Regression with Missing Data, a Comparison Study of Techniques Based on Random Forests

Irving Gómez-Méndez[1,*] and Emilien Joly[2]

[1]Centro de Investigación en Matemáticas, AC (CIMAT)
[*]Corresponding author: irving.gomez@cimat.mx
[2]Centro de Investigación en Matemáticas, AC (CIMAT), emilien.joly@cimat.mx

Abstract

In this paper we present the practical benefits of a new random forest algorithm to deal with missing values in the sample. This follows a first paper on the consistency of this new algorithm. Many different missing value mechanisms (such as MCAR, MAR, NMAR) are considered and simulated. We study the quadratic errors of our algorithm and compare it to the most popular missing value algorithms in the literature. A quick algorithmic complexity study is also given.

1 Introduction

Random forests and recursive trees are widely used methods in applied statistics and computer science. The popularity of recursive trees relies on their easy interpretability, the fact that they can be used for regression and classification tasks, a small number of hyper-parameters to be tuned and their ability to model arbitrarily complex relations between the input and the output space. A random forest combines several randomized recursive trees, maintaining the positive aspects of trees with an improvement in prediction accuracy at a cost of lost in interpretation. This technique accepts a straightforward parallelization which has made it one of the most popular tools for handling high dimensional data sets. It has been successfully involved in various practical problems, including chemioinformatics, ecology, 3D object recognition, bioinformatics and econometrics. A detailed list of applications as well as a review on random forests can be found in Biau and Scornet (2016). The present work is focused on the ability of random forests to deal with missing values.

Approaches to handle missing values regarding the use of random forests could be classified in three groups. In the first group there are algorithms that impute the missing values without using recursive trees or random forests, like k nearest neighbor imputation (knn-imputation) (Troyanskaya et al., 2001) or multiple imputation chained equations (MICE) (Van Buuren et al., 2006), and then construct a random forest with the imputed data set. The second group consists in built-in methodologies which use the observations with missing values directly in the construction of the recursive trees like surrogate splits (Breiman et al., 1984), missing incorporated in attributes (MIA) (Twala, Jones, and Hand, 2008) or the algorithm proposed by Gómez-Méndez and Joly (2020) which is reintroduced in this paper. The third group is composed by algorithms that construct whole random forests and use them directly or through some extra structures like the proximity

Mathematics Subject Classification 2020. Primary ...; Secondary ...
Key words and phrases. ...

matrix to impute the missing values, the methods proposed by Breiman (2003), Ishioka (2013), and Stekhoven and Bühlmann (2011) are algorithms belonging to this third group.

Handle missing values through different algorithms has taken attention for the last decades with several simulation studies comparing the performance of distinct methods (Feelders, 1999; Farhangfar, Kurgan, and Dy, 2008; Rieger, Hothorn, and Strobl, 2010; Hapfelmeier, Hothorn, and Ulm, 2012; Josse et al., 2019). However there is little published works comparing the performance of algorithms based on the construction of whole random forests and built-in methodologies. The motivation of this work is to investigate the performance of these two groups. Of special interest is to study the performance of a computationally exhaustive approach presented here, and for which Gómez-Méndez and Joly (2020) have proven its consistency under an MCAR mechanism and a generalized additive model.

The rest of the paper is organized as follows. In Section 2 we present a review on previous papers that compares techniques based on recursive trees and random forests to handle missing values, we formally introduce the CART criterion and the data-missing mechanisms, describe our proposal to build recursive trees making use of the missing values and calculate its computational complexity. In Section 3 we describe the methods that are compared in the simulation study, which is explained in detail in Section 4 while Section 5 presents the result. Section 6 explores possible simplifications for our proposal. Finally Section 7 presents the conclusions.

2 Literature Review and Introduction of Concepts

2.1 Review of the Literature

The study developed by Feelders (1999) is one of the first to present a result on missing values using recursive trees. This work compares the error rate between surrogate splits in a single decision tree, and the imputation procedure presented by Schafer (1997) and Schafer and Olsen (1998). Two data sets are considered in the study, given by the so-called *Waveform* data set (presented by Breiman et al. (1984)) and the *Prima Indian Datasets* (available at the UCI machine learning repository). The percentage of missing values varies between 10% and 45% for the first data set, where the missing values are introduced accordingly to an MCAR mechanism. On the other hand, 10% of the observations present missing values in the second data set. Feelders (1999) concludes that the imputation procedure yields significantly less missclassification error rate.

The work of Farhangfar, Kurgan, and Dy (2008) presents a comparison of classification methods like support vector machines, k nearest neighbors and the decision tree method C4.5 (Quinlan, 1993) applied to missing data, and imputation algorithms, including single imputation and MICE. In total 15 data sets are considered, inducing missing values with up to 50% of the observations per variable being set missing. The authors conclude that the application of MICE leads to better results in most of the instances.

Rieger, Hothorn, and Strobl (2010) compare surrogate splits introduced in the conditional inference forests (Hothorn, Hornik, and Zeileis, 2006) with *knn*-imputation, with no clear advantage for either of the methods. This study considers classification and regression problems with three different correlation structures and seven schemes for missing values. However, the percentage of missing values is kept constant.

Hapfelmeier, Hothorn, and Ulm (2012) present a comparison study using trees built with the CART criterion, conditional inference trees and their corresponding random forests, focusing on surrogate splits to handle missing values and the use of MICE to impute missing values. The authors consider 12 real life data sets, half of them for regression and the other half for classification, 8 of the data sets already present missing values while in the rest 4 data sets missing values are

induced. Arguing that Rieger, Hothorn, and Strobl (2010) found similar results for MCAR and MAR mechanisms, the missing values are solely introduced according to an MCAR mechanism of missingness, considering missing rates between 0% (benchmark) to 40%. Their results do not show a clear improvement by using multiple imputation, with MICE even producing inferior results when missing values are limited in number and are not arbitrary spread across the data. The results also show a similar result between trees constructed with the CART criterion and conditional inference trees.

The simulation developed by Josse et al. (2019) studies the performance of several methods to handle missing values in regression tasks. In this study 3 different regression functions are consider, one of them being linear, one quadratic and the third being the so-called "friedman1" (Friedman et al., 1991). They consider the MCAR mechanism and censoring, inducing up to 20% of observations with missing values in the first variable. The authors compare conditional inference trees, trees and random forests based on the CART criterion and XGBoost (Chen and Guestrin, 2016). The algorithms to handle missing values include MIA, surrogate splits for both trees built with the CART criterion and conditional inference trees, block propagation (only implemented in XGBoost), surrogate splits, mean-imputation and EM imputation (Dempster, Laird, and Rubin, 1977). The authors clearly favors the usage of MIA for tree-based methods, while block propagation could also be a good method.

Breiman (2003) proposes an algorithm based on random forests to impute the missing values making use of the observed values and the proximity matrix. Ishioka (2013) presents an improvement considering not only the observed values in the imputation procedure, but the nearest neighbors for continuous variables and all the observations for categorical variables. Ishioka (2013) compares these two approaches and *knn*-imputation introducing missing values completely at random in the *Spam* data set and considering missing data rates from 5% to 60%, concluding that the proposed approach outperforms *knn*-imputation and the previous method proposed by Breiman (2003). However, the same author comments that other mechanisms of missingness should be consider.

Stekhoven and Bühlmann (2011) introduce the missForest algorithm which imputes the missing values iteratively considering it as a regression problem in which the imputation of the current variable is done using all the other variables. In the simulation study presented, they consider classification and regression problems in 7 different data sets where 10%, 20% or 30% of the values are removed completely at random, concluding that missForest outperforms *knn*-imputation and MICE. Furthermore, the authors ensure that the full potential of missForest is deployed when data includes interactions or non-linear relations between variables of unequal scales and different types.

2.2 Introduction of Concepts

2.2.1 Random Forests

Throughout this article, we assume to have access to a training data set $\mathcal{D}_n = (\mathbf{X}_i, Y_i)_{i=1,\dots,n}$ where the response variables Y_i are real-valued and the input variables $\mathbf{X}_i$ belong to some space $\mathcal{X} \subseteq \mathbb{R}^p$. The task is then to use the data $\mathcal{D}_n$ to construct a learning model, also called learner, predictor or estimator, $m_n : \mathcal{X} \to \mathbb{R}$ that estimates the regression function $m(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$.

The random forest is made of a set of regression trees that are later aggregated all together (with a simple mean idea). Each branch of the tree will be random (in its construction process) and represents a partition of the input space in smaller regions. Moving along the path of the tree corresponds to a choice of one of the possible regions. To construct this partition of the input space, the trees are built in a recursive way. The root of the tree is the whole input space $\mathcal{X}$, which is split into disjoint regions. Then each region is split into more regions, and this process is continued until some stopping rule is applied. At each step of the tree construction, the partition performed over

a cell (or equivalently its corresponding node) is determined by maximizing some split-criterion. The present work focuses in the so-called CART split criterion. We first introduce some important notations.

- A denotes a general node (or cell).
- $N(A)$ holds for the number of points in A.
- The notation $d = (h, z)$ denotes a cut in A, where

  h is a direction, $h \in \{1, \ldots, p\}$, and

  z is the position of the cut in the hth direction, between the limits of A.
- $\mathcal{C}_A$ is the set of all possible cuts in the node A.
- A cell A is split into two cells denoted $A_L = \{\mathbf{x} \in A : \mathbf{x}^{(h)} < z\}$, $A_R = \{\mathbf{x} \in A : \mathbf{x}^{(h)} \geq z\}$.
- $\bar{Y}_A$ (resp. $\bar{Y}_{A_L}$, $\bar{Y}_{A_R}$) is the empirical mean of the response variable Y_i for the indexes such that $\mathbf{X}_i$ belongs to the cell A (resp. A_L, A_R).

Then, the CART split criterion for a generic cell A is defined as

$$\begin{aligned} L_n(A, d) =& \frac{1}{N(A)} \sum_{i=1}^{n} \left(Y_i - \bar{Y}_A\right)^2 \mathbb{1}_{\mathbf{X}_i \in A} \\ & - \frac{1}{N(A)} \sum_{i=1}^{n} \left(Y_i - \bar{Y}_{A_L} \mathbb{1}_{\mathbf{X}_i^{(h)} < z} - \bar{Y}_{A_R} \mathbb{1}_{\mathbf{X}_i^{(h)} \geq z}\right)^2 \mathbb{1}_{\mathbf{X}_i \in A} \end{aligned} \tag{1}$$

with the convention $0/0 = 0$.

As mentioned above, a random forest is a predictor consisting of $M(> 1)$ randomized trees. The randomization is introduced in two different parts of the tree construction. Prior to the construction of each tree, a_n observations are extracted at random with (or without) replacement from the learning data set $\mathcal{D}_n$. Only these a_n observations are taken into account in the tree construction. Then, at each cell a split is performed by maximizing the split criterion over a number `mtry` of input variables, chosen uniformly at random among the original ones. The tree construction is stopped when each final node contains less or equal than `nodesize` points. Hence, the parameters of this algorithm are:

- $M > 1$, which is the number of trees in the forest.
- $a_n \in \{1, \ldots, n\}$, which is the number of observations in each tree.
- `mtry` $\in \{1, \ldots, p\}$, which is the number of directions (features) chosen, candidates to be split. We denote by $\mathcal{M}_{try}$ the features selected in each step.
- `nodesize` $\in \{1, \ldots, a_n\}$, which is the maximum number of observations for a node to be a final cell.

The randomization introduced in the trees (independent from the original source of randomness in the sample $\mathcal{D}_n$) is represented in a symbolic random variable Θ. To each tree – randomized with the random variable Θ_k – is associated a predicted value at a query point $\mathbf{x}$, denoted as $m_n(\mathbf{x}; \Theta_k)$. The different trees are constructed by the same procedure but with independent randomization, so

the random variables $\Theta_1, \ldots, \Theta_M$ are i.i.d. with common law Θ. The nature and dimension of Θ depends on its use in the tree construction. In our choice of construction rules, Θ consists of the observations selected for the tree and the candidate variables to split at each step. Finally, the kth tree estimate at the point $\mathbf{x}$ is defined as

$$m_n(\mathbf{x}; \Theta_k) = \sum_{i \in \mathcal{I}_{n,\Theta_k}} \frac{Y_i \mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x}; \Theta_k)}}{N(A_n(\mathbf{x}; \Theta_k))}$$

where $\mathcal{I}_{n,\Theta_k}$ is the set of the a_n observations selected prior to the construction of the kth tree, $A_n(\mathbf{x}; \Theta_k)$ is the unique final cell that contains $\mathbf{x}$, and $N(A_n(\mathbf{x}; \Theta_k))$ is the number of observations which belong to the cell $A_n(\mathbf{x}; \Theta_k)$. The average of the trees forms the random forest estimate given by

$$m_{M,n}(\mathbf{x}; \Theta_1, \ldots, \Theta_M) = \frac{1}{M} \sum_{k=1}^{M} m_n(\mathbf{x}; \Theta_k).$$

It is known from the work of Breiman (2001) that the random forest estimate does not overfit when M tends to infinity. This makes the parameter M only restricted by the computational power.

2.2.2 Data-missing Mechanisms

The concept of data-missing mechanisms (introduced by Rubin (1976)) establishes the relationship between missingness and data. Before introducing the data-missing mechanisms, let us define a new variable, called the indicator of missing value

$$\mathbf{M}^{(h)} = \begin{cases} 1 & \text{if } \mathbf{X}^{(h)} \text{ is missing} \\ 0 & \text{otherwise} \end{cases}, \quad 1 \leq h \leq p.$$

We assume throughout this work that the response Y has no missing values which makes unnecessary to define an indicator of missing variable for Y. Then, the mechanisms are fully characterized by the information of the conditional distribution of $\mathbf{M}^{(h)}$ given $(\mathbf{X}, Y)$. There are three possible data-missing mechanisms.

Missing Completely at Random (MCAR) We say that the variable $\mathbf{X}^{(h)}$ is MCAR if $\mathbf{M}^{(h)} \perp (\mathbf{X}, Y)$. In other words, under the MCAR assumption, a coordinate $\mathbf{X}^{(h)}$ has some probability to be missing in the sample and this probability does not depend on the value of $\mathbf{X}$ nor the response variable Y.

Missing at Random (MAR) Let us define the set $h_o = \{h : \mathbb{P}\left[\mathbf{M}^{(h)} = 0\right] = 1\}$, thus $\mathbf{X}^{(h_o)}$ is the vector conformed by those variables that are always observed. The variable $\mathbf{X}^{(h)}$ is MAR if

$$\mathbb{P}\left[\mathbf{M}^{(h)} = 1 | (\mathbf{X}, Y)\right] = \mathbb{P}\left[\mathbf{M}^{(h)} = 1 | (\mathbf{X}^{(h_o)}, Y)\right]$$

That is, the probability of $\mathbf{X}^{(h)}$ being missing only depends on observed data.

Not Missing at Random (NMAR) If the probability of missingness depends on unobserved values we say that $\mathbf{X}^{(h)}$ is NMAR.

2.3 Description of the Algorithm

It is of special interest to study the performance of the algorithm presented by Gómez-Méndez and Joly (2020) for which the authors have proven its consistency for an MCAR mechanism and a generalized additive model, being one of the first results on the consistency of random forests with missing values. This algorithm adapts the original CART criterion to manage missing data directly in the construction of the regression trees. We now reintroduced this proposal.

Assume for now that we have a partition of the input space. When there are missing values in the data set, there is uncertainty on the region to which each observation belongs. Thus, the original CART criterion becomes intractable since the quantities $N(A)$, $N(A_L)$, $N(A_R)$, $\bar{Y}_A$, $\bar{Y}_{A_L}$, $\bar{Y}_{A_R}$, $\mathbb{1}_{\mathbf{X}_i \in A}$, $\mathbb{1}_{\mathbf{X}_i^{(h)} < z}$ and $\mathbb{1}_{\mathbf{X}_i^{(h)} \geq z}$ can not be computed. The proposed approach keeps the form of the CART criterion and makes use of adapted imputations for the intractable parts which allows the computation of a modified version of the CART criterion. Unlike most of the imputation techniques, the imputation step is not performed independently of the evaluation of the CART criterion but is integrated to its later optimization. While a cut is selected by maximizing the original CART criterion, now a couple (cut, imputation) is chosen at each split in the creation of the random tree. The idea is that, for a cut, the observations with missing values are assigned to the child node that maximizes this modified CART criterion. At the end, the missing observations will belong a final node of the tree, which can give an "imputation" of the missing values as a region of the input space, which in turn would be translated into a "cloud" of possible regions for the missing values when the random forest is consider. The pseudo-code for this approach is presented in Algorithm 1, at the beginning of the construction of each tree it is necessary to initialize the list of current not-final cells $\mathcal{P} = \{\mathcal{X}\}$ as well as the induced partitions of the input variables and the target variable $\mathcal{X}_\mathcal{P} = \{(\mathbf{X}_1, \ldots, \mathbf{X}_n)\}$ and $\mathcal{Y}_\mathcal{P} = \{(Y_1, \ldots, Y_n)\}$, respectively. Along with the initialization of these sets, the final partition of the input space $\mathcal{P}_f = \{\}$ and the corresponding partitions of the input variables $\mathcal{X}_f = \{\}$ and the target variable $\mathcal{Y}_f = \{\}$ are initialized. When a cell A satisfies the criteria to be a final cell, it is removed from $\mathcal{P}$ and added to $\mathcal{P}_f$, the input variables $\mathcal{X}_A$ and the target variables $\mathcal{Y}_A$ belonging to A are also removed from $\mathcal{X}_\mathcal{P}$ and $\mathcal{Y}_\mathcal{P}$, and added to $\mathcal{X}_f$ and $\mathcal{Y}_f$, respectively. In the sequel, we denote as $N_{obs}^{(h)}(A)$ the number of observed points in the variable $\mathbf{X}^{(h)}$ belonging to A and $N_{miss}^{(h)}(A)$ the number of observations whose value in the variable $\mathbf{X}^{(h)}$ is missing, which were assigned to cell A.

Algorithm 1 Random forest with assignation of missing entries.

Input: Training sample $\mathcal{D}_n$, number of trees $M > 1$, `mtry`$\in \{1, \ldots, p\}$, $a_n \in \{1, \ldots, n\}$, `nodesize`$\in \{1, \ldots, a_n\}$.

Output: Random forest $m_{M,n}$.

for $i = 1, \ldots, M$ **do**
  Select a_n points uniformly in $\mathcal{D}_n$.
  Initialize $\mathcal{P}$, $\mathcal{X}_\mathcal{P}$, $\mathcal{Y}_\mathcal{P}$, $\mathcal{P}_f$, $\mathcal{X}_f$ and $\mathcal{Y}_f$.
  while $\mathcal{P} \neq \varnothing$ **do**
    Let A, $\mathcal{X}_A$, $\mathcal{Y}_A$ be the first elements of $\mathcal{P}$, $\mathcal{X}_\mathcal{P}$ and $\mathcal{Y}_\mathcal{P}$, resp.
    Set $N(A)$ the number of points which belong or were assigned to A.
    if $N(A) \leq$`nodesize` **then**
      Remove A, $\mathcal{X}_A$ and $\mathcal{Y}_A$ from $\mathcal{P}$, $\mathcal{X}_\mathcal{P}$ and $\mathcal{Y}_\mathcal{P}$, and add them to $\mathcal{P}_f$, $\mathcal{X}_f$ and $\mathcal{Y}_f$.
    else
      for $j = 1, \ldots, p$ **do**
        Compute $N_{obs}^{(j)}(A)$.
      end for
      Let h_{obs} be the features h such that $N_{obs}^{(h)}(A) > 1$.
      /*h_{obs} contains features that can be split.*/
      if $h_{obs} = \varnothing$ **then**
        Remove A, $\mathcal{X}_A$ and $\mathcal{Y}_A$ from $\mathcal{P}$, $\mathcal{X}_\mathcal{P}$ and $\mathcal{Y}_\mathcal{P}$, and add them to $\mathcal{P}_f$, $\mathcal{X}_f$ and $\mathcal{Y}_f$.
      else
        Set $m_{obs} = |h_{obs}|$.
        if $m_{obs} \leq$`mtry` **then**
          Set $\mathcal{M}_{try} = h_{obs}$.
        else
          Select uniformly, without replacement, a subset $\mathcal{M}_{try} \subset h_{obs}$ of cardinality `mtry`.
        end if
        Apply Algorithm 2 on cell A along the features in $\mathcal{M}_{try}$.
        Call A_L and A_R the two resulting cells.
        Remove A, $\mathcal{X}_A$ and $\mathcal{Y}_A$ from $\mathcal{P}$, $\mathcal{X}_\mathcal{P}$ and $\mathcal{Y}_\mathcal{P}$.
        Add A_L, A_R, $\mathcal{X}_{A_L}$, $\mathcal{X}_{A_R}$, $\mathcal{Y}_{A_L}$ and $\mathcal{Y}_{A_R}$ to $\mathcal{P}$, $\mathcal{X}_\mathcal{P}$ and $\mathcal{Y}_\mathcal{P}$.
      end if
    end if
  end while
end for

Algorithm 2 Best cut and assignation.

Input: Cell A, $(\mathcal{X}_A, \mathcal{Y}_A)$, $\mathcal{M}_{try}$.
Output: Best cut and assignation $(\hat{z}, \hat{w})$.

Set $max_{CART} \leftarrow 0$
for $h \in \mathcal{M}_{try}$ **do**
    Compute the midpoint of two consecutive values of $\mathbf{X}^{(h)}$ between those points $\mathbf{X} \in A$, let be $Z_A^{(h)}$ the set of these midpoints.
    Let be $W_A^{(h)}$ the set with all the possible assignation for the missing values in h.
    for $z \in Z_A^{(h)}$ **do**
        for $w \in W_A^{(h)}$ **do**
            Let c_A be the CART-criterion computed with the cut (h, z) and the assignation w.
            if $c_A > max_{CART}$ **then**
                $max_{CART} \leftarrow c_A$.
                $(\hat{z}, \hat{w}) \leftarrow (z, w)$
            end if
        end for
    end for
end for

2.4 Complexity of the Algorithm

Fix a cell A and cut (h, z) in A, to keep a simple notation let $Y_{1,L}, \ldots, Y_{N_{L,obs},L}$ be the values of the response variable for those observations such that $\mathbf{X}^{(h)} < z$, $\mathbf{M}^{(h)} = 0$ and $\mathbf{X} \in A$, respectively $Y_{1,R}, \ldots, Y_{N_{R,obs},R}$ denote the values of the response variable for those observations such that $\mathbf{X}^{(h)} \geq z$, $\mathbf{M}^{(h)} = 0$ and $\mathbf{X} \in A$, and let $\bar{Y}_{L,obs}$ and $\bar{Y}_{R,obs}$ be the average of these values. Suppose without lost of generality that $\bar{Y}_{L,obs} \leq \bar{Y}_{R,obs}$ and denote by $\mathbf{i}_{miss} = \{1, \ldots, N_{miss}\}$ the set of indexes of the observations assigned to the cell A whose variable $\mathbf{X}^{(h)}$ is missing, without lost of generality assume that $Y_{(1)} \leq \cdots \leq Y_{(N_{miss})}$. Because $\bar{Y}_{L,obs} \leq \bar{Y}_{R,obs}$ and maximizing the CART criterion implies to make as different as possible the average of the target on the left child node from the average of the target on the right child node, then observations $i \in \mathbf{i}_{miss}$ with the lowest values Y_i should be assigned to the left child node and the observations $i \in \mathbf{i}_{miss}$ with the largest values Y_i should be assigned to the right child node. Denote by $w \in \{1, \ldots, N_{miss} + 1\}$ the position at which the vector of indexes $\mathbf{i}_{miss}$ is split, where $Y_{(1)}, \ldots, Y_{(w-1)}$ are assigned to the left child node and $Y_{(w)}, \ldots, Y_{(N_{miss})}$ are assigned to the right child node. Therefore, note that the set with all possible assignations $W_A^{(h)}$ has a cardinality of $N_{miss}^{(h)}(A) + 1$ and each element of the set is associated to a value of $n \in \{1, \ldots, N_{miss} + 1\}$.

Denote by N the number of points in the cell A, $N = N_{L,obs} + N_{R,obs} + N_{miss}$, let $N_L(w) = N_{L,obs} + (w - 1)$ and $N_R(w) = N_{R,obs} + (N_{miss} - w + 1)$ be the number of points belonging or assigned to the left child node and the right child node, respectively. Denote by $\bar{Y}_L(w)$ and $\bar{Y}_R(w)$ the average of the response variable for observations on the left child node and right child node, that is,

$$\bar{Y}_L(w) = \frac{1}{N_L(w)} \left(\sum_{j=1}^{N_{L,obs}} Y_{j,L} + \sum_{j=1}^{w-1} Y_{(j)} \right)$$

and

$$\bar{Y}_R(w) = \frac{1}{N_R(w)} \left(\sum_{j=1}^{N_{R,obs}} Y_{j,R} + \sum_{j=w}^{N_{miss}} Y_{(j)} \right)$$

then, the CART criterion as a function of w can be written as

$$L_n(w) = \left(\frac{N_L(w)}{N} \right) \left(\frac{N_R(w)}{N} \right) \left(\bar{Y}_L(w) - \bar{Y}_R(w) \right)^2$$

Let be O_{CART} the number of operations needed to calculate the CART criterion for a given cut (h, z) and assignation w of missing values. Consider the Algorithm 2 and note that $|Z_A^{(h)}| = N_{obs}^{(h)}(A) - 1$ and $|W_A^{(h)}| = N_{miss}^{(h)}(A) + 1$. Thus, the number of necessary operations to the get the best cut and assignation is

$$\sum_{h \in \mathcal{M}_{try}} |Z_A^{(h)}||W_A^{(h)}| O_{CART}$$

On the other hand, denote by $\mathcal{P}_{nf}$ the set of non-final nodes of a tree. Note that on average

$$|\mathcal{P}_{nf}| = \frac{a_n}{\texttt{nodesize}} - 1,$$

to construct a regression tree we need to apply $|\mathcal{P}_{nf}|$ times Algorithm 2. In the worst case $a_n = n$ and $\texttt{nodesize} = 1$, hence $|\mathcal{P}_{nf}| \leq n - 1$. Let be O_{tree} the number of operations needed to build a regression tree with our approach, then

$$\begin{aligned}
O_{tree} &= \sum_{A \in \mathcal{P}_{nf}} \sum_{h \in \mathcal{M}_{try}} |Z_A^{(h)}||W_A^{(h)}| O_{CART} \\
&= \sum_{A \in \mathcal{P}_{nf}} \sum_{h \in \mathcal{M}_{try}} \left(N_{obs}^{(h)}(A) - 1 \right) \left(N_{miss}^{(h)}(A) + 1 \right) O_{CART} \\
&\leq \sum_{A \in \mathcal{P}_{nf}} \sum_{h \in \mathcal{M}_{try}} \left(N(A) - 1 \right) \left(N(A) + 1 \right) O_{CART} \\
&\leq \texttt{mtry}(n^3) O_{CART}
\end{aligned}$$

3 Discussion of Related Methods

Many methods proposed in the literature to handle missing data using random forests operate through imputation in a recursive way. First, they use the original training data set $\mathcal{D}_n$ to fill the blank spaces in a roughly way. For example, with the median of the observed values in the variable. We denote this new data set as $\mathcal{D}_{n,1}$.

The imputed data set $\mathcal{D}_{n,1}$ is used to build a random forest. Then, some structures of the forest are exploited, like the so-called proximity matrix, improving the imputation and resulting in a new data set $\mathcal{D}_{n,2}$. The procedure follows iteratively until some stopping rule is applied, for example when there is little change between the imputed values or when a fix number of iterations is achieved. More formally, let us define

$$\mathbf{X}_{i,\ell}^{(h)} = \begin{cases} \mathbf{X}_i^{(h)} & \text{if } \mathbf{M}_i^{(h)} = 0 \\ \widehat{\mathbf{X}}_{i,\ell}^{(h)} & \text{if } \mathbf{M}_i^{(h)} = 1 \end{cases}$$

where $\widehat{\mathbf{X}}_{i,\ell}^{(h)}$ is the imputation of $\mathbf{X}_i^{(h)}$ at time $\ell \geq 1$, and let $\mathbf{X}_{i,\ell} = \left(\mathbf{X}_{i,\ell}^{(1)}, \ldots, \mathbf{X}_{i,\ell}^{(p)}\right)$.

To properly introduce the methods considered in the simulation study that handle missing data through imputation, we need to define the connectivity between two points in a tree and the proximity matrix of the forest. Let $K_{\Theta,n}(\mathbf{X}, \mathbf{X}')$ be the indicator that $\mathbf{X}$ is in the same final cell that $\mathbf{X}'$ in the tree designed with $\mathcal{D}_n$ and the parameter Θ. If $K_{\Theta,n}(\mathbf{X}, \mathbf{X}') = 1$ we say that $\mathbf{X}$ and $\mathbf{X}'$ are connected in the tree $m_n(\cdot; \Theta)$. The proximity of two points is the average of times in which they are connected in the forest. Formally, let us define the proximity between $\mathbf{X}$ and $\mathbf{X}'$ in the random forest, $m_{M,n}(\cdot; \Theta_1, \ldots, \Theta_M)$, as

$$K_{M,n}(\mathbf{X}, \mathbf{X}') = \frac{1}{M} \sum_{k=1}^{M} K_{\Theta_k,n}(\mathbf{X}, \mathbf{X}')$$

To simplify the notation, let $K_{M,\ell}(i,j)$ be the proximity between $\mathbf{X}_i$ and $\mathbf{X}_j$ at time ℓ. That is, $K_{M,\ell}(i,j)$ is the proximity between $\mathbf{X}_i$ and $\mathbf{X}_j$ in the random forest constructed with the data set $\mathcal{D}_{n,\ell}$. We also define $\mathbf{i}_{miss}^{(h)} \subseteq \{1, \ldots, n\}$ as the indexes where $\mathbf{X}^{(h)}$ is missing, and $\mathbf{i}_{obs}^{(h)} = \{1, \ldots, n\} \setminus \mathbf{i}_{miss}^{(h)}$ as the indexes where $\mathbf{X}^{(h)}$ is observed.

We consider three different approaches which impute missing values through random forests. These methods correspond to the algorithms presented by Breiman (2003), Ishioka (2013), and Stekhoven and Bühlmann (2011), that we have denoted as Breiman's approach, Ishioka's approach and missForest approach, respectively. These algorithms impute the missing values through iterative improvements. We also consider MIA, which is an algorithm that handle the missing values directly in the construction of the trees, assigning all the missing values to the same cell. As simple baselines we consider median-imputation of the missing values and listwise deletion (i.e. removing the observations with missing values) before the construction of the random forest.

Breiman's Approach If $\mathbf{X}^{(h)}$ is a continuous variable, $\widehat{\mathbf{X}}_{j,\ell+1}^{(h)}$ is the weighted mean of the observed values in $\mathbf{X}^{(h)}$, where the weights are defined by the proximity matrix of the previous random forest, that is

$$\widehat{\mathbf{X}}_{j,\ell+1}^{(h)} = \frac{\sum_{i \in \mathbf{i}_{obs}^{(h)}} K_{M,\ell}(i,j) \mathbf{X}_i^{(h)}}{\sum_{i \in \mathbf{i}_{obs}^{(h)}} K_{M,\ell}(i,j)}, \qquad \begin{array}{l} \ell \geq 1 \\ j \in \mathbf{i}_{miss}^{(h)} \end{array}$$

On the other hand, if $\mathbf{X}^{(h)}$ is a categorical variable, $\widehat{\mathbf{X}}_{j,\ell+1}^{(h)}$ is given by

$$\widehat{\mathbf{X}}_{j,\ell+1}^{(h)} = \underset{\mathbf{x} \in \mathcal{X}^{(h)}}{\arg\max} \sum_{i \in \mathbf{i}_{obs}^{(h)}} K_{M,\ell}(i,j) \mathbb{1}_{\mathbf{X}_i^{(h)} = \mathbf{x}}, \qquad \begin{array}{l} \ell \geq 1 \\ j \in \mathbf{i}_{miss}^{(h)} \end{array}$$

That is, $\widehat{\mathbf{X}}_{j,\ell+1}^{(h)}$ is the class that maximizes the sum of the proximity considering the observed values in the class.

Ishioka's Approach If $\mathbf{X}^{(h)}$ is a continuous variable, $\widehat{\mathbf{X}}_{j,\ell+1}^{(h)}$ is the weighted mean of the k nearest neighbors, according to the proximity matrix, over all the values, both imputed and observed. The k closest values are chosen to make more robust the method and avoid values which are outliers.

$$\widehat{\mathbf{X}}_{j,\ell+1}^{(h)} = \frac{\sum_{\substack{i \in \text{neigh}_k \\ i \neq j}} K_{M,\ell}(i,j) \widehat{\mathbf{X}}_{i,\ell}^{(h)}}{\sum_{\substack{i \in \text{neigh}_k \\ i \neq j}} K_{M,\ell}(i,j)}, \qquad \begin{array}{l} \ell \geq 1 \\ j \in \mathbf{i}_{miss}^{(h)} \end{array}$$

For categorical variables, it is not necessary to see only the k closest values because the outliers of $\mathbf{X}$ will have few attention. Meanwhile the proximity with missing values should have more attention, especially when the missing rate is high. Hence, if $\mathbf{X}^{(h)}$ is a categorical variable, $\widehat{\mathbf{X}}^{(h)}_{j,\ell+1}$ is given by

$$\widehat{\mathbf{X}}^{(h)}_{j,\ell+1} = \underset{\mathbf{x}\in\mathcal{X}^{(h)}}{\arg\max} \sum_{i\neq j} K_{M,\ell}(i,j)\mathbb{1}_{\widehat{\mathbf{X}}^{(h)}_{i,\ell}=\mathbf{x}}, \qquad \begin{array}{l}\ell \geq 1 \\ j \in \mathbf{i}^{(h)}_{miss}\end{array}$$

MissForest This algorithm treats the imputation as a regression problem by itself, where the target variable is the input variable with missing values. MissForest predicts the missing values using a random forest trained on the observed parts of the data set. More formally for an arbitrary variable $\mathbf{X}^{(h)}$ we can separate the data set into four parts:

- the observed parts of the variable $\mathbf{X}^{(h)}$, denoted as $\mathbf{y}^{(h)}_{obs}$;
- the missing values of the variable $\mathbf{X}^{(h)}$, denoted as $\mathbf{y}^{(h)}_{miss}$;
- the variables other than $\mathbf{X}^{(h)}$ with the observations $\mathbf{i}^{(h)}_{obs}$, denoted by $\mathbf{x}^{(h)}_{obs}$;
- the variables other than $\mathbf{X}^{(h)}$ with the observations $\mathbf{i}^{(h)}_{miss}$, denoted by $\mathbf{x}^{(h)}_{miss}$.

To begin, the original training data set $\mathcal{D}_n$ is used to fill the blank spaces in a roughly way, for example, with the median of the observed values in the variable. For each variable $\mathbf{X}^{(h)}$ the missing values are imputed by first training a random forest with target $\mathbf{y}^{(h)}_{obs}$ and predictors $\mathbf{x}^{(h)}_{obs}$; then the missing values $\mathbf{y}^{(h)}_{miss}$ are predicted by applying the trained random forest to $\mathbf{x}^{(h)}_{miss}$.

Missing Incorporated in Attributes The Missing Incorporated in Attributes (MIA) consists in keeping all the missing values together when a split is performed. That is, missing values are assigned together to the child node that maximizes the CART criterion (or any other considered criterion). Thus, the splits with this approach assign the values according to one of the following rules:

- $\{\mathbf{X}^{(h)} < z \text{ and } \mathbf{M}^{(h)} = 1\}$ versus $\{\mathbf{X}^{(h)} \geq z\}$
- $\{\mathbf{X}^{(h)} < z\}$ versus $\{\mathbf{X}^{(h)} \geq z \text{ and } \mathbf{M}^{(h)} = 1\}$
- $\{\mathbf{M}^{(h)} = 0\}$ versus $\{\mathbf{M}^{(h)} = 1\}$

4 Explanation of Simulations

The regression function considered in this study is the so-called "friedman1" (Friedman et al., 1991), which has been used in previous simulation studies (Friedman et al., 1991; Breiman, 1996; Rieger, Hothorn, and Strobl, 2010; Josse et al., 2019; Friedberg et al., 2020), given by

$$m(\mathbf{x}) = 10\sin\left(\pi\mathbf{x}^{(1)}\mathbf{x}^{(2)}\right) + 20\left(\mathbf{x}^{(3)} - 0.5\right)^2 + 10\mathbf{x}^{(4)} + 5\mathbf{x}^{(5)}$$

Our simulation study is based on the previous work of Rieger, Hothorn, and Strobl (2010), we simulate $\mathbf{X}$ as a uniformly distributed variable on $[0,1]^5$ and introduce missing values in $\mathbf{X}^{(1)}, \mathbf{X}^{(3)}$ and $\mathbf{X}^{(4)}$, considering 7 different mechanisms of missingness. For each mechanism of missingness

we create 100 training data sets, each one with 200 observations. We also create a testing data set with 2000 observations. This amount of data is to have an appropriate approximation to the mean squared error (MSE)

$$\mathbb{E}_{\mathbf{X}|\mathcal{D}_n}\left[m_{M,n}(\mathbf{X}) - m(\mathbf{X})\right]^2$$

and the bias

$$\mathbb{E}_{\mathbf{X}|\mathcal{D}_n}\left[m_{M,n}(\mathbf{X}) - m(\mathbf{X})\right].$$

Note that these expressions are conditioned on the training sample $\mathcal{D}_n$ and thus they are random variables which take a different value for each of the 100 training data sets.

In $\mathbf{X}^{(1)}$ 20% of data is missing, in $\mathbf{X}^{(3)}$ the amount is 10%, and in $\mathbf{X}^{(4)}$ there is 20% again. A random forest is built for each one of the 100 training data sets for the 7 mechanisms of missingness and the data sets without missing values (that are use as benchmark). We use $M = 50$ trees, which has been seen by simulation to be sufficient to stabilize the error in the case of the complete training data sets, for the rest of parameters we use the default values in the regression mode of the `R` package `randomForests`, the parameter `mtry` is set to $\lfloor p/3 \rfloor$, we have sampled without replacement, so a_n is set to $\lceil 0.632n \rceil$ and `nodesize` is set to 5. These parameters are the same for the median-imputation and MIA approaches.

For Breiman's approach, Ishioka's approach and missForest we initialize the algorithms with the median-imputation and consider the same parameters as before for the construction of the regression trees, the number of iterations is set to 10 and random forests are built with 100 trees in each iteration, corresponding to the default values of the package `missForest` in `R`. We now describe the data-missing mechanisms, which are based on those presented by Rieger, Hothorn, and Strobl (2010).

Missing Completely at Random (MCAR)

We select many locations as desired sampled out of the n observations and replace them by `NA`.

Missing at Random

Methods for generating missing values at random are more complicated. The choice of the locations that are replaced by missing values in the "missing" variable now depends on the value of a second variable, called the "determining" variable. Therefore, the values of the "determining" variable now have influence on whether a value in the "missing" variable is missing or not. For $\mathbf{X}^{(1)}$ the "determining" variable is $\mathbf{X}^{(2)}$, while $\mathbf{X}^{(5)}$ is used as the "determining" variable for $\mathbf{X}^{(3)}$ and $\mathbf{X}^{(4)}$.

Creation of ranks (MAR1) The probability for a missing value in a certain location in the "missing" variable is computed by dividing the rank of the location in the "determining" variable by $n(n+1)/2$. The locations for `NA` in the "missing" variable are then sampled with the resulting probability vector.

Creation of two groups (MAR2) We divide the data set in two groups defined by the "determining" variable. A value belongs to the first group if the value in the "determining" variable is greater than or equal to the median of the "determining" variable, otherwise it belongs to the second group. An observation in the respective group has a missing value with probability of 0.9 or 0.1 divided by the number of members in the group. The locations for `NA` in the "missing" variable are then sampled with the resulting probability vector.

Dexter truncation (MAR3) The observations with the biggest values in the "determining" variable have the "missing" variable replaced by `NA` until the desired fraction of `NA` has been achieved.

Symmetric truncation (MAR4) This method is similar to the previous one but we replace by `NA` the values in the "missing" variable in the observations with the biggest and the smallest values in the "determining" variable.

Missing depending on Y (DEPY) The missing values depend on the value of the response, the probability is 0.1 for observations where $Y \geq 13$, otherwise it is 0.4. The locations for `NA` in the "missing" variable are then sampled with the resulting probability vector.

Not Missing at Random

Logit modelling (LOG) In this method the probability for `NA` no longer depends on a single "determining" variable but in all the other variables. It is modeled as

$$\text{logit}\left(\mathbb{P}\left[\mathbf{M}^{(h)}=1\right]\right)=-0.5+\sum_{\substack{k=1\\k\neq h}}^{5}\mathbf{X}^{(k)}$$

Therefore, the probability of missingness depends on variables with observed values and variables with missing values.

5 Results

5.1 Change of Data-Missing Mechanism

Figures 1 and 2 present the average MSE and average bias over all the training data sets for each of the approaches considered. In green there is listwise deletion, those approaches that implement some imputation in the data set are in blue and those approaches that handle missing values directly in the construction of the trees are in red. Listwise deletion (denoted as "NoRows") generates the largest errors and the estimates with more bias for all the mechanisms. Hence, it could be taken as a bound of the minimum expected performance for a method that attempts to estimate the regression function with missing values. We observe that missForest consistently generates estimators with the lowest errors regardless of the data-missing mechanism. We also observe that our proposed method outperforms MIA, Breiman's approach and Ishioka's approach and can achieve similar errors as missForest. It is worthy to observe that even a simple approach as imputing with the median can outperforms most of the methods considered or with similar behavior in several mechanisms of missingness (see MAR1, MAR2, MAR3, LOG). In terms of bias, we observe that the algorithms that impute the missing values before the construction of the random forests tend to generate less biased results, while MIA tends to generate the second more biased estimators. For the MCAR case, all the methods considered tend to be unbiased and with similar MSE, except for missForest and listwise deletion with the lowest and highest MSE, respectively.[1]

[1]Codes to reproduce our results can be found in: https://github.com/IrvingGomez/RandomForestsSimulations

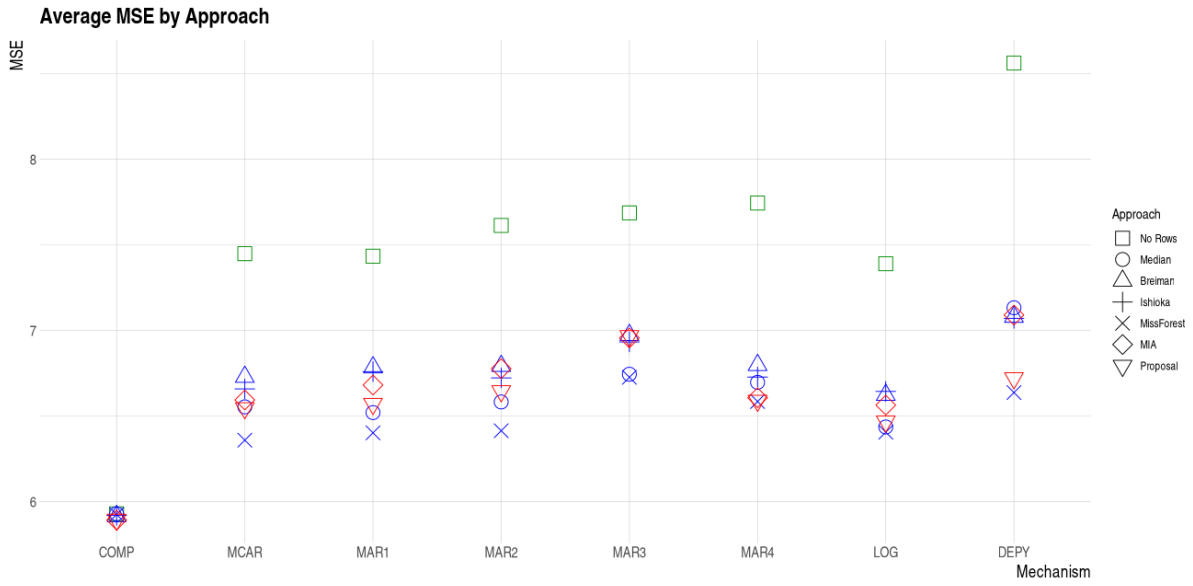

Figure 1: Average MSE of the testing data set for each approach and each mechanism of missingness.

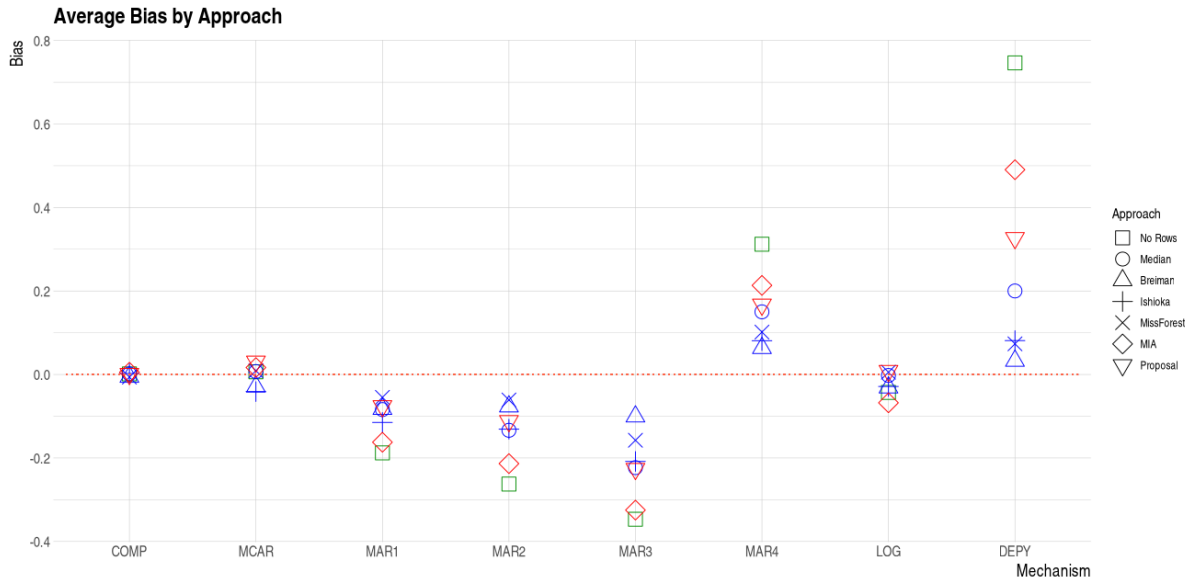

Figure 2: Average bias of the testing data set for each approach and each mechanism of missingness.

5.2 Increasing the Rate of Missingness

Using the 100 training data sets with no missing values, we calculate the importance of the variables with the `R` package `randomForests`, by percentage of increase in mean squared error and by increase

in node purity (Breiman, 2001; Breiman, 2003). Figures 3 and 4 show the violin plots for these important measurements. We can see a consistently order for the variables with missing values in both measures of importance, where $\mathbf{X}^{(4)}$ is considered more important than $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(3)}$. Hence, we decided to change the fraction of missingness in $\mathbf{X}^{(4)}$ to vary between 5%, 10%, 20%, 40%, 60%, 80%, 90% and 95%, without changing the percentage of missingness in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(3)}$. In this part of the study we do not consider anymore listwise deletion.

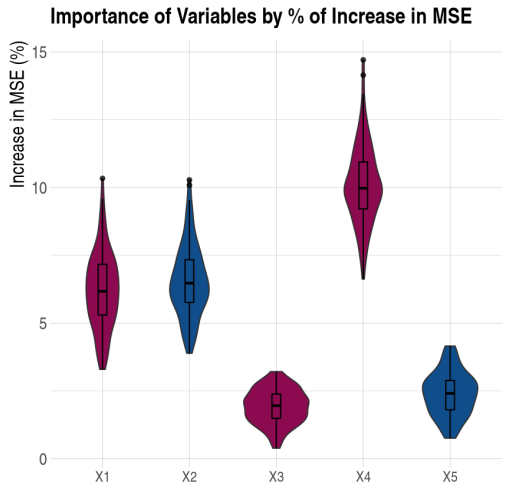

Figure 3: Importance Variable accordingly to percentage increase in MSE.

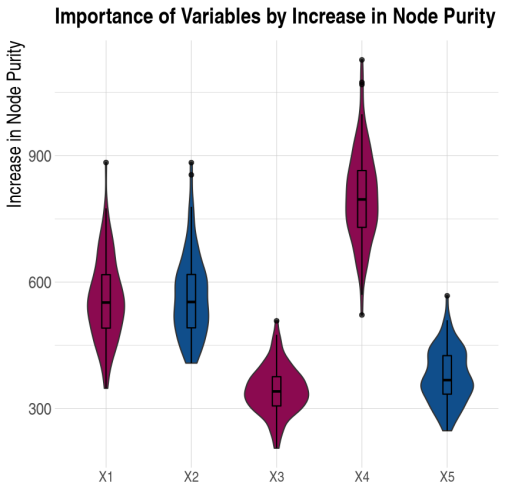

Figure 4: Importance Variable accordingly to increase in node purity.

Figure 5 presents the average MSE over the training data sets varying the percentage of missing data and considering the MAR1 mechanism[2]. Important differences between the performance of the methods become clear with the increasing percentage of missingness. Especially when this value is over 60% there is an order on the performance of the methods induced by the MSE. In this cases missForest, our proposal and MIA represent the methods with less MSE. Moreover, we can see the advantage of searching for the best assignation when the percentage arises between 90% and 95% with our proposal outperforming all the other algorithms (in the case of the DEPY mechanism we observed this phenomenon even for 80%). While all the methods present a similar MSE for percentage of missing values smaller than 40%.

The differences between the mechanisms of missingness are also reflected when we increase the percentage of missing values. For example, the top three algorithms (missForest, our approach and MIA) present a MSE between 7.95 and 8.19 when the percentage of missingness is 90% and the introduction of missing values is done completely at random (MCAR). On the other hand, for the same percentage of missing values and the same algorithms we observe a deterioration in terms of the MSE which varies between 9.05 and 12.86 when we consider the DEPY case. For the DEPY case we consistently see that our approach and missForest outperform the other methods, regardless of the percentage of missing values, while there is no clear advantage for the rest of the algorithms over the others.

[2]Analogous figures for the MSE and bias for all the mechanisms of missingness can be found in the supplementary material at https://irvinggomez.com/publication/supplementary_random_forests_simulation/Supplementary_RandomForestsSimulation.pdf

Figure 6 presents the average bias over the training data sets varying the percentage of missing data and considering the MAR1 mechanism. When we include the bias in the study, the differences between methods and data-missing mechanisms become more evident. We observe that this data-missing mechanism introduced a bias in the methods. MIA and median-imputation create the most biased estimators, while Breiman's approach creates the less biased. Considering other data-missing mechanisms, we observe an unbiased behavior for the MCAR and the LOG cases, with some deterioration when the percentage of missing values affects up to 60% of the observations.

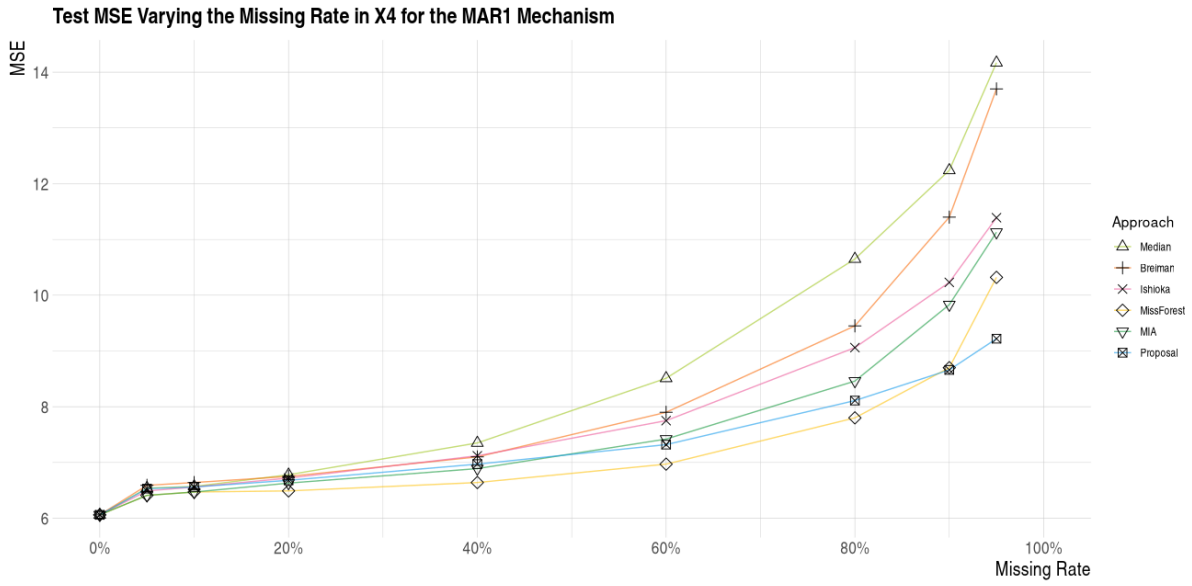

Figure 5: Average MSE for the testing data set for each percentage of missingness, considering the MAR1 mechanism.

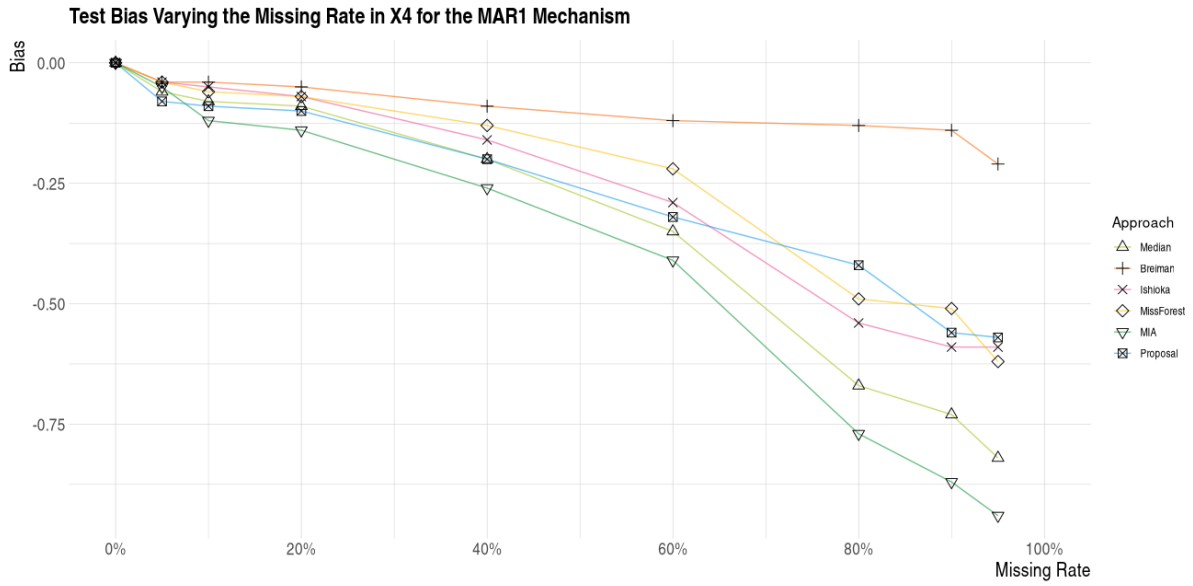

Figure 6: Average bias for the testing data set for each percentage of missingness, considering the MAR1 mechanism.

6 On the Ideas of the Simplifications of our Algorithm

In this section, we discuss a remark that could lead to further simplification of our algorithm in order to reach an algorithmic complexity of the order of the MIA algorithm. Recall from Section 2.4 that the algorithmic complexity of our proposal is of the order $O(n^3)$ times the calculation required to compute one single CART criterion and that MIA algorithm (in $O(n^2)$) is quicker by a linear factor.

From the CART criterion written in Equation (1), we see that the criterion is convex with respect to the variable that counts the number of assigned variables to the left side of a cell A. See Figure 7 for a graphical representation of this fact. This simple remark leads us to opt for a dichotomy strategy for the optimal assignation of the missing variables. Indeed, at first step, we assign half of the missing variables to the left (in A_L) and half to the right (in A_R). The corresponding assignation is then given by a $k \sim N/2$ such that $Y_1 \leq \cdots \leq Y_k \in A_L$ and $Y_{k+1} \leq \cdots \leq Y_N \in A_R$. Such a configuration gives a CART criterion resumed in $CART(k)$. Then a local gradient

$$\nabla CART(k) = 2CART(k) - CART(k-1) - CART(k+1)$$

is computed. We assume that

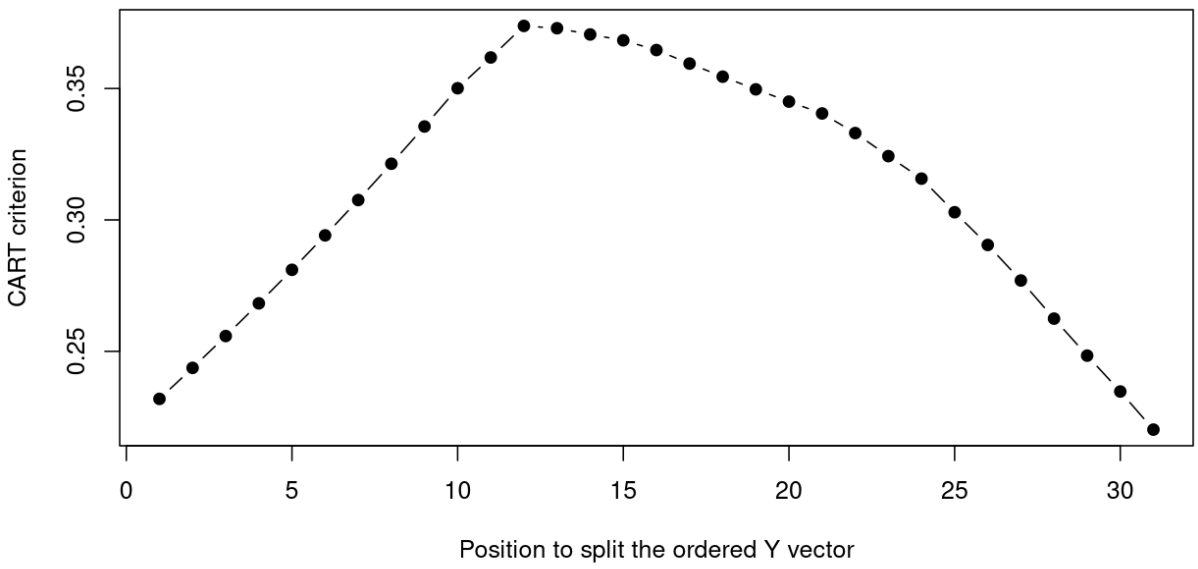

Figure 7: CART criterion as function of the position where we split $Y_(1), \ldots, Y_{(N_{miss})}$.

7 Discussion and Conclusions

We developed a simulation study comparing 7 distinct approaches based on random forests to perform regression with missing entries. The algorithm proposed in this work handle missing values directly in the split criterion considered in the construction of the trees. Two of the random forest-based imputation algorithms rely on the computation of the proximity matrix and improve the imputations iteratively. The remaining imputation algorithm corresponds to missForest where the imputation values of a feature is uploaded using the current imputation of all the other variables, treating the imputation of the missing values as a regression problem by itself, which is a similar strategy for other imputation algorithms like MICE.

With no surprise, listwise elimination was the approach with the worst performance, this method should be avoided unless the percentage of observations with missing values is so low that they can be deleted without a severe harmful. For the rest of the algorithms considered in this simulation there is no clear advantage on either imputation-based methods or approaches that handle missing values directly in the loss function used in the construction of the trees. We observed that the differences between distinct techniques and data-missing mechanisms become clear when the percentage of observations with missing values increases, especially when this value is over 60%. While these differences are diluted for small values of this percentage (less than 40%). The behavior of the methods appear to be dependent on the mechanim of missingness. Computer exhaustive algorithms seem to perform particularly well for large percentage of missing values.

An important observation is that the number of possible assignations for a given cut (h, z) increases extremely rapid with the number of missing values in that variable since there are $2^{n_{miss}^{(h)}}$ possible assignations, which turns the algorithm to be prohibited in real applications. A possible solution for this severe lack is to select at random just a small percentage of all these assignations

and select the assignation that maximizes the loss function between those selected at random. On the other hand, note that MIA only considers three different assignations for the missing values allowing its computation even when there are many observations with missing values. It is important to observe that a simple baseline like median-imputation shows similar performance to more complicated imputation algorithms or exhaustive computing methods, which make us wonder if all the extra effort that it is required is worthy for the little improvement.

Note that the mechanisms of missingness as well as the percentage of missing values considered are not necessarily comparable to real life data. Therefore a further work would include real life data sets that already contain missing values as other studies suggests (Hapfelmeier, Hothorn, and Ulm, 2012).

References

[1] Gérard Biau and Erwan Scornet. "A random forest guided tour". In: *Test* 25.2 (2016), pp. 197–227.

[2] Leo Breiman. "Bagging predictors". In: *Machine learning* 24.2 (1996), pp. 123–140.

[3] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[4] Leo Breiman. *Setting up, using, and understanding random forests V4.0*. 2003. URL: https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf.

[5] Leo Breiman et al. *Classification and regression trees*. Chapman and Hall/CRC, 1984.

[6] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[7] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[8] Alireza Farhangfar, Lukasz Kurgan, and Jennifer Dy. "Impact of imputation of missing values on classification error for discrete data". In: *Pattern Recognition* 41.12 (2008), pp. 3692–3705.

[9] Ad Feelders. "Handling missing data in trees: surrogate splits or statistical imputation?" In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer. 1999, pp. 329–334.

[10] Rina Friedberg et al. "Local linear forests". In: *Journal of Computational and Graphical Statistics* (2020), pp. 1–15.

[11] Jerome H Friedman et al. "Multivariate adaptive regression splines". In: *The annals of statistics* 19.1 (1991), pp. 1–67.

[12] Irving Gómez-Méndez and Emilien Joly. "On the consistency of a random forest algorithm in the presence of missing entries". In: *arXiv preprint arXiv:2011.05433* (2020).

[13] Alexander Hapfelmeier, Torsten Hothorn, and Kurt Ulm. "Recursive partitioning on incomplete data using surrogate decisions and multiple imputation". In: *Computational Statistics & Data Analysis* 56.6 (2012), pp. 1552–1565.

[14] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. "Unbiased recursive partitioning: A conditional inference framework". In: *Journal of Computational and Graphical statistics* 15.3 (2006), pp. 651–674.

[15] Tsunenori Ishioka. "Imputation of missing values for unsupervised data using the proximity in random forests". In: *International Conference on Mobile, Hybrid, and On-line Learning. Nice*. 2013, pp. 30–36.

[16] Julie Josse et al. "On the consistency of supervised learning with missing values". In: *arXiv preprint arXiv:1902.06931* (2019).

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

[18] Anna Rieger, Torsten Hothorn, and Carolin Strobl. "Random forests with missing values in the covariates". In: *Technical report* (2010). URL: http://epub.ub.uni-muenchen.de/11481.

[19] Donald B Rubin. "Inference and missing data". In: *Biometrika* 63.3 (1976), pp. 581–592.

[20] Joseph L Schafer. *Analysis of incomplete multivariate data*. CRC press, 1997.

[21] Joseph L Schafer and Maren K Olsen. "Multiple imputation for multivariate missing-data problems: A data analyst's perspective". In: *Multivariate behavioral research* 33.4 (1998), pp. 545–571.

[22] Daniel J Stekhoven and Peter Bühlmann. "MissForest—non-parametric missing value imputation for mixed-type data". In: *Bioinformatics* 28.1 (2011), pp. 112–118.

[23] Olga Troyanskaya et al. "Missing value estimation methods for DNA microarrays". In: *Bioinformatics* 17.6 (2001), pp. 520–525.

[24] Beth Twala, MC Jones, and David J Hand. "Good methods for coping with missing data in decision trees". In: *Pattern Recognition Letters* 29.7 (2008), pp. 950–956.

[25] Stef Van Buuren et al. "Fully conditional specification in multivariate imputation". In: *Journal of statistical computation and simulation* 76.12 (2006), pp. 1049–1064.