use tools which aid in highlighting interesting aspects of the model fit, such as variable importance or common interactions.

For our third display, we employ multidimensional scaling (MDS) plots, which are a common method for graphically displaying relationships between objects in multidimensional space (Torgerson 1952). Objects that are similar appear closer on the graph, whereas objects objects that are less similar are farther away. MDS can be used to reduce the number of dimensions in high-dimensional data as well as to interpret dis-

similarities as graph distances. We construct an MDS display of a BART fit and extend it to display the uncertainty. For each iteration of the BART fit, we perform MDS on proximities and rotate each plot to match a particular target iteration. From this we get a point cloud, where a confidence ellipse is used to encircle each observation. With this display the analyst can explore, for example, outliers that may require further investigation.

Aside from our three main novel visualisations, we include a selection of standard diagnostic plots, such as trace, residual, and overall model fit plots, that will quickly assess aspects such as convergence and model behaviour. Each of our plots can be run on any of the aforementioned R-packages, despite their differing formats and function arguments. While we make what we believe to be good default choices for the plots we produce, we provide the option to adjust many of the settings. Each aspect of the design of our plots is given careful consideration; we focus on efficient layouts, which includes both clustering and filtering, colour choice, and effectively displaying uncertainty. Our new displays are appropriate for regression and classification fits and are designed to work with the three aforementioned BART packages but, as our approach makes use of individual `R` methods for each BART model fitting package, our proposal could readily be extended to incorporate other BART packages. Our implementation is available as the R package **bartMan** (BART Model ANalysis) which is found at https://github.com/AlanInglis/bartMan.

The outline of this paper is as follows: in Section 2, we describe the formulation of a BART model and provide a brief discussion on how to access variable importance and variable interactions. In Section 3, we describe our new visualisations for assessing variable importance and variable interactions with uncertainty, tree-based analysis, outlier identification with multidimensional scaling, and a selection of enhanced model diagnostic plots on a simple example. In Section 4, we study BART's variable importance and variable interaction methods compared to a model agnostic approach. In Section 5, we demonstrate our new methods on a case study. Finally, in Section 6, we conclude by discussing potential advantages and disadvantages of our approach, as well as potential avenues for further research.

2. Bayesian Additive Regression Trees

2.1. An Introduction to Bayesian Additive Regression Trees

In this section, we provide a brief overview of the BART model to aid the reader in understanding our later visualisations. Those looking for a more complete description should see Chipman et al. (2010). BART is a Bayesian non-parametric model based on an ensemble of trees that can be used for predicting continuous and multi-class responses. Unlike regression models where a linear structure is pre-specified, BART assumes a flexible functional form and so can automatically uncover main and interaction effects, (see Equation 1). Given a continuous response variable y_i with associated

predictors $\mathbf{x}_i$, the BART model, with m trees is expressed as:

$$y_i = \sum_{j=1}^{m} g(\mathbf{x}_i, T_j, M_j) + \epsilon_i, \tag{1}$$

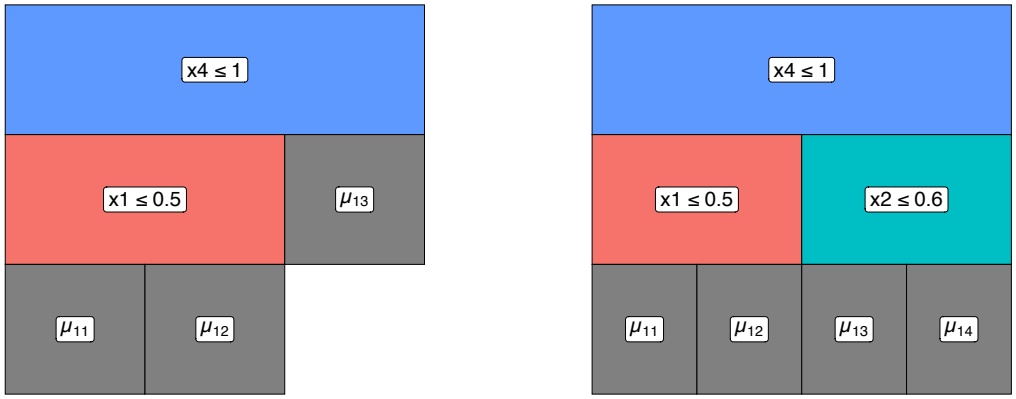
where $\epsilon_i \sim \mathrm{N}(0, \sigma^2)$ and $g(\mathbf{x}_i, T_j, M_j) = \mu_{j\ell}$ is a function that assigns a predicted value for the observations falling into terminal node ℓ of tree j. T_j represents the structure/topology of tree j including the split variables and the values associated with the splits $M_j = (\mu_{j1}, \cdots, \mu_{jb_j})$ represent the set of predicted values at the b_j terminal nodes of the trees. In Chipman et al. (2010), the authors recommend through personal experience a value of $m = 200$, though they mention that the number of trees can also be chosen via cross-validation.

The tree structure T is composed of binary splitting rules of the form $[x_j \leq c]$, where x_j is the variable on which to split and c is the split value. Both quantities are randomly selected and updated as part of the model fitting process, which changes slightly between implementations. The trees are updated at each iteration in a Markov chain Monte Carlo approach where each tree structure is modified by either growing, pruning, changing, or swapping nodes. Growing a tree means that a terminal node is randomly chosen and two new terminal nodes are created, while pruning collapses a pair of terminal nodes to their parent. A splitting rule can also be changed to a different rule, or swapped for another splitting rule in the same tree. In the grow and change moves, a new splitting rule is required, and is proposed by uniformly sampling a splitting variable and a split value though the exact generation of these rules is implementation dependent.

Figure 1 shows an example of the tree structure modifications in action. In Figure 1, a tree, T_1^k, is generated from BART in 4 different instances, where $k = 1, 2, 3, 4$ indicates the iteration number in which the tree is updated. In the full BART model, multiple trees are estimated and the predictions are created from the sum of the μ values across the trees. The tree is displayed as an icicle plot (Kruskal and Landwehr 1983) with the splitting rules (that is, covariates and split points) shown as coloured rectangles, and the terminal nodes $\mu_{j\ell}$ are shown as grey rectangles. Icicle plots were first introduced by Kruskal and Landwehr (1983) as a way to display hierarchical data in a space efficient manner. We use icicle plots to display our tree plots in later sections.

In panel (a) of Figure 1 at iteration 1, observations that satisfy the splitting criterion go left and tree $T_1^{(1)}$ has two internal nodes and three terminal nodes. The grow move is shown going from panel (a) to panel (b), that is $T_1^{(1)}$ to $T_1^{(2)}$. An example prune move would correspond to $T_1^{(2)}$ reverting to $T_1^{(1)}$. In panel (c) we can see the change move as the splitting rule that defines μ_{13} and μ_{14} in $T_1^{(2)}$ is changed. Finally, in (d), the swap move can be seen when comparing the internal nodes of $T_1^{(3)}$ and $T_1^{(4)}$.

As a Bayesian model, BART adopts a set of prior distributions for the tree structure, terminal node parameters, and residual variance. Posterior sampling is based on a Metropolis-within-Gibbs MCMC structure where the trees are sequentially updated through partial residuals. For one MCMC iteration, each tree in the ensemble is modified and then compared to its previous version via a Metropolis-Hastings update. The update involves a marginalised likelihood and the tree prior. Here, the likelihood is marginalised in order to avoid reversible jump MCMC algorithms (Green 1995), which



(a) $T_1^{(1)}$ (b) $T_1^{(2)}$

(c) $T_1^{(3)}$ (d) $T_1^{(4)}$

Figure 1: An example of a tree, T_1^k, generated from BART over $k = 1, 2, 3, 4$ iterations. displayed as an icicle plot with the splitting rules (that is, covariates and split points) shown as coloured rectangles and the terminal nodes $\mu_{j\ell}$ are shown as grey rectangles. In panel (a), tree $T_1^{(1)}$ has two internal nodes and three terminal nodes. Moving from panel (a) to (b) shows the grow move for the tree. Reverting from (b) back to (a) corresponds to a prune move. Panel (c) shows the change move as the splitting rule that defines μ_{13} and μ_{14} in $T_1^{(2)}$ is changed. Finally, in (d) the swap move can be seen when comparing the internal nodes of $T_1^{(3)}$ and $T_1^{(4)}$.

in turn simplifies computation. We highlight that marginal likelihoods, in general, are not analytically easy to obtain, though in BART it is possible due to the use of conjugate priors; see Section 3.1 of Chipman et al. (2010) for further details on the BART full conditionals.

We have described the BART model for a univariate and continuous response variable, however as previously outlined there are BART extensions for many other modelling problems. Our visualisation techniques could in future be extended to these settings.

2.2. Variable Importance and Variable Interaction for BART

Variable importance measures are common and widely used in general machine learning

approaches, for examples see Grömping (2015) or Wei et al. (2015). Variable importance is a measure of a single variable's impact on the response and is used to provide insights into model behaviour. Multiple methods exist for evaluating variable importance, depending on the model; for a comprehensive review of different variable importance techniques see Wei et al. (2015). Measures of variable importance are beneficial for BART models as they give users a more in-depth comprehension of the model's behaviour, allowing them to make more informed decisions and improve the model's performance. Chipman et al. (2010) propose a method called the inclusion proportion to evaluate the variable importance in a BART model from the posterior samples of the tree structures. Their measure of variable importance first calculates for each iteration the proportion of times a variable is used to split nodes considering all m trees, and then averages these proportions across all iterations.

More formally, let K be the number of posterior samples obtained from a BART model. Let c_{rk} be the number of splitting rules using the rth predictor as a split variable in the kth posterior sample of the trees' structure across m trees. Additionally, let $c_{\cdot k} = \sum_{r=1}^{p} c_{rk}$ represent the total number of splitting rules found in the kth posterior sample across the total p variables. Therefore, $z_{rk} = c_{rk}/c_{\cdot k}$ is the proportion of splitting rules for the rth variable, and the average use per splitting rule is given by:

$$\text{VImp}_r = \frac{1}{K} \sum_{k=1}^{K} z_{rk} \tag{2}$$

As noted by Chipman et al. (2010), the value of m can be used to assist with variable selection. A small value of m, for example, will restrict the number of variables that can appear in the trees, and so guide the user as to which are most important, potentially at the expense of a superior model fit. By contrast, as m increases, unimportant or pure noise variables have a greater chance of being included in the trees. This issue occurs because, when m is large, the contribution of each tree to the likelihood is smaller and less important variables can be included by chance in the MCMC updates.

Variable interaction is generally considered as when a pair (or more) of variables jointly impact on the response. In our work, we focus on bivariate interactions only. Note that if the structure of T_j (i.e., a single tree) depends on two variables or more, then T_j may model an interaction, see for example, panel (a) of Figure 1. Following from Chipman et al. (2013), Kapelner and Bleich (2016) suggested a measure of interaction obtained by observing successive splitting rules in each tree. A similar method has been proposed for random forests, which uses the concept of minimal depth (Ishwaran et al. 2010) to assess both importance and interaction strength by examining the position of a variable within the trees (which is implemented in the `randomForestExplainer` package (Paluszynska et al. 2020) in R). Let c_{rqk} be the number of splitting rules using predictors r and q successively (in either order) in the kth posterior sample. Additionally, let $c_{\cdot\cdot k} = \sum_{r=1}^{p} \sum_{q=1}^{p} c_{rqk}$ represent the total number of successive splitting rules found in the kth posterior sample. We follow the convention of Kapelner and Bleich (2016) and we treat the order of successive splits as not important and we sum the r, q counts with the q, r counts. While in random forests the splitting rules are chosen so that they are *the best* given a loss function, the learning process in the BART model takes place through a stochastic search where the splitting rules are *randomly* proposed. That is, they are not optimised to be *the best*, which allows the order of the

splits to be ignored. Therefore, the proportion $z_{rqk} = c_{rqk}/c_{..k}$, provides an estimate of the interaction between variables r and q:

$$\text{VInt}_{rq} = \frac{1}{K}\sum_{k=1}^{K} z_{rqk}. \tag{3}$$

As this method follows a similar technique to evaluating the inclusion proportion, the same pitfalls noted by Chipman et al. (2010) apply, namely that the prior distribution may favour trees containing successive predictor variables where there is no true interaction present if the number of trees is large. For a comparison of both the variable importance and variable interaction methods against a model agnostic approach for evaluating these metrics, see Section 4.

It should be noted that if any of the variables used to build the BART model are categorical, the aforementioned BART packages replace the categorical variables with d dummy variables, where d is the number of factor levels. For some of our plots, the inclusion proportions for variable importance and interaction are then adjusted by aggregating over factor levels. This provides a complete picture of the importance of a factor, rather than that associated with individual factor levels.

Since both the VImp and VInt values are calculated from the full posterior, it is trivial to compute an uncertainty associated with their measurement, simply by storing the importance and interaction calculations per iteration. These can be summarised by the usual means by which posterior distributions are analysed. We will use uncertainty metrics obtained from these distributions in our variable importance and interaction displays of Section 3. In this work we focus on using the structure of trees to assess variable importance and interactions, but other agnostic methods, such as SHAP (Shapley 1997) or those described in Section 4, can be used. For an example of using Shapley values to measure importance in BART models see Schwartz et al. (2022).

3. New Visualisations for BART

To illustrate our new visualisations we use a subset of the iris data (Fisher 1936) where the response is binary and made up of two species (that is, setosa and versicolor). We then fit a BART model to the data using **bartMachine**, using the default setting of 1000 iterations with a burn-in of 250. For simplicity of exposition we set the number of trees to be 20.

We introduce the following visualisations: improved plots of variable importance and interaction which include the uncertainty induced by the posterior distribution of trees; plots of the tree structures which show the splitting variables, the split distribution, and the terminal node values; the ability to identify outlying and influential observations through the terminal node proximity matrix and multi-dimensional scaling; and a set of enhanced model diagnostics for identifying convergence and performance issues.

3.1. Variable Importance and Interaction with Uncertainty

In this section, we present visualisations of the variable importance methods described in Section 2.2. In Figure 2, we show the median of the inclusion proportion as a black

point, with the variables ordered from the largest median importance measure (at the top) and descending. In this case, the 25% to 75% quantile interval extending from each point is displayed as a grey bar. We can see that Petal.Width is the most important variable and that Petal.Length and Sepal.Length have similar inclusion proportions. Sepal.Length importance has a lower degree of uncertainty, as indicated by the relatively small quantile interval, whereas the Petal.Length importance has a larger quantile interval associated with it, and therefore its importance measure should be viewed with a level of caution.

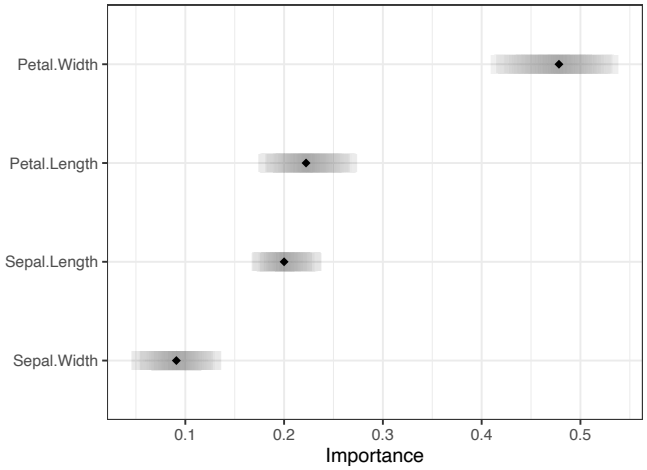
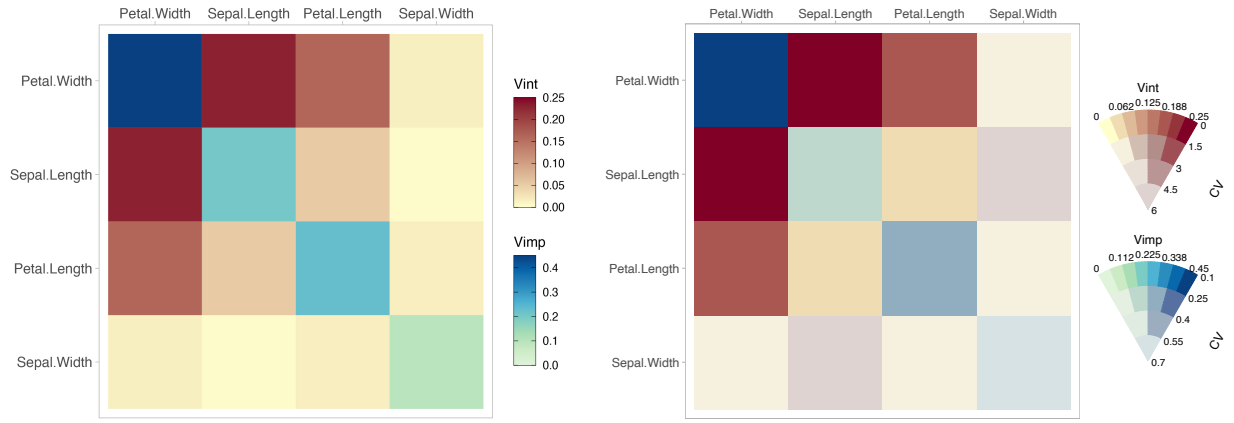


Figure 2: Inclusion proportions for the iris data are shown with the 25% to 75% quantile interval extending from the points. Here Petal.Width is ranked as the most important variable.

In Inglis et al. (2022), the authors propose using a heatmap to display both importance and interactions simultaneously, where the importance values are on the diagonal and interaction values on the off-diagonal. The advantage of such a display is that it allows one to easily identify which variables are relevant as separate predictors while also seeing which variable pairs have high interaction. This method, coupled with the seriation technique described by Inglis et al. (2022), brings predictors with high importance and interaction to the top-left of the heatmap and less relevant predictors to the bottom-right.

Here we adapt the heatmap displays of importance and interactions to include the uncertainty using a VSUP. The colours for the VSUP heatmap were carefully chosen to be distinguishable, colour-blind friendly, and to aid in highlighting high values, while still making the uncertainty prominent. To achieve this, we follow the advice of Strode et al. (2019), who build upon the work of Trumbo (1981), and aim to highlight and focus the reader's attention on the interesting data.

Figure 3 presents a comparison of heatmaps showing the importance and interactions jointly with and without uncertainty. In both heatmaps, the variable importance is displayed on the diagonal and the interactions on the off-diagonal. In (a), we can see that Petal.Width is the most important variable when predicting Species. There also appears to be a strong interaction between Petal.Width and Sepal.Length. In (b), the same values are shown but with a measure of uncertainty included in this case, the coefficient of variation (CV). Other error metrics such as standard deviation



(a) (b)

Figure 3: In (a), the importance values are on the diagonal and interaction values on the off-diagonal. Petal.Width is the most important variable and there is a strong interaction between Petal.Width and Sepal.Length. In (b), the same values values are shown but with the coefficient of variation included by use of a VSUP. Both the importance measure of Petal.Width, and the interaction measure between Petal.Width and Sepal.Length have low coefficient of variation.

can be applied, though in the case of using proportions, larger values tend to have greater uncertainty and so our preference is for the CV. In both (a) and (b), the same method is used to obtain the importance and interaction scores, resulting in comparable scales. Comparing the two plots we observe that in (b), the most important variable, Petal.Width, has a small variation relative to its mean. The Petal.Width and Sepal.Length interaction value has a low coefficient of variation and is consequently highlighted in (b), whereas Petal.Width and Sepal.Width have a low interaction score with relatively high variation.

3.2. Tree-Based Plots

In this section we examine more closely the structure of the decision trees created when building a BART model. Examining the tree structure may yield information on the stability and variability of the tree structures as the algorithm iterates to create the posterior. By sorting and colouring the trees appropriately, we can identify important variables and common interactions between variables for a given iteration. Alternatively we can look at how a single tree evolves through the iteration to explore the fitting algorithm's stability. In Figure 4, we show how a single selected tree changes over all 1000 post burn-in iterations. We use an icicle plot to display the trees. As noted by Barlow and Neville (2001), icicle plots are preferred by users when compared to other methods to display decision trees and use space more efficiently. Additionally, the number of observations within each decision tree node is represented in icicle plots by scaling the node size accordingly. In Figure 4, each parent node is coloured according to the variable with the terminal nodes all coloured a dark grey. A stump is represented by a solid grey square, although stumps can be removed from the plots if desired

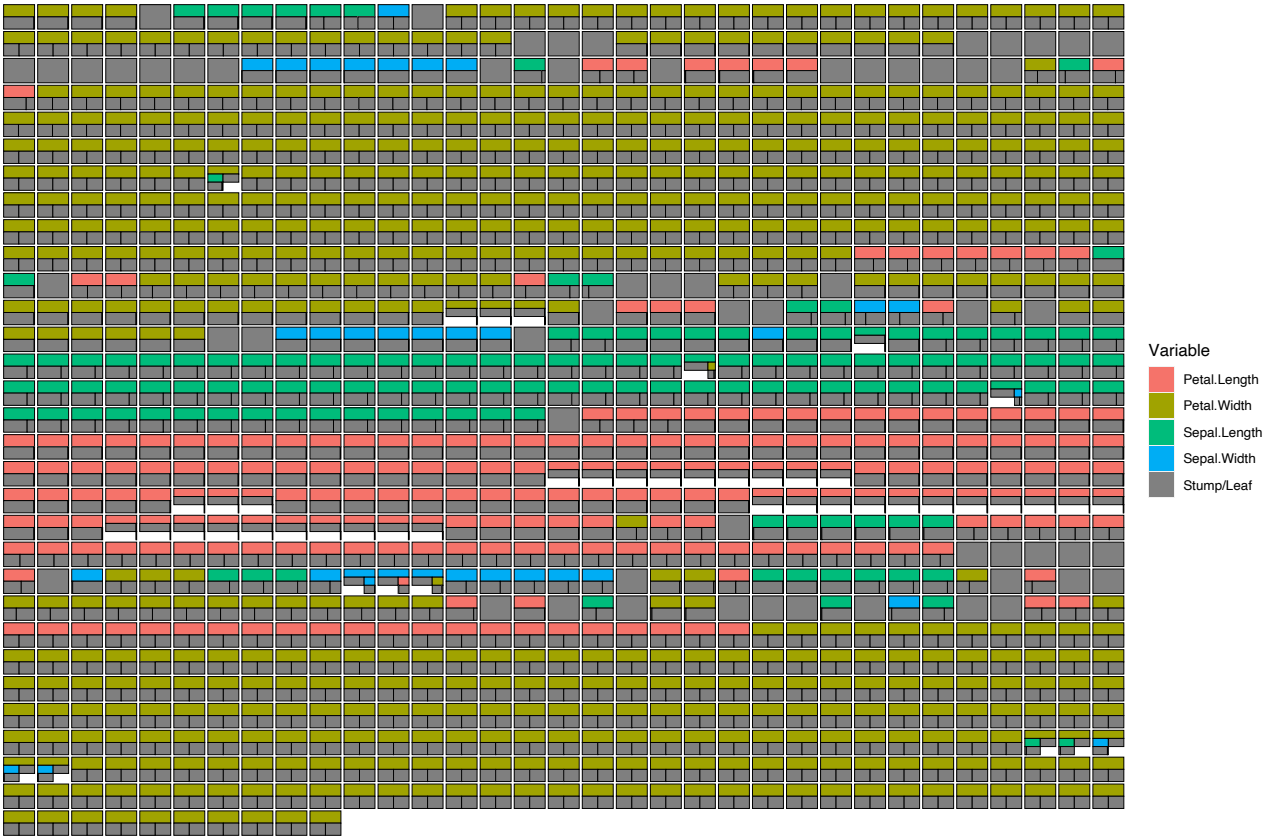


Figure 4: A single tree over 1000 iterations. The coloured bars indicate which variable is used for the split at that point. Grey boxes indicate stumps or terminal nodes. The vertical black lines in the terminal nodes indicate the proportion of the data being split into the left or right terminal node.

(More options to colour the nodes by certain parameters are shown in later plots in this section.) With this display we see how a tree evolves over iterations. Here we see the prevalence of Petal.Width as a splitting variable (lime green rectangles) once again indicating the importance of this predictor.

In our tree displays, it is also useful to view different aspects or metrics. In Figure 5, we explore some of these aspects by displaying all the trees in a selected iteration (in this case, we chose the iteration with lowest residual standard deviation). We consider variations which colour terminal nodes and stumps by the mean response (panel (a)), colour them by the terminal node parameter value (panel (b)), sort the trees by structure starting with the most common tree and descending to the least common tree found for easy identification of the most important splits (panel (c)), or sort the trees by depth (panel (d)). As the μ values in (b) are centred around zero, we use a single-hue, colourblind friendly, diverging colour palette to display the values. For comparison, we use the same palette to represent the mean response values in (a).

Different interesting findings are seen in the four panels. Panel (b) indicates that tree 12 (column two, row three) has a much greater influence on the overall predictions than the others, which seems surprising given the nature of the shrinkage prior used in BART which aims to shrink the terminal node parameters towards zero. From (c), we observe that the most common tree structure in this iteration is Petal.Width as the

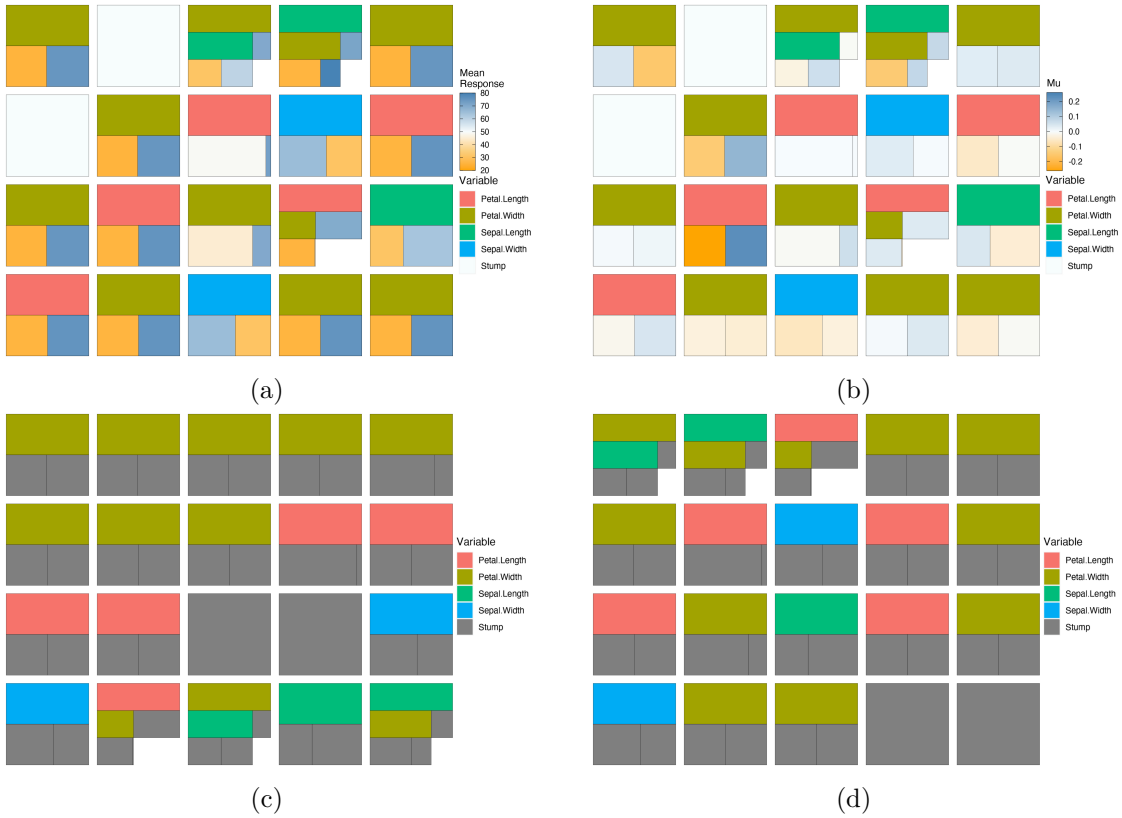


Figure 5: All trees in a selected iteration. In (a) the terminal nodes and stumps are coloured by the mean response. In (b) the terminal nodes and stumps are coloured by the predicted value μ. In (c) we sort the trees by structure starting with the most common tree and descending to the least common tree shape and in (d) we sort the trees by tree depth.

root with a single binary split. The second most common is Petal.Length as the root with a single binary split. In (d), it is quickly identified that the vast majority of trees in this iteration are split only once this is representative of the tree-depth found over the entire ensemble while using the default parameter settings when fitting the model on this data).

When the number of variables or trees is large it can become harder to identify interesting features. We provide a plot that can be used to highlight interesting features by accentuating selected variables by colouring them brightly while uniformly colouring the remaining variables a light grey. When coupled with the sorting shown previously in Figure 5, we have found that this more clearly identifies relationships of interest. As the iris data has very few predictors, we omit this plot here but an example of it can be seen the larger case study example of Figure 14 in Section 5.

Finally, as an alternative to the sorting of the tree structures, seen in Figure 5 (c), we provide a bar plot summarising the tree structures. Figure 6 shows a barplot of the frequency of the tree types over all iterations, filtered to show the top 10 most frequent trees, where the legend indicates the tree structure with the node sizes equally proportioned. To count the tree structures, we use the same sorting algorithm as Figure 5 (c). This seems most useful when summarising a large number of trees (though again

these plots can also be created for a single tree across iterations or to display all trees in a single iteration). We can see that the most common tree type over all iterations is the tree that has a single binary split on Petal.Width, with the second most common being the tree that has a single binary split on Petal.Length. Additionally, we can see that Petal.Width appears in several of the other top 10 most common tree structures. This is in agreement with the inclusion proportion variable importance plot of Figure 2 which tells us that Petal.Width is used as a splitting rule most often. In addition, most of the trees with more than one split involve Petal.Width, indicating that this variable impacts on the response in a more complex way and interacts with other predictors, verifying the findings of Figure 3.

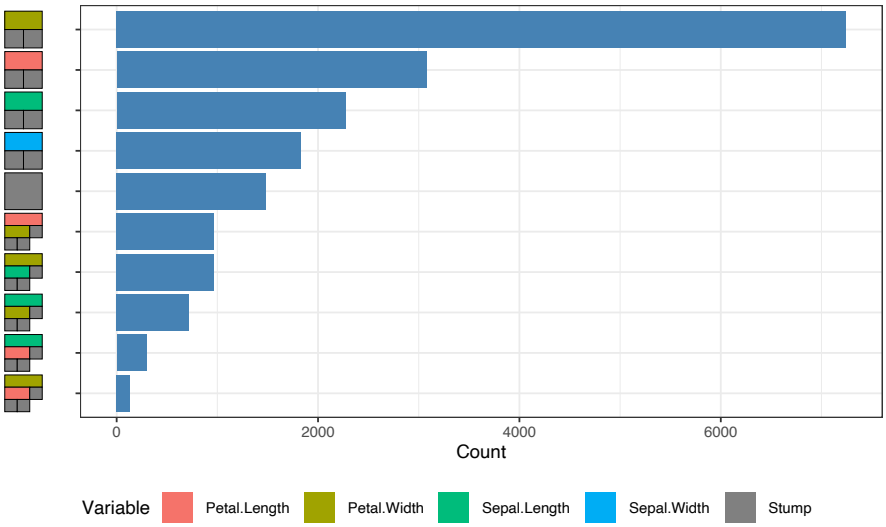


Figure 6: Bar plot of the top 10 most frequent tree types over all iterations. Trees with a single binary split on Petal.Width occur the most often.

As the structure of BART trees can be controlled to favour shallow trees, this mitigates very deep trees from being created. However, as the trees in Figure 4 are plotted on a grid, if deep trees do exist, the visualisation will scale the space automatically. This may result in a visualisation with a lot of white space. In this situation, selecting a subset of trees or using the summary plots (as shown in Figure 5 and 6) may be preferable.

3.3. Outlier Identification with Multidimensional Scaling

Proximity matrices combined with multidimensional scaling (MDS) are commonly used in random forests to identify outlying observations (Breiman 2001). Both proximites and MDS have been shown to be useful tools and can be applied to wide range of data types, including genomic and ecological data (for example, see Englund and Verikas 2012; Cutler et al. 2007). However, to our knowledge, these methods have not yet been implemented for a BART model. When two observations lie in the same terminal node repeatedly they can be said to be similar, and so an $N \times N$ proximity matrix is obtained by accumulating the number of times at which this occurs for each

pair of observations, and subsequently divided by the total number of trees. A higher value indicates that two observations are more similar. The proximity matrix is then visualised using classical MDS (henceforth MDS) to plot their relationship in a lower dimensional projection.

In BART there is a proximity matrix for every iteration and thus a posterior distribution of proximity matrices. While trivial to then apply MDS to each matrix, we introduce a rotational constraint so that we can similarly obtain a posterior distribution of each observation in the lower dimensional space. We first choose a target iteration (we use the iteration with lowest residual standard deviation) and apply MDS. For each subsequent iteration we rotate the MDS solution matrix to match this target as closely as possible using Procrustes' method. We end up with a point for each observation per iteration per MDS dimension. We then group the observations by the mean of each group and produce a scatterplot, where each point represents the centroid of the location of each observation across all the MDS solutions. This allows for an easier to read estimate of potentially outlying data points. We extend this further by displaying the 95% confidence ellipses around each observation's posterior location in the reduced space. Since these are often overlapping we have created an interactive version that highlights an observation's ellipse when hovering the mouse pointer above the ellipse (Figure 7 shows a screenshot of this interaction in use). The observation number is also displayed during this action.

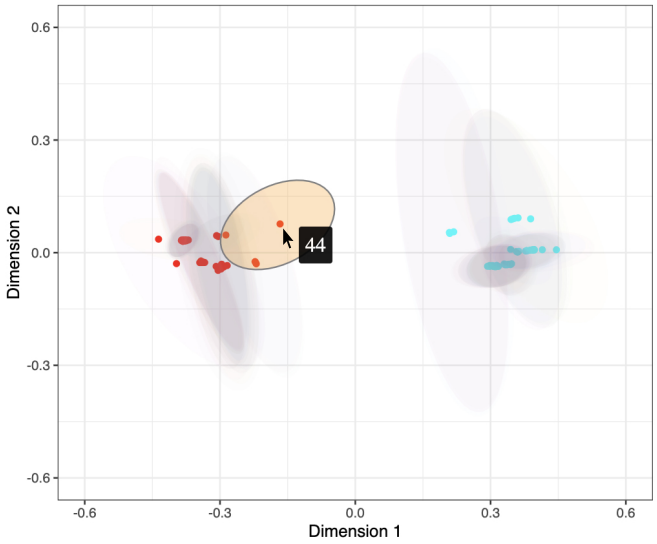


Figure 7: Interactive MDS plot of the iris data where the points are coloured by class (in this case, either Species). Each 95% confidence ellipse corresponds to each observation's posterior location. When hovering the mouse pointer over an ellipse, the ellipse is highlighted and the observation is displayed.

In Figure 7, each point represents the centroid of the location of each observation across all the MDS solutions and are coloured according to their class (in this case, either Species). We can see that most of the variability is, unsurprisingly, in the first-dimension, and while some points have quite different posterior distributions, the uncertainty on many of them is large. Observation 44 appears to have a moderate uncertainty

and is separated by some distance from the other observations in that class, indicating that this observation may be an outlier. This is in agreement with previous studies (such as Acuna and Rodriguez (2004)), which identify this observation as an outlier.

3.4. Enhanced BART model diagnostics

In this section, we examine some of the more common issues a researcher may face when running a BART model. These include checking for convergence, the stability of the trees, the efficiency of the algorithm, and the predictive performance of the model. In our experience, most popular BART R packages are limited in scope for creating informative model visualisations (with the possible exception of **bartMachine** which features versions of Figures 8 and 9). Our goal in these plots is to provide a convenient and useful summary of the model's characteristics which is invariant to the choice of package. A useful side effect of these plots is the ability to compare BART fits from different BART R packages. In the following section we show a selection of diagnostic plots using both the **bartMachine** and **dbarts** packages to build our models. Both models have the same hyperparameters of 1000 iterations with a burn-in of 250 and 20 trees. We use the same two-species subset of the iris data as before.

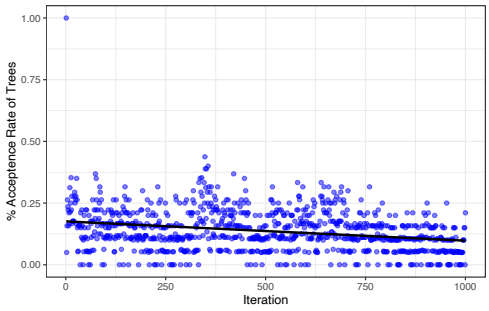
Acceptance Rate of Trees

As discussed in Section 2, BART uses a Metropolis-Hastings algorithm to determine the type of tree structure accepted at each tree in each MCMC iteration. The trees are individually modified by either a grow, prune, change, or swap step and compared to its previous version by calculating the acceptance ratio. The acceptance rate is therefore measured as the percentage of accepted proposed trees across the iterations.

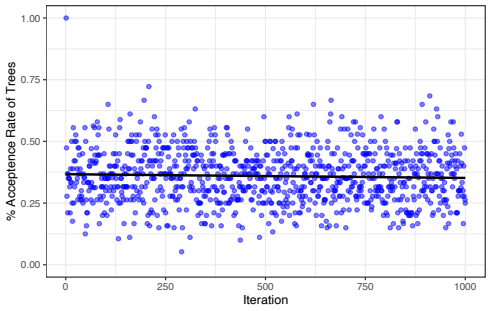
Figure 8 shows the post burn-in percentage acceptance rate across 1000 iterations for both BART models, where each point represents a single iteration. A regression line is shown to indicate the changes in acceptance rate across iterations and to identify the mean rate. Both plots are forced to display the same vertical axis range. Clearly there is a higher acceptance rate (approx 20%) in the **dbarts** fit. None of the iterations in **dbarts** have zero trees accepted, while this occurs commonly for **bartMachine**. This can also be seen in Figure 4 where there are runs of identical trees, indicating that no new trees were accepted during this period.

Tree Depth, Node Number, and Split Distribution

As with the acceptance rate, the average tree depth and average number of all nodes per iteration can give an insight into the fit's stability. Both of these statistics play a key role in the branching process prior distribution used in all the versions of BART and are useful for determining both the behaviour of the posterior distribution in comparison to the prior, and also to check convergence. Figure 9 displays these two metrics for both BART fits. A locally estimated scatterplot smoothing (LOESS) regression line is shown to indicate the changes in both the average tree depth and the average number of nodes across iterations. From Figure 9, we can see that both the post burn-in average tree depth and the average number of nodes per iteration is much more stable in the **dbarts** fit. However, although we use the default number of iterations suggested by the **bartMachine** package, increasing this may improve stability.

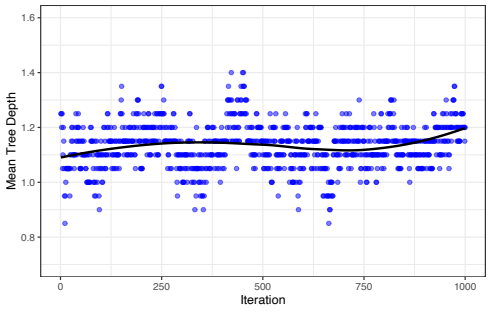


(a) Acceptance rate per iteration using `bartMachine`

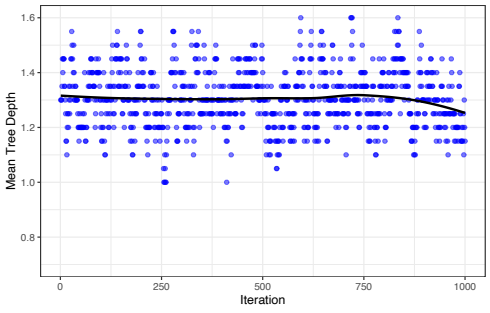


(b) Acceptance rate per iteration using `dbarts`

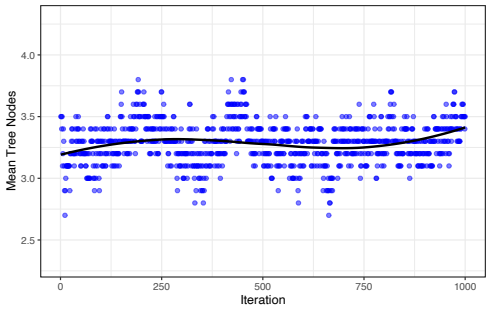
Figure 8: Post burn-in acceptance rate of trees per iteration for a **bartMachine** and **dbarts** fit in (a) and (b), respectively. A black regression line is shown to indicate the changes in acceptance rate across iterations and to identify the mean rate. We can see that the **dbarts** fit has a higher acceptance rate than the **bartMachine** fit.



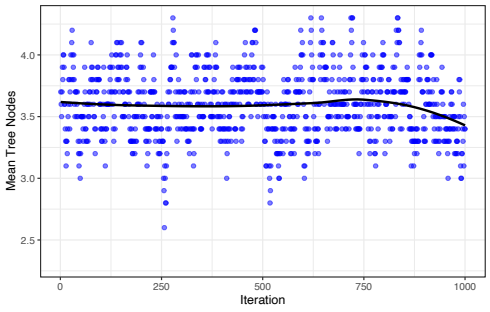
(a) Average tree depth from `bartMachine`.



(b) Average tree depth from `dbarts`.



(c) Average number of nodes from `bartMachine`.

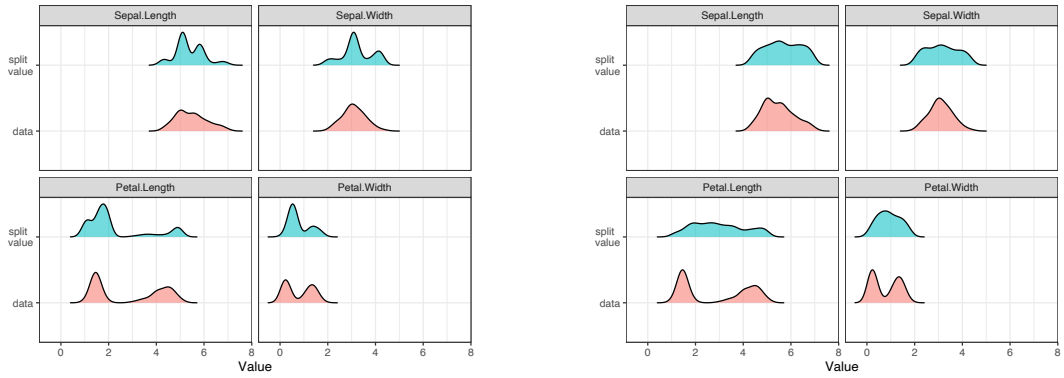


(d) Average number of nodes from `dbarts`.

Figure 9: In the top row we show the post burn-in average tree depth per iteration for a **bartMachine** and **dbarts** fit in (a) and (b), respectively. In the bottom row we show the post burn-in average number of nodes per iteration for a **bartMachine** and **dbarts** fit in (c) and (d), respectively. A black LOESS regression curve is shown to indicate the changes in both the average tree depth and number of nodes across iterations.

Figure 10 shows the densities of split values over all post burn-in iterations for each variable for both models (in green), combined with the densities of the predictor variables (labelled "data", in red). This plot appears to be new; we have not found anything similar in any of the existing packages. We can see that the split value density for

Sepal.Width in the **bartMachine** fit, in (a), has large peak at around 3.2 and the **bartMachine** fit's split values have more modes. In (a), the split density somewhat resembles the data density for some of the variables in the `bartMachine` fit, though it should be noted that in general, there is no specific expectation that the density of splitting values should resemble the density of the data. However, when the number of trees is large there is a potential for noise or unimportant predictors to be included in the tree structure in which case we would expect random splits in the trees which correspond to the predictor data distribution.



(a) Split value distribution obtained from a **bartMachine** fit.

(b) Split value distribution obtained from a **dbarts** fit.

Figure 10: Split values densities (in green) over all iterations for each variable overlayed on the densities of the predictors (in red) for a **bartMachine** fit in (a) and a **dbarts** fit in (b).

In addition to the previous plots, we provide a panel of basic summary diagnostics of the model fit which can be used for both classification and regression models. For the former, we display metrics such as precision-recall and ROC (with uncertainties included), a confusion matrix, fitted value plots, and a histogram of predicted probabilities. For the latter, we show a trace plot of the model variance, a Q-Q plot, and an array of model performance plots and residual plots over all iterations. In the interest of space, we exclude the summary diagnostics for the classification model and display the summary diagnostic plots for the regression model only, as seen in Section 5.

4. Comparing Variable Importance and Interaction Methods for BART

In this section we provide an examination of the variable inclusion proportion methods for evaluating importance and interactions in a BART model (as outlined in Section 2.2) by comparing the raw inclusion proportions with and without uncertainty included against alternative methods used to assess the importance and interactions of variables. These alternative methods do not allow for the inclusion of uncertainty in the metrics they create.

As previously discussed, BART models obtain a measure of importance by observing the proportion of times a variable is used as a split variable across all trees, averaged

over all iterations. The more times a variable is used as a split variable, the more important that variable is deemed to be. Similarly, a measure of interaction can be obtained in a BART model by observing the proportion of successive splits over all trees, averaged over all iterations. However, as noted by Chipman et al. (2010), this method of assessing importance (and interactions) comes with certain pitfalls. Namely, if the number of trees is large, then non-important predictor variables can be preferred as the likelihood is relatively flat and so the tree prior dominates. This can lead to to spurious importance and interactions scores for variables that, in reality, have little influence on the response. This effect can be mitigated somewhat by the inclusion of uncertainty to evaluate the reliability of the measured importance or interaction scores. Additionally, Chipman et al. (2010) state that decreasing the number of trees when building the model diminishes this effect as less important variables get swapped out of the trees for more informative variables.

To compare the usefulness of a BART model's importance and interactions, we compare the BART methodology, with and without uncertainty included, against a model agnostic approach to assess the importance and interactions. To measure the agnostic variable importance we use a permutation method. Permutation importance was first introduced by Breiman (2001) and works by calculating the change in the model's predictive performance after a variable has been randomly permuted. That is, a model score is initially recorded, then a single variable is randomly permuted (this is repeated for each variable) and the model score is recalculated on the new dataset. The difference between the baseline model's performance and the permuted model's performance is taken as the variable importance score. To measure the agnostic interactions we use Friedman's H-statistic (or H-index) (Friedman and Popescu 2008). For this method the partial dependence for a pair of variables is compared to their marginal effects.

Friedman's H-statistic is defined as:

$$H_{jk}^2 = \frac{\sum_{i=1}^{n}[f_{jk}(x_{ij}, x_{ik}) - f_j(x_{ij}) - f_k(x_{ik})]^2}{\sum_{i=1}^{n} f_{jk}^2(x_{ij}, x_{ik})} \tag{4}$$

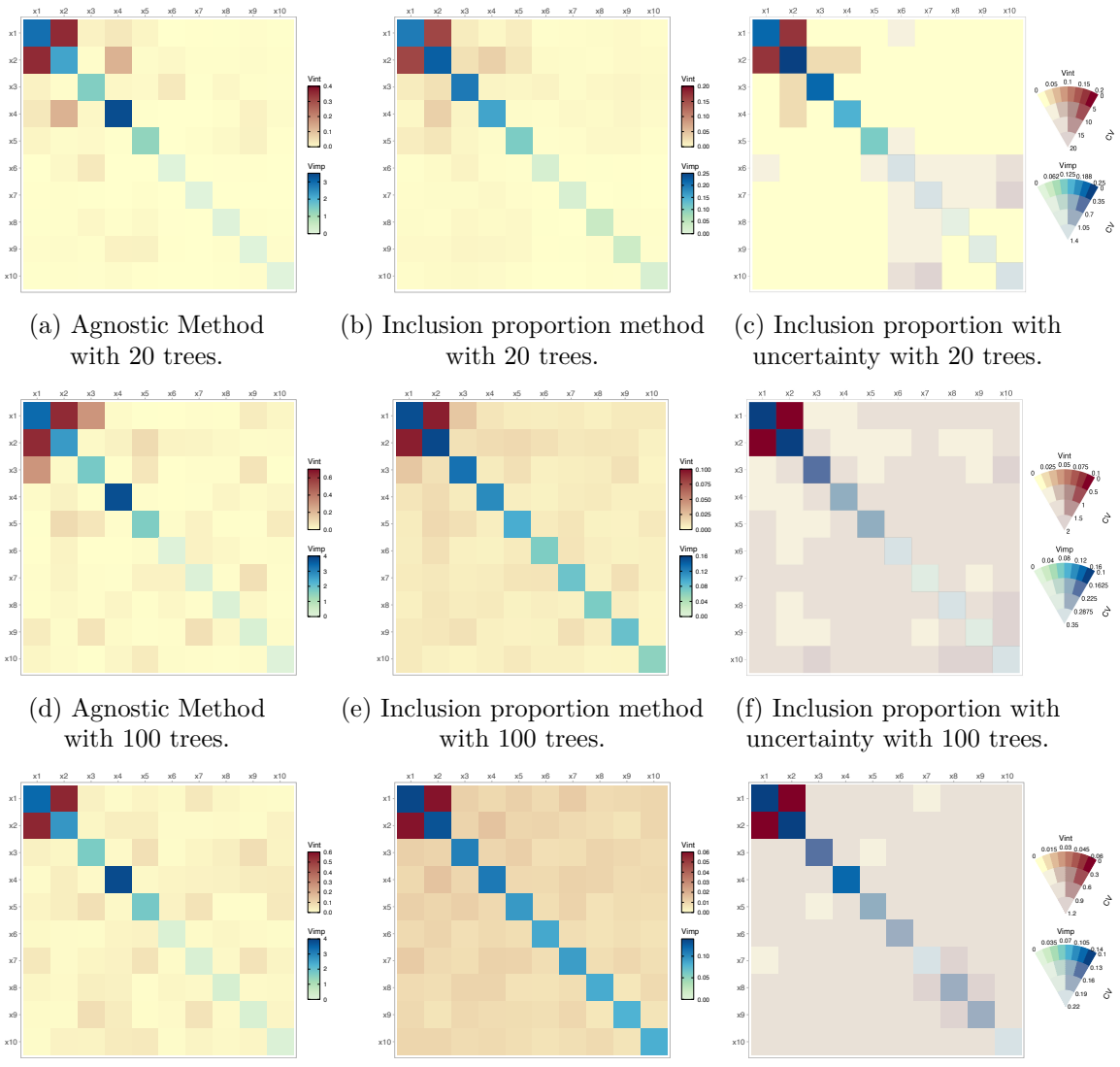
where $f_{jk}(x_j, x_k)$ represents the two-way partial dependence function of both variables, $f_j(x_j)$ and $f_k(x_k)$ represent the partial dependence functions of the single variables, and all partial dependence functions are mean-centered. The obtained measure is scaled in the range (0,1). Inglis et al. (2022) note, however, that variations in the numerator can lead to spuriously high H-values when the denominator in (4) is small because the partial dependence function for the variables j and k is flat in this case. To combat this, the square-root of the average un-normalized (numerator only) version of Friedman's H^2 for calculating pairwise interactions is suggested:

$$H_{jk} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[f_{jk}(x_{ij}, x_{ik}) - f_j(x_{ij}) - f_k(x_{ik})]^2} \tag{5}$$

To explore and compare the variable importance and variable interactions, we generate data using the Friedman benchmark equation (Friedman 1991):

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \epsilon$$
$$\text{where } \; x_j \sim U(0,1), j = 1, 2, \ldots, 10; \;\; \epsilon \sim N(0,1).$$

We simulate 250 observations and fit a BART model using the **dbarts** R package, using the default number of iterations (1000) and burn-in (100). We then set the number of trees to be 20, 100, and 200 to evaluate how well the BART model can capture the importance and interactions. There are five important variables and an interaction between x_1 and x_2 in Equation 6, and five additional predictors $x_6, x_7, \ldots x_{10}$ unrelated to the response.



(a) Agnostic Method with 20 trees.

(b) Inclusion proportion method with 20 trees.

(c) Inclusion proportion with uncertainty with 20 trees.

(d) Agnostic Method with 100 trees.

(e) Inclusion proportion method with 100 trees.

(f) Inclusion proportion with uncertainty with 100 trees.

(g) Agnostic Method with 200 trees.

(h) Inclusion proportion method with 200 trees.

(i) Inclusion proportion with uncertainty with 200 trees.

Figure 11: Comparison of different methods to determine importance and interactions in a BART model with 20, 100, and 200 trees in the first, second, and third rows respectively.

In the first column of Figure 11 (panels (a), (d), and (g)), we use the alternative agnostic permutation approach for measuring importance and Friedman's H-statistic to obtain the interaction measures. In the second column (panels (b), (e), and (h)), we calculate the standard BART model variable inclusion proportion for the importance and interactions. Finally, in the third column (panels (c), (f), and (i)), we display the

same information as in the second column but with uncertainty included, in this case via the coefficient of variation. For each row of Figure 11 we set the number of trees to 20, 100, and 200.

Using the alternative agnostic method (first column) the five important variables are identified with x_4 being ranked as the most important and the interaction between x_1 and x_2 is prominent. This remains consistent, regardless of the number of trees used when building the model. When using the inclusion proportions (second column) the interaction between x_1 and x_2 is strong and individually x_1 and x_2 are the most important. In (b) the five important variables are identified. However, as the number of trees increases (see (e) and (h)) variables $x_6, \ldots, x_{10}$ are incorrectly designated as important. Spurious values are measured for both importance and interactions when increasing the number of trees. Examining the VSUPs (third column) the interaction between x_1 and x_2 is prominent and the five important variables are again evident. Increasing the number of trees has the effect of increasing the relative uncertainty for the spurious values and therefore, highlights the variables of interest. For example, if we compare panels (e) and (f) each based on 100 trees, we see that most of the spurious importance and interaction values in (e) have a moderate degree of relative uncertainty in (f).

It is worth noting that for 20, 100, or 200 trees, although the agnostic method had relatively consistent results, this method may not be computationally practical as it is a slow calculation which gets compounded by the increase in trees. Additionally, the agnostic approach would have be repeated multiple times to allow a measure of uncertainty to be obtained. Conversely, calculating the inclusion proportion is quick. For example, calculating the inclusion proportion for importance and interactions for when the number of trees is 20 (as in panels (b) and (c)) took approximately 1.5 seconds on a MacBook Pro 2.3 GHz Dual-Core Intel Core i5 with 8GB of RAM. Whereas, using the agnostic approach to measure the importance and uncertainty (as in panel (a)) took approximately 43 seconds on the same machine. When viewed with the uncertainty included, the inclusion proportion method performs well when compared to the agnostic method, particularly when the number of trees is low.

5. Case Study: Seoul Bike Sharing Data

In this section we apply our methods on a larger real-world data set. Here we examine and create visualisations concerning bike sharing data from Seoul, South Korea (Sathishkumar 2020) and can be found at `https://data.mendeley.com/datasets/zbdtzxcxvg/2`. The data contains 14 features and includes weather data (for example, humidity, rainfall, snowfall, and several others), the time of the bike rental (in seasons, months, and days), and some local information (such as if the day of rental was a holiday), with the total number of bikes rented per day as the response. The original data contained 8760 hourly observations which we summarise to obtain the daily counts. The modified data can be found at `https://github.com/AlanInglis/bartMan`. The data has been previously studied in Sathishkumar and Yongyun (2020a) and Sathishkumar et al. (2020) who found that the temperature of the day was an important factor for predicting the total number of rentals. Sathishkumar and Yongyun (2020b) also found that the individual month and season play a significant role in predicting bike

rentals and that there is a high degree of collinearity between these variables. It has been shown in that in some tree-based models (such as random forests) that variable importance measures show a bias towards correlated predictor variables (for example, see Strobl et al. (2008)). This can lead to a misrepresentation of the importance of the correlated variables. Consequently, careful consideration should be employed when interpreting the VIVI measures and we recommend evaluating any potential correlation between variables in conjunction with our proposed visualisations.

For our study we fit a BART model, using the **BART** package, with 1000 iterations, a burn-in of 100, and 100 trees, with the goal of investigating which of the predictor variables has a significant impact on the response. We apply a cube root transformation to the response as initially the residuals displayed some evidence of non-normality. As mentioned in Section 2.2, on factor dummy variables, we perform an aggregation of the dummy variables' inclusion proportions for both the importance and the interactions so these metrics can be assessed on the entire factor. The variables treated as factors in the data are Month (the month of the year a bike is rented), Season (season of the year a bike is rented), Wkend (if the day of bike rental is a weekend or not), and Holiday (if the day of bike rental is a public holiday or not).

To begin, Figure 12 shows the model's diagnostics to assess the stability of the model fit. The top two rows indicate a reasonable performance of the residuals with a moderately stable convergence of the residual standard deviation. The black vertical line in the trace plot indicates the separation between the pre and post burn-in period. The bottom row shows that the model fits the training data well and that the Month is clearly the most important variable for predicting the count of bikes rented. However, in the bottom right panel, we can see that Month has a large 25-75% quantile interval when compared to the other variables. The second most important variable is Season followed by Temp (average daily temperature in $^{\circ}C$).

We explore the impact of the variables on the response by examining the importance and interactions jointly in the variable importance and variable interaction plots of Figure 13. For illustration purposes only, we show the plot without uncertainty in the left panel and with uncertainty on the right (as before we use the coefficient of variation). In Figure 13(a), we observe a strong interaction between the variable Month and several others, notably; Rainfall (in mm), Solar.R (Solar radiance in mJ/m^2), Season, and Temp. The strongest interaction can be seen between Month and Rainfall. In Figure 13(b), many of the low importance and interaction scores have high relative uncertainty, so the viewer's attention is drawn to the interesting variables. The most important variable, Month, remains important relative to its uncertainty. Equally, the strong interactions observed in (a) between Month and several others have a low associated variation in (b). In (a), all variables except Month have similar importance scores, but relative to uncertainty, the importance of the last seven variables (Humidity to Wind.Spd) is reduced, represented by greeny-grey colours along the diagonal (b). The interactions between these variables are mostly low and/or with high relative uncertainty, the interaction between Snowfall and Year being an exception.

In Figure 14, we take a deeper look at the structure of the trees for a selected iteration. As before we choose the iteration with the lowest residual standard deviation. As with the importance and interactions, by default we recombine the categorical variables to display the entire factor. With such a large number of predictors, it can become

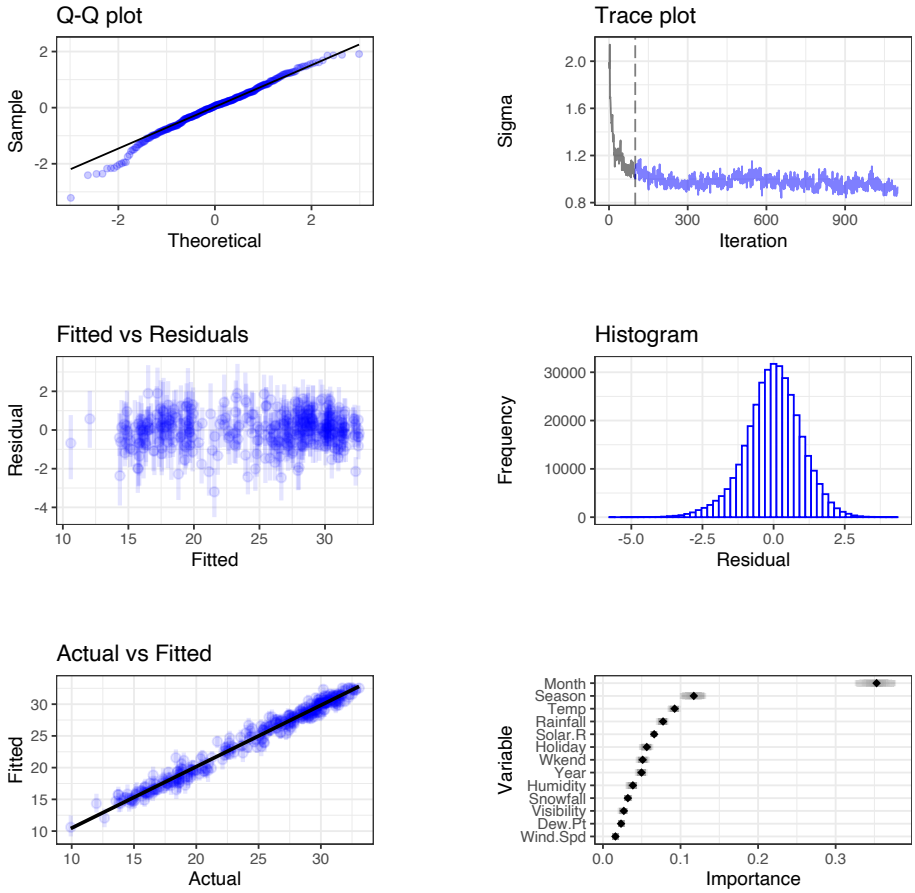
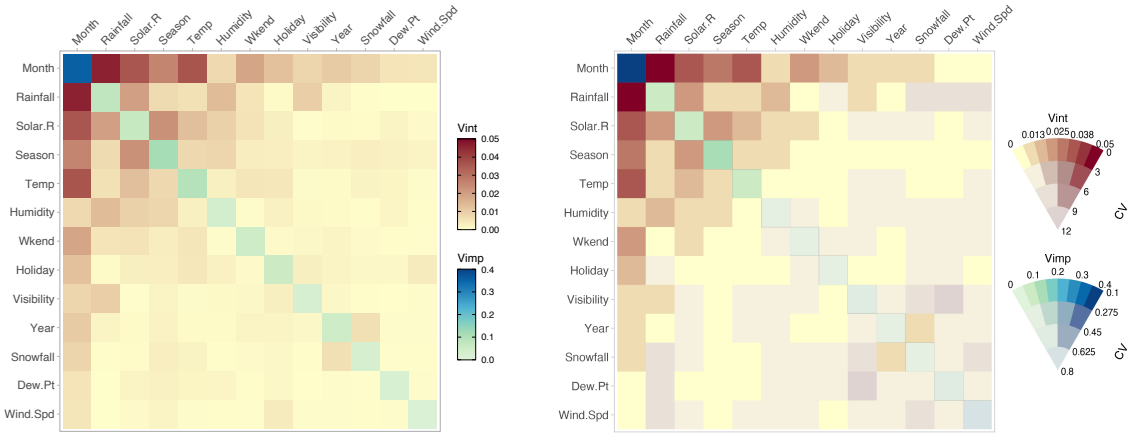


Figure 12: General diagnostic plots for a BART regression fit on bike sharing demand data. Top left: A QQ-plot of the residuals after fitting the model. Top right: σ by MCMC iteration. Middle left: Residuals versus fitted values with 95% credible intervals. Middle right: A histogram of the residuals. Bottom Left: Actual values versus fitted values with 95% credible intervals. Bottom right: Variable importance plot with 25 to 75% quantile interval shown. We can see in the bottom left panel that the model fits the training data reasonably well, with a good convergence seen in the top right panel.

challenging to effectively display a distinguishable hue for each when plotting the trees. To combat this we can select the most interesting variables observed in Figure 13 and highlight them by using bright discernible colours. To aid in efficient examination, we sort the trees by frequency of tree type and remove the stumps.

In this iteration, the most common tree is a single binary split with Month as the parent. It should also be noted that Month is chosen as the root parent more frequently than any other variable and also appears deeper in several other trees and is subsequently the most common variable found in this iteration. The previously noted interactions between Month and the other variables can be observed in the lower portion of the plot. We can also see in Figure 14 that for the variable Month, most of the observations fall into a single terminal node, making one terminal node much larger than the other. Further investigation reveals that the ensemble tends to divide between warmer and colder months, such that the observations corresponding to January and February (the coldest months with the fewest bike rentals) comprise the smaller terminal nodes.



(a) (b)

Figure 13: In (a), Variable importance and interaction plot without uncertainty. In (b), the same values are shown but with the uncertainty included by use of a VSUP. In (b), we can see that the interaction values between Month and several other variables have a low coefficient of variation associated with them.

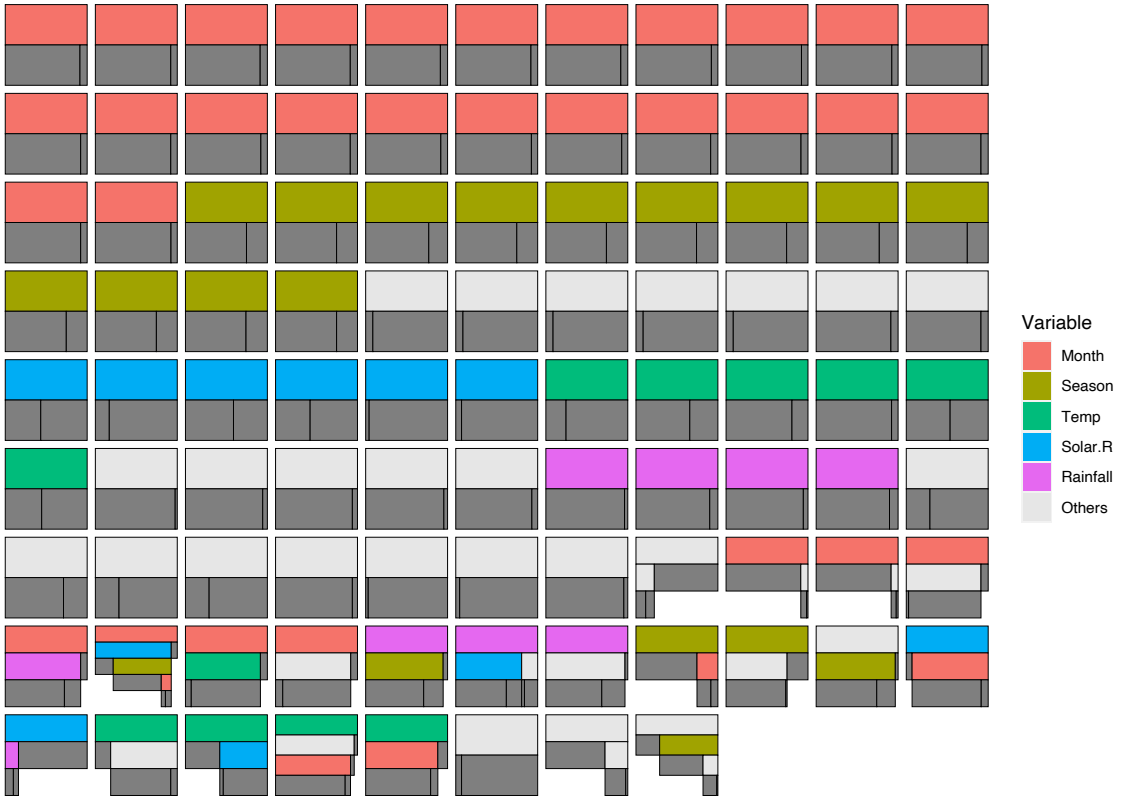


Figure 14: All trees from a selected iteration, highlighting the most interesting variables and sorted by the frequency of tree type. In this case, the terminal nodes are coloured dark grey and the stumps have been removed.

We employ our MDS plot in Figure 15 to help find outliers. Here we can see that each observation has moderate uncertainty, represented by the surrounding 95% uncertainty

ellipses. We have highlighted observation 347 which lies slightly farther away from the group. Inspecting this observation in the data tells us that this observation corresponds to bike rentals on December 24^{th}, which is a public holiday. Bike rentals were well below average for this day, particularly for a public holiday, which has usually high bike rentals. The temperature on this day was also well below average. This may indicate as to why this particular observation lies slightly farther from its group.

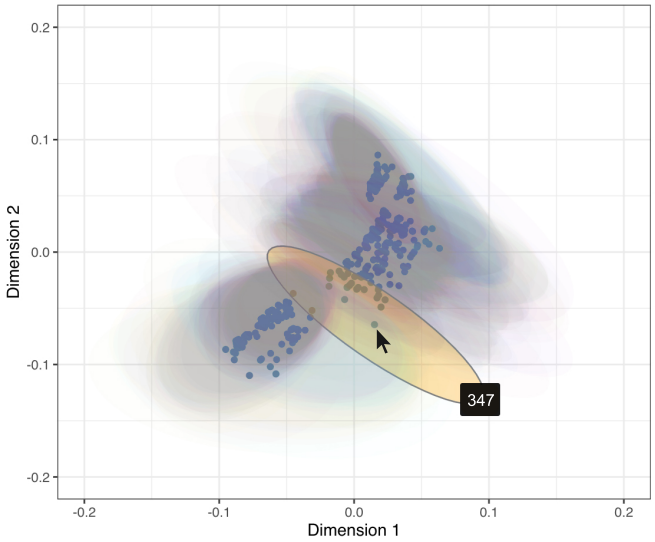


Figure 15: MDS plot of a BART fit on the Seoul bike sharing data. The observations appear to have a moderate degree of uncertainty. Observation 347 (highlighted) appears to be an outlier as it lies slightly farther away from its group.

To summarise, we have used our visualisations to identify and examine variables associated with the prediction of bike rentals in Seoul, South Korea. Our approach allowed us to examine the overall model fit and how individual and pairs of variables impact on the fit. Through our tree-based plots we can examine the inner structure of our fit. Specifically, we found the month a bike was rented was ranked the most important variable. Our methods rated Season as an important predictor, agreeing with previous studies (Sathishkumar and Yongyun 2020b). We also find Temperature to be important, again verifying the findings of Sathishkumar and Yongyun (2020a) and Sathishkumar et al. (2020).

6. Discussion

We have presented new and informative visualisations for posterior evaluation of BART models. We extend the traditional method of assessing variable importance and variable interactions by including the uncertainty that comes with Bayesian models in our point plots and heatmaps that feature the value suppressing uncertainty palettes methods of Correll et al. (2018). With our tree-based plots in Section 3.2, we can examine the structure of the decision trees that are created when building the model as well as providing useful summaries of tree types by way of grouping tree structures by different metrics. We display outlier detection methods by way of an interactive multidimensional scaling plot in Section 3.3 to provide an in-depth examination of a model's fit.

Finally, we provide a selection of enhanced model diagnostic plots in Section 3.4, which are practical for assessing a model fit via a suite of plots that visualise aspects of a model such as stability, tree acceptance rate, average number of nodes, and average tree depth plots. These plots also provide a useful summary of the overall model fit via convergence, residual, and Q-Q plots (for regression), and ROC, precision-recall, and confusion matrices (for classification). Our tools can be used to explore areas of the model that are not well explored. This might involve, looking at particular combinations of variables that are not regularly featured in trees, despite being known to be important, or patches of residuals identified by the MDS that indicate a missing split or covariate. Our approach is simple to use, adaptable, customisable, and can be useful for comparing different BART model fits.

Our importance and interaction plots can be useful in determining which variables have the greatest impact on the response and the inclusion of uncertainty can help in deciding if a given variable's importance is worthwhile. A drawback to this method is that the use of inclusion proportions as an importance/interaction measure relies on the splitting rules in the model. Since BART chooses the splitting rule uniformly across all variables, non-important variables can be included. This effect can be mitigated by selecting a smaller number of trees, however this may limit the predictive performance of the model, as noted by Chipman et al. (2010). The examples of Section 4 show that using the proportions alone as an importance measure can be misleading, but that the use of a VSUPs with relative uncertainty provide a correction.

A current drawback occurs when the number of trees and/or MCMC iterations is large, so that the computational time to build the data frame of trees used for producing these visualisations can vary, depending on the R package used. For example, a model with 20 trees and 500 MCMC iterations took approximately 8.2, 9.2, and 90 seconds for a `BART`, `dbarts`, and `bartMachine` fit, respectively, on a MacBook Pro 2.3 GHz Dual-Core Intel Core i5 with 8GB of RAM. The disparity between `bartMachine` and the other packages is due to the way `bartMachine` uses a Java back-end to extract the raw node data from the model. Although some steps were taken to speed up this process, it remains largely outside of our control.

Our methods are flexible and can be easily extended to work with other BART packages, such as **bayesplot** (Gabry et al. 2019), which is an R package that provides a large library of plotting methods for use with Bayesian models fits. Similarly, our methods could be extended to incorporate different extensions of BART, such as the methods of Prado et al. (2021) for model trees BART (MOTR-BART). Rather than having a single value for the prediction at the node level, MOTR-BART estimates a linear predictor using the covariates that were used as split variables in the relevant tree.

For future work, coupling the acceptance rate (seen in Figure 8) with the average tree type plot (seen in Figure 6) would give an interesting additional check for model space that remains unexplored and could provide insight on problems of the algorithm. Additionally, providing an option to allow the user to select an observation (or subset of observations) and colour the nodes which contain the selected observation(s) would aid interpretability. A different method for measuring and visualising the importance and interactions could also be investigated for future work. A method such as DART (Linero 2018), which modifies a BART model by placing a Dirichlet hyper-prior on the splitting proportions of the regression tree prior, could be used to assess importance. When

using DART, Linero (2018) recommend selecting predictor variables from a so-called median probability model (Barbieri and Berger 2004) to conduct variable selection, where the median probability model is defined as a model containing variables whose posterior inclusion probability is at least 50%. Alternatively, other methods, such as Shapley values or an order dependent measure of importance and interaction (that is, considering the order of splits as important) would be worthy of investigation in future.

Computational Details

The results in this paper were obtained using R 4.1.0 and the R package **bartMan** available at: https://github.com/AlanInglis/bartMan. All additional packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/. The source code and data for generating the experimental results is available at https://github.com/AlanInglis/bartMan/tree/master/paperCode.

Acknowledgments

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.

Supplementary Materials

Here we provide a list of the relevant functions from the `bartMan` package. The main construction function is `extractTreeData` which employs S3 methods and is used to create a list of tree attributes which can be used for the heatmap, tree-plot, proximity, and MDS visualisations. All listed functions work with the `BART`, `dbarts` or `bartMachine` packages.

Table 1: Summary of functions available in the `bartMan` package. The main construction function is `extractTreeData` which is used for the heatmap, MDS, and tree-plot visualisations.

`bartMan` Package Functions		
Function Name	**Description**	**Type**
`extractTreeData`	Creates a list of all tree attributes for a model created by either the `BART`, `dbarts` or `bartMachine` packages.	Construction
`viviBartMatrix`	Returns a matrix or list of matrices which can be used for plotting variable importance and interactions with the uncertainty included.	Construction
`proximityMatrix`	Creates a matrix of proximity values.	Construction
`viviBartPlot`	Plots a Heatmap showing variable importance on the diagonal and variable interaction on the off-diagonal with uncertainty included.	Visualisation
`plotTree`	Plots individual trees from the R-packages `BART`, `dbarts` or `bartMachine`.	Visualisation
`plotAllTrees`	Plots all the trees from a selected iteration or plots a selected tree over all iterations.	Visualisation
`treeBarPlot`	Creates a barplot displaying the frequency of different tree structures.	Visualisation
`plotProximity`	Plots a proximity matrix constructed from the `proximityMatrix` function.	Visualisation.
`mdsBart`	Plots a multi-dimensional scaling plot of a proximity matrix from a BART model.	Visualisation
`bartDiag`	Displays a selection of diagnostic plots.	Visualisation
`acceptRate`	Plots the acceptance rate of trees.	Visualisation
`treeDepth`	Plots the tree depth over iterations.	Visualisation
`treeNodes`	Plots the number of nodes over iterations.	Visualisation
`splitDensity`	Density plots of the split value for each variable used in the model.	Visualisation
`vimpBart`	Plots the variable importance (based on the inclusion proportion) with uncertaity included.	Visualisation

References

Acuna, E. and Rodriguez, C. (2004). A Meta Analysis Study of Outlier Detection Methods in Classification. *Technical paper, Department of Mathematics, University of Puerto Rico at Mayaguez*, 1:25.

Barbieri, M. M. and Berger, J. O. (2004). Optimal Predictive Model Selection. *The Annals of Statistics*, 32(3):870–897, DOI: 10.1214/009053604000000238.

Barlow, T. and Neville, P. (2001). A Comparison of 2-D Visualizations of Hierarchies. In *Information Visualization, IEEE Symposium on*, pages 131–131. Citeseer.

Blattenberger, G. and Fowles, R. (2014). Avalanche Forecasting: Using Bayesian Additive Regression Trees (BART). In *Demand for Communications Services–Insights and Perspectives*, pages 211–227. Springer, DOI: 10.1007/978-1-4614-7993-2_11.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32, DOI: 10.1023/A:1010933404324.

Brodlie, K., Allendes Osorio, R., and Lopes, A. (2012). A Review of Uncertainty in Data Visualization. *Expanding the Frontiers of Visual Analytics and Visualization*, pages 81–109, DOI: https://doi.org/10.1007/978-1-4471-2804-5_6.

Chipman, H. and McCulloch, R. (2016). *BayesTree: Bayesian Additive Regression Trees*, https://CRAN.R-project.org/package=BayesTree. R package version 0.3-1.4.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian Additive Regression Trees. *The Annals of Applied Statistics*, 4(1):266–298, DOI: 10.1214/09-AOAS285.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2013). Bayesian regression structure discovery. In Damien, P., Dellaportas, P., Polson, N. G., and Stephens, D. A., editors, *Bayesian theory and applications*, chapter 22, pages 451–465. OUP Oxford.

Coltman, J. (2020). bartpy. https://github.com/JakeColtman/bartpy.

Correll, M., Moritz, D., and Heer, J. (2018). Value-Suppressing Uncertainty Palettes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11. DOI: 10.1145/3173574.3174216.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., and Lawler, J. J. (2007). Random Forests for Classification in Ecology. *Ecology*, 88(11):2783–2792, DOI: https://doi.org/10.1890/07-0539.1.

Deng, W., Coker, B., Liu, J. Z., and Coull, B. A. (2022). Towards a unified framework for uncertainty-aware nonlinear variable selection with theoretical guarantees. *arXiv preprint arXiv:2204.07293*.

Dorie, V. (2020). **dbarts**: Discrete Bayesian Additive Regression Trees Sampler. https://CRAN.R-project.org/package=dbarts. R package version 0.9-19.

Englund, C. and Verikas, A. (2012). A Novel Approach to Estimate Proximity in a Random Forest: An Exploratory Study. *Expert systems with applications*, 39(17):13046–13050, DOI: https://doi.org/10.1016/j.eswa.2012.05.094.

Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2):179–188.

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–67, DOI: 10.1214/aos/1176347963.

Friedman, J. H. (2000). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29, DOI: 10.1214/aos/1013203451.

Friedman, J. H. and Popescu, B. E. (2008). Predictive Learning via Rule Ensembles. *The Annals of Applied Statistics.*, pages 916–954, DOI: 10.1214/07-AOAS148.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. (2019). Visualization in Bayesian Workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182:389–402, DOI: 10.1111/rssa.12378.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, DOI: https://doi.org/10.1093/biomet/82.4.711.

Grömping, U. (2015). Variable importance in regression models. *Wiley interdisciplinary reviews: Computational statistics*, 7(2):137–152, DOI: https://doi.org/10.1002/wics.1346.

Hahn, P. R., Murray, J. S., Carvalho, C. M., et al. (2020). Bayesian Regression Trees Models for Causal Inference: Regularization, Confounding, and Heterogeneous Effects. *Bayesian Analysis*, DOI: 10.1214/19-BA1195.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, DOI: https://doi.org/10.1007/978-0-387-21606-5.

Hernández, B., Pennington, S. R., and Parnell, A. C. (2015). Bayesian Methods for Proteomic Biomarker Development. *EuPA Open Proteomics*, 9:54–64, DOI: https://doi.org/10.1016/j.euprot.2015.08.001.

Hill, J. L. (2011). Bayesian Nonparametric Modeling for Causal Inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, DOI: 10.1198/jcgs.2010.08162.

Inglis, A., Parnell, A., and Hurley, C. B. (2022). Visualizing Variable Importance and Variable Interaction Effects in Machine Learning Models. *Journal of Computational and Graphical Statistics*, pages 1–13, DOI: 10.1080/10618600.2021.2007935.

Ishwaran, H., Kogalur, U. B., Gorodeski, E. Z., Minn, A. J., and Lauer, M. S. (2010). High-dimensional variable selection for survival data. *Journal of the American Statistical Association*, 105(489):205–217, DOI: 10.1198/jasa.2009.tm08622.

Kapelner, A. and Bleich, J. (2016). **bartMachine**: Machine Learning with Bayesian Additive Regression Trees. *Journal of Statistical Software*, 70(4):1–40, DOI: 10.18637/jss.v070.i04.

Kruskal, J. B. and Landwehr, J. M. (1983). Icicle Plots: Better Displays for Hierarchical Clustering. *The American Statistician*, 37(2):162–168, DOI: 10.1080/00031305.1983.10482733.

Linero, A. R. (2018). Bayesian Regression Trees for High-Dimensional Prediction and Variable Selection. *Journal of the American Statistical Association*, 113(522):626–636, DOI: 10.1080/01621459.2016.1264957.

Liu, Y., Traskin, M., Lorch, S. A., George, E. I., and Small, D. (2015). Ensemble of Trees Approaches to Risk Adjustment for Evaluating a Hospital's Performance. *Health Care Management Science*, 18(1):58–66.

Paluszynska, A., Biecek, P., and Jiang, Y. (2020). *randomForestExplainer: Explaining and Visualizing Random Forests in Terms of Variable Importance*, https://CRAN.R-project.org/package=randomForestExplainer. R package version 0.10.1.

Pang, A. T., Wittenbrink, C. M., Lodha, S. K., et al. (1997). Approaches to Uncertainty Visualization. *The Visual Computer*, 13(8):370–390, DOI: 10.1007/s003710050111.

Prado, E. B., Moral, R. A., and Parnell, A. C. (2021). Bayesian Additive Regression Trees with Model Trees. *Statistics and Computing*, 31(3):1–13, DOI: https://doi.org/10.1007/s11222-021-09997-3.

Robertson, P. K. and O'Callaghan, J. F. (1986). The Generation of Color Sequences for Univariate and Bivariate Mapping. *IEEE Computer Graphics and Applications*, 6(2):24–32, DOI: 10.1109/MCG.1986.276688.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, DOI: 10.7717/peerj-cs.55.

Sathishkumar, V. (2020). Seoul Bike Sharing Demand Prediction. DOI: 10.17632/zbdtzxcxvg.2.

Sathishkumar, V., Park, J., and Cho, Y. (2020). Using Data Mining Techniques for Bike Sharing Demand Prediction in Metropolitan City. *Computer Communications*, 153:353–366, DOI: https://doi.org/10.1016/j.comcom.2020.02.007.

Sathishkumar, V. and Yongyun, C. (2020a). A Rule-Based Model for Seoul Bike Sharing Demand Prediction using Weather Data. *European Journal of Remote Sensing*, 53(sup1):166–183, DOI: 10.1080/22797254.2020.1725789.

Sathishkumar, V. and Yongyun, C. (2020b). Season Wise Bike Sharing Demand Analysis using Random Forest Algorithm. *Computational Intelligence*, DOI: https://doi.org/10.1111/coin.12287.

Schwartz, M. H., Steele, K. M., Ries, A. J., Georgiadis, A. G., and MacWilliams, B. A. (2022). A model for understanding the causes and consequences of walking impairments. *PLoS one*, 17(12):e0270731, DOI: https://doi.org/10.1371/journal.pone.0270731.

Shapley, L. S. (1997). A Value for n-Person Games. *Classics in Game Theory*, 69.

Sparapani, R., Spanbauer, C., and McCulloch, R. (2021). Nonparametric Machine Learning and Efficient Computation with Bayesian Additive Regression Trees: The **BART** R Package. *Journal of Statistical Software*, 97(1):1–66, DOI: 10.18637/jss.v097.i01.

Sparapani, R. A., Logan, B. R., McCulloch, R. E., and Laud, P. W. (2016). Nonparametric Survival Analysis using Bayesian Additive Regression Trees (BART). *Statistics in Mdicine*, 35(16):2741–2753, DOI: 10.1002/sim.6893.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC bioinformatics*, 9:1–11, DOI: https://doi.org/10.1186/1471-2105-9-307.

Strode, G., Morgan, J. D., Thornton, B., Mesev, V., Rau, E., Shortes, S., and Johnson, N. (2019). Operationalizing Trumbo's Principles of Bivariate Choropleth Map Design. *Cartographic Perspectives*, 94:5–24, DOI: https://doi.org/10.14714/CP94.1538.

Teuling, A., Stöckli, R., and Seneviratne, S. I. (2011). Bivariate Colour Maps for Visualizing Climate Data. *International Journal of Climatology*, 31(9):1408–1412, DOI: https://doi.org/10.1002/joc.2153.

Torgerson, W. S. (1952). Multidimensional Scaling: I. Theory and Method. *Psychometrika*, 17(4):401–419, DOI: https://doi.org/10.1007/BF02288916.

Trumbo, B. E. (1981). A Theory for Coloring Bivariate Statistical Maps. *The American Statistician*, 35(4):220–226, DOI: 10.1080/00031305.1981.10479360.

Wei, P., Lu, Z., and Song, J. (2015). Variable Importance Analysis: A Comprehensive Review. *Reliability Engineering & System Safety*, 142:399–432, DOI: https://doi.org/10.1016/j.ress.2015.05.018.

Affiliation:

Alan Inglis
Hamilton Institute
Maynooth University
Maynooth
Co.Kildare
Ireland
E-mail: alan.n.inglis@gmail.com

Andrew Parnell
Hamilton Institute
Maynooth University
Maynooth
Co.Kildare
Ireland
E-mail: andrew.parnell@mu.ie

Catherine Hurley
Department of Mathematics and Statistics
Maynooth University
Maynooth
Co.Kildare
Ireland
E-mail: catherine.hurley@mu.ie

Journal of Data Science, Statistics, and Visualisation https://jdssv.org/
published by the International Association for Statistical Computing
http://iasc-isi.org/

ISSN 2773-0689
February 2024, Volume IV, Issue I *Submitted:* 2022-09-20
doi:10.52933/jdssv.v4i1.79 *Accepted:* 2023-06-28