On the consistency of supervised learning with missing values

Julie Josse, Nicolas Prost, Erwan Scornet, Gaël Varoquaux

On the consistency of supervised learning with missing values

Julie Josse julie.josse@polytechnique.edu
Centre de Mathématiques Appliquées, Ecole Polytechnique
XPOP project-team, INRIA Saclay

Nicolas Prost nicolas.prost@inria.fr
Centre de Mathématiques Appliquées, Ecole Polytechnique
Parietal project-team, INRIA Saclay

Erwan Scornet erwan.scornet@polytechnique.edu
Centre de Mathématiques Appliquées, Ecole Polytechnique

Gaël Varoquaux gael.varoquaux@inria.fr
Parietal project-team, INRIA Saclay
1 Rue Honoré d'Estienne d'Orves, 91120 Palaiseau – France

Abstract

In many application settings, the data have missing features which make data analysis challenging. An abundant literature addresses missing data in an inferential framework: estimating parameters and their variance from incomplete tables. Here, we consider supervised-learning settings: predicting a target when missing values appear in both training and testing data.

We show the consistency of two approaches in prediction. A striking result is that the widely-used method of imputing with the mean prior to learning is consistent when missing values are not informative. This contrasts with inferential settings where mean imputation is pointed at for distorting the distribution of the data. That such a simple approach can be consistent is important in practice. We also show that a predictor suited for complete observations can predict optimally on incomplete data, through multiple imputation.

We analyze further decision trees. These can naturally tackle empirical risk minimization with missing values, due to their ability to handle the half-discrete nature of incomplete variables. After comparing theoretically and empirically different missing values strategies

in trees, we recommend using the "missing incorporated in attribute" method as it can handle both non-informative and informative missing values.

keywords Bayes consistency, empirical risk minimization, decision trees, imputation, missing incorporated in attribute

1 Introduction

As volumes of data increase, they are harder to curate and clean. They may come from the aggregation of various sources (*e.g.* merging multiple databases) and contain variables of different natures (*e.g.* different sensors). Such heterogeneous data collection can lead to many missing values: samples only come with a fraction of the features observed. Though there is a vast literature on treating missing values, it focuses on estimating parameters and their variance in the presence of missing values in a single data set. In contrast, there are few studies of supervised-learning settings where the aim is to predict a target variable given input variables. Rather than generative models, these settings only require *discriminative* (or conditional) models. Also, they separate training and testing.

Aside from the aggregation of multiple sources, missing values can also appear for a variety of reasons. For sensor data, missing data can arise from device failure. On the contrary, informative missing values can be found in poll data for instance where participants may not answer sensitive questions related to unpopular opinions. In medical studies, some measurements may be impractical on patients in a critical state, in which case the presence of missing values can be related to the variable of interest, target of the prediction (patient status). These various scenarios lead to different missing value mechanisms.

The classical literature on missing data, led by Rubin (1976), defines missing data mechanisms based on the relationship between missingness and observed values: if they are independent, the mechanism is said to be Missing Completely At Random (MCAR); if the missingness only depends on the observed values, then it is Missing At Random (MAR); otherwise it is Missing Not At Random (MNAR). However, adapting this nomenclature to supervised learning, to take into account the target variable of the prediction, has seldom been discussed.

Many methods are available to deal with missing values (Josse and Reiter, 2018). Listwise deletion, *i.e.* the removal of incomplete observations, may

allow to train the model on complete data. Yet it may not suffice for supervised learning, as the test set may also contain incomplete data. Hence the prediction procedure should handle missing data. A popular solution suitable with any existing learning algorithm is to impute missing values, that is to replace them with plausible values to produce a completed data set. The widespread practice of imputing with the mean of the variable on the observed entries is known to have serious drawbacks as it distorts the joint and marginal distributions of the data which induces bias in estimators (Little and Rubin, 2002). Interestingly, the performance of mean imputation has never been really assessed when aiming at predicting an output. In practice, users resort to different strategies such as imputing separately the training and test sets or imputing them jointly. More elaborate strategies rely on using expectation maximization (EM) to fit a model on incomplete data (Dempster et al., 1977; Little and Rubin, 2002; Little, 1992). However, such a model cannot readily be applied to new incomplete data. In addition, the EM relies on strong parametric assumptions. Alternatively, some learning algorithms, such as decision trees, can be adapted to handle missing values.

In this paper, we study the classic tools of missing data in the context of supervised learning. We start in Section 2 by setting the notations and briefly summarizing the missing data literature. Our first contribution, detailed in Section 3, is to suggest a formalism for missing data adapted to supervised learning; we write existing methods with this formalism and show how to make predictions on a test set with missing values. Section 4 presents our main contribution which consists in studying the consistency of two approaches to estimate the prediction function with missing values. The first theorem states that, given an optimal predictor for the completely-observed data, a consistent procedure can be built by predicting on a test set where missing entries are replaced by multiple imputation. The second approach, which is the most striking and has important consequences in practice, shows that mean imputation prior to learning is consistent for supervised learning. This is as far as we know the first result justifying this very convenient practice of handling missing values. In Section 5, we then analyze decision trees, as their greedy and discrete natures allow to adapt them to handle missing values directly. We compare various missing data methods for trees: surrogate splits, the default in Classification and Regression Trees (CART, Breiman et al. 1984), probabilistic splits, the default in C4.5 (Quinlan, 2014), block propagation, the method used in XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017), a method called "missing incorporated in attribute" (MIA, Twala et al. 2008) and also conditional inference trees

(Hothorn et al., 2006). Theoretical analysis of toy examples justifies some empirical results observed in Kapelner and Bleich (2015), one of the few papers that studied trees with missing values for supervised learning. We recommend MIA as a way to exploit the missing pattern in the estimation. Finally, Section 6 compares the different tree methods on simulated data with missing values. We also show the benefits for prediction of an approach often used in practice, which consists in "adding the mask", *i.e.* adding binary variables that code for the missingness of each variables as new covariates, even though this method has been recommended against for estimation (Jones, 1996).

2 Definitions, problem setting, prior art

Notation Throughout the paper, **bold** letters refer to vectors; CAPITAL letters refer to real-valued or vector-valued random variables, while lowercase letters are realisations. In addition, as usual, for any two variables A and B of joint density f,

$$f(b) \coloneqq f_B(b) \coloneqq \int f(\alpha, b)\, \mathrm{d}\mu(\alpha), \qquad f(a|b) \coloneqq f_{A|B=b}(a) \coloneqq \frac{f(a,b)}{f(b)}.$$

2.1 Supervised learning

Supervised learning is typically focused on learning to predict a *target* $Y \in \mathcal{Y}$ from inputs $\mathbf{X} \in \mathcal{X} = \bigotimes_{j=1}^{d} \mathcal{X}_j$, where the pair $(\mathbf{X}, Y)$ is considered as random, drawn from a distribution P. Formally, the goal is to find a function $f : \mathcal{X} \to \mathcal{Y}$, that minimizes $\mathbb{E}[\ell(f(\mathbf{X}), Y)]$ given a cost function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, called the *loss* (Vapnik, 1999). The best possible prediction function is known as the *Bayes rule*, given by

$$f^\star \in \operatorname*{argmin}_{f:\, \mathcal{X} \to \mathcal{Y}} \mathbb{E}\left[\ell(f(\mathbf{X}), Y)\right], \tag{1}$$

and its error rate is the *Bayes rate* (Devroye et al., 2013). A *learning* procedure is used to create the function f from a set of *training* pairs $\mathcal{D}_{n,\text{train}} = \{(\mathbf{X}_i, Y_i), i = 1, \ldots, n\}$. f is therefore itself a function of $\mathcal{D}_{n,\text{train}}$, and can be write $\hat{f}_{\mathcal{D}_{n,\text{train}}}$ or simply $\hat{f}_n$. There are many different learning procedures, including random forests (Breiman, 2001) or support vector machines (Cortes and Vapnik, 1995). A learning procedures that, given an infinite amount of data, yields a function that achieves the Bayes rate is said

to be *Bayes consistent*. In other words, $\hat{f}_n$ is Bayes consistent if

$$\mathbb{E}[\ell(\hat{f}_n(\mathbf{X}), Y)] \xrightarrow[n\to\infty]{} \mathbb{E}[\ell(f^\star(\mathbf{X}), Y)].$$

In a classification setting, Y is drawn from a finite set of discrete values, and the cost ℓ is typically the zero-one loss: $\ell(Y_1, Y_2) = \mathbb{1}_{Y_1 \neq Y_2}$. In a regression setting, Y is drawn from continuous values in $\mathbb{R}$ and is assumed to satisfy $\mathbb{E}[Y^2] < \infty$. A common cost is then the square loss, $\ell(Y_1, Y_2) = (Y_1 - Y_2)^2$. Considering the zero-one loss (Rosasco et al., 2004) or the square loss (see *e.g.* sec 1.5.5 of Bishop (2006)), the Bayes-optimal function $f^\star$, that minimizes the expected loss, satisfies $f^\star(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}]$.
Note that the learning procedure has access to a finite sample, $\mathcal{D}_{n,\text{train}}$, and not to the distribution P hence, it can only use the *empirical* risk, $\sum_{i=1\ldots n} \ell(f(\mathbf{X}_i), Y)$, rather than the expected risk. A typical learning procedure is therefore the *empirical risk minimization* defined as the following optimization problem

$$\hat{f}_n \in \underset{f:\mathcal{X}\to\mathcal{Y}}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{i=1}^{n} \ell\left(f(\mathbf{X}_i), Y_i\right) \right).$$

A new data set $\mathcal{D}_{n,\text{test}}$ is then needed to estimate the generalization error rate of the resulting function f.

2.2 Prior art on missing values

In this section, we review the different missing data mechanisms. We then summarize the main methods to handle missing values: imputation methods and likelihood-based ones. Most of this prior art to deal with missing values is based on a single data set with no distinction between training and test set.

2.2.1 Missing data mechanisms

To follow the historical definitions which do not give the response Y a particular role, we temporarily consider Y as part of the input vector $\mathbf{X}$, though we assume that Y has no missing values. Rubin (1976) defines three missing data mechanisms and fundamental results for working with likelihood models in the presence of missing data. These are defined by considering the realised data set $(\mathbf{x}_i) = (x_{ij}) \in \mathbb{R}^{n\times d}$, as one realisation from a distribution in $\mathbb{R}^{n\times d}$. The missingness is encoded as an *indicator matrix*

$(\mathbf{m}_i) = (m_{ij}) \in \mathbb{R}^{n \times d}$ where, for all i and j, $m_{ij} = 0$ if x_{ij} is observed, and $m_{ij} = 1$ if x_{ij} is missing.
The definitions of Rubin (1976) can naturally be adapted to the i.i.d. setting. All rows $(\mathbf{x}_i, \mathbf{m}_i)$ are assumed to be sampled i.i.d. from a distribution in $\mathscr{P} = \{f_\theta(\mathbf{x})g_\phi(\mathbf{m}|\mathbf{x}) : (\theta, \phi) \in \Omega_{\theta,\phi}\}$ where, marginally, $\theta \in \Theta$ and $\phi \in \Phi$. The goal in statistical inference is to estimate the parameter θ. This is usually done by maximizing the likelihood $\mathcal{L}(\theta) = \prod_{i=1}^n f_\theta(\mathbf{x}_i)$, which is well defined when the $\mathbf{x}_i$ are fully observed. Here, the likelihood is integrated over the missing values, resulting in

$$\text{(full likelihood)} \qquad \mathcal{L}_1(\theta, \phi) = \prod_{i=1}^{n} \int g_\phi(\mathbf{m}_i|\mathbf{x}) f_\theta(\mathbf{x}) \, \mathrm{d}\delta_{o(\cdot,\mathbf{m}_i)=o(\mathbf{x}_i,\mathbf{m}_i)}(\mathbf{x})$$

where $o(\mathbf{x}, \mathbf{m})$ denotes the observed values for any realisation $(\mathbf{x}, \mathbf{m})$ of $(\mathbf{X}, \mathbf{M})$ (Seaman et al., 2013), and δ the Dirac measure. The parameter ϕ is generally not considered as of interest. In addition, modeling the missing values mechanism requires strong parametric assumptions. An easier quantity would be

$$\text{(likelihood of observed data)} \qquad \mathcal{L}_2(\theta) = \prod_{i=1}^{n} \int f_\theta(\mathbf{x}) \, \mathrm{d}\delta_{o(\cdot,\mathbf{m}_i)=o(\mathbf{x}_i,\mathbf{m}_i)}(\mathbf{x})$$

ignoring the missing data mechanism. To leave the difficult term, *i.e.* the missing values mechanism, out of the expectation, Rubin (1976) introduces an *ad hoc* assumption, called *Missing At Random (MAR)*, which is that for all $\phi \in \Phi$, for all $i \in [\![1, n]\!]$, for all $\mathbf{x}' \in \mathcal{X}$,

$$o(\mathbf{x}', \mathbf{m}_i) = o(\mathbf{x}_i, \mathbf{m}_i) \Rightarrow g_\phi(\mathbf{m}_i|\mathbf{x}') = g_\phi(\mathbf{m}_i|\mathbf{x}_i),$$

and states the following result.

Theorem 1 (Theorem 7.1 in Rubin (1976))**.** *Let ϕ such that for all $1 \leq i \leq n$, $g_\phi(\mathbf{m}_i|\mathbf{x}_i) > 0$. Assuming (a) MAR, (b) $\Omega_{\theta,\phi} = \Theta \times \Phi$, $\mathcal{L}_2(\theta)$ is proportional to $\mathcal{L}_1(\theta, \phi)$ with respect to θ, so that the inference for θ can be obtained by maximizing the likelihood $\mathcal{L}_2$ which ignores the mechanism.*

MAR has a stronger version, more intuitive: *Missing Completely At Random (MCAR)*. In its simplest and strongest form, it states that $\mathbf{M} \perp\!\!\!\perp \mathbf{X}$ (the model's density is $f_\theta(\mathbf{x})g_\phi(\mathbf{m})$). At the other end of the spectrum, if it is not possible to ignore the mechanism, the corresponding model is called *Missing Not At Random (MNAR)*.

There is little literature on missing data mechanism for supervised learning or discriminative models. Kapelner and Bleich (2015) formalise the problem by separating the role of the response y, factorising the likelihood as $f_\theta(\mathbf{x})g_\phi(\mathbf{m}|\mathbf{x})h_\chi(y|\mathbf{x},\mathbf{m})$. Note that they do not write $g_\phi(\mathbf{m}|\mathbf{x},y)$. They justify this factorisation with the – somewhat causal – consideration that the missing values are part of the features, which precede the response. The need to represent the response variable in the factorization show that it may be useful to extend the traditional mechanisms for a supervised learning setting: the link between the mechanism and the output variable can have a significant impact on the results. Davidian (2017) and Arel-Bundock and Pelc (2018) noticed that as long as $\mathbf{M}$ does not depend on Y, it is possible to estimate regression coefficients without bias even with listwise deletion and MNAR values. Ding and Simonoff (2010) generalise the MAR assumption with the following nomenclature `MXY`: the missing mechanism can marginally depend on the target ($**\mathtt{Y}$), on the features that are always observed ($*\mathtt{X}*$) or on the features that can be missing ($\mathtt{M}**$).

2.2.2 Imputation prior to analysis

Most statistical models and machine learning procedures are not designed for incomplete data. It is therefore useful to *impute* the data, forming a completed data set that can be analysed by any procedure, *e.g.* supervised learning methods. To impute data, *joint modeling* (JM) approaches capture the joint distribution across features (Little and Rubin, 2002). A simple example of joint modeling imputation is to assume a Gaussian distribution of the data, to estimate the mean vector and covariance matrix from the incomplete data (using an EM algorithm, see Section 2.2.3). Missing entries can then be imputed with their conditional expectation knowing the observed data and the estimated parameters. More powerful methods can be based on low-rank models (Hastie et al., 2015; Josse et al., 2016), or deep learning approaches such as denoising autoencoders (DAEs, Vincent et al. 2008; Gondara and Wang 2018) and generative adversarial networks (Li et al., 2017; Yoon et al., 2018). Another popular approach to impute data is called fully conditional specification (FCS) also known as imputation with conditional equation (ICE) (van Buuren, 2018). It also assumes a joint distribution for the data, but defines it implicitly by the conditional distributions of each variable. This approach is popular because it is flexible and can easily handle variables of a different nature such as ordinal, categorical, numerical, etc. A powerful example of this class uses iterative imputation of each variable by random forests (Stekhoven and Bühlmann, 2011).

The role of the dependent variable Y and whether or not to include it in the imputation model has been a rather controversial point. Indeed, it is quite counter-intuitive to include it when the aim is to apply a conditional model on the imputed data set to predict the outcome Y. Nevertheless, it is recommended as it can provide information for imputing covariates (Allison, 2001, p.57). Sterne et al. (2009) illustrated the point for the simple case of a bivariate Gaussian data (X, Y) with a positive structure of correlation and missing values on X. Imputing using only X is not appropriate when the aim is to estimate the parameters of the linear regression model of Y given X.

One important issue with "single" imputation, *i.e.* predicting only one value for each missing entries, is that it forgets that some values were missing and considers imputed values and observed values in the same way. It leads to underestimation of the variance of the parameters (Little and Rubin, 2002) estimated on the completed data. One solution, to incorporate the uncertainty of the prediction of values is to use multiple imputation (MI, Rubin 1987) where many plausible values are generated for each missing entries, leading to many imputed data sets. Then, MI consists in applying an analysis on each imputed data sets and combining the results. Although many procedures to generate multiple imputed data sets are available (Murray, 2018), here again, the case of discriminatory models is only rarely considered, with the exception of Wood et al. (2008) who use a variable selection procedure on each imputed data set and propose to keep the variables selected in all imputed data sets to construct the final model (see also Liu et al., 2016). We note that even when imputing data, the objective is to make an inference with missing data, *e.g.* to best estimate parameters and their variance in the presence of an incomplete data set.

2.2.3 EM algorithm

Imputation leads to two-step methods that are generic in the sense that any analysis can be performed from the same imputed data set. On the contrary, the expectation maximization (EM) algorithm (Dempster et al., 1977) proceeds directly in one step. It can thus be better suited to a specific problem but requires the development of a dedicated algorithm.

The EM algorithm can be used in missing data settings to compute maximum likelihood estimates from an incomplete data set. Indeed, with the assumptions of Theorem 1 (MAR settings), maximizing the observed likelihood $\mathcal{L}_2$ gives principle estimation of parameters θ. The log-likelihood of

the observed data is

$$\ell_2(\theta) = \sum_{i=1}^{n} \log \int f_\theta(\mathbf{x}) \, \mathrm{d}\delta_{o(\cdot,\mathbf{m}_i)=o(\mathbf{x}_i,\mathbf{m}_i)}(\mathbf{x}).$$

Starting from an initial parameter $\theta^{(0)}$, the algorithm alternates the two following steps,

$$\textbf{(E-step)} \qquad Q(\theta|\theta^{(t)}) = \sum_{i=1}^{n} \int (\log f_\theta(\mathbf{x})) f_{\theta^{(t)}}(\mathbf{x}) \, \mathrm{d}\delta_{o(\cdot,\mathbf{m}_i)=o(\mathbf{x}_i,\mathbf{m}_i)}(\mathbf{x}).$$

$$\textbf{(M-step)} \qquad \theta^{(t+1)} \in \operatorname*{argmax}_{\theta\in\Theta} Q(\theta|\theta^{(t)}).$$

The well-known property of the EM algorithm states that at each step t, the observed log-likelihood increases, although there is no guarantee to find the global maximum. In Appendix B.2 we give an example of an EM algorithm to estimate the parameters of a bivariate Gaussian distribution from incomplete data.

3 Supervised learning procedures with missing data on train and test set

Supervised learning typically assumes that the data are i.i.d. In particular, an out-of-sample observation is supposed to be drawn from the same distribution as the original sample. Hence it has the same missing data mechanism. An appropriate method should be able to predict on new data with missing values. Here we discuss how to adapt classic missing data techniques to machine learning settings, and vice versa.

Notations Following Rubin (1976) and others (Rosenbaum and Rubin 1984, appendix B; Mohan and Pearl 2018; Yoon et al. 2018), we define the incomplete feature vector $\widetilde{\mathbf{X}}$ as $\widetilde{X}_j = \texttt{NA}$ if $M_j = 1$, and $\widetilde{X}_j = X_j$ otherwise. As $\mathcal{X}$ is a cartesian product, $\widetilde{\mathbf{X}}$ belongs to the space $\widetilde{\mathcal{X}} = \bigotimes_{j=1}^{d}(\mathcal{X}_j \cup \{\texttt{NA}\})$. We have

$$\widetilde{\mathbf{X}} = \mathbf{X} \odot (\mathbf{1} - \mathbf{M}) + \texttt{NA} \odot \mathbf{M},$$

where $\odot$ is the term-by-term product, with the convention that, for all one-dimensional x, $\texttt{NA} \cdot x = \texttt{NA}$. As such, when the data are real, $\widetilde{\mathbf{X}}$ can be seen as a mixed categorical and continuous variable, taking values in $\mathbb{R} \cup \{\texttt{NA}\}$. The observed training set, which is available for statistical analysis, is then defined as $\widetilde{\mathcal{D}}_{n,\text{train}} = ((\widetilde{\mathbf{X}}_i, Y_i))_{1\le i\le n}$.

3.1 Out-of-sample imputation

Using missing value imputation in a supervised learning setting is not straightforward as it requires to impute new, out-of-sample, test data, where the target is unavailable.

A simple strategy is to fit an imputation model on the training set, yielding the estimated parameter $\hat{\alpha}$. $\hat{\mathbf{X}}_{\text{train}}$ denoting the imputed training data set, a predictive model can then be learned using $\hat{\mathbf{X}}_{\text{train}}$ and Y_{train}, yielding the estimated parameter $\hat{\beta}$. Finally, on the test set, the covariates must be imputed with the same imputation model (using $\hat{\alpha}$) and the dependent variable predicted using the imputed test set and the estimated learning model (using $\hat{\beta}$).

This approach is easy to implement for *univariate imputation* methods that consider each feature separately, for instance with mean imputation: the parameter $\hat{\alpha}$ corresponds to the mean $\hat{\mu}_i$ of each column which is learned on the training set, and any new observation on the test set can be imputed by $(\hat{\mu}_1, \ldots, \hat{\mu}_d)$. The imputation with joint Gaussian model on $(\mathbf{X}, Y)$ – which parameters are learned by the EM algorithm on the training set– is also appropriate as one can impute the test set using the conditional expectations of the missing features given the observed features (and without a Y) and the estimated parameters.

For more general imputation methods, two issues hinder out-of-sample imputation. First, many available imputation methods are "black-boxes" that take as input an incomplete data set and output a completed data set: they do not separate the estimation of model parameters from their usage to complete the data. This is the case for many implementations of iterative conditional imputation such as MICE (van Buuren, 2018) or missForest (Stekhoven and Bühlmann, 2011). It is also difficult for powerful imputers presented in Section 2.2.2 such as low-rank matrix completion, as they cannot be easily marginalised on $\mathbf{X}$ alone.

As most existing implementations cannot easily impute a new data set with the same imputation model, some analysts resort to performing separate imputation of the training set and the test set. But the smaller the test set, the more suboptimal this strategy is, and it completely fails in the case where only one observation has to be predicted. Another option is to consider semi-supervised settings, where the test set is available at train time: grouped imputation can then simultaneously impute the train and the test set (Kapelner and Bleich, 2015), while the predictive model is subsequently learned on the training set only.

3.2 EM and out-of-sample prediction

The likelihood framework (Section 2.2.1) enables predicting new observation, though it has not been much discussed. Jiang et al. (2018) consider a special case of this approach for a logistic regression and by assuming a Gaussian model on the covariates $\mathbf{X}$.

Let the assumptions of Theorem (1) be verified (MAR settings). Model parameters $\theta^\star$ can then be estimated by maximizing the observed log-likelihood ℓ_2 with an EM algorithm (Section 2.2.3). The corresponding maximum likelihood estimate $\hat{\theta}_n$ can be used for out-of-sample prediction with missing values. More precisely, for a fixed missing indicator $\mathbf{m}$, we write $\mathbf{x}_o = o(\mathbf{x}, \mathbf{m})$ the observed values and $\mathbf{x}_m$ the missing values. The probability of y as a function of the observed values $\mathbf{x}_o$ only, can be related to that on a fully-observed data set:

$$
\begin{aligned}
p_{\hat{\theta}_n}(y|\mathbf{x}_o) = \frac{p_{\hat{\theta}_n}(y, \mathbf{x}_o)}{p_{\hat{\theta}_n}(\mathbf{x}_o)} \qquad &= \frac{1}{p_{\hat{\theta}_n}(\mathbf{x}_o)} \int p_{\hat{\theta}_n}(y, \mathbf{x}_m, \mathbf{x}_o) \, \mathrm{d}\mathbf{x}_m \\
&= \frac{1}{p_{\hat{\theta}_n}(\mathbf{x}_o)} \int p_{\hat{\theta}_n}(y|\mathbf{x}_m, \mathbf{x}_o) p_{\hat{\theta}_n}(\mathbf{x}_m|\mathbf{x}_o) p_{\hat{\theta}_n}(\mathbf{x}_o) \, \mathrm{d}\mathbf{x}_m \\
&= \int p_{\hat{\theta}_n}(y|\mathbf{x}_m, \mathbf{x}_o) p_{\hat{\theta}_n}(\mathbf{x}_m|\mathbf{x}_o) \, \mathrm{d}\mathbf{x}_m \\
&= \mathbb{E}_{\mathbf{X}_m|\mathbf{X}_o=\mathbf{x}_o} \left[p_{\hat{\theta}_n}(y|\mathbf{X}_m, \mathbf{x}_o) \right] \qquad (2)
\end{aligned}
$$

It is then possible to approximate the expectation with Monte Carlo sampling from the distribution $p_{\hat{\theta}_n}(\mathbf{X}_m|\mathbf{X}_o = \mathbf{x}_o)$. Such a sampling is easy in simple models, *e.g.* using Schur's complements for Gaussian distributions in linear regression settings. But in more complex settings, such as logistic regression, there is no explicit solution and one option is Metropolis Hasting Monte Carlo.

3.3 Empirical risk minimization with missing data

The two previous approaches that we discussed are specifically designed to fix the missing data issue: imputing or specifying a parametric model and computing the probability of the response given the observed values. However, in supervised learning settings, the goal is rather to build a prediction function that minimizes an expected risk. Empirical risk minimization, the workhorse of machine learning, can be adapted to deal with missing data.

Recall that in missing data settings, we do not have access to $\mathbf{X}$ but rather to $\widetilde{\mathbf{X}}$. Therefore, given a class of functions $\mathscr{F}$ from $\widetilde{\mathcal{X}}$ to $\mathcal{Y}$, we aim at

minimizing the empirical risk on this class, that is

$$\widehat{f}_n \in \underset{f \in \mathscr{F}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f(\widetilde{\mathbf{X}}_i), Y_i\right). \tag{3}$$

Unfortunately, the half-discrete nature of $\widetilde{\mathcal{X}} = \bigotimes_{j=1}^{d} (\mathcal{X}_j \cup \{\texttt{NA}\})$, makes the problem difficult. Indeed, many learning algorithms do not work with mixed data types, such as $\mathbb{R} \cup \texttt{NA}$, but rather require a vector space. This is true in particular for gradient-based algorithms. As a result, the optimization problem (3) is hard to solve with typical learning tools.

Another point of view can be adopted for losses which leads to Bayes-optimal solutions such that $\widetilde{f}^{\star}(\widetilde{\mathbf{X}}) = \mathbb{E}[Y|\widetilde{\mathbf{X}}]$. As there are at most 2^d admissible missing patterns, we can rewrite the Bayes estimate as

$$\mathbb{E}\left[Y \middle| \widetilde{\mathbf{X}}\right] = \sum_{\mathbf{m} \in \{0,1\}^d} \mathbb{E}\left[Y | o(\mathbf{X}, \mathbf{m}), \mathbf{M} = \mathbf{m}\right] \; \mathbb{1}_{\mathbf{M}=\mathbf{m}}, \tag{4}$$

This formulation highlights the combinatorial issues: solving (3) may require, as suggested by Rosenbaum and Rubin (1984, Appendix B), to estimate 2^d different submodels, that is $\mathbb{E}\left[Y | o(\mathbf{X}, \mathbf{m}), \mathbf{M} = \mathbf{m}\right]$ appearing in (4) for each $\mathbf{m} \in \{0,1\}^d$, which grows exponentially with the number of variables.

Modifying existing algorithms or creating new ones to deal with the optimization problem (3) is in general a difficult task due to the numerous possible missing data patterns. We will see in Section 5 that decision trees are particularly well suited to address this problem.

Remark 1. *Note that in practice, not all patterns may be possible in the training and test sets. For instance, if there is only complete data in the train set, the only submodel of interest is* $\mathbb{E}\left[Y | o(\mathbf{X}, \mathbf{m}), \mathbf{M} = \mathbf{m}\right]$ *for* $\mathbf{m} = (0, \ldots, 0)$*, which boils down to the regular supervised learning scenario on a complete data. However, the train and test set are assumed to be drawn from the same data distribution. Hence, we expect to observe similar patterns of missingness in train and test sets. If this is not the case, it corresponds to a distributional shift, and should be tackled with dedicated methods (see, e.g., Sugiyama et al., 2017). This may happen for instance, when a study conducted on past data led to operational recommendations, making the practitioners measure systematically the variables of interest.*

4 Bayes-risk consistency of imputation procedures

In this section, we show theoretically that, without assuming any parametric distribution for the data, single imputation procedures can lead to a Bayes-optimal predictor in the presence of missing data on covariates (in both train and test sets), *i.e.* they asymptotically target the function $\widetilde{f}^{\star}(\widetilde{\mathbf{X}}) = \mathbb{E}[Y|\widetilde{\mathbf{X}}]$. We first focus on the risk of a predictor, consistent for the complete data, computed on test data with missing values, using several imputation strategies: unconditional mean, conditional mean and multiple imputation. We then consider the full problem of tackling missing values in the train and the test set. We study a classical approach, described in Section 3.1, which consists first in imputing the training set, learning on the imputed data, and predicting on a test set which has been imputed with the same method. Although mean imputation of variables is one of the most widely used approaches, it is highly criticised in the classic literature for missing data (Little and Rubin, 2002). Indeed, it leads to a distortion of the data distribution and consequently statistics calculated on the imputed data table are biased. A simple example is the correlation coefficient between two variables, which is biased towards zero if the missing data are imputed by the mean. However, in a supervised learning setting the aim is not to compute statistics representative of the data set, but to minimize a prediction risk by estimating a regression function. For this purpose, we show in Section 4.2 that mean imputation may be completely appropriate and leads to consistent estimation of the prediction function. This result is remarkable and extremely useful in practice.

4.1 Test-time imputation

Here we consider that we have an optimal (Bayes-consistent) predictor f for the complete data, *i.e.* $f(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}]$, and we show that when there is missing data in the test set, in MAR settings, multiple imputation with f can give the optimal prediction, *i.e.* Bayes consistent for incomplete data. In the case of MCAR values, *i.e.* where the complete data is a random subsample from the sample, the function f can be obtained for instance by "listwise deletion" in the train set: fitting a supervised learning procedure on data for which the samples with missing data have been removed.

4.1.1 Test-time conditional multiple imputation is consistent

Let us first make explicit the multiple imputation procedure for prediction. For a given vector $\widetilde{\mathbf{x}} \in (\mathbb{R} \cup \{\texttt{NA}\})^d$, we let $\mathbf{m}$ be the missing indicator and

write $\mathbf{x}_o = o(\mathbf{x}, \mathbf{m})$ for observed values and $\mathbf{x}_m$ the missing values. We then draw the missing values $\mathbf{X}_m$ from their distribution conditional on $\mathbf{X}_o = \mathbf{x}_o$ and compute the regression function on these completed observations. The resulting multiple imputation function is given by:

$$f^{\star}_{mult\ imput}(\widetilde{\mathbf{x}}) = \mathbb{E}_{\mathbf{X}_m|\mathbf{X}_o=\mathbf{x}_o}[f(\mathbf{X}_m, \mathbf{x}_o)]. \tag{5}$$

Note that this expression is similar to the expression Equation 2 given for EM but assuming that we know the true nonparametric distribution of the data.

Theorem 2. *Consider the regression model*

$$Y = f(\mathbf{X}) + \varepsilon,$$

where $\mathbf{X} = (X_1, \ldots, X_d)$ *takes values in* $\mathbb{R}^d$*, for all subset* $\mathcal{S} \subset \{1, \ldots, d\}$*,* $(M_j)_{j\in\mathcal{S}} \perp\!\!\!\perp (X_j)_{j\in\mathcal{S}}$ *conditional on* $(X_k)_{k\in\mathcal{S}^c}$ *(MAR mechanism) and where* $\varepsilon \perp\!\!\!\perp (M_1, X_1, \ldots, M_d, X_d)$ *is a centred noise. Then the multiple imputation procedure, defined in* (5)*, is consistent, that is, for all* $\widetilde{\mathbf{x}} \in (\mathbb{R} \cup \texttt{NA})^d$*,*

$$f^{\star}_{mult\ imput}(\widetilde{\mathbf{x}}) = \mathbb{E}[Y|\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}].$$

The proof is given in Appendix A.

4.1.2 Single mean imputation is not consistent

Given the success of multiple imputation, it is worth checking that single imputation is not sufficient. We show with two simple examples that indeed, single imputation on the test set is not consistent even in MAR setting.
We first show, that (unconditional) mean imputation is not consistent, if the learning algorithm has been trained on the complete cases only.

Example 1. *In one dimension, consider the following simple example,*

$$X_1 \sim U(0,1), \quad Y = X_1^2 + \varepsilon, \quad M_1 \sim \mathcal{B}(1/2) \perp\!\!\!\perp (X_1, Y),$$

with ε *an independent centered Gaussian noise. Here,* $\mathbb{E}[Y|X_1] = X_1^2$*, and the regression function* $\widetilde{f}^{\star}(\widetilde{\mathbf{X}}) = \mathbb{E}[Y|\widetilde{\mathbf{X}}]$ *satisfies*

$$\begin{aligned}
\widetilde{f}^{\star}(\widetilde{\mathbf{X}}) &= X_1^2 \cdot \mathbb{1}_{M_1=0} + \mathbb{E}[Y|\widetilde{X} = \texttt{NA}] \cdot \mathbb{1}_{M_1=1} \\
&= X_1^2 \cdot \mathbb{1}_{M_1=0} + \mathbb{E}[X_1^2] \cdot \mathbb{1}_{M_1=1} \\
&= X_1^2 \cdot \mathbb{1}_{M_1=0} + (1/3) \cdot \mathbb{1}_{M_1=1}.
\end{aligned} \tag{6}$$

In the oracle setting where the distribution of (X_1, Y, M_1) is known, "plugging in" the mean imputation of X_1 yields the prediction

$$\begin{aligned} f_{imputation}(\widetilde{X}) &= X_1^2 \cdot \mathbb{1}_{M_1=0} + (\mathbb{E}[X_1])^2 \cdot \mathbb{1}_{M_1=1} \\ &= X_1^2 \cdot \mathbb{1}_{M_1=0} + (1/4) \cdot \mathbb{1}_{M_1=1}. \end{aligned} \tag{7}$$

In this example, mean imputation is not optimal: when X_1 is missing, the prediction obtained by mean imputation is $1/4$, whereas the optimal prediction (the one which minimizes the square loss) is $1/3$ as seen in (6).
Inspecting (6) and (7) reveals that the poor performance of mean imputation are due to the fact that $\mathbb{E}[X_1^2] \neq (\mathbb{E}[X_1])^2$. The non-linear relation between Y and X_1 breaks mean imputation. This highlights the fact that the imputation method should be chosen in accordance with the learning algorithm that will be applied later on. This is related to the concept of congeniality (Meng, 1994) defined in multiple imputation.

4.1.3 Conditional mean imputation is consistent if there are deterministic relations between input variables

We now consider conditional mean imputation, using information of other observed variables to impute. Conditional mean imputation may work in situations where there is redundancy between variables, as highlighted in Example 2. However, we give a simple example below stressing that using it to impute the test may not be Bayes optimal.

Example 2. *Consider the following regression problem with two identical input variables:*

$$X_1 = X_2 \sim \mathcal{U}([0,1]), \quad Y = X_1 + X_2^2 + \varepsilon, \quad M_2 \sim \mathcal{B}(1/2) \perp\!\!\!\perp (X_1, X_2, Y)$$

The Bayes-optimal predictor is then given by

$$\begin{aligned} \widetilde{f}^{\star}(\widetilde{\mathbf{X}}) &= \begin{cases} X_1 + X_2^2 & \text{if } \widetilde{X}_2 \neq \texttt{NA} \\ X_1 + \mathbb{E}[X_2^2 | \widetilde{X}_2 = \texttt{NA}] & \text{if } \widetilde{X}_2 = \texttt{NA} \end{cases} \\ &= \begin{cases} X_1 + X_2^2 & \text{if } \widetilde{X}_2 \neq \texttt{NA} \\ X_1 + X_1^2 & \text{if } \widetilde{X}_2 = \texttt{NA} \end{cases} \end{aligned}$$

Imputation with the mean of X_2 in this function leads to

$$f_{imputation}(\widetilde{\mathbf{X}}) = \begin{cases} X_1 + X_2^2 & \text{if } \widetilde{X}_2 \neq \texttt{NA} \\ X_1 + (1/4) & \text{if } \widetilde{X}_2 = \texttt{NA} \end{cases}$$

whereas, imputing X_2 by its mean conditional on X_1 gives

$$f_{\textit{imputation using } X_1}(\widetilde{\mathbf{X}}) = \begin{cases} X_1 + X_2^2 & \textit{if } \widetilde{X}_2 \neq \texttt{NA} \\ X_1 + X_1^2 & \textit{if } \widetilde{X}_2 = \texttt{NA} \end{cases},$$

as $(\mathbb{E}[X_2|X_1])^2 = X_1^2$.

If there is no deterministic link between variables, conditional mean imputation fails to recover the regression function, in the case where the regression function is not linear (see Example 2, where $X_1 = X_2$ is replaced by $X_1 = X_2 + \varepsilon$).

4.1.4 Pathological case: missingness is a covariate

Example 3 shows a situation in which any imputation methods single or multiple fail, since missingness contains information about the response variable Y.

Example 3. *Consider the following regression model,*

$$X_1 \sim \mathcal{U}(0,1) \qquad M_1 \sim \mathcal{B}(1/2) \perp\!\!\!\perp X_1 \qquad Y = X_1 \cdot \mathbb{1}_{M_1=0} + 3X_1 \cdot \mathbb{1}_{M_1=1} + \varepsilon.$$

Here,

$$\mathbb{E}[Y|X_1] = X_1 \cdot \mathbb{P}(M_1 = 0) + 3X_1 \cdot \mathbb{P}(M_1 = 1) = 2X_1\,.$$

Unconditional mean imputation prediction is given by

$$\begin{aligned} f_{\textit{imputation}}(\widetilde{X}) &= X_1 \cdot \mathbb{1}_{M_1=0} + \mathbb{E}[X_1] \cdot \mathbb{1}_{M_1=1} \\ &= X_1 \cdot \mathbb{1}_{M_1=0} + (1/2) \cdot \mathbb{1}_{M_1=1}, \end{aligned}$$

whereas, the regression function satisfies

$$\begin{aligned} \widetilde{f}^{\star}(\widetilde{X}) &= X_1 \cdot \mathbb{1}_{M_1=0} + 3\mathbb{E}[X_1|\widetilde{X} = \texttt{NA}] \cdot \mathbb{1}_{M_1=1} \\ &= X_1 \cdot \mathbb{1}_{M_1=0} + (3/2) \cdot \mathbb{1}_{M_1=1}. \end{aligned}$$

In this case, the presence of missing values is informative in itself, and having access to the complete data set (all values of X_1) does not provide enough information. Such scenario advocates for considering the missingness as an additional input variable. Indeed, in such situation, single and multiple imputation fail to recover the targeted regression function, without adding a missingness indicator to the input variables.

4.2 Mean imputation at train and test time is consistent

We now show that learning on the mean-imputed training data, imputing the test set with the means (of the variables on the training data), and predicting is optimal if the missing data are MAR and if the learning algorithm is consistent when trained on complete data only. Theorem 3 deals with missing data on X_1 only. In that scenario, for each observed $\tilde{\mathbf{x}} \in \mathbb{R} \cup \texttt{NA}$, the imputed entry is defined as $\mathbf{x}' = (x_1', x_2, \ldots, x_d)$ where

$$x_1' = x_1 \mathbb{1}_{M_1=0} + \mathbb{E}[X_1]\mathbb{1}_{M_1=1}.$$

We need the following assumption.

Assumption 1 (Missingness pattern)**.** *The variables $X_2, \ldots, X_d$ are fully observed and the missingness pattern M_1 on variable X_1 satisfies $M_1 \perp\!\!\!\perp X_1 | X_2, \ldots, X_d$ and is such that the function $(x_2, \ldots, x_d) \mapsto \mathbb{P}[M_1 = 1 | X_2 = x_2, \ldots, X_d = x_d]$ is continuous.*

Theorem 3. *Consider the input vector $\mathbf{X} = (X_1, \ldots, X_d)$ which has a continuous density $g > 0$ on $[0,1]^d$, the response*

$$Y = f(\mathbf{X}) + \varepsilon$$

such that $\|f\|_\infty < \infty$. Assume that the missingness pattern satisfies Assumption 1 and that ε is a centered noise independent of $(\mathbf{X}, M_1)$. If, after mean imputation, one uses a learning algorithm that is universally consistent when trained on any fully observed data set, then, the overall procedure will predict, for all (possibly partially missing) entries $\mathbf{x}' \in \mathbb{R}^d$,

$$\begin{aligned} f^\star_{impute}(\mathbf{x}') = {} & \mathbb{E}[Y | X_2 = x_2, \ldots, X_d = x_d, M_1 = 1]\mathbb{1}_{x_1' = \mathbb{E}[X_1]}\mathbb{1}_{\mathbb{P}[M_1=1|X_2=x_2,\ldots,X_d=x_d]>0} \\ & + \mathbb{E}[Y | \mathbf{X} = \mathbf{x}']\mathbb{1}_{x_1' = \mathbb{E}[X_1]}\mathbb{1}_{\mathbb{P}[M_1=1|X_2=x_2,\ldots,X_d=x_d]=0} \\ & + \mathbb{E}[Y | \mathbf{X} = \mathbf{x}', M_1 = 0]\mathbb{1}_{x_1' \neq \mathbb{E}[X_1]}. \end{aligned}$$

Letting

$$\widetilde{\mathbf{X}} = \begin{cases} \mathbf{X}' & \text{if } X_1' \neq \mathbb{E}[X_1] \\ (\texttt{NA}, X_2, \ldots, X_d) & \text{if } X_1' = \mathbb{E}[X_1] \end{cases},$$

the mean imputation prediction is equal to the Bayes function almost everywhere, that is

$$f^\star_{impute}(\mathbf{X}') = \widetilde{f}^\star(\widetilde{\mathbf{X}}). \tag{8}$$

The proof is given in Appendix A. Theorem 3 confirms that it is preferable to use the same imputation for the train and the test set. Indeed, the learning algorithm can learn the imputed value (here the mean) and use that information to detect that the entry was initially missing. If the imputed value changes from train set to test set (for example, if instead of imputing the test set with the mean of the variables of the train set, one imputes by the mean of the variables on the test set), the learning algorithm may fail, since the imputed data distribution would differ between train and test sets. Note that the precise imputed value does not matter if the learning algorithm is consistent when trained on fully observed data. By default, the mean is not a bad choice even if it only preserves the first order statistic (mean) of the sample. Theorem 3 remains valid when missing values occur for variables $X_1, \ldots, X_j$ under the assumption that $(M_1, \ldots, M_j) \perp\!\!\!\perp (X_1, \ldots, X_j)$ conditional on $(X_{j+1}, \ldots, X_d)$ and if for every pattern $\mathbf{m} \in \{0,1\}^j \times \{0\}^{d-j}$, the functions $(x_{j+1}, \ldots, x_d) \mapsto \mathbb{P}[\mathbf{M} = \mathbf{m} | X_{j+1} = x_{j+1}, \ldots, X_d = x_d]$ are continuous.

Almost everywhere consistency. Note that the equality between the mean imputation learner and the Bayes function holds almost surely but not for every $\tilde{\mathbf{x}}$. Indeed, under the setting of Theorem 3, let $\tilde{\mathbf{x}} = (\mathbb{E}[X_1], x_2, \ldots, x_d)$, for any $x_2, \ldots, x_d \in [0,1]$ such that

$$\mathbb{P}[M_1 = 1 | X_2 = x_2, \ldots, X_d = x_d] > 0.$$

In this case, $\mathbf{x}' = (\mathbb{E}[X_1], x_2, \ldots, x_d)$ and

$$f^\star_{impute}(\mathbf{x}') = \mathbb{E}[Y | X_2 = \mathbf{x}_2, \ldots, \mathbf{X}_d = \mathbf{x}_d, M_1 = 1],$$

which is different, in general, from

$$\widetilde{f}^\star(\widetilde{\mathbf{x}}) = \mathbb{E}[Y | X_1 = \mathbb{E}[X_1], X_2 = x_2, \ldots, X_d = x_d].$$

Therefore, on the event $A_1 = \{\tilde{\mathbf{X}}, \tilde{X}_1 = \mathbb{E}[X_1]\}$, the two functions $f^\star_{impute}$ and $\widetilde{f}^\star$ differ. Since A_1 is a zero probability event, the equality $f^\star_{impute}(\mathbf{X}') = \widetilde{f}^\star(\widetilde{\mathbf{X}})$ does not hold pointwise (as shown above) but hold almost everywhere (as stated in Theorem 3). However, a simple way to obtain the pointwise equality in equation (8) is to impute missing data by values that are out of the range of the true distribution, which echoes the "separate class" method advocated by Ding and Simonoff (2010).

5 Decision trees: an example of empirical risk minimization with missing data

Decision trees offer a natural way for empirical risk minimization with missing values. This is in part due to their ability to handle the half-discrete nature of $\tilde{\mathbf{X}}$.

We first present the different approaches available to handle missing values in tree-based methods in Sections 5.2 and 5.3. We then compare them theoretically in Section 5.4 and showing the interest of using the "missing incorporated in attribute" approach whether the missing values are MCAR or informative.

5.1 Tree construction with CART

The algorithm CART (Classification And Regression Trees, Breiman et al. 1984) is one of the most popular tree algorithm, originally designed for complete data sets. Each leaf of the tree defines an interval on each variable $A = \prod_{j=1}^{d} [a_{j,L}, a_{j,R}] \subset \mathbb{R}^d$. On A, the algorithm finds the best split $(j^\star, z^\star) \in \mathcal{S}$, where a split is defined by the choice of a feature j along which the split is performed and the position z of the split. Writing $\mathcal{S} = \{(j, z), j \in [\![1, d]\!],\ z \in \mathbb{R}, z_j \in [a_{j,L}, a_{j,R}]\}$ the set of all possible splits in the cell A, the best split is defined as the solution of the following optimization problem

$$(j^\star, z^\star) \in \underset{(j,z)\in\mathcal{S}}{\operatorname{argmin}}\ \mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X_j \le z, \mathbf{X} \in A]\big)^2 \cdot \mathbb{1}_{X_j \le z, \mathbf{X}\in A} + \big(Y - \mathbb{E}[Y|X_j > z, \mathbf{X} \in A]\big)^2 \cdot \mathbb{1}_{X_j > z, \mathbf{X}\in A}\Big]. \tag{9}$$

For each cell A, the problem (9) can be rewritten as

$$f^\star \in \underset{f\in\mathcal{P}_c}{\operatorname{argmin}}\ \mathbb{E}\Big[\big(Y - f(\mathbf{X})\big)^2 \mathbb{1}_{\mathbf{X}\in A}\Big], \tag{10}$$

where $\mathcal{P}_c$ is the set of piecewise-constant functions on $A \cap \{x_j \le s\}$ and $A \cap \{x_j > s\}$ for $(j, s) \in \mathcal{S}$. Therefore the optimization problem (10) amounts to solving a least square problem on the subclass of functions $\mathcal{P}_c$. Thus, by minimizing the mean squared error, the CART procedure targets the quantity $\mathbb{E}[Y|\mathbf{X}]$. In the presence of missing values, this criterion must be adapted and several ways to do so have been proposed, as detailed below.

5.2 Splitting criterion discarding missing values

The most popular option is to select the split only on the available cases for each variable:

$$(j^\star, z^\star) \in \underset{(j,z)\in\mathcal{S}}{\operatorname{argmin}}\, C(j,s), \tag{11}$$

$$\begin{aligned}\text{where}\quad C(j,s) =& \mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X_j \le z, \mathbf{X}\in A, M_j = 0]\big)^2 \cdot \mathbb{1}_{X_j\le z, \mathbf{X}\in A, M_j=0} \\ &+ \big(Y - \mathbb{E}[Y|X_j > z, \mathbf{X}\in A, M_j = 0]\big)^2 \cdot \mathbb{1}_{X_j> z, \mathbf{X}\in A, M_j=0}\Big].\end{aligned}$$

As the missing values were not used to construct the criterion, it is still necessary to specify to which cell they are sent. Indeed, the solution of discarding missing data at each step would lead to a drastic reduction of the data set. The different methods to propagate missing data down the tree are detailed below.

Surrogate splits Once the best split is chosen, surrogate splits search for a split on another variable that induces a data partition close to the original one. More precisely, let $(j_0^\star, z_0^\star)$ be the selected split. To send down the tree observations with no $j_0^\star$th variable, a new stump, *i.e.*, a tree with one cut, is fitted to the response $\mathbb{1}_{X_{j_0^\star}\le z_0^\star}$, using variables $(X_j)_{j\neq j_0^\star}$. The split $(j_1^\star, z_1^\star)$ which minimizes the misclassification error is selected, and observations are split accordingly. Those that lack both variables $j_0^\star$ and $j_1^\star$ are split with the second best, $j_2^\star$, and so on until the proposed split has a worse misclassification error than the blind rule of sending all remaining missing values to the same daughter, the most populated one. In the predicting phase, the training surrogates are kept. They are the default method in the `rpart` implementation (Therneau et al., 1997). Surrogate method is expected to be appropriate when there are relationship between the covariates.

Probabilistic splits Another possibility is to propagate missing observations according to a Bernoulli distribution $\mathcal{B}(\frac{\#L}{\#L+\#R})$, where $\#L$ (resp. $\#R$) is the number of points already on the left (resp. right). This is the default method in the C4.5 algorithm (Quinlan, 2014).

Block propagation The third choice is to send all incomplete observations as a block, to a side chosen by minimizing the error. This is the method

in XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017).

Note that Hothorn et al. (2006) proposed conditional trees, which adapt the criterion (11) to missing values. Indeed, this criterion implies a selection bias: it leads to underselecting the variables with many missing values due to the multiple comparison effects (Strobl et al., 2007). As a result, it favors variables where many splits are available, and therefore those with fewer missing values. Conditional trees are based on the calculation of a linear statistic of association between Y and each feature X_j, $T = \langle X_j, Y \rangle$ on the observed feature. Then, its distribution under the null hypothesis of independence between Y and X_j is estimated by permutation and the variable with the smallest p-value is selected. Note that the improvement, illustrated in Appendix B.1, is meant to be on the selection of the variables but does not ensure that it improves the prediction performance. Once the variables have been selected, Hothorn et al. (2006) use surrogate splits to propagate the missing entries. One potential drawback of this approach is the use of a linear statistic for association.

5.3 Splitting criterion with missing values: MIA

The second important class of methods uses missing values to compute the criterion for each split and thus best split location. Its most common instance is "missing incorporated in attribute" (MIA, Twala et al. 2008). More specifically, MIA selects

$$f^\star \in \underset{f \in \mathcal{P}_{c,miss}}{\operatorname{argmin}} \ \mathbb{E}\Big[\big(Y - f(\widetilde{\mathbf{X}})\big)^2 \mathbb{1}_{\widetilde{\mathbf{X}} \in A}\Big], \tag{12}$$

where $\mathcal{P}_{c,miss} = \mathcal{P}_{c,miss,L} \cup \mathcal{P}_{c,miss,R} \cup \mathcal{P}_{c,miss,sep}$ with

- $\mathcal{P}_{c,miss,L}$ is the set of all functions piecewise constant on a partition of the form $\{\{\widetilde{X}_j \leq z \vee \widetilde{X}_j = \texttt{NA}\}, \{\widetilde{X}_j > z\}\}$, for any $z \in \mathbb{R}, j \in [\![1, d]\!]\}$.
- $\mathcal{P}_{c,miss,R}$ is the set of all functions piecewise constant on a partition of the form $\{\{\widetilde{X}_j \leq z\}, \{\widetilde{X}_j > z \vee \widetilde{X}_j = \texttt{NA}\}\}$, for any $z \in \mathbb{R}, j \in [\![1, d]\!]\}$.
- $\mathcal{P}_{c,miss,sep}$ is the set of all functions piecewise constant on a partition of the form $\{\{\widetilde{X}_j \neq \texttt{NA}\}, \{\widetilde{X}_j = \texttt{NA}\}\}$, for any $j \in [\![1, d]\!]\}$.

This means that the missing values are treated like a category by the algorithm, they are simply distinct from real numbers which is appropriate to handle the space $\mathbb{R} \cup \texttt{NA}$. It is a greedy algorithm to minimize a square loss

between Y and a function of $\widetilde{\mathbf{X}}$ and consequently targets the quantity (4) which separate $\mathbb{E}[Y|\widetilde{\mathbf{X}}]$ into 2^d terms. However, it is not exhaustive: at each step, the tree can cut for each variable according to missing or non missing and selects this cut when it is relevant, *i.e.* when it minimizes the prediction error. The final leaves can correspond to a cluster of missing values patterns (observations with missing values on the two first variables for instance and any missing patterns for the other variables). MIA is thought to be a good method to apply when missing pattern is informative, as this procedure allows to cut with respect to missing/ non missing and uses missing data to compute the best splits. Note this latter property implies that the MIA approach does not require a different method to propagate missing data down the tree.

Remark 2. *Implicit imputation: Whether it is in the case where the missing values are propagated in the available case method (Section 5.2), or incorporated in the split choice in MIA, missing values are assigned either to the left or the right interval. Consequently, handling missing values in a tree can be seen as implicit imputation by an interval.*

5.4 Theoretical comparison of CART versus MIA

We now compare theoretically the positions of the splitting point at the root and the prediction errors on simple examples with MCAR values. Proposition 1 computes the splitting position of MIA and CART, and highlights that the splitting position of MIA varies even for MCAR missing data. Proposition 2 then compares the risk of the different splitting strategies: probabilistic split, block propagation, surrogate split, and MIA. We prove that MIA and surrogate splits are the two best strategies, one of which may be better than the other depending on the dependence structure of covariables.

Proposition 1. *Let $p \in [0,1]$. Consider the regression model*

$$\left\{ \begin{array}{rcl} Y & = & X_1 \\ X_1 & \sim & U([0,1]) \end{array} \right. , \quad \left\{ \begin{array}{rcl} \mathbb{P}[M_1 = 0] & = & 1-p \\ \mathbb{P}[M_1 = 1] & = & p \end{array} \right. ,$$

where $M_1 \perp\!\!\!\perp (X_1, Y)$ is the missingness pattern on X_1. Let $C_{MIA}(j, s, q, p)$ be the value of the splitting MIA criterion computed at $(1, s) \in \mathcal{S}, q \in \{L, R\}$, where q stands for the side where missing values are sent. Therefore,

1. *The best split $s^\star$ given by the CART criterion (11) is $s^\star = 1/2$.*

2. *The best splits* $s^\star_{MIA,L}(p)$ *and* $s^\star_{MIA,R}(p)$ *given by the MIA criterion (12), assuming that all missing values are sent to the left node (resp. to the right node), satisfy*

$$s^\star_{MIA,L}(p) = \underset{s\in[0,1]}{\operatorname{argmin}}\ C_{MIA}(1, s, L, p), \tag{13}$$

where

$$C_{MIA}(1, s, L, p) = \frac{1}{3} - \frac{1}{p + (1-p)s}\Big(\frac{p}{2} + \frac{(1-p)s^2}{2}\Big)^2 - (1-p)(1-s)\Big(\frac{1+s}{2}\Big)^2,$$

and $s^\star_{MIA,R}(p) = 1 - s^\star_{MIA,L}(p)$.

The proof is given in Appendix A. Proposition 1 shows that the split given by optimizing the CART criterion does not depend on the percentage p of missing values since the pattern is independent of (X, Y). A numerical solution to equation (13) is displayed in Figure 1. When there are no missing values ($p = 0$), the split occur at $s = 1/2$ as expected. When p increases, the threshold does not correspond anymore to the one calculated using observed values only as it is influenced by the missing entries even in the MCAR setting. Indeed, with MIA the threshold is selected as the one minimizing the prediction error. Hence MIA optimizes both the threshold and the side of the split on which it sends all the missing entries such that the prediction error is the smallest. This is important as it allows to propagate a new observation in the test set with missing values.
Recall that the quadratic risk R of a function $f^\star$ is defined as $R(f^\star) = \mathbb{E}[(Y - f^\star(X))^2]$. Proposition 2 enables us to compare the risk of a single split performed with the different strategies. It highlights that even in the simple case of MCAR, MIA gives more accurate predictions than block propagation or probabilistic split.

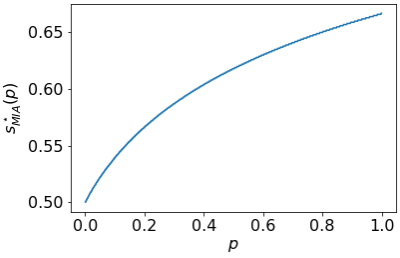


Figure 1: $s^\star_{MIA}$, the split chosen by the MIA criterion, as a function of the fraction p of missing values on X_1, assuming values are sent left.

Proposition 2. *Consider the regression model*

$$\left\{\begin{array}{rcl} Y & = & X_1 \\ X_1 & \sim & U([0,1]) \\ X_2 & = & X_1 \mathbb{1}_{W=1} \end{array}\right. , \quad \left\{\begin{array}{rcl} \mathbb{P}[W=0] & = & \eta \\ \mathbb{P}[W=1] & = & 1-\eta \end{array}\right. , \quad \left\{\begin{array}{rcl} \mathbb{P}[M_1=0] & = & 1-p \\ \mathbb{P}[M_1=1] & = & p, \end{array}\right. ,$$

where $(M_1, W) \perp\!\!\!\perp (X_1, Y)$. *The random variable* M_1 *is the pattern of missingness for* X_1 *and* W *stands for the link between* X_1 *and* X_2. *Let* $f^\star_{MIA}$, $f^\star_{block}$, $f^\star_{prob}$, $f^\star_{surr}$ *be respectively, the theoretical prediction resulting from one split according to MIA, CART with block propagation and CART with probabilistic splitting strategy, and a single split, where missing data are handled via surrogate split (in the infinite sample setting). We have*

$$\begin{aligned} R(f^\star_{MIA}) &= \min_{s\in[0,1]} C_{MIA}(1,s,L,p)\mathbb{1}_{p\leq 1-\eta} + \min_{s\in[0,1]} C_{MIA}(1,s,L,1-\eta)\mathbb{1}_{p>1-\eta}, \\ R(f^\star_{block}) &= C_{MIA}(p,1/2) \\ R(f^\star_{prob}) &= -\frac{p^2}{16} + \frac{p}{8} + \frac{1}{48}, \\ R(f^\star_{surr}) &= \frac{1}{48} + \frac{6}{48}\eta p. \end{aligned}$$

where $C_{MIA}(1,s,L,p)$ *is defined in Proposition 1. In particular,*

$$R(f^\star_{MIA}) \leq R(f^\star_{block}) \quad \textit{and} \quad R(f^\star_{MIA}) \leq R(f^\star_{prob}).$$

Proof is given in Appendix A.
Figure 2 depicts the risk of each estimate, in the context of proposition 2, resulting from a split computed via one of the four methods described above. Only surrogate and MIA risks depend on the value η which measures the independence between X_1 and X_2. As proved, the risk of probabilistic split and block propagation is larger than that of MIA. Besides, surrogate split is better than MIA if the link between X_1 and X_2 is strong (small values of η) and worse if this link is weak (high values of η).

6 Simulations

This section illustrates experimentally the take-home messages of the article. The code for these experiments is available online[1]. First, mean imputation can be appropriate and is consistent in a supervised learning setting when

[1] https://github.com/nprost/supervised_missing

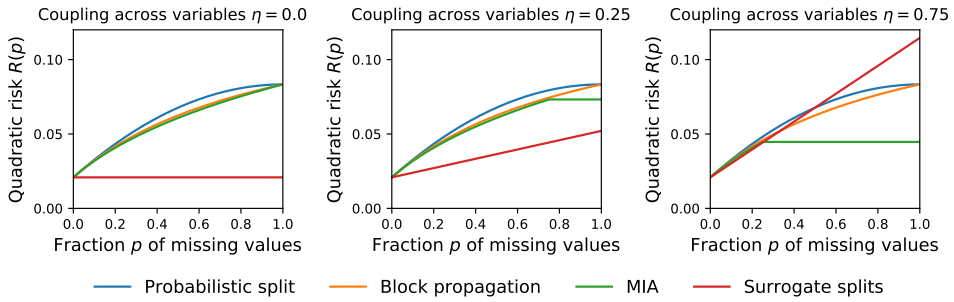


Figure 2: Theoretical risk of the four splitting methods in function of p, for three values of η parameter that controls the amount of coupling between X_1 and X_2 in the model of Proposition 2.

missing values are MAR and not related to the outcome. Second, tree-based methods are an efficient way to target $\widetilde{f}^{\star}(\widetilde{\mathbf{X}}) = \mathbb{E}[Y|\widetilde{\mathbf{X}}]$ especially when using MIA (Section 5.3) and can handle well informative pattern of missing values. While Proposition 2 compares the risk of the tree methods for a single split, simulations allow us to study grown trees.

We compare imputation methods, using the "proper way" to impute as described in Section 3.1, *i.e.*, where imputation values from the training set are used to impute the test set.

In addition, we consider imputation with the missing indicator $\mathbf{M}$ in the features. The rationale behind this indicator is that it can be useful to improve the prediction when going beyond the hypothesis of Theorem 3, *i.e.* considering a finite sample, a learning algorithm with a low approximation capacity (as linear regression) and with missing values that can either be MNAR or depend on Y.

6.1 Simulation setting

We consider three regression models, with covariates $(X_1, \ldots, X_d)$ distributed as $\mathcal{N}(\mu, \Sigma)$ with $\mu = \mathbf{1}_d$ and $\Sigma = \rho \mathbf{1}\mathbf{1}^T + (1-\rho) I_d$. The first model is quadratic, the second one is linear, and the third one has been used as a benchmark for tree methods by several authors, including Friedman (1991) and Breiman (1996). We also consider a last regression model where the relationship between covariables are nonlinear. In all four models, ε is a centered gaussian noise with standard deviation 0.1, which makes it proportionally smaller in model 4 than in the others.

Model 1 (Quadratic)**.** $Y = X_1^2 + \varepsilon$

Model 2 (Linear). $Y = X\beta + \varepsilon$ *with* $\beta = (1, 2, -1, 3, -0.5, -1, 0.3, 1.7, 0.4, -0.3)$.

Model 3 (Friedman). $Y = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \varepsilon$

Model 4 (Friedman, Nonlinear). $Y = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \varepsilon$ *where* X *is a hidden uniform variable on* $[-3, 0]$ *and the covariates* $(X_1, \ldots, X_d)$ *are distributed as*

$$\begin{cases} X_1 &= X^2 + \varepsilon_1 \\ X_2 &= \sin(X) + \varepsilon_2 \\ X_3 &= \tanh(X)\exp(X)\sin(X) + \varepsilon_3 \\ X_4 &= \sin(X-1) + \cos(X-3)^3 + \varepsilon_4 \\ X_5 &= (1-X)^3 + \varepsilon_5 \end{cases} \quad \begin{cases} X_6 &= \sqrt{\sin(X^2) + 2} + \varepsilon_6 \\ X_7 &= X - 3 + \varepsilon_7 \\ X_8 &= (1-X)\sin(X)\cosh(X) + \varepsilon_8 \\ X_9 &= \frac{1}{\sin(2X) - 2} + \varepsilon_9 \\ X_{10} &= X^4 + \varepsilon_{10} \end{cases},$$

where ε_i *are independent centered Gaussian with standard deviation 0.05.*

In the first experiment, we use Model 1 with $d = 3$ and introduce missing values on X_1 according to the following mechanisms. Results are depicted in Figure 3.

Missing Pattern 1 (MCAR). *For* $p \in [0, 1]$ *the missingness is generated according to a Bernoulli distribution*

$$\forall i \in [\![1, n]\!], M_{i,1} \sim \mathcal{B}(p).$$

Missing Pattern 2 (Censoring MNAR). *A direct way to select a proportion* p *of missing values on a variable, that depends on the underlying value, is to crop them above the* $1 - p$*-th quantile*

$$\forall i \in [\![1, n]\!], M_{i,1} = \mathbb{1}_{X_{i,1} > [X_1]_{(1-p)n}}.$$

Missing Pattern 3 (Predictive missingness). *Last, we can consider a pattern mixture model, letting* M_1 *be part of the regression function, with* $M_1 \perp\!\!\!\perp \mathbf{X}$ *and*

$$Y = X_1^2 + 3M_1 + \varepsilon.$$

In the second experiment, the other three models are used with $d = 10$, with a MCAR mechanism on all variables. Results are shown in Figure 4. We compare the following methods using implementation in the R (R Core Team, 2018) software and default values for the tuning parameters. We run a first comparison with decision trees, a second with random forests, and a third with gradient boosting. Indeed, while we have mostly covered single decision trees in this paper, their aggregation into boosting or random forests

is much more powerful in practice. Unless stated otherwise, the package used for decision trees is rpart (Therneau and Atkinson, 2018), the package used for random forests is ranger (Wright and Ziegler, 2015) and for gradient boosting XGBoost (Chen and Guestrin, 2016). Note that we have used surrogate splits only with single decision trees.

- **MIA**: missing in attributes, implemented as described in Remark 3
- **block**: block propagation, implemented only in XGBoost (Chen and Guestrin, 2016)
- **rpart+mask/ rpart**: CART with surrogate splits, with or without the indicator $\mathbf{M}$ in the covariates
- **ctree+mask/ ctree**: conditional trees, implemented in package partykit (Hothorn and Zeileis, 2015) with or without the indicator $\mathbf{M}$ in the covariates
- **impute mean+mask/ impute mean**: CART when missing values are imputed by unconditional mean with or without the indicator $\mathbf{M}$ added in the covariates
- **impute Gaussian**: CART when missing values are imputed by conditional expectation when data are assume to follow a Gaussian multivariate distribution. More precisely, the parameters of the Gaussian distribution are estimated with an EM algorithm (R package norm (Fox, 2013)). Note that for numerical reasons, we shrink the estimated covariance matrix (replacing $\hat{\Sigma}$ by $0.99 \times \hat{\Sigma} + 0.01 \times \mathrm{tr}(\hat{\Sigma})\mathrm{I}_\mathrm{d}$) before imputing.

To train and evaluate the performance of the methods, we decompose the observations into a training set (80%) and testing set (20%), and repeat the process 500 times. The metric is the percentage of explained variance (*i.e.* the R^2 statistic) computed on the test set. For visual purposes, we display the *relative* explained variance: for each of the 500 repetitions separately, we center the scores of all the methods by substracting the mean. This is also done separately for trees, forests, boosting.

Remark 3. *From a practical point of view, a simple way to implement MIA consists in duplicating the incomplete columns, and replacing the missing entries once by $+\infty$ and once by $-\infty$ (or an extreme out-of-range value). This creates two dummy variables for each original one containing missing values. Splitting along a variable and sending all missing data to the left (for example) is the same as splitting along the corresponding dummy variable where missing entries have been completed by $-\infty$.*

6.2 Results

Figure 3 presents the results for one choice of correlation between covariables and percentage of missing entries, as others give similar interpretation.

In the MCAR case, all decision tree methods behave in the same way. Having previously performed a "good" imputation, *i.e.* one that captures the relationships between variables such as *impute Gaussian* in relation to mean imputation, slightly helps prediction. This is all the most true as the correlation between variables increases, which we have not displayed.

When the pattern is more complex (MNAR or predictive missingness), there are clear differences between methods. As expected, MIA achieves excellent performance, together with the other methods that code for missing values by adding the mask. The classic versions of rpart and ctree (cut-off on observed data and surrogate split) are not sufficient and need the addition of the mask. What is perhaps remarkable is that mean imputation with rpart still has excellent performances. Boosting exhibits the same behaviour as single decision trees. For random forests however mean imputation underperforms without the addition of a mask.

Figure 4 compares the results for different datasets with values missing com-

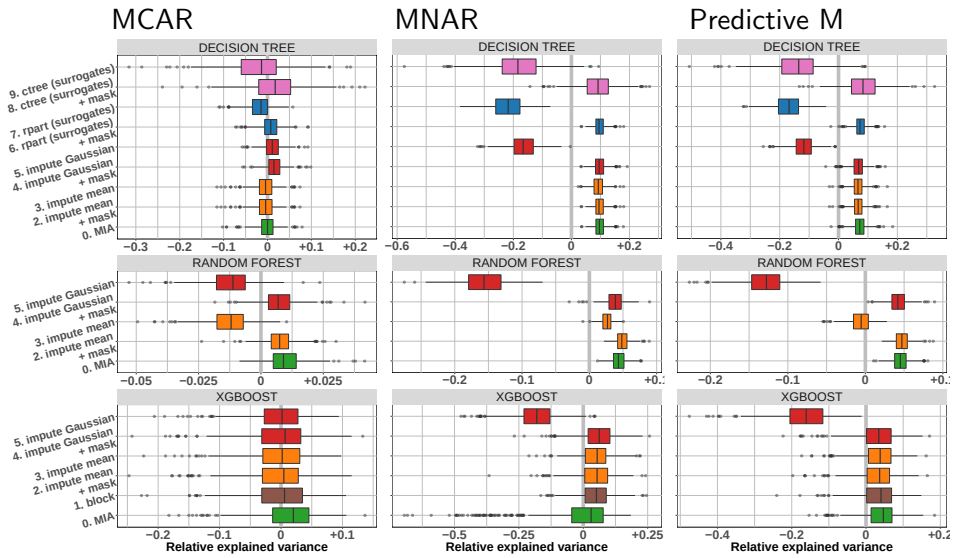


Figure 3: **Relative scores on model 1** • Relative explained variance for different mechanisms with 20% of missing values, $n = 1000$, $d = 3$ and $\rho = 0.5$.

pletely at random. The non-linearity does not seem to hinder the Gaussian imputation, as it performs better than mean imputation and MIA in most cases. All in all, MIA proves to be a strong option in all the scenarios that we have experimented, although Gaussian imputation with the mask is always equally good, and better in some cases. In these experiments, MIA and block propagation give similar results.

6.3 Consistency

In the third experiment, we compare the methods of Section 6.1 varying sample size to assess their asymptotic performances, on models 2, 3 and 4. We wish to compare the tree performance with respect to the Bayes risk. For each sample size (between 300 and 10^5), we summarize 200 repetitions by their median and quartiles (as in the boxplots). Assuming MCAR, the Bayes estimator is the expectation of Y conditionally to the observed values,

$$\mathbb{E}[Y|\widetilde{\mathbf{X}}] = \mathbb{E}[f(\mathbf{X})|\widetilde{\mathbf{X}}] = \mathbb{E}[f(\mathbf{X})|o(\mathbf{X},\mathbf{M})].$$

It has an easy closed expression only if the joint distribution of $(\mathbf{X}, Y)$ is Gaussian. To compute an approximate Bayes rate for a nonlinear regression

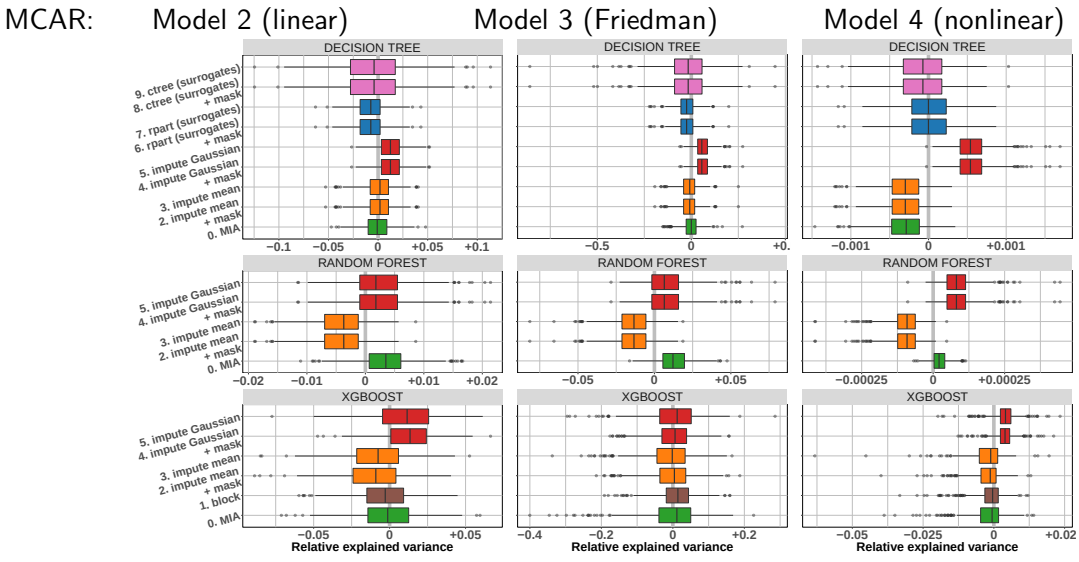


Figure 4: **Relative scores on different models in MCAR** • Relative explained variance for models 2, 3, 4, MCAR with 20% of missing values, $n = 1000$, $d = 10$ and $\rho = 0.5$.

with Gaussian features, we apply joint Gaussian multiple imputation, as justified in Section 4.1.1, on a very large sample. For the third case scenario where the features are not Gaussian, we have not computed the Bayes rate. In linear settings, Figure 5 (left) shows that *impute Gaussian* benefits from correlations between features and is the best-performing method; For decision trees and forests, mean imputation, MIA and surrogate splits are also consistent but with a slower convergence rate (we have not displayed con-

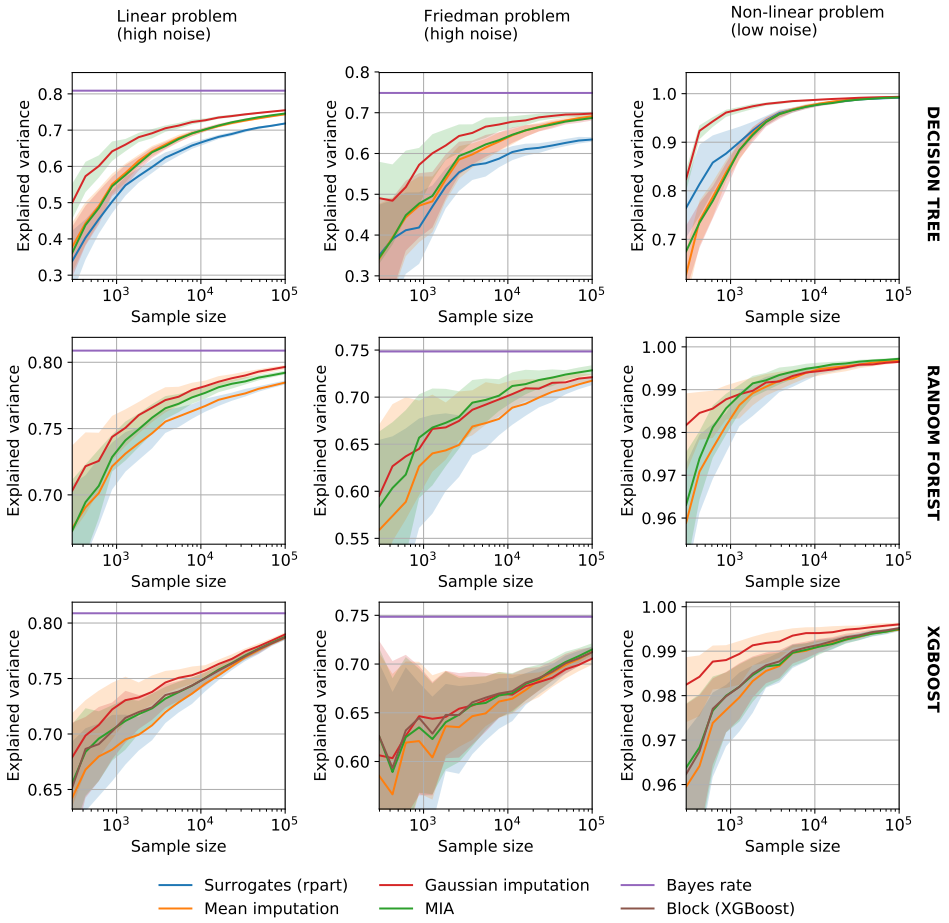


Figure 5: **Bayes consistency in MCAR** • Consistency with 40% of MCAR values on all variables, on models 2 (linear), 3 (Friedman), 4 (non-linear). Note that on the last row, the lines for MIA and block almost overlap.

ditional trees as they exhibit the same behaviour as rpart with surrogate splits). Adding the indicator matrix in the data changes almost nothing here, so we have not displayed the corresponding curves. For non-linear associations (Figure 5, middle and right), the benefit brought by Gaussian imputation over the others methods seems to carry over though it is less pronounced for random forests and boosting. For low-noise settings (Figure 5, right) MIA and mean imputation seem equivalent.
For boosting, the difference between methods vanishes with large n, as can be expected from boosting's ability to turn weak learners into strong ones (Schapire, 1990). Gaussian imputation is still beneficial for small sample sizes. Here again, block propagation overlaps with MIA, showing equally good performances for all sample sizes. Block propagation is interesting from a practical point of view, because it is implemented in computationally-efficient way in XGBoost and LightGBM. Here it performs well empirically, but proposition 2 and Figure 2 show that it can underperform compared to MIA for strongly coupled variables. Note that MIA can easily be implemented as a preprocessing step, as detailed in remark 3.

7 Discussion and conclusion

We have studied procedures for supervised learning with missing data. Unlike in the classic missing data literature, the goal of the procedures is to yield the best possible prediction on test data with missing values. Our theoretical and empirical results outline simple practical recommendations:

- Given a model suitable for the fully observed data, good prediction can be achieved on a test set by multiple imputation of its missing values with a conditional imputation model fit on the train set (Theorem 2).
- To train and test on data with missing values, the same imputation model should be used. Single mean imputation is consistent, provided a powerful, non-linear model (Theorem 3).
- For tree-based models, a good solution for missing values is Missing Incorporated in Attribute (MIA, Twala et al. 2008, see implementation Remark 3), which optimizes not only the split but also the handling of the missing values (Proposition 2 and experimental results). Experiments show that block propagation can also be a good option.
- Empirically, good imputation methods applied at train and test time reduce the number of samples required to reach good prediction (Figure 5).
- When missingness is related to the prediction target, imputation does not suffice and it is useful to add indicator variables of missing entries as

features (Example 3 and Figure 3).

These recommendations hold to minimize the prediction error in an asymptotic regime. More work is needed to establish theoretical results in the finite sample regime. In addition, different practices may be needed to also control for the uncertainty associated to a prediction.

Acknowledgement We thank Stefan Wager for fruitful discussion and Julie Tibshirani for the suggestion to implement quickly MIA. We also thank Antoine Ogier who initiated our work on imputation in supervised learning during his internship. We acknowledge funding from ANR under grant ANR-17-CE23-0018 – DirtyData, and DataIA, under grant MissingBigData.

References

Allison, P. D. (2001). *Missing data*, volume 136. Sage publications.

Arel-Bundock, V. and Pelc, K. J. (2018). When can multiple imputation improve regression estimates? *Political Analysis*, 26(2):240–245.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer Cambridge, UK.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *sigkdd international conference on knowledge discovery and data mining*, page 785. ACM.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Davidian, M. (2017). Statistical methods for analysis with missing data - course.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Devroye, L., Györfi, L., and Lugosi, G. (2013). *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media.

Ding, Y. and Simonoff, J. S. (2010). An investigation of missing data methods for classification trees applied to binary response data. *Journal of Machine Learning Research*, 11(Jan):131–170.

Fox, M. J. (2013). Package 'norm'.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67.

Gondara, L. and Wang, K. (2018). Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, page 260.

Hastie, T., Mazumder, R., Lee, J. D., and Zadeh, R. (2015). Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674.

Hothorn, T. and Zeileis, A. (2015). partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, 16:3905.

Jiang, W., Josse, J., and Lavielle, M. (2018). Logistic regression with missing covariates–parameter estimation, model selection and prediction. *arXiv preprint arXiv:1805.04602*.

Jones, M. P. (1996). Indicator and stratification methods for missing explanatory variables in multiple linear regression. *Journal of the American statistical association*, 91(433):222–230.

Josse, J. and Reiter, J. P. (2018). Introduction to the special section on missing data. *Statist. Sci.*, 33(2):139–141.

Josse, J., Sardy, S., and Wager, S. (2016). denoiser: A package for low rank matrix estimation. *Journal of Statistical Software*.

Kapelner, A. and Bleich, J. (2015). Prediction with missing data via bayesian additive regression trees. *Canadian Journal of Statistics*, 43(2):224–239.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154.

Li, Y., Liu, S., Yang, J., and Yang, M.-H. (2017). Generative face completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3911–3919.

Little, R. J. (1992). Regression with missing x's: a review. *Journal of the American Statistical Association*, 87(420):1227–1237.

Little, R. J. and Rubin, D. B. (1987, 2002). *Statistical analysis with missing data*. John Wiley & Sons.

Liu, Y., Wang, Y., Feng, Y., and Wall, M. M. (2016). Variable selection and prediction with incomplete high-dimensional data. *Ann. Appl. Stat.*, 10(1):418–450.

Meng, X.-L. (1994). Multiple-imputation inferences with uncongenial sources of input. *Statistical Science*, pages 538–558.

Mohan, K. and Pearl, J. (2018). Graphical models for processing missing data. *arXiv preprint arXiv:1801.03583*.

Murray, J. S. (2018). Multiple imputation: A review of practical and theoretical findings. *Statist. Sci.*, 33(2):142–159.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rosasco, L., Vito, E. D., Caponnetto, A., Piana, M., and Verri, A. (2004). Are loss functions all the same? *Neural Computation*, 16(5):1063–1076.

Rosenbaum, P. R. and Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American statistical Association*, 79(387):516–524.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. Wiley.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5(2):197–227.

Seaman, S., Galati, J., Jackson, D., and Carlin, J. (2013). What is meant by" missing at random"? *Statistical Science*, pages 257–268.

Stekhoven, D. J. and Bühlmann, P. (2011). Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118.

Sterne, J. A. C., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., Wood, A. M., and Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *BMJ*, 338.

Strobl, C., Boulesteix, A.-L., and Augustin, T. (2007). Unbiased split selection for classification trees based on the gini index. *Computational Statistics and Data Analysis*, 52(1):483–501.

Sugiyama, M., Lawrence, N. D., Schwaighofer, A., et al. (2017). *Dataset shift in machine learning*. The MIT Press.

Therneau, T. and Atkinson, B. (2018). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-13.

Therneau, T. M., Atkinson, E. J., et al. (1997). An introduction to recursive partitioning using the rpart routines.

Twala, B. E. T. H., Jones, M. C., and Hand, D. J. (2008). Good methods for coping with missing data in decision trees. *Pattern Recogn. Lett.*, 29(7):950–956.

van Buuren, S. (2018). *Flexible Imputation of Missing Data*. Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press LLC.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

Wood, A. M., White, I. R., and Royston, P. (2008). How should variable selection be performed with multiply imputed data? *Statistics in Medicine*, 27(17):3227–3246.

Wright, M. N. and Ziegler, A. (2015). ranger: A fast implementation of random forests for high dimensional data in c++ and r. *arXiv preprint arXiv:1508.04409*.

Yoon, J., Jordon, J., and van der Schaar, M. (2018). Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*.

A Proofs

A.1 Proof of Theorem 2

Proof of Theorem 2: consistency of test-time conditional multiple imputation. Let $\widetilde{\mathbf{x}} \in (\mathbb{R} \cup \texttt{NA})^d$. Without loss of generality, assume that only $\tilde{x}_1, \ldots, \tilde{x}_j$ are equal to NA, for some $j \in \{1, \ldots, d\}$. Let $g_{\tilde{x}_{j+1}, \ldots, \tilde{x}_d}$ be the distribution of $(X_1, \ldots, X_j)$ conditional on the event $\{\widetilde{X}_{j+1} = \tilde{x}_{j+1}, \ldots, \widetilde{X}_d = \tilde{x}_d\}$. Let $(\hat{X}_1, \ldots, \hat{X}_j)$ be a random vector with distribution $g_{\tilde{x}_{j+1}, \ldots, \tilde{x}_d}$. By definition, the multiple imputation prediction described in Theorem 2 is given by

$$\begin{aligned} f^{\star}_{mult\ imput}(\widetilde{\mathbf{x}}) &= \mathbb{E}_{\hat{X}_1, \ldots, \hat{X}_j}[f(\hat{X}_1, \ldots, \hat{X}_j, \tilde{x}_{j+1}, \ldots, \tilde{x}_d)] \\ &= \mathbb{E}[f(\hat{X}_1, \ldots, \hat{X}_j, \widetilde{X}_{j+1}, \ldots, \widetilde{X}_d) | \widetilde{X}_{j+1} = \tilde{x}_{j+1}, \ldots, \widetilde{X}_d = \tilde{x}_d] \\ &= \mathbb{E}[Y | \widetilde{X}_{j+1} = \tilde{x}_{j+1}, \ldots, \widetilde{X}_d = \tilde{x}_d], \end{aligned} \tag{14}$$

since $Y = f(\mathbf{X}) + \varepsilon$, where $\varepsilon \perp\!\!\!\perp (\mathbf{X}, \mathbf{M})$. On the other hand, note that, since the missing pattern is MAR,

$$\begin{aligned} \mathbb{E}[Y | \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}] &= \mathbb{E}[Y | \widetilde{X}_1 = \texttt{NA}, \ldots, \widetilde{X}_j = \texttt{NA}, \widetilde{X}_{j+1} = \tilde{x}_{j+1}, \ldots, \widetilde{X}_d = \tilde{x}_d] \\ &= \mathbb{E}[Y | M_1 = 1, \ldots, M_j = 1, \tilde{X}_{j+1} = \tilde{x}_{j+1}, \ldots, \tilde{X}_d = \tilde{x}_d] \\ &= \mathbb{E}[Y | \tilde{X}_{j+1} = \tilde{x}_{j+1}, \ldots, \tilde{X}_d = \tilde{x}_d]. \end{aligned} \tag{15}$$

Combining (14) and (15), we finally obtain

$$f^{\star}_{mult\ imput}(\widetilde{\mathbf{x}}) = \mathbb{E}[Y | \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}].$$

□

A.2 Proof of Theorem 3

Proof of Theorem 3: consistency of mean imputation at train and test time. Let $\mathbf{x} \in [0,1]^d$ such that $x_1 \neq \mathbb{E}[X_1]$. Thus, for $0 < h < |x_1 - \mathbb{E}[X_1]|$, letting $B(\mathbf{x}, h)$ be the euclidean ball centered at $\mathbf{x}$ of radius h,

$$\begin{aligned} \mathbb{E}[Y | \mathbf{X}' \in B(\mathbf{x}, h)] &= \frac{\mathbb{E}[Y \mathbb{1}_{\mathbf{X}' \in B(\mathbf{x}, h)}]}{\mathbb{P}[\mathbf{X}' \in B(\mathbf{x}, h)]} \\ &= \frac{\mathbb{E}[Y \mathbb{1}_{\mathbf{X} \in B(\mathbf{x}, h)} \mathbb{1}_{M_1 = 0}]}{\mathbb{P}[\mathbf{X} \in B(\mathbf{x}, h), M_1 = 0]} \\ &= \mathbb{E}[Y | \mathbf{X} \in B(\mathbf{x}, h), M_1 = 0]\,. \end{aligned} \tag{16}$$

Taking the limit of (16) when h tends to zero,

$$\begin{aligned}\mathbb{E}[Y|\mathbf{X}'=\mathbf{x}] &= \lim_{h\to 0}\mathbb{E}[Y|\mathbf{X}'\in B(\mathbf{x},h)]\\ &= \mathbb{E}[Y|\mathbf{X}=\mathbf{x}, M_1=0]. \end{aligned} \tag{17}$$

Now, let $\mathbf{x}\in[0,1]^d$ such that $x_1=\mathbb{E}[X_1]$. If $\mathbb{P}[M_1=1|X_2=x_2,\ldots,X_d=x_d]=0$, then $\{\mathbf{X}'=\mathbf{x}\}=\{\mathbf{X}'=\mathbf{x},M_1=0\}=\{\mathbf{X}=\mathbf{x}\}$, and consequently,

$$\mathbb{E}[Y|\mathbf{X}'=\mathbf{x}]=\mathbb{E}[Y|\mathbf{X}=\mathbf{x}]. \tag{18}$$

Now, if $\mathbb{P}[M_1=1|X_2=x_2,\ldots,X_d=x_d]=\eta>0$, we have

$$\begin{aligned}\mathbb{P}[\mathbf{X}'\in B(\mathbf{x},h)] &= \mathbb{E}[\mathbb{1}_{\mathbf{X}'\in B(\mathbf{x},h)}\mathbb{1}_{M_1=0}]+\mathbb{E}[\mathbb{1}_{\mathbf{X}'\in B(\mathbf{x},h)}\mathbb{1}_{M_1=1}]\\ &= \mathbb{E}[\mathbb{1}_{\mathbf{X}\in B(\mathbf{x},h)}\mathbb{1}_{M_1=0}]+\mathbb{E}[\mathbb{1}_{(X_2,\ldots,X_d)\in B((x_2,\ldots,x_d),h)}\mathbb{1}_{M_1=1}],\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}[f(\mathbf{X})\mathbb{1}_{\mathbf{X}'\in B(\mathbf{x},h)}] = E[f(\mathbf{X})\mathbb{1}_{\mathbf{X}\in B(\mathbf{x},h)}\mathbb{1}_{M_1=0}]\\ + E[f(\mathbf{X})\mathbb{1}_{(X_2,\ldots,X_d)\in B((x_2,\ldots,x_d),h)}\mathbb{1}_{M_1=1}].\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}[Y|\mathbf{X}'\in B(\mathbf{x},h)] =& \frac{\mathbb{E}[f(\mathbf{X})\mathbb{1}_{\mathbf{X}'\in B(\mathbf{x},h)}]}{\mathbb{P}[\mathbf{X}'\in B(\mathbf{x},h)]}\\ =& \frac{\mathbb{E}[f(\mathbf{X})\mathbb{1}_{\mathbf{X}\in B(\mathbf{x},h)}\mathbb{1}_{M_1=0}]+\mathbb{E}[f(\mathbf{X})\mathbb{1}_{(X_2,\ldots,X_d)\in B((x_2,\ldots,x_d),h)}\mathbb{1}_{M_1=1}]}{\mathbb{E}[\mathbb{1}_{\mathbf{X}\in B(\mathbf{x},h)}\mathbb{1}_{M_1=0}]+\mathbb{E}[\mathbb{1}_{(X_2,\ldots,X_d)\in B((x_2,\ldots,x_d),h)}\mathbb{1}_{M_1=1}]}.\end{aligned} \tag{19}$$

The terms in (19) involving $M_1=0$ satisfy

$$\mathbb{E}[\mathbb{1}_{\mathbf{X}\in B(\mathbf{x},h)}\mathbb{1}_{M_1=0}]\le\mu(B(\mathbf{x},h))\le\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}\|g\|_\infty h^d, \tag{20}$$

and

$$\begin{aligned}|\mathbb{E}[f(\mathbf{X})\mathbb{1}_{\mathbf{X}\in B(\mathbf{x},h)}\mathbb{1}_{M_1=0}]| &\le E[|f(\mathbf{X})|\mathbb{1}_{\mathbf{X}\in B(\mathbf{x},h)}]|\\ &\le \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}\|g\|_\infty\|f\|_\infty h^d.\end{aligned} \tag{21}$$

The second term of the denominator in (19) can be bounded from below,

$$\begin{aligned}
&\mathbb{E}[\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{1}_{M_1=1}] \\
&= \mathbb{E}[\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{P}[M_1=1|X_2,\dots,X_d]] \\
&\geq \frac{\pi^{(d-1)/2}}{\Gamma(\frac{d-1}{2}+1)}\big(\inf_{[0,1]^d} g\big)h^{d-1}\eta. \qquad (22)
\end{aligned}$$

The second term of the numerator in (19) verifies

$$\begin{aligned}
&\mathbb{E}[f(\mathbf{X})\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{1}_{M_1=1}] \\
&= \mathbb{E}[\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{E}[f(\mathbf{X})\mathbb{1}_{M_1=1}|X_2,\dots,X_d]] \\
&= \mathbb{E}[\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{E}[f(\mathbf{X})|X_2,\dots,X_d]\mathbb{E}[\mathbb{1}_{M_1=1}|X_2,\dots,X_d]].
\end{aligned}$$

If $\mathbb{E}[f(\mathbf{X})|X_2=x_2,\dots,X_d=x_d]>0$, by uniform continuity of f and g,

$$\begin{aligned}
&\mathbb{E}[\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{E}[f(\mathbf{X})|X_2,\dots,X_d]\mathbb{E}[\mathbb{1}_{M_1=1}|X_2,\dots,X_d]] \\
&\geq \mathbb{E}[f(\mathbf{X})|X_2=x_2,\dots,X_d=x_d]\frac{\pi^{(d-1)/2}}{\Gamma(\frac{d-1}{2}+1)}\big(\inf_{[0,1]^d} g\big)h^{d-1}\eta.
\end{aligned}$$

Similarly, if $\mathbb{E}[f(\mathbf{X})|X_2=x_2,\dots,X_d=x_d]<0$, we have

$$\begin{aligned}
&\mathbb{E}[\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{E}[f(\mathbf{X})|X_2,\dots,X_d]\mathbb{E}[\mathbb{1}_{M_1=1}|X_2,\dots,X_d]] \\
&\leq \mathbb{E}[f(\mathbf{X})|X_2=x_2,\dots,X_d=x_d]\frac{\pi^{(d-1)/2}}{\Gamma(\frac{d-1}{2}+1)}\big(\inf_{[0,1]^d} g\big)h^{d-1}\eta \\
&\leq 0.
\end{aligned}$$

Hence, if $\mathbb{E}[f(\mathbf{X})|X_2=x_2,\dots,X_d=x_d]\neq 0$

$$\begin{aligned}
&|\mathbb{E}[f(\mathbf{X})\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{1}_{M_1=1}]| \\
&\geq |\mathbb{E}[f(\mathbf{X})|X_2=x_2,\dots,X_d=x_d]|\frac{\pi^{(d-1)/2}}{\Gamma(\frac{d-1}{2}+1)}\big(\inf_{[0,1]^d} g\big)h^{d-1}\eta. \qquad (23)
\end{aligned}$$

Gathering inequalities (20)-(23) and using equation (19), we have, if $\mathbb{E}[f(\mathbf{X})|X_2=x_2,\dots,X_d=x_d]\neq 0$

$$\begin{aligned}
\lim_{h\to 0}\mathbb{E}[Y|\mathbf{X}'\in B(\mathbf{x},h)] &= \lim_{h\to 0}\frac{E[f(\mathbf{X})\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{1}_{M_1=1}]}{\mathbb{E}[\mathbb{1}_{(X_2,\dots,X_d)\in B((x_2,\dots,x_d),h)}\mathbb{1}_{M_1=1}]} \\
&= \mathbb{E}[f(\mathbf{X})|X_2=x_2,\dots,X_d=x_d,M_1=1].
\end{aligned}$$

Finally, if $\mathbb{E}[f(\mathbf{X})|X_2 = x_2, \ldots, X_d = x_d] = 0$ then by uniform continuity of f, there exists ε_h such that $\varepsilon_h \to 0$ as $h \to 0$ satisfying,

$$|\mathbb{E}[f(\mathbf{X})\mathbb{1}_{(X_2,\ldots,X_d)\in B((x_2,\ldots,x_d),h)}\mathbb{1}_{M_1=1}]| \leq \varepsilon_h h^{d-1}\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}\|g\|_\infty,$$

hence

$$\begin{aligned}\lim_{h\to 0}\mathbb{E}[Y|\mathbf{X}' \in B(\mathbf{x},h)] &= 0 \\ &= \mathbb{E}[f(\mathbf{X})|X_2 = x_2, \ldots, X_d = x_d] \\ &= \mathbb{E}[f(\mathbf{X})|X_2 = x_2, \ldots, X_d = x_d, M_1 = 1],\end{aligned}$$

since $M_1 \perp\!\!\!\perp X_1|(X_2, \ldots, X_d)$. Consequently, for all $\mathbf{x} \in [0,1]^d$ such that $x_1 = \mathbb{E}[X_1]$,

$$\lim_{h\to 0}\mathbb{E}[Y|\mathbf{X}' \in B(\mathbf{x},h)] = \mathbb{E}[f(\mathbf{X})|X_2 = x_2, \ldots, X_d = x_d, M_1 = 1]. \quad (24)$$

Combining equations (17), (18) and (24), the prediction given by the mean imputation followed by learning is, for all $\mathbf{x}' \in \mathbb{R}^d$,

$$\begin{aligned}f^\star_{impute}(\mathbf{x}') = &\mathbb{E}[Y|X_2 = x_2, \ldots, X_d = x_d, M_1 = 1]\mathbb{1}_{x'_1=\mathbb{E}[X_1]}\mathbb{1}_{\mathbb{P}[M_1=1|X_2=x_2,\ldots,X_d=x_d]>0} \\ &+ \mathbb{E}[Y|\mathbf{X} = \mathbf{x}']\mathbb{1}_{x'_1=\mathbb{E}[X_1]}\mathbb{1}_{\mathbb{P}[M_1=1|X_2=x_2,\ldots,X_d=x_d]=0} \\ &+ \mathbb{E}[Y|X_2 = x_2, \ldots, X_d = x_d, M_1 = 0]\mathbb{1}_{x'_1\neq\mathbb{E}[X_1]},\end{aligned}$$

which concludes the proof. □

A.3 Proof of Proposition 1

Cart splitting criterion. Under the model given in Proposition 1, simple calculations show that

$$\begin{aligned}\mathbb{E}[Y|X \in [0,s]] = \frac{s}{2}, \quad \mathbb{E}[Y^2|X \in [0,s]] = \frac{s^2}{3} \\ \mathbb{E}[Y|X \in [s,1]] = \frac{1+s}{2}, \quad \mathbb{E}[Y^2|X \in [s,1]] = \frac{1-s^3}{3(1-s)} \\ \mathbb{P}[X \in [0,s]] = s, \quad \mathbb{P}[X \in [s,1]] = 1-s.\end{aligned}$$

Thus the CART spltting criterion can be written as

$$\begin{aligned}C(1,s) = &\ \mathbb{E}[Y^2] - (\mathbb{P}[X \in [0,s]](\mathbb{E}[Y|X \in [0,s]])^2 + \mathbb{P}[X \in [s,1]](\mathbb{E}[Y|X \in [s,1]])^2) \\ = &\ \frac{1}{3} - \left(s\left(\frac{s}{2}\right)^2 + (1-s)\left(\frac{1+s}{2}\right)^2\right) \\ = &\ \frac{s(s-1)}{4} + \frac{1}{12}.\end{aligned}$$

By definition,

$$s^\star = \underset{s\in[0,1]}{\text{argmin}} \left(\frac{1}{4}s(s-1) + \frac{1}{12}\right) = 1/2,$$

and the criterion evaluated in $s = 1/2$ is equal to $1/48$. The calculations are exactly the same when a percentage of missing value is added if $M_1 \perp\!\!\!\perp X_1$.

MIA splitting criterion. By symmetry, we can assume than missing values are sent left. It is equivalent to observing

$$X' = 0\mathbb{1}_{M=1} + X\mathbb{1}_{M=0}.$$

The MIA splitting criterion is then defined as

$$\begin{aligned}
s^\star_{\text{MIA,L}} &= \underset{s\in[0,1]}{\text{argmin}}\ \mathbb{E}\left[\left(Y - \mathbb{E}[Y|X' \le s]\mathbb{1}_{X'\le s} - \mathbb{E}[Y|X' > s]\mathbb{1}_{X'>s}\right)^2\right] \\
&= \underset{s\in[0,1]}{\text{argmin}}\ \mathbb{P}(X' \le s)\mathbb{E}\left[\left(Y - \mathbb{E}[Y|X' \le s]\right)^2 \middle| X' \le s\right] \\
&\quad + \mathbb{P}(X' > s)\mathbb{E}\left[\left(Y - \mathbb{E}[Y|X' > s]\right)^2 \middle| X' > s\right].
\end{aligned}$$

We have

$$\begin{aligned}
\mathbb{E}[Y|X' \in [0,s]] &= \mathbb{E}[X|X' \in [0,s]] \\
&= \mathbb{E}[X\mathbb{1}_{M=1} + X\mathbb{1}_{M=0}|X' \in [0,s]] \\
&= \frac{1}{\mathbb{P}[X' \in [0,s]]}\mathbb{E}[X\mathbb{1}_{M=1,X'\in[0,s]} + X\mathbb{1}_{M=0,X'\in[0,s]}] \\
&= \frac{1}{p + (1-p)s}\Big(\frac{p}{2} + \frac{(1-p)s^2}{2}\Big).
\end{aligned}$$

Besides,

$$\begin{aligned}
\mathbb{E}[Y^2|X' \in [0,s]] &= \mathbb{E}[X^2|X' \in [0,s]] \\
&= \mathbb{E}[X^2\mathbb{1}_{M=1} + X^2\mathbb{1}_{M=0}|X' \in [0,s]] \\
&= \frac{1}{p + (1-p)s}\mathbb{E}[X^2\mathbb{1}_{M=1,X'\in[0,s]} + X^2\mathbb{1}_{M=0,X'\in[0,s]}] \\
&= \frac{1}{p + (1-p)s}\Big(\frac{p}{3} + \frac{(1-p)s^3}{3}\Big)
\end{aligned}$$

Thus the left-part of the criterion is given by

$$\begin{aligned}
&\mathbb{P}(X' \in [0,s])\mathbb{E}[(Y - \mathbb{E}[Y|X' \in [0,s]])^2 | X' \in [0,s]] \\
&= \Big(p + (1-p)s\Big)\Big(\mathbb{E}[Y^2|X' \in [0,s]] - (\mathbb{E}[Y|X' \in [0,s]])^2\Big) \\
&= \Big(p + (1-p)s\Big)\Big(\frac{1}{p + (1-p)s}\Big(\frac{p}{3} + \frac{(1-p)s^3}{3}\Big) \\
&\quad - \Big(\frac{1}{p + (1-p)s}\Big(\frac{p}{2} + \frac{(1-p)s^2}{2}\Big)\Big)^2\Big) \\
&= \Big(\frac{p}{3} + \frac{(1-p)s^3}{3}\Big) - \frac{1}{p + (1-p)s}\Big(\frac{p}{2} + \frac{(1-p)s^2}{2}\Big)^2
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
\mathbb{E}[Y|X' \in [s,1]] &= \mathbb{E}[X|X' \in [s,1]] \\
&= \mathbb{E}[X\mathbb{1}_{M=1} + X\mathbb{1}_{M=0}|X' \in [s,1]] \\
&= \frac{1}{(1-p)(1-s)}\mathbb{E}[X\mathbb{1}_{M=1, X' \in [s,1]} + X\mathbb{1}_{M=0, X' \in [s,1]}] \\
&= \frac{1}{(1-p)(1-s)}\Big((1-p)\frac{1-s^2}{2}\Big) \\
&= \frac{1+s}{2}.
\end{aligned}$$

Besides,

$$\begin{aligned}
\mathbb{E}[Y^2|X' \in [s,1]] &= \mathbb{E}[X^2|X' \in [s,1]] \\
&= \mathbb{E}[X^2\mathbb{1}_{M=1} + X^2\mathbb{1}_{M=0}|X' \in [s,1]] \\
&= \frac{1}{(1-p)(1-s)}\mathbb{E}[X^2\mathbb{1}_{M=1, X' \in [s,1]} + X^2\mathbb{1}_{M=0, X' \in [s,1]}] \\
&= \frac{1}{(1-p)(1-s)}\Big((1-p)\frac{1-s^3}{3}\Big) \\
&= \frac{1-s^3}{3(1-s)}.
\end{aligned}$$

Thus the right-part of the criterion is given by

$$\begin{aligned}
&\mathbb{P}(X' \in [s,1])\mathbb{E}[(Y - \mathbb{E}[Y|X' \in [s,1]])^2 | X' \in [s,1]] \\
&= \Big((1-p)(1-s)\Big)\Big(\mathbb{E}[Y^2|X' \in [s,1]] - (\mathbb{E}[Y|X' \in [s,1]])^2\Big) \\
&= \Big((1-p)(1-s)\Big)\Big(\frac{1-s^3}{3(1-s)} - (\frac{1+s}{2})^2\Big) \\
&= (1-p)\frac{1-s^3}{3} - (1-p)(1-s)\Big(\frac{1+s}{2}\Big)^2.
\end{aligned}$$

Finally,

$$\begin{aligned}
s^\star_{\text{MIA,L}} = \underset{s\in[0,1]}{\text{argmin}} \Bigg\{ &\Big(\frac{p}{3} + \frac{(1-p)s^3}{3}\Big) - \frac{1}{p+(1-p)s}\Big(\frac{p}{2} + \frac{(1-p)s^2}{2}\Big)^2 \\
&+ (1-p)\frac{1-s^3}{3} - (1-p)(1-s)\Big(\frac{1+s}{2}\Big)^2\Bigg\},
\end{aligned}$$

which concludes the proof.

A.4 Proof of proposition 2

Probabilistic and block propagation. First, note that the variable $X_2 = X_1 \mathbb{1}_{W=1}$ is similar to the variable studied for the computation of the MIA criterion in Proposition 1. Therefore, the value of the CART splitting criterion along the first variable is $C_{\text{MIA}}(1, 1/2, \text{L}, 0)$ and its value along the second variable is $C_{\text{MIA}}(2, s^\star_{\text{MIA,L}}, \text{L}, \eta)$. Since the function

$$\alpha \mapsto C_{\text{MIA}}(\cdot, s^\star_{\text{MIA,L}}, \text{L}, \alpha)$$

is increasing, splitting along the first variable leads to the largest variance reduction. Thus, for probabilistic and block propagation, splits occur along the first variable. Let us now compare the value of these criteria. We have

$$\mathbb{P}[X_1 \leq 1/2] = \mathbb{P}[X_1 \geq 1/2] = 1/2.$$

The quantities related to the left cell are given by

$$\mathbb{E}[Y|X_1 \leq 1/2] = \frac{p+1}{4} \quad \text{and} \quad \mathbb{E}[Y^2|X_1 \leq 1/2] = \frac{p}{4} + \frac{1}{12}.$$

The quantities related to the left cell are given by

$$\mathbb{E}[Y|X_1 \geq 1/2] = \frac{3-p}{4} \quad \text{and} \quad \mathbb{E}[Y^2|X_1 \geq 1/2] = \frac{7}{12} - \frac{p}{4}.$$

Thus, the value of the criterion satisfies

$$\begin{aligned} R(f^\star_{\text{prob}}) &= \frac{1}{2}\Big(\frac{p}{4} + \frac{1}{12} - \left(\frac{p+1}{4}\right)^2\Big) \\ &\quad + \frac{1}{2}\Big(\frac{7}{12} - \frac{p}{4} - \left(\frac{3-p}{4}\right)^2\Big) \\ &= -\frac{p^2}{16} + \frac{p}{8} + \frac{1}{48}. \end{aligned}$$

Let, for all $p \in [0,1]$,

$$\begin{aligned} h(p) &= R(f^\star_{\text{prob}}) - R(f^\star_{\text{block}}) \\ &= -\frac{p^2}{16} + \frac{p}{8} + \frac{1}{48} - \left(-\frac{11}{48} + \frac{1}{8}\frac{3p+2}{2p+1}\right) \\ &= -\frac{p^2}{16} + \frac{p}{8} + \frac{1}{16} - \frac{1}{16}\frac{1}{2p+1}. \end{aligned}$$

We have,

$$h'(p) = -\frac{p}{8} + \frac{1}{8} + \frac{1}{8(2p+1)^2},$$

and consequently,

$$h''(p) = -\frac{1}{8} - \frac{1}{2(2p+1)^3}.$$

An inspection of the variation of h reveals that $h(p) \geq 0$ for all $p \in [0,1]$, which concludes the first part of the proof.

MIA. As noticed above, the criterion computed along the second variable is given by

$$C_{\text{MIA}}(2, s^\star_{\text{MIA,L}}, \text{L}, \eta)$$

Since the function

$$\alpha \mapsto C_{\text{MIA}}(\cdot, s^\star_{\text{MIA,L}}, \text{L}, \alpha)$$

is increasing, MIA split will occur along the first variable if $p \leq \eta$ and along the second variable if $p \geq \eta$. Therefore, the risk of the MIA splitting procedure is given by

$$R(f^\star_{\mathrm{MIA}}) = \min_{s\in[0,1]} C_{\mathrm{MIA}}(1, s, \mathrm{L}, p)\mathbb{1}_{p\leq\eta} + \min_{s\in[0,1]} C_{\mathrm{MIA}}(1, s, \mathrm{L}, \eta)\mathbb{1}_{p>\eta}.$$

Surrogate split. Consider the model $Y = X_1$ and $X_2 = X_1\mathbb{1}_{W=1}$, where $\mathbb{P}[W = 0] = \eta$. Let us determine the best split along X_2 to predict $Z = \mathbb{1}_{X_1<0.5}$. Since $\{X_2 \leq s\} = \{X_1 \leq s, W = 1\} \cup \{W = 0\}$, and $\{X_2 > s\} = \{X_1 > s, W = 1\}$,

$$\mathbb{P}[X_2 \leq s] = s(1-p) + p \quad \text{and} \quad \mathbb{P}[X_2 > s] = (1-s)(1-p).$$

Consequently,

$$\begin{aligned}
\mathbb{E}[Z|X_2 \leq s] &= \frac{\mathbb{E}[\mathbb{1}_{X_1\leq 0.5, X_2\leq s}}{\mathbb{P}[X_2 \leq s]} \\
&= \frac{1}{s(1-p)+p}\mathbb{E}[\mathbb{1}_{X_1\leq 0.5, X_1\leq s, W=1} + \mathbb{1}_{X_1\leq 0.5, W=0}] \\
&= \frac{1}{s(1-p)+p}\big[(1-p)\min(0.5, s) + \frac{p}{2}\big].
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[Z|X_2 \geq s] &= \frac{\mathbb{E}[\mathbb{1}_{X_1\leq 0.5, X_2> s}}{\mathbb{P}[X_2 > s]} \\
&= \frac{1}{(1-s)(1-p)}\mathbb{P}[X_1 \leq 0.5, W = 1, X_1 \geq s] \\
&= \frac{(0.5-s)_+}{1-s}.
\end{aligned}$$

Besides, note that $\mathbb{E}[Z^2] = \mathbb{P}[X_1 \leq 0.5] = 0.5$. Therefore, the splitting criterion to predict $\mathbb{1}_{X_1\leq 0.5}$ with X_2 is given by

$$\begin{aligned}
f(s) &= \frac{1}{2} - \mathbb{P}[X_2 \leq s](\mathbb{E}[Z|X_2 \leq s])^2 - \mathbb{P}[X_2 > s](\mathbb{E}[Z|X_2 > s])^2 \\
&= \frac{1}{2} - \frac{1}{s(1-p)+p}\Big((1-p)\min(0.5, s) + \frac{p}{2}\Big)^2 - \frac{1-p}{1-s}((0.5-s)_+)^2.
\end{aligned}$$

For $s \geq 1/2$,

$$h(s) = \frac{1}{2} - \frac{1}{4(s(1-p)+p)},$$

which is minimal for $s = 1/2$. For $s \leq 1/2$,

$$h(s) = \frac{1}{2} - \frac{1}{4}\left(\frac{p^2}{p + s(1-p)} + \frac{1-p}{1-s}\right).$$

Hence,

$$h'(s) = -\frac{1-p}{4}\frac{(1-2p)s^2 + 2ps}{(1-s)^2(s(1-p)+p)^2}.$$

Let $g(s) = (1-2p)s^2 + 2ps$. If $p \leq 1/2$, the solutions of $g(s) = 0$ are negative, thus, $g(s) \geq 0$ for all $s \in [0, 1/2]$ and thus the minimum of h is reached at $s = 1/2$. If $p \geq 1/2$, one solution of $g(s) = 0$ is zero and the other is $s = 2p/(2p-1) > 1$. Thus, $g(s) \geq 0$ for all $s \in [0, 1/2]$ and the minimum of h is reached at $s = 1/2$. Finally, the minimum of h is reached at $s = 1/2$. The risk of the surrogate estimate is then given by

$$\begin{aligned} R(f^\star_{\text{surr}}) &= \mathbb{E}[(Y - f^\star_{\text{surr}}(\mathbf{X}))^2] \\ &= \mathbb{E}[(Y - f^\star_{\text{surr}}(\mathbf{X}))^2 \mathbb{1}_{M_1=1} + (Y - f^\star_{\text{surr}}(\mathbf{X}))^2 \mathbb{1}_{M_1=1}]. \end{aligned}$$

Here,

$$\begin{aligned} & \mathbb{E}[(Y - f^\star_{\text{surr}}(\mathbf{X}))^2 | M_1 = 1] \\ = \; & \mathbb{E}[(X_1 - 0.25)^2 \mathbb{1}_{X_2<0.5} + (X_1 - 0.75)^2 \mathbb{1}_{X_2 \geq 0.5}] \\ = \; & \eta \mathbb{E}[(X_1 - 0.25)^2] + (1-\eta)\mathbb{E}[(X_1 - 0.25)^2 \mathbb{1}_{X_1 \leq 0.5}] \\ & \quad + (1-\eta)\mathbb{E}[(X_1 - 0.75)^2 \mathbb{1}_{X_1 > 0.5}] \\ = \; & \frac{1}{48} + \frac{6\eta}{48}. \end{aligned}$$

Finally,

$$R(f^\star_{\text{surr}}) = \frac{1-p}{48} + p\Big(\frac{1}{48} + \frac{6\eta}{48}\Big) = \frac{1}{48} + \frac{6}{48}\eta p.$$

B Miscellaneous

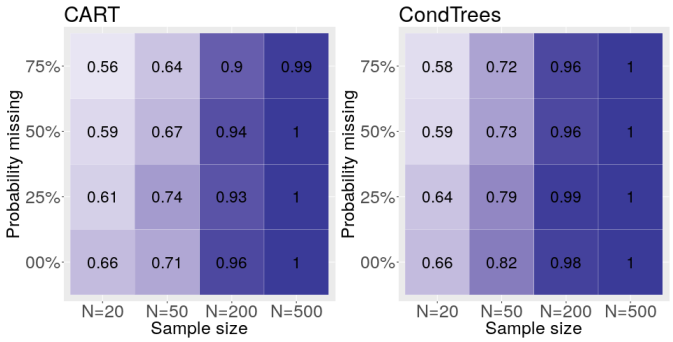
B.1 Variable selection properties of the tree methods with missing values

Decision trees based on the CART criterion (implemented in the R library rpart) and on conditional trees (implemented in the the R library partykit)

lead to different ways of selecting splitting variables. We illustrate this behaviour on the simple following model:

$$\begin{cases} X_1 \perp\!\!\!\perp X_2 & \sim \mathcal{N}(0,1) \\ \varepsilon & \sim \mathcal{N}(0,1) \\ Y & = 0.25X_1 + \varepsilon. \end{cases}$$

We insert MCAR values, either on the first variable or on both variables. Stumps (decision trees of depth one) are fit on 500 Monte-Carlo repetitions. We vary the sample size and the percentage of missing values. Figure 6 show that CART and conditional trees give similar results when there are missing values on both variables. However, Figure 7 shows that CART has a tendency to underselect X_1 when there are missing values only on X_1. For instance, for a sample of size 50 with 75% missing values, CART selects the non-informative variable X_2 more frequently than X_1, while conditional trees keep selecting X_1 more often.

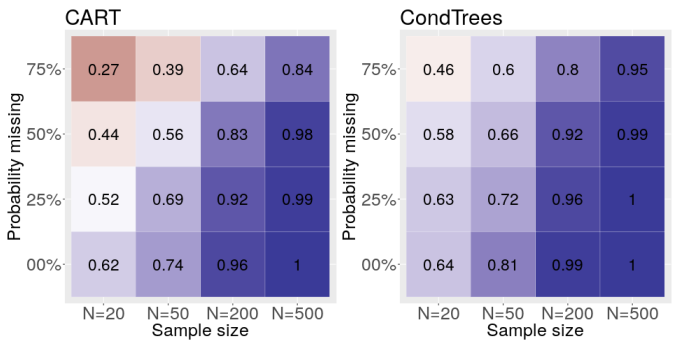


(a) CART (b) Conditional trees

Figure 6: Frequency of selection of X_1 when there are missing values on X_1 and X_2

B.2 Example of EM algorithm

Let us consider a simple case of n observations $(\mathbf{x}_1, \mathbf{x}_2) = (x_{i1}, x_{i2})_{1 \leq i \leq n}$ sampled from the distribution of (X_1, X_2), a bivariate Gaussian distribution with parameters (μ, Σ). We assume that X_2 is subjected to missing values and that only r values are observed. The aim is to get the maximum likelihood estimates of (μ, Σ) from the incomplete data set. The algorithm described below can be straightforwardly extended to the multivariate case.



(a) CART (b) Conditional trees

Figure 7: Frequency of selection of X_1 when there are missing values on X_1 only

Note that from $(\hat{\mu}, \hat{\Sigma})$, it is then possible to directly estimate the parameters of a linear regression model and thus to perform linear regression with missing values.
We denote by $f_{1,2}(\mathbf{x}_1, \mathbf{x}_2; \mu, \Sigma)$, $f_1(\mathbf{x}_1; \mu_1, \sigma_{11})$ and $f_{2|1}(\mathbf{x}_2|\mathbf{x}_1; \mu, \Sigma)$, respectively, the probability of joint distribution of (X_1, X_2), marginal distribution of X_1 and conditional distribution of $X_2|X_1$. The joint distribution of observed data can be decomposed as:

$$f_{1,2}(\mathbf{x}_1, \mathbf{x}_2; \mu, \Sigma) = \prod_{i=1}^{n} f_1(x_{i1}; \mu_1, \sigma_{11}) \prod_{j=1}^{r} f_{2|1}(x_{j2}|x_{j1}; \mu, \Sigma),$$

and the observed log-likelihood is written (up to an additional constant that does not appear in the maximization and that we therefore drop):

$$\ell(\mu, \Sigma; \mathbf{x}_1, \mathbf{x}_2) = -\frac{n}{2}\log(\sigma_{11}^2) - \frac{1}{2}\sum_{i=1}^{n}\frac{(x_{i1}-\mu_1)^2}{\sigma_{11}^2} - \frac{r}{2}\log\left((\sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}})^2\right)$$

$$-\frac{1}{2}\sum_{i=1}^{r}\frac{\left(x_{i2} - \mu_2 - \frac{\sigma_{12}}{\sigma_{11}}(x_{i1}-\mu_1)\right)^2}{(\sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}})^2}$$

We skip the computations and directly give the expression of the closed form maximum likelihood estimates of the mean:

$$\hat{\mu}_1 = n^{-1}\sum_{i=1}^{n} x_{i1}$$

$$\hat{\mu}_2 = \hat{\beta}_{20.1} + \hat{\beta}_{21.1}\hat{\mu}_1,$$

where

$$\hat{\beta}_{21.1} = s_{12}/s_{11}, \quad \hat{\beta}_{20.1} = \bar{x}_2 - \hat{\beta}_{21.1}\bar{x}_1,$$

$$\bar{x}_j = r^{-1}\sum_{i=1}^{r} x_{ij} \quad \text{and} \quad s_{jk} = r^{-1}\sum_{i=1}^{r}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k), \quad j,k = 1,2.$$

In this simple setting, we have an explicit expression of the maximum likelihood estimator despite missing values. However, this is not always the case but it is possible to use an EM algorithm to get the maximum likelihood estimators in the cases where data are missing.
The EM algorithm consists in maximizing the observed likelihood through successive maximization of the complete likelihood (if we had observed all n realizations of $\mathbf{x}_1$ and $\mathbf{x}_2$). Maximizing the complete likelihood

$$\ell_c(\mu, \Sigma; \mathbf{x}_1, \mathbf{x}_2) = -\frac{n}{2}\log\left(\det(\Sigma)\right) - \frac{1}{2}\sum_{i=1}^{n}(x_{i1} - \mu_1)^T\Sigma^{-1}(x_{i1} - \mu_1)$$

would be straightforward if we had all the observations. However elements of this likelihood are not available. Therefore, we replace them by the conditional expectation given observed data and the parameters of the current iteration. These two steps of computation of the conditional expectation (E-step) and maximization of the completed likelihood (M step) are repeated until convergence. The update formulas for the E and M steps are as follows:
E step: The sufficient statistics of the likelihood are:

$$s_1 = \sum_{i=1}^{n} x_{i1}, \quad s_2 = \sum_{i=1}^{n} x_{i2}, \quad s_{11} = \sum_{i=1}^{n} x_{i1}^2, \quad s_{22} = \sum_{i=1}^{n} x_{i2}^2, \quad s_{12} = \sum_{i=1}^{n} x_{i1}x_{i2}.$$

Since some values of x_2 are not available, we fill in the sufficient statistics with:

$$\mathbb{E}[x_{i2}|x_{i1}; \mu, \Sigma] = \beta_{20.1} + \beta_{21.1}x_{i1}$$

$$\mathbb{E}[x_{i2}^2|x_{i1}; \mu, \Sigma] = (\beta_{20.1} + \beta_{21.1}x_{i1})^2 + \sigma_{22.1}$$

$$\mathbb{E}[x_{i2}x_{i2}|x_{i1}; \mu, \Sigma] = (\beta_{20.1} + \beta_{21.1}x_{i1})x_{i1}.$$

with, $\beta_{21.1} = \sigma_{12}/\sigma_{11}$, $\beta_{20.1} = \mu_2 - \beta_{21.1}\mu_1$, and $\sigma_{22.1} = \sigma_{22} - \sigma_{12}^2/\sigma_{11}$.
M step: The M step consists in computing the maximum likelihood estimates as usual. Given s_1, s_2, s_{11}, s_{22}, and s_{12}, update $\hat{\mu}$ and $\hat{\sigma}$ with

$$\hat{\mu}_1 = s_1/n, \; \hat{\mu}_2 = s_2/n,$$

$$\hat{\sigma}_1 = s_{11}/n - \hat{\mu}_1^2,\ \hat{\sigma}_2 = s_{22}/n - \hat{\mu}_2^2,\ \hat{\sigma}_{12} = s_{12}/n - \hat{\mu}_1\hat{\mu}_2$$

Note that s_1, s_{11}, $\hat{\mu}_1$ and $\hat{\sigma}_1$ are constant across iterations since we do not have missing values on $\mathbf{x}_1$.

Remark 4. *Note that EM imputes the sufficient statistics and not the data.*