Handling Missing Values using Decision Trees with Branch-Exclusive Splits

Cédric Beaulac [1] Jeffrey S. Rosenthal [2]

April 27, 2018

Abstract

In this article we propose a new decision tree construction algorithm. The proposed approach allows the algorithm to interact with some predictors that are only defined in subspaces of the feature space. One way to utilize this new tool is to create or use one of the predictors to keep track of missing values. This predictor can later be used to define the subspace where predictors with missing values are available for the data partitioning process. By doing so, this new classification tree can handle missing values for both modelling and prediction. The algorithm is tested against simulated and real data. The result is a classification procedure that efficiently handles missing values and produces results that are more accurate and more interpretable than most common procedures.

Keywords : Classification and Regression Tree, Missing Data, Applied Machine Learning, Data analysis, Interpretable Models, Variable Importance Analysis

[1]University of Toronto
Department of Statistical Sciences
Sidney Smith Hall, 100 St George St, Toronto, ON M5S 3G3
E-mail : beaulac.cedric@gmail.com
ORCID : 0000-0002-6050-5313
[2]University of Toronto
Department of Statistical Sciences
Sidney Smith Hall, 100 St George St, Toronto, ON M5S 3G3
E-mail : jeff@math.toronto.edu

1 Introduction

Machine learning algorithms are used in many exciting real data applications, but may have problems handling predictors with missing values. Many solutions have been proposed to deal with observations that are missing completely at random (MCAR). Here, we propose a solution that uses the tree structure of Classification and Regression Trees (CART) to deal in an intuitive manner with observations that are missing in patterns which are not completely at random.

Our proposed new tree construction procedure was inspired by a data set where the missing pattern of one subset of predictors could be perfectly explained by another subset. A decision tree is an algorithm that partitions the feature space recursively, splitting it into two subspaces. The proposed algorithm is a general framework that allows the researcher to impose a structure on the variables available for the partitioning process. By doing so, we construct Branch-Exclusive Splits Trees (BEST). When a predictor x_j contains missing values, we can use other predictors to identify the region where the predictor x_j contains no missing value. We can therefore use the proposed algorithm to consider splitting on a predictor only when it contains no missing value based on previous partitioning. BEST can be easily adapted to any splitting rule and any forest forming procedure [4][5][10]. BEST also has other applications. It can be used by researchers that would like to utilize some knowledge they have on the data generating distribution in order to guide the algorithm in selecting a more accurate and more interpretable classifier.

In this article we will introduce the classification problem and its notation and we will explain how classification trees solve that problem. We will then do a quick review of the missing values treatments that are currently being used. Afterwards, we will introduce the proposed algorithm and some motivating examples before explaining in detail how the algorithm functions. Finally, some tests will be performed on simulated data sets and on the real data that inspired this new algorithm.

2 The classification problem

In a typical supervised statistical learning problem we are interested in understanding the relationship between a response variable $\mathbf{Y}$ and an associated m-dimensional predictor vector $\mathbf{X} = (X_1, ..., X_m)$. When the response variable is categorical and takes k different possible values, this problem is defined as a k-class classification problem. In that set up, an interesting challenge is to use a data set $S = \{(Y_i, X_{1,i}, ..., X_{m,i}); i = 1, ..., n\}$ in order to construct a classifier h. Most of the time, it is assumed that the observations within our data set were drawn independently from the same unknown and true distribution $\mathcal{D}$, i.e. $\mathbf{X} \times \mathbf{Y} \sim \mathcal{D}$. A classifier is built to emit a class prediction for any new data point X that belongs in the feature space $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_m$. Therefore a classifier divides the feature space $\mathcal{X}$ into k disjoint regions R such that $\cup_{j=1}^{k} R_j = \mathcal{X}$, i.e. $h(X) = \sum_{j=1}^{k} j\mathbf{1}\{X \in R_j\}$.

2.1 Classification and Regression Trees

A classification tree [3] is an algorithm that forms these regions by recursively dividing the predictor space, more precisely, this procedure performs recursive binary partition. Beginning with the entire feature space, the algorithm selects the variable to split upon and the location of the split that minimizes some impurity measure. Then the resulting two regions are each split into two more regions until some stopping rule is applied. The classifier will label each region with one of the k possible classes.

The traditional labelling process goes as follows; let $p_{rk} = \frac{1}{n_r} \sum_{X_i \in R_r} \mathbf{1}\{Y_i = k\}$, the proportion of the class k in the region r. Then, the label of the region r is the majority class in that region, i.e. if $X \in R_r$, $h_S(X) = \text{argmax}_k(p_{kr})$. For regression trees, the output mean within a leaf node is used as prediction for observations that belong in that node. The impurity measure function can take many forms. Hastie & al. [11] introduces three possible region impurity measures Q_r, the *Gini Index*, the *Deviance* and the *Misclassification error*. For regression trees, the mean squared error is one possible region impurity measure.

When splitting a region into two new regions R_1 and R_2 the algorithm will compute the total impurity of the new regions ; $n_1Q_1 + n_2Q_2$ and will pick the split variable j and split location s that minimizes that total impurity. If the predictor j is continuous, the possible splits are of the form $X_j \leq s$ and $X_j > s$ which usually results in $n_r - 1$ possible splits. For a categorical predictor having q possible values, we usually consider all of the $2^{q-1} - 1$ possible partitions.

The partitioning continues until a stopping rule is applied. In some cases, the algorithm stops whenever every terminal node of the tree contains less than β observations, in other cases it stops when all observations within a region belong to the same class. To prevent overfitting, a deep tree is built and then the tree can be pruned. Tree-pruning is a cost-complexity procedure that relies on considering that each leaf, region, is associated with a cost α. That is, the tree pruning procedure consists of minimizing the cost complexity criterion defined as :

$$C_\alpha = \sum_{r=1}^{|R|} n_r Q_r + \alpha |R|, \tag{1}$$

where $|R|$ is the number of regions, i.e. the number of leaves in the tree. The procedure begins by collapsing leaves that produce the smallest increase in total impurity and this technique will collapse leaves as long as the increase in impurity is less than the cost α of the additional leaf. The α parameter can be determined by cross-validation or with the use of a validation set.

3 Background about missing values

As described in the previous section, a standard assumption in data analysis is that all observations are distributed according to the data generation distribution $\mathcal{D}$. We could think of the missingness itself as a random variable $\mathbf{M}$ also of dimension m that is distributed according to some missingness generating distribution which is a part of $\mathcal{D}$, i.e. $\mathbf{X} \times \mathbf{M} \times \mathbf{Y} \sim \mathcal{D}$.

Different relationships between $\mathbf{M}$ and $\mathbf{X}$ will lead to different missingness structure. Rubin [16] and Little and Rubin [14] defined three types of relationship. Missing completely at random (MCAR) is the simplest structure: $\mathbf{M}$ and $\mathbf{X}$ are independent. Even though this is a very restrictive assumption, many missing value techniques based on imputations were constructed under that assumption. Missing at random (MAR) is much more complicated; it essentially means that the missing pattern is independent of missing observations but can still depend on observed predictors, though this terminology has not always been used consistently [18]. More rigorously, the distribution of the missing pattern is independent of predictors with missing values conditionally on observed values. If the missingness depends on the unobserved values, we say that the data is missing not at random (MNAR). As we will see, the relationship between $\mathbf{M}$ and $\mathbf{X}$ has a considerable effect on the efficiency of the many missing values techniques that exist. As we will see in section 5 (see also Ding and Simonoff [6]), when the missingness depends on the response variable, the data is MAR but the results are quite different than they would be if the missingness depends on observed predictors instead.

3.1 Missing values techniques for Decision Trees

In order the handle missing values, a wide variety of solutions have been proposed for classification trees. Recent surveys ([17],[6],[9],[20]) define in detail most of the techniques that are currently used and compare them using various simulated and real data. Some techniques are only suitable for training, some for prediction and finally, some can deal with missing value in both. Under the assumption that both the observations obtained for training and the observations that need to be predicted are distributed according to the same true data generating distribution $\mathcal{D}$, we would like to use a technique that can handle missing values for both training and prediction.

An important family of approaches to deal with missing value is predictive value imputation (PVI) methods. They aim at estimating the missing value and impute them within both the training and the test set. The simplest imputation consists of replacing the missing values with the mean for numerical predictors or the mode for categorical

predictors. More advanced prediction models have also been proposed, such a linear model, k-nearest neighbours or expectation-maximization (EM). These models use the known predictors to impute values for the missing ones. If the predictors are independent, these approaches will have close to no predictive power. Even though Gavankar [9] and Saar-Tsechansky and Provost [17] raise other problems concerning those techniques, they tend to perform well when there exist correlation between the predictors. Twala [20] demonstrated using simulated data sets the great performances of EMMI [13], an expectation-maximization based imputation algorithm that produces multiple imputations and aggregates the results.

The popular C4.5 implementation [15] has its own way to manage missing data, defined as a distribution-based imputation (DBI). When selecting the predictor to split upon, only the observations with known values are considered. After choosing the best predictor to split upon, observations with known values are split as usual. Observations with unknown values are distributed among the two children nodes in proportionate to the split on known values. Similarly, for prediction, a new test observation with missing value is split intro branches according to the portions of training example falling into those branches. The prediction is then based upon a weighted vote among possible leaves.

The surrogate variable (SV) approach [3] is a special case of predictive value imputation. As explained in [11], during the training process, when considering a predictor for a split, only the observations for which that predictor is not missing are used. After the primary predictor and split point have been selected, a list of surrogate predictors and split points is constructed. The first surrogate split is the predictor and split point pair that best mimics the split of the training data achieved by the primary split. Then the second surrogate split is determined among the leftovers predictors and so on. When splitting the training set during the tree-building procedure or when sending and observation down the tree during prediction, the surrogate splits are used in order if the primary splitting predictor value is missing. Many articles ([7],[6],[17],[20]) showed that the results aren't satisfactory in many cases and Kim and Loh [12] noted the variable selection biased caused by this approach.

The Separate Class (SC) method replaces the missing value with a new value or a new class for all observations. For categorical predictors we can simply define *missing value* as a category on its own and for continuous predictors any value out of the interval of observed value can be used. This technique is proved to be the best by Ding [6] when there is missing values in both the training and the test set and when observations are missing not at random (MNAR) or when this missing value depends on the response. Twala and al. [21] also came up with similar results with a generalization of the separate class method named Missing Incorporated in Attribute (MIA).

Finally, reduced-feature models are praised by Saar [17] when missing values appears only for the prediction process. This technique relies on using only know predictors of the new observation we are trying to classify. A tree is built using only the know predictors of the new observation. If multiple observations contain different missing pattern then multiple trees are built to classify the various observations. It shares a great deal of similarities with lazy decision trees [8] as both models tailor a classifier to a specific observation and uses only known predictors to do so.

Our proposed algorithm differs from imputation methods as it only uses known information to build the classifier instead of using prediction to replace missing values. It also differs from reduced-feature models as it not only uses the known values but also utilize the fact that we know some predictors are missing. Finally, our algorithm shares similarities with separate class models, they both perform well under the same conditions and can both lead to the exact some tree structure. On the other hand, BEST identifies the missing pattern using other predictors rather than including this information about missingness within the predictor containing missing value. Doing so, our approach leads to more interpretable results but also the ability to identify the importance of the missing pattern rather than credit the effect of the missing pattern to the predictor containing missing values.

4 Branch-Exclusive Splits Trees

We now introduce our proposed new algorithm, BEST. The purpose of BEST is to utilize the tree structure itself in order to manage some missing data or some special structure among predictors.

As we earlier explained, a classification tree aims at partitioning the feature space and labelling the resulting regions. CART does so by looking through all the possible splits and selecting the one that minimizes some prespecified error measure. When using BEST some predictors are available to split upon only within some regions of the feature space. From a tree perspective, some split variables are exclusive to some branches in the classification tree. By doing so, some missing data can be managed as they will never be considered for splitting. Similarly, some insight on the data structure can be used to force some variable to be split upon before others. The result is a tree-structured classification model where certain split variables are branch-exclusive. All of the construction described below could also be used for regression trees.

4.1 Motivating examples

Let us now give some examples of data sets for which the algorithm BEST is suitable. Our first example is the motivating data set, it concerns information regarding the performances of students in a university. The data set was provided to us by the Univeristy of Toronto and was first introduced and analysed by Bailey and al. [1]. It was later analysed by Beaulac and Rosenthal [2] where the goal was to predict whether or not a student would complete its program and if it does, which department would the student major in. The predictors available represent the number of credits and grades obtained in the departments during the first two semesters. Understanding the importance of these predictors was also a question raised by the authors. Obviously a student did not obtain grades in many departments has he can only register to a limited number of courses within a year. In this situation, many grade variables were missing for every student. The proposed algorithm handles that problem by considering the averaged grade obtained

in a department only for a student registered in that department. BEST will force the classification tree algorithm to split upon the *registered* predictor to begin, and will then allow splits on the *grade* predictor for the group of students that took at least 1 course within the department. Therefore, the *registered* variable is used to define the region where the *grade* variable is available for the partitioning process.

Our second example concerns a medical data set containing a list of conditions for each patient, together with details about those conditions as needed. For example, does the patient have diabetes, and if it does, then is it type I or type II ? The reason why *diabetes type* is missing for patients with no diabetes is explained by the *diabetes* variable. We could also have a variable indicating the degree of sickness and a variable representing the treatment selected. If patients are only treated if they are very sick, then the treatment selected is only defined for patients with a degree of sickness high enough. If both the degree of sickness and the treatment are important predictors for the recovery of patients, we cannot combine them in one predictor because information would be lost, but we can make sure to analyse the effect of the treatment only for patients who received treatment with the proposed algorithm.

If a data set contains missing values on predictor X_j but no predictors can help define the region with no missing value, we can add a new predictor X_{m+1} to the model as our *gating variable*. This new predictor is a dummy variable such that $X_{m+1}(i) = 0$ if $X_j(i)$ is missing and 1 if not. Then, BEST will only consider splitting on X_j in the subspace defined by $X_{m+1} = 1$. Multiple dummy variables are added to the model if multiple predictors contain missing values. Doing so allows us to analyse the individual importance of the missing patterns.

4.2 Theoretical justifications

Formally, the loss of a classifier h is defined as :

$$L_{\mathcal{D}}(h) = \mathbf{P}_{\mathcal{D}}[h(X) \neq Y]$$

Since the data generating distribution $\mathcal{D}$ is unknown, the empirical loss computed with the data set S is typically used as an estimator of the true loss :

$$L_{\mathcal{S}}(h) = \frac{|\{i \in [n] : h(X_i) \neq Y_i\}|}{n}$$

Usually, a subset of models $\mathcal{H}$ is selected in advance and since we don't know the true data generating distribution $\mathcal{D}$, most learning algorithms, including classification trees, are trying to identify the classifier $h \in \mathcal{H}$ that minimizes the empirical loss $L_S(h)$. $\mathcal{H}$, the hypothesis class, is a set of function mapping from $\mathcal{X}$ to $\mathcal{Y}$. Because we cannot use the true loss, we can decompose the empirical loss in a manner to observe a bias-complexity tradeoff. Suppose $h_S = \underset{h \in \mathcal{H}}{\operatorname{argmin}} L_S(h)$, then :

$$\begin{aligned} L_D(h_S) &= \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + (L_D(h_S) - \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)). \\ &= e_{\text{app}}(\mathcal{H}) + e_{\text{est}}(h_S). \end{aligned} \tag{2}$$

In [19], $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) = e_{\text{app}}$ is defined as the approximation error. It is the minimum achievable loss within the hypothesis class. This term measures the impact of the bias produced by the choice of $\mathcal{H}$. The second term , $(L_D(h_S) - \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)) = e_{\text{est}}$, is defined as the estimation error. This error is caused by the use of the empirical loss instead of the true loss when selecting the best classifier h. This error term decreases as the sample size increases and it increases as the size of the hypothesis class $\mathcal{H}$ increases.

Since the goal is to minimize the total loss a natural tradeoff emerges from equation 2. A vast and large hypothesis class $\mathcal{H}$ leads to a wider choice of functions and therefore decreases the inductive bias and the approximation error with it. On the other hand,

the larger the hypothesis class is, the more difficult it is to identify the best hypothesis within this class and thus leads to larger estimation error.

Our proposed algorithm aims at obtaining a better classifier by restricting the hypothesis class to a smaller one without increasing the bias. Suppose $\mathcal{H}_T$ is defined as the set of all tree-structured classifiers that can be constructed using the methodology introduced by Breiman & al. [3]. Then, what BEST really consists of is a new algorithm that aims to find the best classifier in a new hypothesis class $\mathcal{H}_B$ that contains all the tree-structured classifiers that respect a set of conditions regarding the order that variables can be split upon. Therefore, we have $\mathcal{H}_B \subset \mathcal{H}_T$.

Since the complexity of $\mathcal{H}_B$ is smaller than the complexity of $\mathcal{H}_T$, with the same sample size, the estimation error of BEST should be smaller. In order for the total loss of the proposed method to be smaller we must also look at the approximation error : $\min_{h \in \mathcal{H}_B} L_{\mathcal{D}}(h)$. If our assumptions about the real data generating process $\mathcal{D}$ are true, then the best classifier in the hypothesis class $\mathcal{H}_T$ of all classification tree is also contained in $\mathcal{H}_B$, i.e. $\operatorname{argmin}_{h \in \mathcal{H}_T} L_D(h) \in \mathcal{H}_B$. This would imply that $\min_{h \in \mathcal{H}_T} L_{\mathcal{D}}(h) = \min_{h \in \mathcal{H}_B} L_{\mathcal{D}}(h)$. Of course, assuming $\operatorname{argmin}_{h \in \mathcal{H}_T} L_D(h) \in \mathcal{H}_B$ is audacious but in some cases this assumption is really natural. Suppose S is a data set, $h_S(\mathcal{H}_T)$ is the classifier that minimizes the empirical loss on $\mathcal{H}_T$ and $h_S(\mathcal{H}_B)$ is the classifier that minimizes the empirical loss on $\mathcal{H}_B$, the two results discuss above lead to :

$$\begin{aligned} L_D(h_S(\mathcal{H}_T)) &= \min_{h \in \mathcal{H}_T} L_{\mathcal{D}}(h) + e_{\text{est}}(h_S(\mathcal{H}_T)). \\ &= \min_{h \in \mathcal{H}_B} L_{\mathcal{D}}(h) + e_{\text{est}}(h_S(\mathcal{H}_T)). \\ &\geq \min_{h \in \mathcal{H}_B} L_{\mathcal{D}}(h) + e_{\text{est}}(h_S(\mathcal{H}_B)). \\ &= L_D(h_S(\mathcal{H}_B)), \end{aligned} \tag{3}$$

which implies that the under the assumption we've made we would not only naturally manage missing values but also reduce the loss. If the assumption about the best classifier is false, we might increase the loss, and the assumption itself is impossible to verify.

Therefore, the behaviour of the algorithm under multiple scenarios will be tested in section 5 with simulated data.

4.3 Algorithm

With the exception of a few steps, BEST works exactly like CART. The algorithm takes as input the full data set S, the tuning parameter β and a list containing the predictor availability structure V. First, S is set as the root node, the first node that goes through the following steps. The algorithm begins by verifying a set of conditions before proceeding with the partitioning process. The first condition (C1) is that the node contains more than β observations. Then, the next condition (C2) is that the observations in the node have different labels; this condition makes sure that the algorithm has a reason to split the data. Finally, the last condition (C3) is that at least one of the available predictors takes different values among the observations in the node; this is to guarantee that the algorithm can actually split the data.

If at least one condition is not respected, then the node is defined as a leaf node, a label is assigned to that node for prediction purposes and the partitioning process is stopped. Usually the class that represents the majority in a leaf node is selected as label for that node, but one could define different label assignment rules.

If all the conditions (C1, C2 and C3) are respected then the partitioning process begins. The algorithm will go through all the available predictors. For a predictor j, the algorithm will go through all the possible partition s of the node with respect to the predictor j and will compute the total impurity of the resulting two nodes $n_{r_1}Q_{r_1}+n_{r_2}Q_{r_2}$. Any node impurity measure Q can be used. BEST then selects the predictors j and the split s that minimizes the total impurity and create two children nodes by splitting the data according to s.

The last step is to update the list of available predictors for the children nodes. There exist multiple possible structures that can contain this information but we've settled on

a list V. Suppose we have m predictors, then $V[0]$ is a vector of size m where $V[0](j) = 1$ if the jth predictor is available to be split upon in the root node.

Suppose now that the jth predictor is selected as the splitting predictor. $V[j]$ contains the information necessary to update the list of available predictors. If j is a continuous predictor, $V[j]$ contains a threshold value, if the splitting point s is greater (or less) than the threshold, then the child node containing the observations such that $X_j > s$ ($X_j < s$) have their *available predictors* updated according to the vector contained in $V[j]$. If j is a categorical predictor, then $V[j]$ contains a matrix where the line i represents the update needed on the node containing the value i after the partitioning.

Here is a pseudo-code of the proposed algorithm :

Algorithm : BEST(S,β,V)

1. Starting with the entire data set S and the set of available predictors $V[0]$.
2. Check conditions (C1, C2 and C3).
3. **if** (any condition is false) :
   Assign a label to the node and exit.

else if (all conditions are true) :
   for (j in all available predictors):
      for (s in all possible splits) :
         Compute total impurity measure.
   Select variable j and split s with minimum impurity measure and split the node into two children nodes.
   Update the available predictors for both children nodes.
   Repeat steps 2 & 3 on the two children nodes.

In addition, the resulting tree can be pruned as explained in section 2.1, and constructed with any splitting rule and any stopping rule. Since one of the goals of this new algorithm is to produce accurate but also interpretable models we did not discuss forests so far, but the proposed tree construction procedure can be used to build any type of forest.

5 Experiments : Simulated data sets

Since we observed that the data size isn't an important factor when comparing different missing value treatments to each other, we fix the training set size to 1000 observations, and the test set size to 500 observations, throughout our simulations. Our data is generated from a classification tree. The tree is of depth 4 and the response variable can take on 4 different classes. Labels are assigned according to a set of 8 predictors, 4 of them are categorical, 4 of them are numerical. They all have different degrees of importance, in fact, the response is actually totally independent of 2 predictors.

To begin, we will generate a complete data set, fit a single pruned decision tree and compute the accuracy on the test set. Then, a missing pattern is applied on both the training and the test set and we will successively use one missing value treatment and compute the prediction accuracy on the test set. We will compare 6 methods; (1) the Distribution Based Imputation proposed by Quinlan [15], (2) a simple single variable imputation, either the mode for a categorical predictor or the mean for a numerical one, (3) a predictive value imputation using know predictors; EM for numerical predictors and multinomial logistic regression for categorical one [22], (4) the separate class approach, (5) the surrogate variable technique introduced by Breiman and al. [3] and finally (6) BEST, our proposed approach. Since the Reduced-Feature Model was the least accurate in every single experiment we've done, we decided not to include it in the following plots to improve readability. Multiple Imputation methods were also left out as they actually create forests.

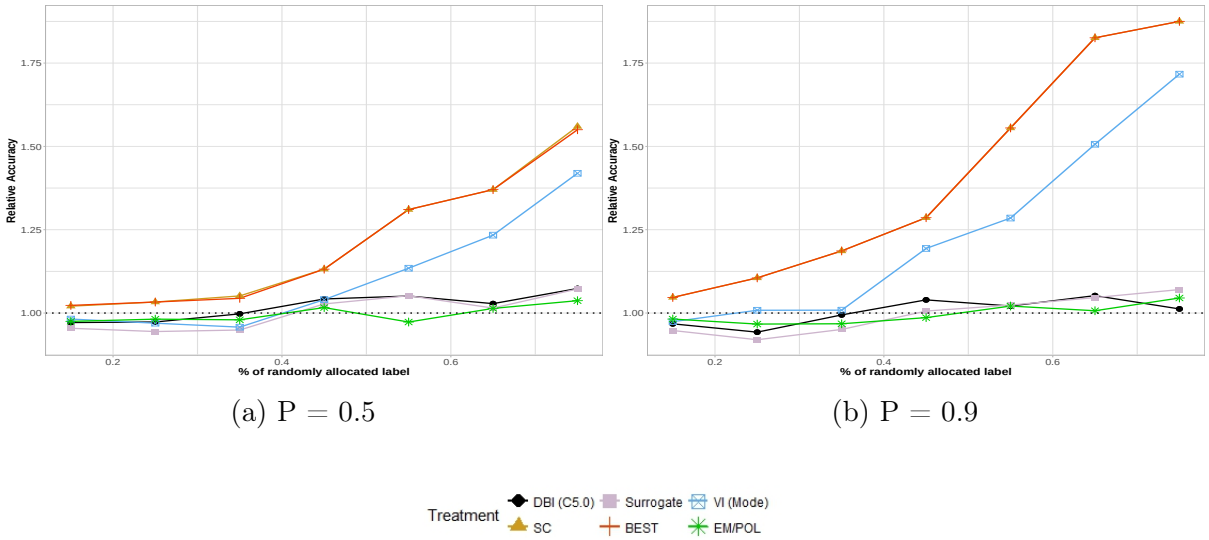
We will examine the relative accuracy; which is the ratio of the prediction accuracy on the test set for the model trained with missing values over the accuracy obtained by the model trained without missing value. For every set of characteristics tested, the relative accuracy will be averaged over 10 randomly generated data sets.

Finally, we did not test the algorithm run time for two reasons. First, our code is entirely written in R and is not optimized yet but more importantly, there is no theoretical reason to believe that BEST is either slower or faster than any other Decision

Tree algorithm. The variable availability is considered only once when a new region is created and is executed pretty efficiently. In some cases BEST is even faster since it only analyse a subset of predictors.

5.1 MAR : Missingness depends on the response

According to Ding and al. [6], the relationship between the missing pattern and the response variable has a great effect on the results obtained from different missing value treatments. The overall relation between the covariate and the response is also important since the less correlated they are the more information we relatively gain from analysing the missing pattern. So we begin by analysing the effect of these two parameters. One predictor is randomly selected every iteration and the censoring process is then applied. The censoring process goes as follows; one of the four labels is randomly selected, and a fixed proportion of the observations with this label gets their previously selected predictor rendered missing. The following plots represent two different proportions of missing observations and the horizontal axis represents different proportions of randomly labelled data, the less randomly assigned label there is, the more the set of predictors explain the response. Since the distribution of the missing pattern is conditionally independent of the predictors with missing values, this is considered MAR. In this experiment we've used a dummy variable as the *gating variable*.



(a) P = 0.5 (b) P = 0.9

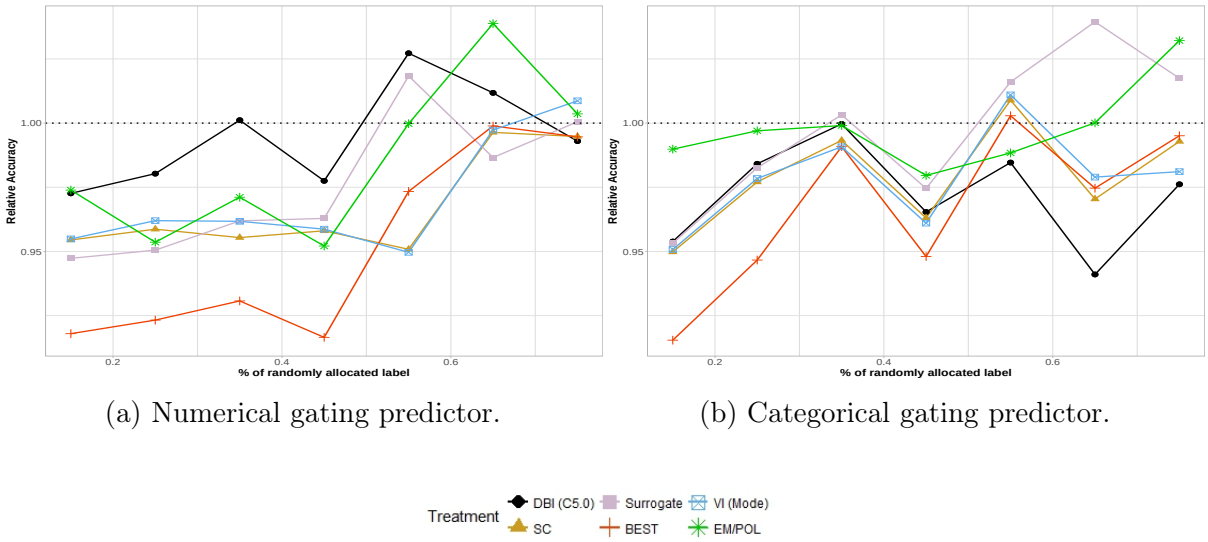
Figure 1: Missingness depends on the response.

The results are somehow as expected. As the proportion of randomly assigned label increases, so does the relative information gained from analysing the missing pattern. Therefore, models like BEST and SC shine as they utilize the fact that there is missing values instead of trying to impute them. BEST and SC approaches have similar results as their relative accuracy curves are almost perfectly superimposed and we can see in both 1a and 1b that these approaches have relative accuracy curves that are greatly superior to any other missing value technique. It is also interesting to notice the high performances of the simple single value imputation. Our experiments revealed that when the predictor containing missing value is continuous, replacing missing values with the mean actually behave like the separate class approach because only the missing values will exactly take the value of the mean. If the predictor with missing value is categorical, replacing missing values with the mode will make the observation with missing value undistinguishable from observations that truly belong to that class.

5.2 MAR : Missingness depends on observed predictors

In this set up, the missing values depend on an observed predictor. For the first experiment two predictors are randomly picked, one will have missing value based upon the other. If the predictor guiding the missing process is categorical, a subset of class

is picked and when the guiding predictor value is one of those classes, then the other predictor selected will have 75% of it's observation rendered missing. If the guiding predictor is is numerical, a value is randomly selected within its range and when the value of the guiding predictor is bellow that threshold, 75% of the other predictor selected will be missing . In our second experiment, the guiding predictor must be numerical, and once again a threshold is randomly drawn, but this time, everytime the guiding predictor value is below the threshold, the other predictor is rendered missing. Therefore we can use the guiding predictor as the gating predictor in the BEST algorithm.



(a) Numerical gating predictor.

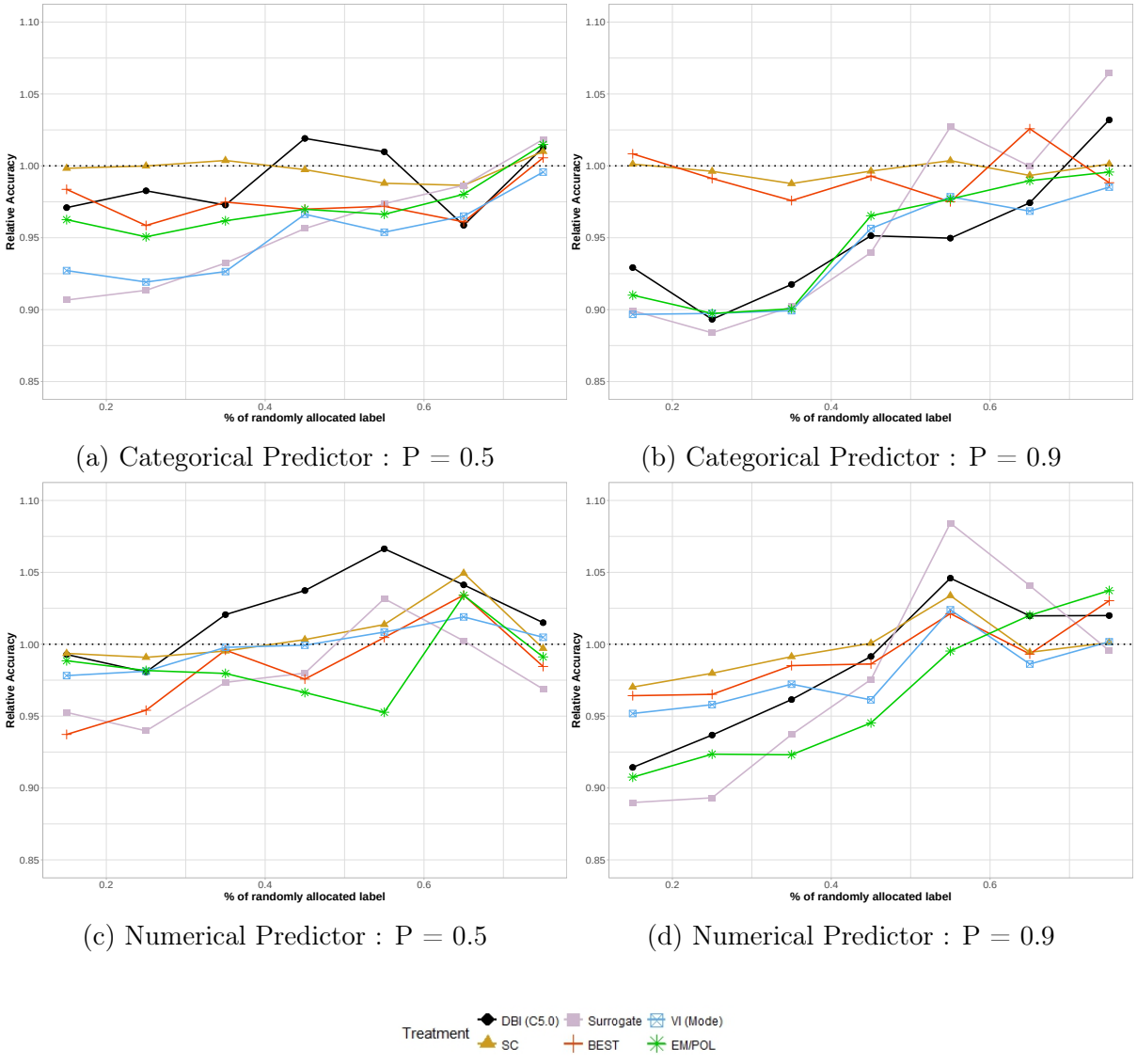
(b) Categorical gating predictor.

Figure 2: Missingness depends on a numerical predictors.

When the missing pattern depends on observed predictors we've noted that BEST is underperforming on average as shown in figure 2a and 2b. The reason for BEST's lower average performance is that, while it performs well in most cases, it performs extremely poorly in a small number of cases when the gating variable has no predictive power. In that case the algorithm never partitions the data according to this variable and thus the variable with missing value is never available for the data partitioning process. Our proposed algorithm therefore looses access to a variable that might be informative, which is a limitation of BEST.

5.3 MNAR

To test how the proposed algorithm would perform when values are missing not at random, we've set up four experiments. To begin we sample a categorical predictor, then a random subset of categories are selected and either 50% or all observations with those categories are rendered as missing. For the last two tests, a numerical predictor is randomly selected, then a threshold is randomly selected within the range of the predictor and then either 50% or all observations below that threshold are rendered missing.



(a) Categorical Predictor : P = 0.5

(b) Categorical Predictor : P = 0.9

(c) Numerical Predictor : P = 0.5

(d) Numerical Predictor : P = 0.9

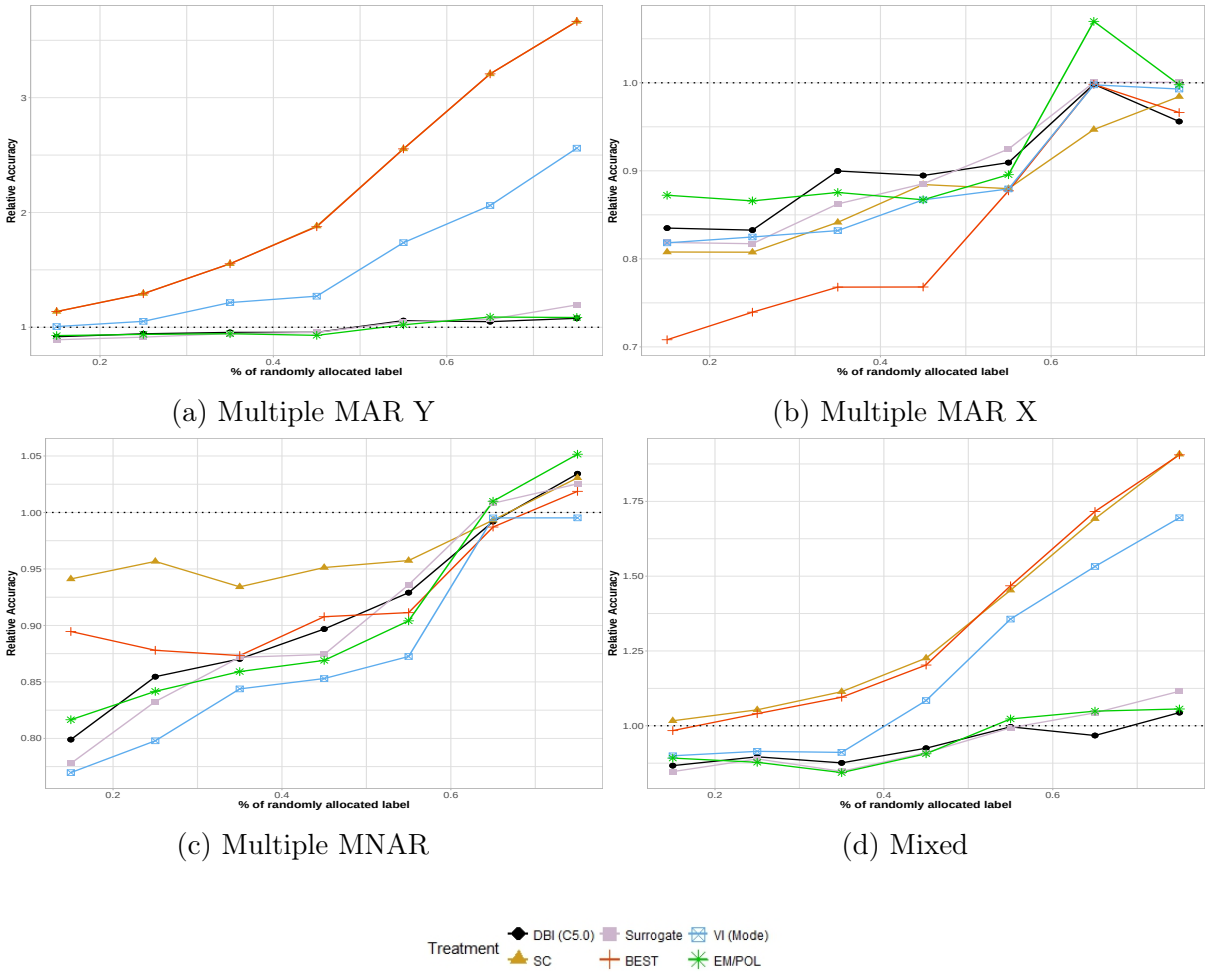
Figure 3: Missingness depends on missing values.

Here we observe that relative accuracy for BEST is above most other techniques in all

four plots. We notice in figure 3b and 3d that BEST offers better performances when more observations are missing. In those cases, the gating variable becomes a more important predictor and is more likely to get partitioned upon.

5.4 Multiple missing predictors

For this experiment we wanted to see how the various technique to handle missing value would perform compared to BEST when there is multiple predictors containing missing value. Once again, we will be looking at the relative accuracy while varying the amount of randomly assigned labels. For the 4 experiments we've set up, three predictors contain missing values. The first graph represent the case when there is three missing predictors that are MAR but explained by the response. The second graph presents the results when there is missing values at random based of other observed predictors for three predictors. The third graph contains the results when three predictors contains missing value that are missing not at random. Finally, in the forth test, the data contains one predictors missing at random based of the response, one predictors with missing value based of other predictors and finally one predictor with value missing not at random.



(a) Multiple MAR Y

(b) Multiple MAR X

(c) Multiple MNAR

(d) Mixed

Figure 4: Missingness depends on missing values.

Imposing missing value on multiple variables highlights the properties we've already observed. When the missingness pattern is dictated by the response, both the SC approach and BEST are performing better than anything else. In figure 4a, their curves are perfectly superimposed, and the relative accuracy for those methods are way higher than one, which really exposes well the strength of those methods. In figure 4b the weaknesses of BEST are once again noticeable. When the randomly selected missing guiding variables is a non-informative predictor, then the dummy gating variable has no predictive power, thus it is never selected has the partitioning variable which results in BEST algorithm never gaining access to the variables with missing values.

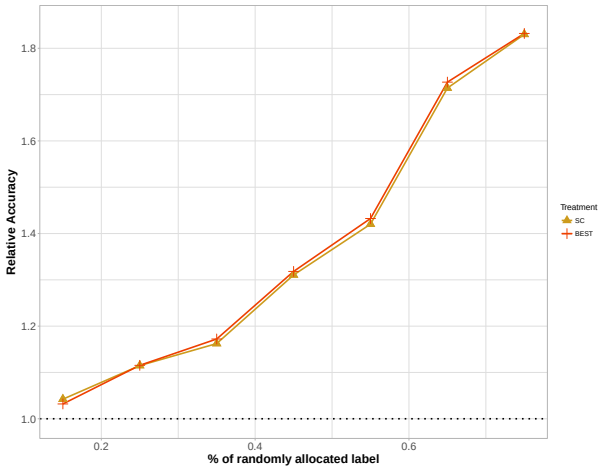
We also observe in figure 4c slightly above average relative accuracy for the BEST algorithm when the data contains values that are missing not at random. Finally, as it

is observed in figure 4d, when there is multiple types of missing pattern, the information that can be gained by considering the missing pattern as potentially informative is really important. Discarding the possibility that the missing pattern might be informative could cost a lot as we observe in figure 4d since BEST and the SC approach do capture the effect of these patterns and out performs every other techniques.

5.5 Random Forests and Variable Importance

Here we wanted to build a small example where Random Forests are fit and used to analyse the variable importance. Random Forests are popular in exploratory analysis as the variable importance tools that were developed for this model became popular. Here we will quickly discuss how BEST produces more accurate variable importance computation than the Separate Class approach.

As we've seen in the previous experiments, when the missing pattern depends on the response itself, both the Separate Class approach and BEST outperform any other Missing Value techniques. Here we've created an example where the missing pattern depends on the response, used either the SC approach of BEST to handle missing value and we've built Forest out of those trees. As expected, both technique offers similar performances :



When the values for a predictor are conditionally missing at random given the response, the missing pattern is itself a good predictor. If we would like to identify the usefulness of the pattern as a predictor, a Random Forest of trees built under the SC approach

would fail to distinguish between the effect of the observed value for that predictor and the effect of the missing pattern rendered on it. Since BEST actually uses a variable to define the region with missing values, either with another predictor or a user-created dummy variable, this gating variable importance will better represent the predictive power of the missing pattern.

% Rand.	Data	X_1	X_2	X_3	X_4	$\mathbf{X_5}$	X_6	X_7	X_8	$\mathbf{X_9}$
15 %	Complete	25.74	34.15	20.24	30.33	**4.98**	78.81	46.43	45.73	-
	SC	31.83	43.18	28.17	33.15	**155.12**	69.53	39.40	35.09	-
	BEST	30.31	40.89	26.10	33.02	**3.09**	64.56	45.04	35.72	**145.84**
45 %	Complete	15.16	26.66	12.12	24.12	**10.77**	32.38	22.28	20.36	-
	SC	17.09	31.64	15.31	29.05	**190.93**	26.23	15.12	16.05	-
	BEST	16.43	25.42	13.33	25.91	**5.92**	23.34	15.57	18.73	**189.14**
100 %	Complete	8.93	16.30	8.46	16.38	**21.17**	20.28	19.33	20.20	-
	SC	11.04	19.47	10.49	19.98	**207.31**	11.03	10.88	11.04	-
	BEST	9.97	16.89	9.56	16.73	**8.94**	11.70	10.40	10.91	**198.72**

Table 1: Variable Importance table : Computed using the GINI decrease importance

Once again, the data is generated according to a tree and we've looked at the variable importance for various proportions of randomly assigned labels in each leaf. We've built a Random Forest using the complete data set and computed the GINI decrease importance. Then we've randomly selected one of the four labels, and the predictor X_5, a predictor of low importance according to the GINI decreases under the complete data set, is rendered missing depending on the value of the response. Since the SC approach uses the predictor containing missing value to identify observations containing missing value then it incorrectly identifies X_5 as the most important predictor. Using BEST, we can easily observe that the missing pattern is the important predictor and that X_5 is actually of low importance when observed as it should be according to the complete data variable importance.

6 Experiments : Grades Data set

The data set mentioned in section 4.1 was analysed using BEST. Once again, the accuracy of the proposed algorithm is compared to other missing value techniques. To begin, we've

tried predicting if a student completed its program using its first year of courses and results. The data set contains 38842 observations. Our set of predictors consist of the number of credit attempted in all the departments and the average grade obtained in those. The number of credit is a numerical variable that serves as the gating variable. If the number of credits attempted in a department is greater than 0 for every observation in a region then BEST acquires access to the grade variable. We've sampled the data set to form training sets of different sizes and used all the remaining observations to assess the accuracy. We've repeated the process 10 times and averaged the results. We did not includ the single value imputation because we expect this technique to produce the same result as the SC approach since all predictors are numerical. We did not include the imputation produced by the mice package [22] as the package was incapable of handling the data set.

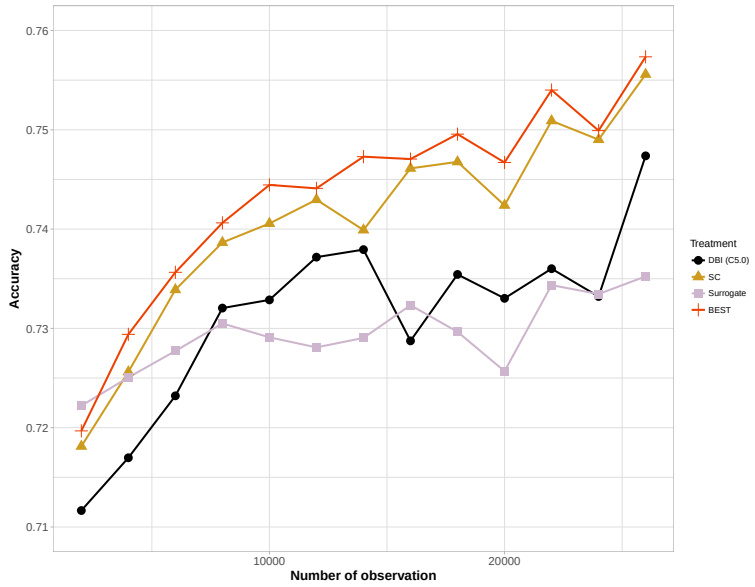


Figure 5: Accuracy when predicting the program completion.

As we can observe, BEST is the most accurate technique to handle missing value for that true data set. This data set also provides a good example for the increased interpretability of the BEST classifier as we previously discussed. For the SC approach we've replaced missing grades by 101. In most of the trees created using this approach, splits performed on the grades variable were often of the form : *students with grades above SP and students with missing grades* are partitioned from *students with grades*

below SP. Even though the algorithm can use the split point 100 later on to isolate students with missing grade the first few splits generated by the SC approach often feels counter-intuitive. BEST achieves the same accuracy while keeping the splits logical and interpretable by looking at the number of courses attempted first and looking at the grade only in a region where every student attended at least one course. If interpretability is considered a strength of Decision Trees, then BEST is better at preserving this strength.

Another reason why we might prefer using BEST in this analysis is its ability to rightfully identify the variable importance. As we discussed in section 4.1, we were interested in the importance of the predictors. Therefore BEST is an improvement as it truly identifies the importance of the gating variables as we've shown in section 5.5. In this case we were able to distinguish the importance of registering to courses in a department from the importance of the grades obtained in that department.

Here, we will look at the 26 488 who completed their program. Using the same set of explanatory variables, we will try to predict the department they majored in :

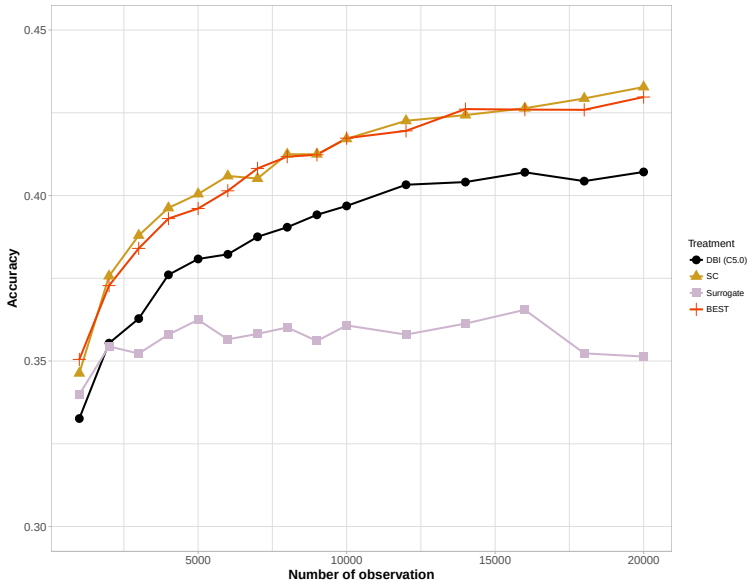


Figure 6: Accuracy when predicting the choice of major.

Here we observe closer results from the two best performing algorithm. BEST and SC have statistically indistinguishable performances and are the top performers. Even thought the results are a lot closer between BEST and SC, our proposed algorithm pro-

duced trees that were more interpretable and could be used to produces a non-biased variable importance analysis.

7 Discussion and conclusion

We've constructed a modified tree-building algorithm that lets the users decide the regions of the feature space where variables are available for the data partitioning process. We have focused on using this feature to manage missing values. BEST has the elegant property of analysing a variable only when values are known without assuming any missingness dependence structure. It produces highly interpretable trees and achieves higher accuracy than most missing value handling technique in cases we've identified using simulated data. Even though BEST shares similarities with the separate class technique, BEST leads to a more accurate variable importance analysis and produces more interpretable and intuitive classifiers.

BEST suffers from a weakness when the gating variable has no predictive power. In those cases, the algorithm will never choose to split upon this variable, thus never reaching the branch-exclusive variable and almost surely reducing the accuracy of the model. This problem can lead to serious decrease in accuracy in some simple cases where the data is MCAR. In this scenario, the missing pattern is non-informative and thus the dummy gating variable will never be selected as the split variable by the algorithm. The same problem can also be observed when the missing pattern depends on other predictors, if those guiding predictors are uninformative. This weakness is intrinsic to the algorithm as it is caused by the greedy nature of decision trees. Since it can only see the reduction in impurity gained with a single partition, a classic decision tree approach cannot perceive the accuracy gained by the combination of two successive partitioning. Fortunately, as we've previously discussed, there already exist multiple techniques to handle data MCAR and we can count on cross-validation in order to help us select the best missing data handling technique. This weakness is also barely noticeable if the predictors we analyse are all informative.

The results produced by BEST were quiet satisfactory when the algorithm was used on the motivating student-grades real data set. We were able to achieve higher accuracy than with any other technique while obtaining a more interpretable classifier. Since we wanted to identify if grades within a department were more important than grades in other departments, BEST was an improvement as it answers that research question by providing a more reliable variable importance analysis than the separate class approach previously used [2].

For future work, considering pairs of consecutive splits would be a great improvement and would negate the limitation caused by the greedy procedure of tree construction algorithm. This would overcome the above weakness concerning gating variables with no predictive power. However, naive implementation of this modification would lead to much larger run times, so we plan to investigate it further elsewhere.

Acknowledgements

We are very grateful to Glenn Loney and Sinisa Markovic of the University of Toronto for providing us with the anonymised students grade data. The authors also gratefully acknowledge the financial support from NSERC of Canada.

References

[1] Michael A. Bailey, Jeffrey S. Rosenthal, and Albert H. Yoon. Grades and incentives: assessing competing grade point average measures and postgraduate outcomes. *Studies in Higher Education*, 41(9):1548–1562, 2016.

[2] C. Beaulac and J. S. Rosenthal. Predicting University Students' Academic Success and Choice of Major using Random Forests. *ArXiv e-prints*, February 2018.

[3] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984. new edition [**?**]?

[4] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] Yufeng Ding and Jeffrey S. Simonoff. An investigation of missing data methods for classification trees applied to binary response data. *J. Mach. Learn. Res.*, 11:131–170, March 2010.

[7] A. J. Feelders. Handling missing data in trees: Surrogate splits or statistical imputation. In *PKDD*, 1999.

[8] Jerome Friedman, Ron Kohavi, and Yeogirl Yun. Lazy decision trees. 1, 09 1997.

[9] S. Gavankar and S. Sawarkar. Decision tree: Review of techniques for missing values at training, testing and compatibility. In *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, pages 122–126, Dec 2015.

[10] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, Apr 2006.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.

[12] H. Kim and W.-Y. Loh. Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96:589–604, 2001.

[13] Joseph L. Schafer and Maren K. Olsen. Multiple imputation for multivariate missing-data problems: A data analyst's perspective. 33, 07 2000.

[14] Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc., 2002.

[15] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[16] Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[17] Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. *J. Mach. Learn. Res.*, 8:1623–1657, December 2007.

[18] Shaun Seaman, John Galati, Dan Jackson, and John Carlin. What is meant by âĂIJmissing at randomâĂİ? *Statist. Sci.*, 28(2):257–268, 05 2013.

[19] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

[20] Bhekisipho Twala. An empirical comparison of techniques for handling incomplete data using decision trees. *Appl. Artif. Intell.*, 23(5):373–405, May 2009.

[21] Bhekisipho Twala, M.C. Jones, and David Hand. Good methods for coping with missing data in decision trees. 29:950–956, 05 2008.

[22] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011.