MDA for random forests: inconsistency, and a practical solution via the Sobol-MDA

Clément Bénard* Sébastien Da Veiga † Erwan Scornet‡

Abstract

Variable importance measures are the main tools to analyze the black-box mechanism of random forests. Although the Mean Decrease Accuracy (MDA) is widely accepted as the most efficient variable importance measure for random forests, little is known about its theoretical properties. In fact, the exact MDA definition varies across the main random forest software. In this article, our objective is to rigorously analyze the behavior of the main MDA implementations. Consequently, we mathematically formalize the various implemented MDA algorithms, and then establish their limits when the sample size increases. In particular, we break down these limits in three components: the first two are related to Sobol indices, which are well-defined measures of a variable contribution to the output variance, widely used in the sensitivity analysis field, as opposed to the third term, whose value increases with dependence within input variables. Thus, we theoretically demonstrate that the MDA does not target the right quantity when inputs are dependent, a fact that has already been noticed experimentally. To address this issue, we define a new importance measure for random forests, the Sobol-MDA, which fixes the flaws of the original MDA. We prove the consistency of the Sobol-MDA and show its good empirical performance through experiments on both simulated and real data. An open source implementation in `R` and `C++` is available online.

1 Introduction

Random forests (Breiman, 2001) are an ensemble learning algorithm, which aggregates a large number of trees to perform regression and classification tasks, and achieve state-of-the-art accuracy on a wide range of problems. In particular, random forests exhibit a good behavior on high-dimensional or noisy data, do not require tuning procedures, and are also well known for their robustness. All in all, random forests are widely used in practice thanks to these remarkable features. However, they suffer from a major drawback: a given prediction is generated through a large number of operations, typically ten thousands, which makes the interpretation of the prediction mechanism impossible. Because of this complexity, random forests are often qualified as black-boxes. More generally, the interpretability of learning

*Safran Tech, Sorbonne Université
†Safran Tech
‡Ecole Polytechnique

algorithms is receiving an increasingly high interest since this black-box characteristic is a strong practical limitation. For example, applications involving critical decisions, typically healthcare, require predictions to be justified. The most popular way to interpret random forests is variable importance analysis: input variables are ranked by decreasing order of their importance in the algorithm prediction process. Thus, specific variable importance measures were developed along with random forests (Breiman, 2001, 2003a). However, we will see that they may not target the right variable ranking when input variables are dependent, and could therefore be improved. First, we review the existing variable importance measures for random forests.

Variable importance. There are essentially two importance measures for random forests: the Mean Decrease Accuracy (MDA) (Breiman, 2001) and the Mean Decrease Impurity (MDI) (Breiman, 2003a). The MDA measures the decrease of accuracy when the values of a given input variable are permuted, thus breaking its relation to the output and to the other input variables. On the other hand, the MDI sums the weighted decreases of impurity over all nodes that split on a given variable, averaged over all trees in the forest. In both cases, a high value of the metric means that the variable is used in many important operations of the prediction mechanism of the forest. Unfortunately, there is no precise and rigorous interpretation since these two definitions are purely empirical. Furthermore, in the last decade, many empirical analysis have highlighted the flaws of the MDI—see Strobl et al. (2007) for example. Li et al. (2019) and Zhou and Hooker (2019) recently improved the MDI to partially remove its bias. However, Scornet (2020) demonstrated that the MDI is consistent only under a strong and restrictive assumption: the regression function is additive and the input variables are independent. Otherwise, the MDI is ill-defined. Overall, the MDA is widely considered as the most efficient variable importance measure for random forests (Strobl et al., 2007; Ishwaran, 2007; Genuer et al., 2010; Boulesteix et al., 2012), and we therefore focus on the MDA. Although it is extensively used in practice, little is known about its theoretical properties. To our knowledge, only Ishwaran (2007) and Zhu et al. (2015) provide theoretical analyses of modified versions of the MDA, but the asymptotic behavior of the original MDA algorithm (Breiman, 2001) is unknown: Ishwaran (2007) considers Breiman's forests but simplifies the MDA procedure, whereas Zhu et al. (2015) considers the original MDA but assumes the independence of the input variables and an exponential concentration inequality on the random forest estimate, the latter being proved only for purely random forests (which do not use the data to build the tree partitions). On the practical side, many empirical analyses provide evidence that when input variables are dependent, the MDA may fail to detect some relevant variables (Archer and Kimes, 2008; Strobl et al., 2008; Nicodemus and Malley, 2009; Genuer et al., 2010; Auret and Aldrich, 2011; Toloşi and Lengauer, 2011; Gregorutti et al., 2017; Hooker and Mentch, 2019). It is critical to assess that the properties of a variable importance measure are in line with the final objective of the conducted analysis. In the following paragraphs, we review the possible goals of variable importance, and then introduce sensitivity analysis to deepen the theoretical understanding of the MDA.

Variable importance objectives. The analysis of variable importance is not an end in itself, the goal is essentially to perform variable selection, with usually two final aims (Genuer et al., 2010): (i) find a small number of variables with a maximized accuracy, or (ii) detect

and rank all influential variables to focus on for further exploration with domain experts. Depending on which of these two objectives is of interest, different strategies should be used as the following example shows: if two influential variables are strongly correlated, one must be discarded in the first case, while the two must be kept in the second case. Indeed, if two variables convey the same statistical information, only one should be selected if the goal is to maximize the predictive accuracy with a small number of variables, i.e., objective (i). On the other hand, these two variables may be acquired differently and represent distinct physical quantities. Therefore, they may have different interpretations for domain experts, and both should be kept for objective (ii).

Sensitivity analysis. Sensitivity analysis is the study of uncertainties in a system. The main goal is to apportion the uncertainty of a system output to the uncertainty of the different inputs. Iooss and Lemaître (2015) and Ghanem et al. (2017) provide detailed reviews of global sensitivity analysis (GSA). In particular, GSA introduces well-defined importance measures of input contributions to the output variance: Sobol indices (Sobol, 1993; Saltelli, 2002; Mara et al., 2015) and Shapley effects (Shapley, 1953; Owen, 2014; Iooss and Prieur, 2017). These metrics are widely used to analyze computer code experiments, especially for the design of industrial systems. However, the literature about variable importance in the fields of statistical learning and machine learning rarely mentions sensitivity analysis. The reason of this hiatus is clear: until quite recently, GSA was focused on independent inputs, whereas the machine learning community essentially works with dependent inputs. In the last years, Gregorutti (2015) first established a link between GSA and the MDA: in the case of independent inputs the theoretical counterpart of the MDA is the unnormalized total Sobol index, i.e., twice the amount of explained variance lost when a given input variable is removed from the model, which is the expected quantity for both objectives (i) and (ii) in this independent setting. Additionally, Mara et al. (2015) extended Sobol indices to the case of dependence, named "full Sobol indices", while Owen (2014) reintroduced Shapley effects. Originally proposed in game theory (Shapley, 1953), Shapley effects exhibit very interesting properties as they equitably allocate the mutual contribution due to dependence and interactions to individual inputs. The main limitation of Shapley effects is the computational complexity which is exponential with the number of input variables. While full Sobol indices are confined to GSA, Shapley effects are now widely used by the machine learning community to interpret both tree ensembles and neural networks. In particular, SHAP values (Lundberg and Lee, 2017) adapt Shapley effects for local interpretation of model predictions, and Lundberg et al. (2018) provide a fast algorithm for tree ensembles. Finally, Covert et al. (2020) introduce SAGE, based on Shapley effects applied to any loss function, as a global importance measure for machine learning models. A detailed literature review of random forests and sensitivity analysis can be found in Antoniadis et al. (2020).

Article outline. In Section 2, we review and clarify the different MDA algorithms implemented in the main random forest software: several definitions coexist, and we first formalize them mathematically. Then, we conduct an asymptotic analysis to demonstrate that all MDA versions are indeed inappropriate for the two possible objectives of variable importance analysis. We first establish the limits of the empirical MDA algorithms—see the Supplementary Material for the proofs. Next, we analyze these limits and extend the result of Gregorutti

(2015) to the general dependent case: two additional terms in the theoretical counterpart of the MDA appear because of the permutation trick in the procedure. One is the full total Sobol index (Mara et al., 2015), but the other one is not directly related to a measure of importance. Thus, it is clear that the MDA is misleading for objectives (i) and (ii) when inputs are dependent, which is very often the case with real data. To our knowledge, this is the first asymptotic result on Breiman's MDA, which sheds light on the empirical limitations observed in practice. We also clarify the different MDA implementations, highlight that they have different meanings, and provide guidelines to the most appropriate one depending on the data distribution. Next, for objective (ii), it is widely accepted that Shapley effects are the most relevant importance measure as they equitably handle interactions and dependence. On the other hand, when one is using variable importance to select a small number of variables while maximizing predictive accuracy—objective (i), the total Sobol index is clearly the relevant measure to eliminate the less influential variables. However, no appropriate estimate of this quantity exists for random forests when inputs are dependent as demonstrated in Section 2. Therefore, we focus on objective (i) throughout the article. In Section 3, we propose the Sobol-MDA, an augmented version of the MDA which consistently estimates the total Sobol index even when input variables are dependent. We show the good empirical performance of the procedure on both simulated and real data, and prove the consistency of the Sobol-MDA. An implementation in `R` and `C++` of the Sobol-MDA is available at `https://gitlab.com/cbenard/sobol-mda`, and is based on `ranger` (Wright and Ziegler, 2017), a fast implementation of random forests. Thus, the Sobol-MDA enjoys good properties that make it a more efficient importance variable measure than the original MDA in a dependent setting.

2 MDA Theoretical Limitations

2.1 MDA Literature Review

The MDA was originally proposed by Breiman in his seminal article (Breiman, 2001), and works as follows. The values of a specific variable are permuted to break its relation to the output. Then, the predictive accuracy is computed for this perturbed dataset. The difference between this degraded accuracy and the original one gives the importance of the variable: a high decrease of accuracy means that the considered variable has a strong influence on the prediction mechanism. However, a review of the literature on random forests and their software implementations reveals that there is no consensus on the exact mathematical formulation of the MDA. We focus on the most popular random forest algorithms:

- the `R` package `randomForests` (Liaw and Wiener, 2002) based on the original `Fortran` code from Breiman and Cutler
- the fast `R/C++` implementation `ranger` (Wright and Ziegler, 2017)
- the most widely used `python` machine learning library `scikit-learn` (Pedregosa et al., 2011) (`RandomForestClassifier`/`RandomForestRegressor`)
- the `R` package `randomForestSRC` (Ishwaran and Kogalur, 2020) which implements survival forests in addition to the original algorithm.

Algorithm	Package	Error Estimate	Data
Train-Test MDA	`scikit-learn` `randomForestSRC`	Forest	Testing dataset
Breiman-Cutler MDA	`randomForest` (normalized) `ranger` / `randomForestSRC`	Tree	OOB sample
Ishwaran-Kogalur MDA	`randomForestSRC`	Forest	OOB sample

Table 1: Summary of the different MDA characteristics.

To give an order of magnitude, the typical number of users of each of these packages during the year 2020 is about half a million. A close inspection of their code exhibits that essentially three distinct definitions of the MDA are widely used—see the Supplementary Material for references and details about the MDA implementation in the package codes. The differences between the three MDA versions are twofold: the MDA can be computed based on the tree error or the whole forest error, and via a test set or out-of-bag samples—see Table 1 for a summary. We first give an overview of these different definitions, and then formalize them mathematically in the next subsection.

The most simple approach is taken by `scikit-learn` where the forest is fit with a training sample and the accuracy decrease is estimated with an independent testing sample. Throughout the article, we call the generalization error of the forest the expected quadratic risk for a new query point, usually estimated with an independent sample. Thus, forest predictions are run for both the testing sample and its permuted version, and the corresponding quadratic risks are subtracted to give the generalization error increase, denoted the **Train-Test MDA**. This procedure is also one of the options provided by `randomForestSRC`. However in practice, splitting the sample in two parts for training and testing often hurts the accuracy of the model, and then decreases the accuracy of the MDA estimate.

Since the data is bootstrapped prior to the construction of each tree, a portion of the sample is left out, and can be used to measure accuracy: the out-of-bag (OOB) sample. This principle is originally introduced by Breiman (Breiman, 2001), and to be precise, let us quote the original definition:

"Suppose there are M input variables. After each tree is constructed, the values of the m-th variable in the out-of-bag examples are randomly permuted and the out-of-bag data is run down the corresponding tree. The classification given for each $\boldsymbol{x}_n$ that is out of bag is saved. This is repeated for $m = 1, 2, \ldots, M$. At the end of the run, the plurality of out-of-bag class votes for $\boldsymbol{x}_n$ with the m-th variable noised up is compared with the true class label of $\boldsymbol{x}_n$ to give a misclassification rate. The output is the percent increase in misclassification rate as compared to the out-of-bag rate (with all variables intact)."

Despite the lack of mathematical formulation, it seems clear that for each tree, the generalization error is estimated using its OOB sample and the permuted version. Then, the two errors are subtracted and this difference is averaged across all trees to give the **Breiman-Cutler MDA**. Among the four main random forest implementations introduced above, only `ranger` and `randomForestSRC` exactly follow this definition. In `randomForests`, the final quan-

tity is normalized by the standard deviation of the generalization error differences. However, this procedure is questionable (Díaz-Uriarte and De Andres, 2006; Strobl and Zeileis, 2008): a non-influential variable would constantly have a small risk difference with a standard deviation close to zero, potentially leading to a high normalized MDA.

More importantly, observe that Breiman's MDA definition is in fact a Monte-Carlo estimate of a random tree decrease of accuracy when a variable is noised up. Since we are interested in the variable influence in the entire forest, and not only in a single tree, it seems natural to extend the OOB procedure to estimate the forest risk (Ishwaran, 2007; Ishwaran et al., 2008) and implemented in `randomForestSRC`: for each data point, we retrieve the set of trees which do not involve the considered point in their construction. The predictions are run for each tree of this collection and averaged to generate the OOB forest prediction for the considered point. Repeating this for the full sample enables to estimate the OOB quadratic risk of the forest. Then, a component of each out-of-bag sample is independently permuted, and the same procedure gives the inflated OOB forest risk. Finally, the difference between these two risks forms the **Ishwaran-Kogalur MDA**. From an algorithmic point of view, notice that the only difference with Breiman's definition is the mechanisms to aggregate tree predictions and compute the errors.

Overall, all these MDA definitions coexist in the main random forest implementations, and are widely used interchangeably. However, their subtle differences lead to their convergence towards distinct quantities. Consequently, the MDA versions are not equivalent and each of them is appropriate depending on the data distribution. To deepen the discussion, we mathematically formalize the three MDA versions.

2.2 Mathematical Formalization

We first need to define a standard regression setting with the following Assumption (A1), and introduce random forest notations below.

(A1) *The response* $Y \in \mathbb{R}$ *follows*

$$Y = m(\boldsymbol{X}) + \varepsilon$$

where $\boldsymbol{X} = (X^{(1)}, \dots, X^{(p)}) \in [0,1]^p$ *admits a density over* $[0,1]^p$ *bounded from above and below by stricly positive constants,* m *is continuous, and the noise* ε *is sub-Gaussian, independent of* $\boldsymbol{X}$*, and centered. A sample* $\mathscr{D}_n = \{(\boldsymbol{X}_1, Y_1), \dots, (\boldsymbol{X}_n, Y_n)\}$ *of* n *independent random variables distributed as* $(\boldsymbol{X}, Y)$ *is available.*

The random CART estimate $m_n(\mathbf{x}, \Theta)$ is trained with $\mathscr{D}_n$, and the bootstrap sampling and the split randomization are generated by Θ, and $\mathbf{x} \in [0,1]^p$ is the query point. The component of Θ used to resample the data is denoted $\Theta^{(S)} \subset \{1, \dots, n\}$. The random forest estimate $m_{M,n}(\mathbf{x}, \boldsymbol{\Theta}_M)$ aggregates M Θ-random CART, each of which is randomized by a component of $\boldsymbol{\Theta}_M = (\Theta_1, \dots, \Theta_M)$. In the sequel, we consider a fixed index $j \in \{1, \dots, p\}$. Next, we define $\mathbf{X}_{i,\pi_j}$ as the vector $\mathbf{X}_i$ where the j-th component is permuted between observations. Similarly, $\mathbf{X}_{\pi_j}$ is the vector $\mathbf{X}$ where the j-th component is replaced by an independent copy of $X^{(j)}$. Finally, we also introduce $\mathbf{X}^{(-j)}$, as the random vector $\mathbf{X}$ without the j-th component. Now, we can detail the three MDA definitions, summarized in Table 1.

Train/Test MDA. In this version of the MDA, the forest is trained with the available sample $\mathscr{D}_n$, and we assume that an independent testing sample $\mathscr{D}'_n = \{(\mathbf{X}'_1, Y'_1), \ldots, (\mathbf{X}'_n, Y'_n)\}$ is also available to estimate the quadratic risk of the forest, and the associated risk when a variable is noised up. Thus, the Train/Test MDA (TT-MDA) is formally defined by

$$\widehat{\mathrm{MDA}}^{(TT)}_{M,n}(X^{(j)}) = \frac{1}{n}\sum_{i=1}^{n} \big(Y'_i - m_{M,n}(\mathbf{X}'_{i,\pi_j}, \boldsymbol{\Theta}_M)\big)^2 - \big(Y'_i - m_{M,n}(\mathbf{X}'_i, \boldsymbol{\Theta}_M)\big)^2.$$

This algorithm is the only MDA version implemented in `scikit-learn`, and is one possibility in `randomForestSRC`. Note that the TT-MDA is straightforward to implement with any random forest package by simply running predictions.

Breiman-Cutler MDA. In the original definition, the quadratic risk of each tree is estimated for both the out-of-bag sample and the permuted out-of-bag sample. The average difference between these two risks is averaged across all trees to define the Breiman-Cutler MDA (Breiman, 2001). More precisely, for each Θ_ℓ-random tree, we randomly permute the j-th component of the out-of-bag dataset, and denote $\mathbf{X}_{i,\pi_{j\ell}}$ the i-th permuted sample for the ℓ-th tree and for $i \in \{1, \ldots, n\} \setminus \Theta^{(S)}_\ell$. Then, the Breiman-Cutler MDA (BC-MDA) is formally given by

$$\widehat{\mathrm{MDA}}^{(BC)}_{M,n}(X^{(j)}) = \frac{1}{M}\sum_{\ell=1}^{M}\frac{1}{N_{n,\ell}}\sum_{i=1}^{n}\big[(Y_i - m_n(\mathbf{X}_{i,\pi_{j\ell}}, \Theta_\ell))^2 - (Y_i - m_n(\mathbf{X}_i, \Theta_\ell))^2\big]\mathbb{1}_{i\notin\Theta^{(S)}_\ell},$$

where $N_{n,\ell} = \sum_{i=1}^{n}\mathbb{1}_{i\neq\Theta^{(S)}_\ell}$ is the size of the out-of-bag sample of the ℓ-th tree. This algorithm is available in `ranger` and `randomForestSRC`. In `randomForest`, by default, the BC-MDA is normalized by the standard deviation of the tree risk difference. Note that `ranger` also provides the possibility to normalize the BC-MDA.

Ishwaran-Kogalur MDA. Since the training data $\mathscr{D}_n$ is resampled prior to a tree construction, a portion of $\mathscr{D}_n$ is not involved in the growing of each tree. It is therefore possible to estimate the random forest error using $\mathscr{D}_n$ alone. More precisely, any sample $\mathbf{X}_i$ is not involved in the training of a random batch of trees, defined by

$$\Lambda_{n,i} = \{\ell \in \{1, \ldots, M\} : i \notin \Theta^{(S)}_\ell\}.$$

We can take advantage of such batch of trees to define the out-of-bag random forest estimate by averaging the tree predictions considering only trees that belong to $\Lambda_{n,i}$. Formally, for $i \in \{1, \ldots, n\}$,

$$m^{(OOB)}_{M,n}(\mathbf{X}_i, \boldsymbol{\Theta}_M) = \frac{1}{|\Lambda_{n,i}|}\sum_{\ell\in\Lambda_{n,i}} m_n(\mathbf{X}_i, \Theta_\ell)\mathbb{1}_{|\Lambda_{n,i}|>0}.$$

Recall that for each Θ_ℓ-random tree, we randomly permute the j-th component of the out-of-bag dataset to define $\mathbf{X}_{i,\pi_{j\ell}}$. We insist that the permutation is independent for each tree. Then, we define the permuted OOB forest estimate as

$$m^{(OOB)}_{M,n,\pi_j}(\mathbf{X}_i, \boldsymbol{\Theta}_M) = \frac{1}{|\Lambda_{n,i}|}\sum_{\ell\in\Lambda_{n,i}} m_n(\mathbf{X}_{i,\pi_{j\ell}}, \Theta_\ell)\mathbb{1}_{|\Lambda_{n,i}|>0}.$$

Finally, the Ishwaran-Kogalur MDA (IK-MDA) (Ishwaran, 2007; Ishwaran et al., 2008) is defined as

$$\widehat{\mathrm{MDA}}^{(IK)}_{M,n}(X^{(j)}) = \frac{1}{N_{M,n}} \sum_{i=1}^{n} (Y_i - m^{(OOB)}_{M,n,\pi_j}(\mathbf{X}_i, \mathbf{\Theta}_M))^2 - (Y_i - m^{(OOB)}_{M,n}(\mathbf{X}_i, \mathbf{\Theta}_M))^2,$$

where $N_{M,n} = \sum_{i=1}^{n} \mathbb{1}_{|\Lambda_{n,i}|>0}$ is the number of points which are not used in all tree constructions. This algorithm is implemented in `randomForestSRC`. Besides, this package also provides the possibility to define the IK-MDA by blocks: the trees of the forest are divided in a fixed number of blocks. The IK-MDA is estimated for each block and then averaged. Thus, the BC-MDA can be seen as a specific case where the number of blocks is the number of trees M.

An asymptotic analysis of these three MDA versions, summarized in Table 1, reveals that they do not share the same theoretical counterpart. Consequently, they have different meanings and generate different variable rankings, from which divergent conclusions can be drawn. However, these MDA versions are used interchangeably in practice. The convergence of the MDA is established in the next subsection, and then the different theoretical counterparts are analyzed in the following subsection.

2.3 MDA Inconsistency

The OOB estimate is involved in both the BC-MDA and IK-MDA, but is also used in practice to provide a fast estimate of the random forest error. We begin our asymptotic analysis by a result on the efficiency of the OOB estimate, stated in Proposition 1 below, which shows that the OOB error consistently estimates the generalization error of the forest. This result will be later used to establish the convergence of the IK-MDA. First observe that, by construction of the set of trees $\Lambda_{n,i}$, the OOB estimate aggregates a smaller number of trees than in the standard forest: $\mathbb{E}[|\Lambda_{n,i}|] = (1-a_n/n)M$ trees in average. Therefore, the risks of the OOB and standard forest estimates are different quantities. The following proposition states that for a fixed sample size n, the OOB risk converges towards the standard forest risk as the number of trees increases, with a fast rate of $1/M$. The only difference between the implemented algorithms and our theoretical results, is that the resampling in the forest growing is done without replacement to alleviate the mathematical analysis. We define a_n the number of subsampled training observations used to build each tree.

Proposition 1 *If Assumption (A1) is satisfied, for a fixed sample size n and $i \in \{1, \dots, n\}$, we have*

$$\left| \mathbb{E}\left[\left(m^{(OOB)}_{M,a_n,n}(\boldsymbol{X}_i, \mathbf{\Theta}_M) - m(\boldsymbol{X}_i) \right)^2 \right] - \mathbb{E}\left[\left(m_{M,a_n,n-1}(\boldsymbol{X}, \mathbf{\Theta}_M) - m(\boldsymbol{X}) \right)^2 \right] \right| = O\left(\frac{1}{M}\right).$$

To our knowledge, this is the first result which states the convergence of the OOB error towards the forest error for any fixed sample size. This suggests that growing a large number of trees in the forest—which is computationally possible and what is done in practice—ensures that the OOB estimate provides a good approximation of the forest error.

Next, the convergence of the three versions of the MDA holds under the following Assumption (A2) of the consistency of a theoretical randomized CART. Since we are interested

in the random forest interpretation through the MDA, it seems natural to conduct our analysis assuming that each tree of the forest is an efficient learner, i.e., consistent. To formalize such an assumption, we first define the variation of the regression function within a cell $A \subset [0,1]^p$ by

$$\Delta(m, A) = \sup_{x,x' \in A} |m(x) - m(x')|,$$

and secondly, we introduce $A_k^\star(\mathbf{x}, \Theta)$ the cell of the theoretical CART of depth k (randomized with Θ) in which the query point $\mathbf{x} \in [0,1]^p$ falls.

(A2) *The randomized theoretical CART tree built with the distribution of* $(\boldsymbol{X}, Y)$ *is consistent, that is, for all* $\boldsymbol{x} \in [0,1]^p$*, almost surely,*

$$\lim_{k \to \infty} \Delta(m, A_k^\star(\boldsymbol{x}, \Theta)) = 0.$$

At first glance, Assumption (A2) seems quite obscure since it involves the theoretical CART. However, Scornet et al. (2015) show that (A2) holds if the regression function is additive. Because the original CART (Breiman et al., 1984) is a greedy algorithm, (A2) may not always be satisfied when the regression function m has interaction terms. However, it holds if the CART algorithm is slightly modified to avoid splits to be close to the edges of cells, and the split randomization is slightly increased to have a positive probability to split in all directions at all nodes (Meinshausen, 2006; Wager and Athey, 2018). Indeed in that case, all cells become infinitely small as the tree depth k increases, and therefore (A2) holds by continuity of m. Such modifications of CART have a negligible impact in practice on the random forest estimate since the cut threshold and the split randomization increase can be chosen arbitrarily small. Notice that such asymptotic regime is specifically analyzed in the next section.

As specified above, a_n is the number of training observations subsampled without replacement to build each tree, and we define t_n as the final number of terminal leaves in every tree. Notice that we can specify a_n in $m_{M,a_n,n}(\mathbf{x}, \boldsymbol{\Theta}_M)$ or $m_{a_n,n}(\mathbf{x}, \Theta)$ when needed, but we omit it in general to avoid cumbersome notations. In order to properly define the MDA procedures, the out-of-bag sample needs to be at least of size 2 to enable permutations, i.e., $a_n \leq n - 2$. Finally, we need the following Assumption (A3) on the asymptotic regime of the empirical forest as stated in Scornet et al. (2015), which essentially controls the number of terminal leaves with respect to the sample size n to enforce the random forest consistency.

(A3) *The asymptotic regime of* a_n*, the size of the subsampling without replacement, and the number of terminal leaves* t_n *is such that* $a_n \leq n - 2$*,* $a_n/n < 1 - \kappa$ *for a fixed* $\kappa > 0$*,* $\lim_{n\to\infty} a_n = \infty$*,* $\lim_{n\to\infty} t_n = \infty$*, and* $\lim_{n\to\infty} t_n \frac{(\log(a_n))^9}{a_n} = 0$*.*

In the case of the IK-MDA, the number of trees has to tend to infinity with the sample size to ensure convergence. To lighten notations, we drop the dependence of M_n to n.

(A4) *The number of trees grows to infinity with the sample size* n*:* $M \underset{n\to\infty}{\longrightarrow} \infty$*.*

Now, we can state the convergence of all MDA algorithms. In particular, this asymptotic analysis reveals that the theoretical MDA counterparts are not identical across the different MDA definitions. Thus, input variables are ranked according to different criteria when the BC-MDA or IK-MDA is used. We deepen this discussion in the following subsection.

Theorem 1 *If Assumptions (A1), (A2), and (A3) are satisfied, then, for all* $M \in \mathbb{N}^\star$ *and* $j \in \{1, \ldots, p\}$ *we have*

$$(i) \quad \widehat{MDA}^{(TT)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{E}[(m(\boldsymbol{X}) - m(\boldsymbol{X}_{\pi_j}))^2]$$
$$(ii) \quad \widehat{MDA}^{(BC)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{E}[(m(\boldsymbol{X}) - m(\boldsymbol{X}_{\pi_j}))^2].$$

If Assumption (A4) is additionally satisfied, then

$$(iii) \quad \widehat{MDA}^{(IK)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{E}[(m(\boldsymbol{X}) - \mathbb{E}[m(\boldsymbol{X}_{\pi_j})|\boldsymbol{X}^{(-j)}])^2].$$

Sketch of proof of Theorem 1. The complete proof is to be found in the Supplementary Material and is based on the exact derivation of the MDA expressions defined above. Remarkably, the generalization error of the OOB forest, which appears in the IK-MDA, is upper bounded by the standard forest error, multiplied by the factor $2/(1 - a_n/n)$. Thus, the consistency of the original forest implies that the OOB forest error tends to zero. This bound is derived by controlling the randomness of the observation selection process in the tree construction. □

Besides, the package `randomForest` uses a modified version of the BC-MDA where it is normalized by the standard deviation of the risk differences across all trees. Since the risk difference converges towards the same constant for each tree, the theoretical counterpart of the standard deviation of the tree risk is null, and therefore the theoretical normalized BC-MDA is undefined. Note that `ranger` also provides the possibility to normalize the BC-MDA, but it is not the default setting. Futhermore, as we have already mentioned, the package `randomForestSRC` also provides the possibility to define the IK-MDA by blocks: the trees of the forest are divided in several blocks, and the IK-MDA is estimated for each block and then averaged. If the number of blocks is fixed and Assumption (A4) is satisfied, the number of trees in each block grows to infinity, and therefore Theorem 1-(iii) still holds.

2.4 MDA Analysis

The theoretical counterparts of the MDA established in Theorem 1 are hard to interpret since $\mathbf{X}_{\pi_j}$ has a different distribution than the original input data $\mathbf{X}$ whenever components of $\mathbf{X}$ are dependent. These different MDA versions are widely used in practice to assess the variable importance of random forests, but the relevance of such analyses completely relies on the ranking criteria $\mathbb{E}[(m(\mathbf{X}) - m(\mathbf{X}_{\pi_j}))^2]$ or $\mathbb{E}[(m(\mathbf{X}) - \mathbb{E}[m(\mathbf{X}_{\pi_j})|\mathbf{X}^{(-j)}])^2]$. It is possible to deepen the discussion, observing that $\mathbf{X}$ and $\mathbf{X}_{\pi_j}$ are independent conditionally on $\mathbf{X}^{(-j)}$ by construction. It enables to break down the MDA limit using Sobol indices that are well-defined quantity to measure the contribution of an input to the output variance.

Definition 1 (Total Sobol Index) *The total Sobol index of variable* $X^{(j)}$ *(Sobol, 1993; Saltelli, 2002) gives the proportion of explained output variance lost when* $X^{(j)}$ *is removed from the model, that is*

$$ST^{(j)} = \frac{\mathbb{E}[\mathbb{V}(m(\boldsymbol{X})|\boldsymbol{X}^{(-j)})]}{\mathbb{V}(Y)}.$$

Notice that $ST^{(j)}$ *is also called the independent total Sobol index in Mara et al. (2015) and Benoumechiara (2019).*

Definition 2 (Full Total Sobol Index) *The full total Sobol index of variable* $X^{(j)}$ *(Mara et al., 2015) gives the proportion of output variance explained by* $X^{(j)}$ *including the contribution due to its dependence and interactions with other inputs, that is*

$$ST^{(j)}_{full} = \frac{\mathbb{E}[\mathbb{V}(m(\boldsymbol{X}_{\pi_j})|\boldsymbol{X}^{(-j)})]}{\mathbb{V}(Y)}.$$

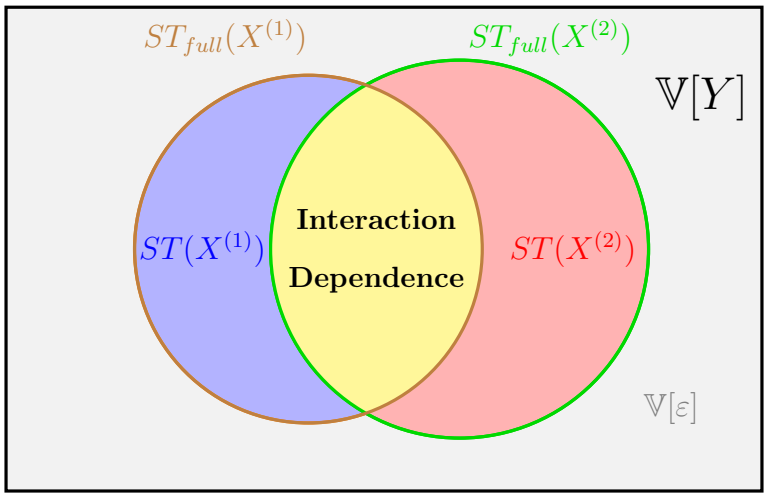

Figure 1: Illustration of the standard and full total Sobol indices for $Y = m(X^{(1)}, X^{(2)}) + \varepsilon$.

Figure 1 illustrates the total Sobol indices for an input dimension of $p = 2$. In particular, we see that a portion of the output variance is explained by $X^{(1)}$ alone, $ST(X^{(1)})$, another portion by $X^{(2)}$ alone, $ST(X^{(2)})$, and a last portion by the interaction and dependence between $X^{(1)}$ and $X^{(2)}$, which is added to the total Sobol index of each variable to define the full total Sobol indices. Thus, it is possible to break down the MDA limits using these total Sobol indices and the following quantity $\text{MDA}_3^{\star(j)}$, further discussed below and defined as

$$\text{MDA}_3^{\star(j)} = \mathbb{E}[(\mathbb{E}[m(\mathbf{X})|\mathbf{X}^{(-j)}] - \mathbb{E}[m(\mathbf{X}_{\pi_j})|\mathbf{X}^{(-j)}])^2].$$

Proposition 2 *If Assumptions (A1), (A2) and (A3) are satisfied, then for all* $M \in \mathbb{N}^\star$ *and* $j \in \{1, \dots, p\}$ *we have*

$$(i) \quad \widehat{MDA}^{(TT)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + \mathbb{V}[Y] \times ST^{(j)}_{full} + MDA_3^{\star(j)}$$

$$(ii) \quad \widehat{MDA}^{(BC)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + \mathbb{V}[Y] \times ST^{(j)}_{full} + MDA_3^{\star(j)}.$$

If Assumption (A4) is additionally satisfied, then

$$(iii) \quad \widehat{MDA}_{M,n}^{(IK)}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + MDA_3^{\star(j)}.$$

The proof is to be found in the Supplementary Material and is based on Theorem 1 and the independence of $m(\mathbf{X})$ and $m(\mathbf{X}_{\pi_j})$ conditionally on $\mathbf{X}^{(-j)}$. In the sequel, we denote $\text{MDA}_1^{\star(j)} = \mathbb{V}[Y] \times ST^{(j)}$ and $\text{MDA}_2^{\star(j)} = \mathbb{V}[Y] \times ST_{full}^{(j)}$. Each term of the decompositions of Proposition 2 can be interpreted alone. $\mathbf{MDA}_1^{\star(j)}$ is the non-normalized total Sobol index that has a straightforward interpretation: the amount of explained output variance lost when $X^{(j)}$ is removed from the model. This quantity is really the information one is looking for when computing the MDA for objective (i). $\mathbf{MDA}_2^{\star(j)}$ is the non-normalized full total Sobol index. Its interpretation is more difficult: it gives the "full" contribution of $X^{(j)}$ to the output variance, including the contribution due to its dependence and interactions with other variables. For example, if the regression function m does not depend on $X^{(j)}$ which is correlated to another influential input, then $ST^{(j)} = 0$ but $ST_{full}^{(j)} > 0$. For objective (i), one wants to keep only one variable of a group of highly influential and correlated inputs, and therefore $ST_{full}^{(j)}$ is a misleading component. $\mathbf{MDA}_3^{\star(j)}$ is not a known measure of importance, and seems to have no clear interpretation: it measures how the permutation shifts the average of m over the j-th input, and thus characterizes the structure of m and the dependence of $\mathbf{X}$ combined. $\text{MDA}_3^{\star(j)}$ is null if inputs are independent or if the regression function is additive. Regions of the input space $[0,1]^p$ that combine strong interactions with strong dependence contribute to increase the value of this third term, as illustrated in the analytical example in the following subsection.

Overall, all MDA definitions are misleading with respect to both objectives (i) and (ii) since they include $\text{MDA}_3^{\star(j)}$ in their theoretical counterparts. From a practical perspective, it is only possible to conclude in general that the BC-MDA or IK-MDA should be used rather than the TT-MDA. Indeed, on the one hand we only have access to one finite sample $\mathscr{D}_n$ in practice, which has to be split in two parts to use the TT-MDA, hurting the forest accuracy. On the other hand, it is possible to grow many trees at a reasonable linear computational cost, and Proposition 1 ensures that the OOB estimate is efficient in this case. With additional assumptions on the data distribution, the BC-MDA and the IK-MDA recover meaningful theoretical counterparts. In particular, when inputs are independent, the theoretical MDA is the unnormalized total Sobol index, as stated in Gregorutti (2015) and formalized in the following corollary.

Corollary 1 *If $\boldsymbol{X}$ has independent components, and if Assumptions (A1)-(A3) are satisfied, for all $M \in \mathbb{N}^\star$ and $j \in \{1, \ldots, p\}$ we have*

$$\widehat{MDA}_{M,n}^{(TT)}(X^{(j)}) \xrightarrow{\mathbb{L}^1} 2\mathbb{V}[Y] \times ST^{(j)}$$
$$\widehat{MDA}_{M,n}^{(BC)}(X^{(j)}) \xrightarrow{\mathbb{L}^1} 2\mathbb{V}[Y] \times ST^{(j)}.$$

In addition, if Assumptions (A4) is satisfied,

$$\widehat{MDA}_{M,n}^{(IK)}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)}.$$

Thus, Corollary 1 states that when inputs are independent, all MDA versions estimate the same quantity (up to a factor 2). However, since the TT-MDA is based on a portion

Algorithm	Settings		
	Independent inputs	**Additivity of m**	**Dependent inputs & Interactions**
TT-MDA	Objectives (i) & (ii)	Objective (ii)	None
BC-MDA	Objectives (i) & (ii)	Objective (ii)	None
IK-MDA	Objectives (i) & (ii)	Objective (i)	None

Table 2: Valid MDA objectives depending on the data characteristics.

of the training sample, the BC-MDA on the accuracy of a single tree, and the IK-MDA on the accuracy of the forest, the IK-MDA appears to be a more efficient estimate than the two others in this independent setting.

Interestingly, when inputs are dependent but without interactions, all MDA versions are well defined quantities, but with different theoretical counterparts. Such specific settings are quite frequent in practice and the BC-MDA and IK-MDA lead to drastically different conclusions as the following corollary shows.

Corollary 2 *If the regression function m is additive, and if Assumptions (A1)-(A3) are satisfied, for all $M \in \mathbb{N}^\star$ and $j \in \{1, \dots, p\}$ we have*

$$\widehat{MDA}^{(TT)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + \mathbb{V}[Y] \times ST^{(j)}_{full}$$
$$\widehat{MDA}^{(BC)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + \mathbb{V}[Y] \times ST^{(j)}_{full}.$$

In addition, if Assumptions (A4) is satisfied,

$$\widehat{MDA}^{(IK)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)}.$$

Thus, without interactions, the BC-MDA or IK-MDA should be used depending on the final objective. As already mentioned, the total Sobol index is the appropriate measure for our objective (i), and therefore the IK-MDA is the corresponding estimate. However, the IK-MDA would discard highly influential variables which are strongly correlated to other variables. Then, if one wants to identify all highly influential variables—objective (ii), the BC-MDA should rather be used in this additive setting. If we further assume that the regression function is linear, the MDA limits can be explicitly written with the linear coefficients and the input variances as stated in Gregorutti et al. (2015); Hooker and Mentch (2019), and also left as an exercise in chapter 15 of Friedman et al. (2001).

Proposition 2, Corollary 1, and Corollary 2 are summarized in Table 2 with respect to objectives (i) and (ii). Notice that in the case of independent input variables, the total Sobol index is a relevant measure for both objectives (i) and (ii). Next, in the following subsection, we provide an analytical example to show how the MDA can fail to detect relevant variables when the data has both dependence and interactions.

Remark 1 (Full Total Sobol Index) *One can observe that under Assumptions (A1)-(A4), for all* $j \in \{1, \dots, p\}$ *we have*

$$\widehat{MDA}^{(BC)}_{M,n}(X^{(j)}) - \widehat{MDA}^{(IK)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)}_{full}.$$

Thus, subtracting the BC-MDA and the IK-MDA provides a consistent estimate of the full total Sobol index, which is a quantity of interest for objective (ii). However, the BC-MDA is based on the tree error, whereas the IK-MDA is based on the forest error. Consequently both the terms $\mathbb{V}[Y] \times ST^{(j)}$ *and* $MDA^{\star(j)}_3$ *are based on different estimates using the BC-MDA or the IK-MDA and their subtraction gives an estimate with a strong bias.*

Remark 2 (Distribution Support) *Our asymptotic analysis relies on Assumption (A1), which states that the support of the distribution of the input* $\boldsymbol{X}$ *is a hypercube. Without such geometrical assumption, the support of* $\boldsymbol{X}_{\pi_j}$ *may differ from the support of* $\boldsymbol{X}$ *in the dependent case. It means that the permuted samples may query the random forest in regions with no training samples, resulting in inconsistent forest and MDA estimates, and then in a poor empirical performance (Hooker and Mentch, 2019). This is an additional source of confusion of the MDA when inputs are dependent, induced by the permutation trick.*

2.5 Analytical Example

To illustrate the behavior of the MDA, we take a simple example and analytically derive the MDA limit and its three associated components $\text{MDA}^{\star(j)}_1$, $\text{MDA}^{\star(j)}_2$, and $\text{MDA}^{\star(j)}_3$. This example shows how the MDA is misleading when input variables are dependent. We consider the BC-MDA, denoted as MDA to lighten notations. The TT-MDA or IK-MDA lead to identical conclusions.

Example description. The input $\mathbf{X}$ is a Gaussian vector of dimension $p = 5$. Its covariance matrix is defined by $\mathbb{V}[X^{(j)}] = \sigma_j^2$ for $j \in \{1, \dots, 5\}$, and all covariance terms are null except $\text{Cov}[X^{(1)}, X^{(2)}] = \rho_{1,2}\sigma_1\sigma_2$ and $\text{Cov}[X^{(4)}, X^{(5)}] = \rho_{4,5}\sigma_4\sigma_5$. The regression function m is given by

$$m(\mathbf{X}) = \alpha X^{(1)} X^{(2)} \mathbb{1}_{X^{(3)}>0} + \beta X^{(4)} X^{(5)} \mathbb{1}_{X^{(3)}<0}.$$

Notice that m has a simple form to enable an easy interpretation of the importance measures, but that interaction terms are required to highlight the different behaviors of the three MDA components in a correlated setting. Simple calculations give the analytical expression $\text{MDA}^{\star(1)}$ of the MDA limit for $X^{(1)}$ as

$$\text{MDA}^{\star(1)} = \underbrace{\frac{1}{2}(\alpha\sigma_1\sigma_2)^2(1-\rho_{1,2}^2)}_{\text{MDA}^{\star(1)}_1} + \underbrace{\frac{1}{2}(\alpha\sigma_1\sigma_2)^2}_{\text{MDA}^{\star(1)}_2} + \underbrace{\frac{3}{2}\rho_{1,2}^2(\alpha\sigma_1\sigma_2)^2}_{\text{MDA}^{\star(1)}_3}.$$

First, observe that $\text{MDA}^{\star(1)}_1$ decreases with the correlation between $X^{(1)}$ and $X^{(2)}$. Indeed, $\text{MDA}^{\star(1)}_1$ is the total Sobol index and when these two variables are strongly dependent, the additional information provided by $X^{(1)}$ alone is small. In the extreme case, $\rho_{1,2} = 1$ implies

that $\text{MDA}_1^{\star(1)} = 0$, i.e., $X^{(1)}$ can be removed from the model without hurting the model accuracy since all its information is contained in $X^{(2)}$. On the other hand, $\text{MDA}_2^{\star(1)}$ does not rely on the dependence between $X^{(1)}$ and $X^{(2)}$. Indeed, this term is the full total Sobol index that considers the contribution of $X^{(1)}$ including its dependence and interactions with other variables. It is clear that the MDA mixes two terms with opposite meanings: the marginal contribution and the full contribution to the output variance. Finally, the third term $\text{MDA}_3^{\star(1)}$ measures how the permutation of $X^{(1)}$ shifts the mean value of the regression function averaged over $X^{(1)}$, which is not a quantity of interest to rank variables. However, in a high correlation setting $\left(\rho_{1,2} > \frac{\sqrt{2}}{2}\right)$, we have $\text{MDA}_3^{\star(1)} > \text{MDA}_1^{\star(1)} + \text{MDA}_2^{\star(1)}$, which means that the meaningless third term is the main contribution of the MDA value of variable $X^{(1)}$. Besides, symmetrically for the other input variables, we have $\text{MDA}^{\star(1)} = \text{MDA}^{\star(2)}$, and the same formula for $X^{(4)}$ and $X^{(5)}$ with the appropriate parameters. MDA formulas for variables $3, 4$, and 5 are to be found in the Supplementary Material.

Inaccurate variable selection. As stated in the introduction, one of the main objective of variable importance analysis is usually to select a small number of variables while maximizing the model accuracy. In our example, we show how the MDA fails for this purpose. Let say we want to remove the less relevant input variable in a setting where the two vectors $\mathbf{X}^{(1,2)}$ and $\mathbf{X}^{(4,5)}$ are interchangeable ($\alpha\sigma_1\sigma_2 = \beta\sigma_4\sigma_5$), except that their dependence strengths differ and satisfy $\rho_{1,2} < \rho_{4,5}$. Since the correlation between variables 4 and 5 is higher than between variables 1 and 2, we should remove $X^{(4)}$ or $X^{(5)}$ to minimize the information loss, as suggested by the total Sobol index ranking

$$ST^{(4)} = ST^{(5)} < ST^{(1)} = ST^{(2)} < ST^{(3)}.$$

However, in such setting we have

$$\text{MDA}^{\star(1)} = \text{MDA}^{\star(2)} < \text{MDA}^{\star(3)} < \text{MDA}^{\star(4)} = \text{MDA}^{\star(5)},$$

that would lead to discard $X^{(1)}$ or $X^{(2)}$, which is suboptimal—see the Supplementary Material for computation details. On the other hand, using only $\text{MDA}_1^{\star(j)}$ or $\text{MDA}_1^{\star(j)} + \text{MDA}_2^{\star(j)}$ as importance measures gives the accurate variable selection. The term $\text{MDA}_3^{\star(j)}$ artificially increases the MDA value because of correlation, and is thus misleading for both objectives (i) and (ii).

3 Sobol-MDA

When input variables are dependent, the MDA fails to estimate the total Sobol index, which is our true target to solve problem (i), as shown in Section 2. Therefore, we introduce an improved MDA procedure for random forests: the Sobol-MDA, that consistently estimates the total Sobol index even when input variables are dependent and have interactions. The Sobol-MDA is able to identify the less relevant variable among the input data, as the total Sobol index is the proportion of output explained variance lost when a given variable is removed from the model. Therefore, a recursive feature elimination procedure based on the Sobol-MDA is highly efficient for our objective (i) of selecting a small number of variables while

maximizing predictive accuracy. Notice that training a random forest without the variable of interest would also enable to get an estimate of the total Sobol index. However, the Sobol-MDA only requires to perform forest predictions, which is computationally faster than the forest growing. It is also possible to estimate total Sobol indices with existing algorithms which are not specific to random forests. Indeed, this type of methods only requires a black-box estimate to generate predictions from given values of the input variables. Initially, Mara et al. (2015) introduce Monte-Carlo algorithms for the estimation of total Sobol indices in a dependent setting. The first step of the method is to generate a sample from the conditional distributions of the inputs. However, in our setting defined in Assumption (A1), we do not have access to these conditional distributions, and their estimation is a difficult problem when only a limited sample $\mathscr{D}_n$ is available. Consequently, the approach of Mara et al. (2015) is not really appropriate for our setting.

In the first subsection, we introduce the Sobol-MDA algorithm. Next, we focus on the associated properties: the computational complexity and the algorithm consistency. In the third subsection, we show the good empirical behavior of the proposed algorithm through experiments on both simulated and real data, especially when used in a recursive feature elimination procedure.

3.1 Sobol-MDA Algorithm

The key feature of the original MDA procedures is to permute the values of the j-th component of the data to break its relation to the output, and then compute the degraded accuracy of the forest. Observe that this is strictly equivalent to drop the original dataset down each tree of the forest, but when a sample hits a split involving variable j, it is randomly sent to the left or right side with a probability equal to the proportion of points in each child node. This fact highlights that the goal of the MDA is simply to perturb the tree prediction process to cancel out the splits on variable j. Besides, notice that this point of view on the MDA procedure (using the original dataset and noisy trees) is introduced by Ishwaran (2007) to conduct a theoretical analysis of a modified version of the MDA. Here, our Sobol-MDA algorithm builds on the same principle of ignoring splits on variable j, such that the noisy CART tree predicts $\mathbb{E}[m(\mathbf{X})|\mathbf{X}^{(-j)}]$ (instead of $m(\mathbf{X})$ for the original CART). It enables to recover the proper theoretical counterpart: the unnormalized total Sobol index, i.e., $\mathbb{E}[\mathbb{V}(m(\mathbf{X})|\mathbf{X}^{(-j)})]$. To achieve this, we leave aside the permutation trick, and use another approach to cancel out a given variable j in the tree prediction process: the partition of the input space obtained with the terminal leaves of the original tree is projected along the j-th direction—see Figure 2, and the outputs of the cells of this new projected partition are recomputed with the training data. From an algorithmic point of view, this procedure is quite straight-forward as we will see below, and enables to get rid of variable $X^{(j)}$ in the tree estimate. Then, it is possible to compute the accuracy of the associated OOB projected forest estimate, subtract it from the original accuracy, and normalize the obtained difference by $\mathbb{V}[Y]$ to obtain the Sobol-MDA for variable $X^{(j)}$.

Interestingly, to compute SHAP values for tree ensembles, Lundberg et al. (2018) also introduce an algorithm to modify the CART predictions to estimate $\mathbb{E}[m(\mathbf{X})|\mathbf{X}^{(-j)}]$. More precisely, they propose the following recursive algorithm: the query point $\mathbf{x}$ is dropped down

the tree, but when a split on variable j is hit, $\mathbf{x}$ is sent to both the left and right children nodes. Then, $\mathbf{x}$ falls in multiple terminal cells of the tree. The final prediction is the weighted average of the cell outputs, where the weight associated to a terminal leave A is given by an estimate of $\mathbb{P}(\mathbf{X} \in A | \mathbf{X}^{(-j)} = \mathbf{x}^{(-j)})$: the product of the empirical probabilities to choose the side that leads to A at each split on variable j in the path of the original tree. At first sight, their approach seems suited to estimate total Sobol indices, but unfortunately, the weights are properly estimated by such procedure only if the components of $\mathbf{X}$ are independent. Therefore, as highlighted in Aas et al. (2019), this algorithm gives biased predictions in a correlated setting.

We improve over Lundberg et al. (2018) with the Projected-CART Algorithm 1: both training and out-of-bag samples are dropped down the tree and sent on both right and left children nodes when a split on variable j is met. Again, each data point may belong to multiple cells at each level of the tree. For each out-of-bag sample, the associated prediction is the output average over all training samples that belong to the same collection of terminal leaves. This mechanism is equivalent to projecting the tree partition on the subspace span by $\mathbf{X}^{(-j)}$, as illustrated in Figure 2 for $p = 2$ and $j = 2$. Recall that $A_n(\mathbf{X}, \Theta)$ is the cell of the original tree partition where $\mathbf{X}$ falls, whereas the associated cell of the projected partition is denoted $A_n^{(-j)}(\mathbf{X}^{(-j)}, \Theta)$. Formally, we respectively denote the associated projected tree and projected out-of-bag forest estimates as $m_n^{(-j)}(\mathbf{X}^{(-j)}, \Theta)$ and $m_{M,n}^{(-j,OOB)}(\mathbf{X}_i^{(-j)}, \mathbf{\Theta}_M)$, respectively defined by

$$m_n^{(-j)}(\mathbf{X}^{(-j)}, \Theta) = \frac{\sum_{i=1}^{a_n} Y_i \mathbb{1}_{\mathbf{X}_i \in A_n^{(-j)}(\mathbf{X}^{(-j)}, \Theta)}}{\sum_{i=1}^{a_n} \mathbb{1}_{\mathbf{X}_i \in A_n^{(-j)}(\mathbf{X}^{(-j)}, \Theta)}},$$

and for $i \in \{1, \dots, n\}$,

$$m_{M,n}^{(-j,OOB)}(\mathbf{X}_i^{(-j)}, \mathbf{\Theta}_M) = \frac{1}{|\Lambda_{n,i}|} \sum_{\ell \in \Lambda_{n,i}} m_n^{(-j)}(\mathbf{X}_i^{(-j)}, \Theta_\ell) \mathbb{1}_{|\Lambda_{n,i}| > 0}.$$

The Projected-CART algorithm provides two sources of improvements over Lundberg et al. (2018): first, the training data points are dropped down the modified tree to recompute the cell outputs, and thus $\mathbb{E}[m(\mathbf{X}) | \mathbf{X}^{(-j)} \in A]$ is directly estimated in each cell. Secondly, the projected partition is finer than in the original tree, which mitigates masking effects (when an influential variable is not often selected in the tree splits because of other highly correlated variables).

Finally, the Sobol-MDA estimate is given by the normalized difference of the quadratic error of the OOB projected forest with the OOB error of the original forest. Formally, we define the Sobol-MDA as

$$\widehat{\text{S-MDA}}_{M,n}(X^{(j)}) = \frac{1}{\hat{\sigma}_Y^2} \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - m_{M,n}^{(-j,OOB)}(\mathbf{X}_i^{(-j)}, \mathbf{\Theta}_M)\right)^2 - \left(Y_i - m_{M,n}^{(OOB)}(\mathbf{X}_i, \mathbf{\Theta}_M)\right)^2,$$

where $\hat{\sigma}_Y^2 = \frac{1}{n-1} \sum_{i=1}^{n} (Y_i - \bar{Y})^2$ is the standard variance estimate of the output Y. An implementation in `R` and `C++` of the Sobol-MDA is available at `https://gitlab.com/cbenard/sobol-mda` and is based on `ranger` (Wright and Ziegler, 2017), a fast implementation of random forests.

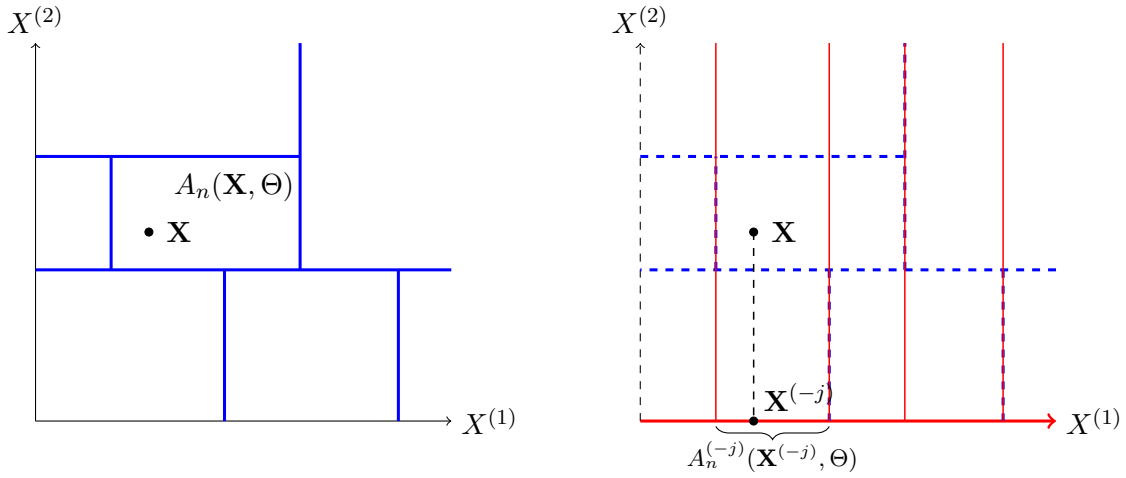

Figure 2: Example of the partition of $[0,1]^2$ by a random CART tree (left side) projected on the subspace span by $\mathbf{X}^{(-2)} = X^{(1)}$ (right side). Here, $p = 2$ and $j = 2$.

Remark 3 (Empty Cells) *Some cells of the projected partition may contain no training samples. Consequently, the prediction for a new query point falling in such cells is undefined. In practice, the Projected-CART algorithm 1 uses the following strategy to avoid empty cells. Recall that each level of the tree defines a partition of the input space (if a terminal leave occurs before the final tree level, it is copied down the tree at each level), and that a projected partition can thus be associated to each tree level. When a new query point is dropped down the tree, if it falls in an empty cell of the projected partition at a given tree level, the prediction is computed using the previous level. Notice that empty cells cannot occur in the partitions associated to the root and the first level of the tree by construction. Therefore, this mechanism enforces that the projected tree estimate is well defined over the full input space.*

3.2 Sobol-MDA Properties

Computational complexity. By definition, an estimate of the total Sobol index is given by the following procedure: retrain the random forest without the j-th variable, and subtract the associated explained variance to the original accuracy with all variables. However, this brute force approach is computationally expensive since it requires to fit p forests to get the total Sobol index of each variable. Louppe (2014) states that the average computational complexity of the forest growing is $O(Mpn\log^2(n))$. Thus, the total complexity of the brute force approach is $O(Mp^2n\log^2(n))$, which is quadratic with the dimension p and therefore intractable in high-dimensional settings.

On the other hand, the original MDA procedure has an average complexity of $O(Mpn\log(n))$: to run a balanced tree prediction for a given data point, it is dropped down the $\log(n)$ levels of the tree, which makes a complexity of $O(n\log(n))$ for the full OOB sample, repeated for the M trees of the forest and the p variables. In the Sobol-MDA procedure, the complexity analysis is similar, except that when a point is dropped down the tree, it can be sent to both

Algorithm 1 Projected-CART

Input: A Θ-random CART built with $\mathscr{D}_n$, and a variable index $j \in \{1, \ldots, p\}$. (Note that if a terminal leave occurs before the final tree level, it is copied at each level down the tree.)
Initialize both in-bag and OOB samples at the root node of the tree;
for all tree levels:
    for all level nodes:
        if the splitting variable is not j:
            send each data point to the right or left children node according to the node split;
        if the splitting variable is j:
            send the node sample to both the right and left children node ignoring the split;
    for all data points:
        retrieve the collection of nodes where the data point falls at the current tree level;
    for all OOB data points:
        retrieve the set of in-bag points which fall in the same node collection;
        if all nodes in the considered node collection are terminal:
            compute the output average of the in-bag points;
            set this average as the prediction for the considered OOB observation;
        if no in-bag points fall in the same node collection:
            retrieve the corresponding in-bag data points at the previous tree level;
            set the output average of these in-bag points as the prediction for the considered OOB observation;
return predictions;

the left and right children nodes, generating multiple operations at a given tree level and then an additional multiplicative factor of $\log(n)$. However, it is not necessary to run the Projected-CART algorithm for each of the p variables. Indeed, when a given observation is dropped down the tree, it meets at most $\log(n)$ different variables in the original tree path. Therefore, the Projected-CART prediction has to be computed only for $\log(n)$ variables for each observation. Thus, the Sobol-MDA algorithm has a computational complexity of $O(Mn\log^3(n))$, which is in particular independent of the dimension p, and quasi-linear with the sample size n.

Consistency. The original MDA versions do not converge towards the total Sobol index, which is the relevant quantity for our objective (i)—see Proposition 2. On the other hand, the Sobol-MDA is consistent as stated below. Before introducing this convergence result, we need to introduce additional assumptions. Indeed, in Section 2, we show the convergence of the different MDA versions provided that the forest is an efficient estimate, i.e. consistent. To enforce the consistency of random forests, we used Assumption (A2) which controls the variation of the regression function in each cell of the theoretical tree: $\Delta(m, A^\star(\mathbf{x}, \Theta)) \xrightarrow{a.s.} 0$. Because the components of $\mathbf{X}$ may be dependent, Assumption (A2) does not imply the same property for the projected partition. Therefore, we cannot directly build on the consistency result from Scornet et al. (2015) to prove the consistency of the Sobol-MDA. Thus, we take another route and define a new Assumption (A2') which brings two modifications to the random forest algorithm.

(A2') *A node split is constrained to generate child nodes with at least a small fraction $\gamma > 0$ of the parent node observations. Secondly, the split selection is slightly modified: at each tree node, the number* `mtry` *of candidate variables drawn to optimize the split is set to* `mtry` $= 1$ *with a small probability $\delta > 0$. Otherwise, with probability $1 - \delta$, the default value of* `mtry` *is used.*

Importantly, since γ and δ can be chosen arbitrarily small, the modifications of assumption (A2') are mild. Besides, notice that this assumption follows Meinshausen (2006) and Wager and Athey (2018): we slightly modify the random forest algorithm to enforce empirical cells to become infinitely small as the sample size increases. The projected forest inherits this property and an asymptotic analysis from Györfi et al. (2006) gives the consistency of the Sobol-MDA, provided that the complexity of tree partitions is appropriately controlled. If an original tree has t_n terminal leaves, the associated projected partition may have a higher number of terminal leaves, at most 2^{t_n}. Thus, we introduce Assumption (A3'), which slightly modifies (A3) with a more restrictive regime for the number of terminal leaves t_n in the original trees.

(A3') *The asymptotic regime of a_n, the size of the subsampling without replacement, and the number of terminal leaves t_n is such that $a_n \leq n - 2$, $a_n/n < 1 - \kappa$ for a fixed $\kappa > 0$,* $\lim_{n\to\infty} a_n = \infty$, $\lim_{n\to\infty} t_n = \infty$, *and* $\lim_{n\to\infty} 2^{t_n} \frac{(\log(a_n))^9}{a_n} = 0$.

The Projected-CART algorithm ignores the splits based on the j-th variable, and the associated OOB projected forest consistently estimates $\mathbb{E}[m(\mathbf{X})|\mathbf{X}^{(-j)}]$ under Assumptions (A1), (A2'), and (A3'), which leads to the consistency of the Sobol-MDA as stated in the theorem below. The proof is to be found in the Supplementary Material.

Theorem 2 *If Assumptions (A1), (A2'), and (A3') are satisfied, for all* $M \in \mathbb{N}^\star$ *and* $j \in \{1, \ldots, p\}$

$$\widehat{S\text{-}MDA}_{M,n}(X^{(j)}) \xrightarrow{p} ST^{(j)}.$$

Theorem 2 shows that the proposed Sobol-MDA algorithm consistently estimates the total Sobol index, which gives the proportion of output explained variance lost when a given variable is removed from the model. Therefore, the Sobol-MDA targets the appropriate quantity for objective (i), of selecting a small number of variables while maximizing accuracy, as opposed to the original MDA versions—see Proposition 2. Besides, we also insist that the Sobol-MDA estimate is normalized by the output variance, and is thus easily interpretable since it gives a proportion of output variance allocated to a given input variable.

3.3 Experiments

We conduct three batches of experiments. First, we come back to the analytical example of the previous section, and show empirically that the Sobol-MDA leads to the accurate importance variable ranking, while original MDA versions do not. Next, we simulate a typical setting where several groups of variables are strongly correlated and only few inputs are involved in the regression function. In such difficult setting, the Sobol-MDA identifies the relevant variables, as opposed to the original MDA versions. Finally, we apply the RFE on real data to show the performance improvement of the Sobol-MDA for variable selection.

Simulated data: example 1. We consider the same example as in Section 2, where the data has both dependence and interactions. In our example, recall that the input is a Gaussian vector with $p = 5$, and the regression function is given by

$$m(\mathbf{X}) = \alpha X^{(1)} X^{(2)} \mathbb{1}_{X^{(3)}>0} + \beta X^{(4)} X^{(5)} \mathbb{1}_{X^{(3)}<0}.$$

Here, we set $\alpha = 1.5$, $\beta = 1$, $\mathbb{V}[X^{(j)}] = 1$ for all variables $j \in \{1, \ldots, 5\}$, and the correlation coefficients are set to $\rho_{1,2} = 0.9$ and $\rho_{4,5} = 0.6$ (other covariance terms are null). Finally, we define the model output as $Y = m(\mathbf{X}) + \varepsilon$, where ε is an independent centered gaussian noise whose variance verifies $\mathbb{V}[\varepsilon]/\mathbb{V}[Y] = 10\%$. Then, we run the following experiment: first, we generate a sample $\mathscr{D}_n$ of size $n = 3000$ and distributed as the Gaussian vector $\mathbf{X}$. Next, a random forest of $M = 300$ trees is fit with $\mathscr{D}_n$ and we compute the BC-MDA, IK-MDA, and Sobol-MDA. To enable comparisons, the BC-MDA is normalized by $2\mathbb{V}[Y]$, and the IK-MDA by $\mathbb{V}[Y]$—see Proposition 2. To show the improvement of our Projected-CART algorithm, we also compute the Sobol-MDA using the algorithm from Lundberg et al. (2018), denoted $\widehat{\text{S-MDA}}_{Ldg}$. All results are reported in Table 3 and the theoretical counterparts of the estimates are also provided. Notice that the associated standard deviations are gathered in Table 4, and that the variables are ranked by decreasing values of the theoretical total Sobol index since it is the value of interest: $\mathbf{X}^{(3)}$, then $\mathbf{X}^{(4)}$ and $\mathbf{X}^{(5)}$, and finally $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$.

Only the Sobol-MDA computed with the Projected-CART algorithm ranks the variables in the same appropriate order than the total Sobol index. In particular, $\mathbf{X}^{(4)}$ and $\mathbf{X}^{(5)}$ have a higher total Sobol index than variables 1 and 2 because of the stronger correlation between

	BC-MDA*	$\widehat{\text{BC-MDA}}$	IK-MDA*	$\widehat{\text{IK-MDA}}$	ST*	$\widehat{\text{S-MDA}}$	$\widehat{\text{S-MDA}}_{Ldg}$
$\mathbf{X}^{(3)}$	0.47	0.37	0.47	0.43	0.47	0.45	0.43
$\mathbf{X}^{(4)}$	0.21	0.10	0.37	0.14	0.10	0.08	0.13
$\mathbf{X}^{(5)}$	0.21	0.09	0.37	0.13	0.10	0.08	0.13
$\mathbf{X}^{(1)}$	0.64	0.24	1.0	0.29	0.07	0.05	0.22
$\mathbf{X}^{(2)}$	0.64	0.24	1.0	0.28	0.07	0.05	0.23

Table 3: Normalized BC-MDA, normalized IK-MDA, and Sobol-MDA estimates for Example 1.

	$\widehat{\text{IK-MDA}}$	$\widehat{\text{BC-MDA}}$	$\widehat{\text{S-MDA}}$	$\widehat{\text{S-MDA}}_{Ldg}$
$\mathbf{X}^{(3)}$	0.02	0.03	0.03	0.03
$\mathbf{X}^{(4)}$	0.01	0.02	0.01	0.01
$\mathbf{X}^{(5)}$	0.01	0.01	0.01	0.01
$\mathbf{X}^{(1)}$	0.02	0.02	0.01	0.02
$\mathbf{X}^{(2)}$	0.02	0.02	0.01	0.01

Table 4: Standard deviations of the normalized BC-MDA, normalized IK-MDA, and Sobol-MDA estimates over 10 repetitions for Example 1.

$\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ than between $\mathbf{X}^{(4)}$ and $\mathbf{X}^{(5)}$. For all the other importance measures, $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are more important than $\mathbf{X}^{(4)}$ and $\mathbf{X}^{(5)}$. For the original MDA, this is due to the higher coefficient $\alpha = 1.5 > \beta = 1$, to the term $\text{MDA}_2^{\star(j)}$, and especially to $\text{MDA}_3^{\star(j)}$ which increases with correlation. Since the explained variance of the random forest is 82% in this experiment, all estimates have a negative bias. The bias of the BC-MDA and IK-MDA dramatically increases with correlation. Indeed, a strong correlation between variables leaves some regions of the input space free of training data. However, the OOB permuted sample queries the forest in these regions where the forest extrapolates. This phenomenon combined with the $\text{MDA}_3^{\star(j)}$ component explains the high bias of the BC-MDA and IK-MDA for correlated inputs. Also observe that since $\mathbf{X}^{(3)}$ is independent of the other variables, the bias is small for both the BC/IK-MDA, and it is smaller for the IK-MDA than the BC-MDA as the forest estimate is more accurate than a single tree. Finally, the Sobol-MDA computed with the algorithm of (Lundberg et al., 2018) is biased as suggested by (Aas et al., 2019), and the bias also seems to increase with correlation.

Simulated data: example 2. We consider the following problem inspired by Archer and Kimes (2008); Gregorutti et al. (2017) and related to gene expressions. The goal is to identify relevant variables among several groups of many strongly correlated inputs, where the output is a linear combination of only one variable per group. In this dependent and additive setting, the BC-MDA is expected to behave poorly because of the full total Sobol index component—see Corollary 2, whereas the IK-MDA has the appropriate theoretical counterpart. We will see that the Sobol-MDA also outperforms the IK-MDA in practice. More precisely, we define $\mathbf{X}$, a random vector of dimension $p = 200$, composed of 5 independent groups of 40 variables. Each

$\widehat{\text{S-MDA}}$		$\widehat{\text{BC-MDA}}/2\mathbb{V}[Y]$		$\widehat{\text{IK-MDA}}/\mathbb{V}[Y]$	
$\mathbf{X}^{(1)}$	0.035	$\mathbf{X}^{(1)}$	0.048	$\mathbf{X}^{(1)}$	0.056
$\mathbf{X}^{(161)}$	0.005	$\mathbf{X}^{(25)}$	0.010	$\mathbf{X}^{(5)}$	0.009
$\mathbf{X}^{(81)}$	0.004	$\mathbf{X}^{(31)}$	0.008	$\mathbf{X}^{(81)}$	0.007
$\mathbf{X}^{(121)}$	0.004	$\mathbf{X}^{(14)}$	0.008	$\mathbf{X}^{(41)}$	0.005
$\mathbf{X}^{(41)}$	0.002	$\mathbf{X}^{(40)}$	0.007	$\mathbf{X}^{(161)}$	0.005
$\mathbf{X}^{(179)}$	0.002	$\mathbf{X}^{(3)}$	0.007	$\mathbf{X}^{(15)}$	0.005
$\mathbf{X}^{(13)}$	0.001	$\mathbf{X}^{(17)}$	0.006	$\mathbf{X}^{(121)}$	0.005
$\mathbf{X}^{(25)}$	0.001	$\mathbf{X}^{(26)}$	0.006	$\mathbf{X}^{(7)}$	0.005
$\mathbf{X}^{(73)}$	0.001	$\mathbf{X}^{(41)}$	0.006	$\mathbf{X}^{(4)}$	0.004
$\mathbf{X}^{(155)}$	0.001	$\mathbf{X}^{(121)}$	0.006	$\mathbf{X}^{(28)}$	0.004

Table 5: Normalized BC-MDA, normalized IK-MDA, and Sobol-MDA estimates (influential variables in blue) for Example 2.

group is a centered gaussian random vector where two distinct components have a correlation of 0.8 and the variance of each input is 1. The regression function m only involves one variable from each group, and is simply defined by

$$m(\mathbf{X}) = 2X^{(1)} + X^{(41)} + X^{(81)} + X^{(121)} + X^{(161)}.$$

Finally, we define the model output as $Y = m(\mathbf{X}) + \varepsilon$, where ε is an independent gaussian noise ($\mathbb{V}[\varepsilon]/\mathbb{V}[Y] = 10\%$). Next, a sample of size $n = 1000$ is generated based on the distribution of $\mathbf{X}$, and a random forest of $M = 300$ trees is fit.

Table 5 shows that the Sobol-MDA identifies the 5 relevant variables, whereas both the BC-MDA and IK-MDA identify some noisy variables among the top 5. As expected from Corollary 2, the IK-MDA performs much better than the BC-MDA in this additive and dependent setting. In this case, the IK-MDA converges towards the total Sobol index, as the Sobol-MDA, whereas the BC-MDA limit has an additional term: the full total Sobol index, which increases the importance of all variables of the first group (1 to 40) because of their correlation with the most influential variable $X^{(1)}$.

Recursive feature elimination. The Recursive Feature Elimination algorithm (RFE) is originally introduced by Guyon et al. (2002) to perform variable selection with SVM. Gregorutti et al. (2017) apply RFE to random forests with the MDA as importance measure. The principle of the RFE algorithm is to discard the less relevant input variables one by one, and is summarized in Algorithm 2. Thus, the RFE is a relevant strategy for our objective (i) of building a model with a high accuracy and a small number of variables. At each step of the RFE, the goal is to detect the less relevant input variable based on the trained model. Since the total Sobol index measures the proportion of explained output variance lost when a given variable is removed, the optimal strategy is therefore to discard the variable with the smallest total Sobol index. As the Sobol-MDA directly estimates the total Sobol index whereas existing MDA all have additional noisy terms—see Section 2, using the Sobol-MDA improves the performance of the RFE, as shown in the following experiments.

Algorithm 2 Recursive Feature Elimination

1: for j in $1, \ldots, p$:
2: train a random forest
3: compute the MDA for all variables
4: remove the variable with the smallest MDA
5: return the ordered list of removed variables

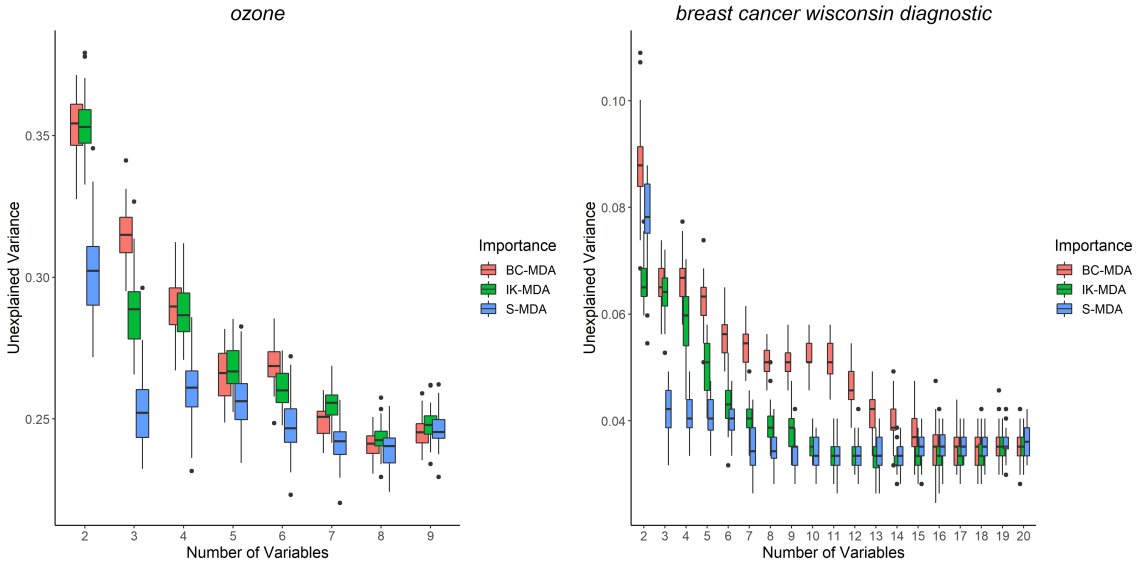

Figure 3: Random forest error versus the number of variables for the "Ozone" and "Breast Cancer Wisconsin Diagnostic" datasets at each step of the RFE, using different importance measures: BC-MDA, IK-MDA, and Sobol-MDA.

The RFE algorithm is illustrated with four real datasets: following (Genuer et al., 2010) we use the "Ozone" data (Dua and Graff, 2017) for a regression example, as well as two other datasets from the UCI repository: "Galaxy" and "Prostate". We also use the "Breast Cancer Wisconsin Diagnosis" data for a binary classification case as in Song et al. (2007). The RFE is run three times, respectively using the BC-MDA, IK-MDA, and the Sobol-MDA as importance measures to iteratively discard the less relevant variable. At each step of the RFE, the explained variance of the forest is retrieved. Following Gregorutti et al. (2017), we do not use the OOB error since it gives optimistically bias results, but use instead a 10-fold cross-validation: the forest and the associated importance measure are computed with 9 folds, and the error is estimated with the 10-th fold. For each dataset, the cross-validation is repeated 40 times to get the result uncertainties, displayed as boxplots in the figures. Figures 3 and 4 highlight that the Sobol-MDA leads to a more efficient variable selection than the BC-MDA and the IK-MDA for the "Ozone", "Breast Cancer Wisconsin Diagnosis", and "Galaxy" datasets. Notice that the IK-MDA performs better than the BC-MDA, as expected from their theoretical counterparts—see Proposition 2. We also insist that the Sobol-MDA can perform a significant improvement over the BC-MDA as soon as inputs are dependent, which is very often the case for real data. On the other hand, the Sobol-MDA also outperforms the IK-MDA

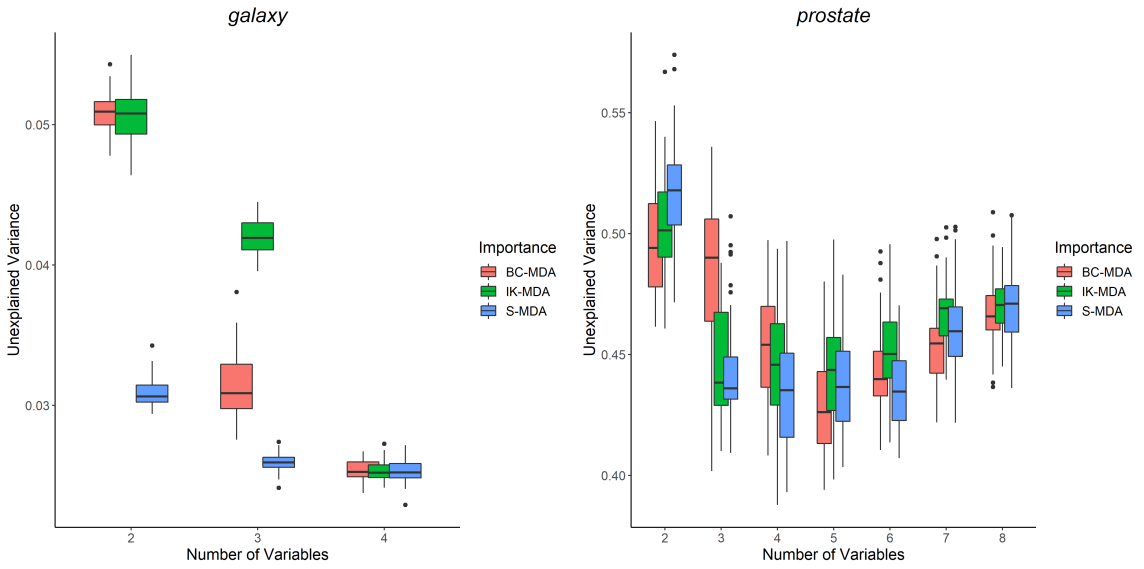

Figure 4: Random forest error versus the number of variables for the "Galaxy" and "Prostate" datasets at each step of the RFE, using different importance measures: BC-MDA, IK-MDA, and Sobol-MDA.

if the data exhibits both interactions and dependence—see Corollaries 1 and 2. The "Prostate" dataset in Figure 4 is an example where the Sobol-MDA does not significantly improve over the original MDA. Indeed, a generalized additive model has an unexplained variance of 0.42 for this dataset, which is slightly better than the 0.45 of random forests, and suggests that the forest does not identify significant interactions. Consequently, the Sobol-MDA and IK-MDA have a very close performance, as expected from Corollary 2. We can also observe that the BC-MDA performs similarly, except for a selection of 3 variables, where the associated error is significantly higher. For the Sobol-MDA, the 3 selected variables are systematically "lcavol", "svi", and "lweight" across all folds of the cross-validation. For the BC-MDA, "lweight" is sometimes replaced by "pgg45" or "lcp", leading to a degraded performance. The reason of such selection is given by Corollary 2: "lcavol", "svi", "pgg45", and "lcp" are strongly correlated, which inflates their BC-MDA, but are quite independent from "lweight".

4 Conclusion

Variable importance is the main approach to analyze the black-box mechanism of random forests, and the MDA is the most widely used importance measure. However, many empirical studies have shown that when input variables are dependent, the MDA fails to detect influential variables. We conducted a theoretical analysis to understand this undesirable behavior. First, a close inspection of the literature and the main random forest software show that different definitions coexist: the Train-Test MDA, the Breiman-Cutler MDA, and the Ishwaran-Kogalur MDA. An asymptotic analysis shows that these different MDA versions do not converge towards the appropriate theoretical quantity when input variables are dependent,

and are thus misleading for both objectives (i) and (ii) of variable importance. Therefore, we propose an augmented MDA algorithm: the Sobol-MDA, which consistently estimates the total Sobol index, i.e. the appropriate theoretical counterpart which tells how much explained variance of the output is lost when a given variable is removed from the model, at an efficient computational cost. We run many experiments to show the good empirical performance of the Sobol-MDA, especially to perform variable selection through the Recursive Feature Elimination algorithm (RFE). An implementation in `R` and `C++` of the Sobol-MDA is available at https://gitlab.com/cbenard/sobol-mda.

References

K. Aas, M. Jullum, and A. Løland. Explaining individual predictions when features are dependent: more accurate approximations to shapley values. *arXiv preprint arXiv:1903.10464*, 2019.

A. Antoniadis, S. Lambert-Lacroix, and J.-M. Poggi. Random forests for global sensitivity analysis: a selective review. *Reliability Engineering & System Safety*, 206:107–312, 2020.

K.J. Archer and R.V. Kimes. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52:2249–2260, 2008.

L. Auret and C. Aldrich. Empirical comparison of tree ensemble variable importance measures. *Chemometrics and Intelligent Laboratory Systems*, 105:157–170, 2011.

N. Benoumechiara. *Treatment of dependency in sensitivity analysis for industrial reliability*. PhD thesis, Sorbonne Université ; EDF R&D, 2019.

A.-L. Boulesteix, S. Janitza, J. Kruppa, and I.R. König. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2:493–507, 2012.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman. Setting up, using, and understanding random forests v3.1. 2003a.

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, Boca Raton, 1984.

I. Covert, S. Lundberg, and S.-I. Lee. Understanding global feature contributions through additive importance measures. *arXiv preprint arXiv:2004.00668*, 2020.

R. Díaz-Uriarte and S.A. De Andres. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics New York, 2001.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31:2225–2236, 2010.

R. Ghanem, D. Higdon, and H. Owhadi. *Handbook of Uncertainty Quantification*. Springer, New York, 2017.

B. Gregorutti. *Random forests and variable selection : analysis of the flight data recorders for aviation safety*. PhD thesis, Université Pierre et Marie Curie - Paris VI, 2015.

B. Gregorutti, B. Michel, and P. Saint-Pierre. Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis*, 90:15–35, 2015.

B. Gregorutti, B. Michel, and P. Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27:659–678, 2017.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:389–422, 2002.

L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer, New York, 2006.

G. Hooker and L. Mentch. Please stop permuting features: an explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019.

B. Iooss and P. Lemaître. *A Review on Global Sensitivity Analysis Methods*, pages 101–122. Springer, Boston, 2015.

B. Iooss and C. Prieur. Shapley effects for sensitivity analysis with correlated inputs: Comparisons with sobol'indices, numerical estimation and applications. *arXiv:1707.01334*, 2017.

H. Ishwaran. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1:519–537, 2007.

H. Ishwaran and U.B. Kogalur. *Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC)*, 2020. URL https://cran.r-project.org/package=randomForestSRC. R package version 2.9.3.

H. Ishwaran, U.B. Kogalur, E.H. Blackstone, and M.S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2:841–860, 2008.

X. Li, Y. Wang, S. Basu, K. Kumbier, and B. Yu. A debiased mdi feature importance measure for random forests. In *Advances in Neural Information Processing Systems*, pages 8049–8059, New York, 2019.

A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2:18–22, 2002.

G. Louppe. Understanding random forests: from theory to practice. *arXiv preprint arXiv:1407.7502*, 2014.

S.M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, New York, 2017.

S.M. Lundberg, G.G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

T. A Mara, S. Tarantola, and P. Annoni. Non-parametric methods for global sensitivity analysis of model output with dependent inputs. *Environmental Modelling & Software*, 72: 173–183, 2015.

N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999, 2006.

K.K. Nicodemus and J.D. Malley. Predictor correlation impacts machine learning algorithms: implications for genomic studies. *Bioinformatics*, 25:1884–1890, 2009.

A.B. Owen. Sobol'indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2:245–251, 2014.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

A. Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145:280–297, 2002.

E. Scornet. Trees, forests, and impurity-based variable importance. *arXiv preprint arXiv:2001.04295*, 2020.

E. Scornet, G. Biau, and J.-P. Vert. Consistency of random forests. *The Annals of Statistics*, 43:1716–1741, 2015.

L.S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2:307–317, 1953.

I.M. Sobol. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1:407–414, 1993.

L. Song, A. Smola, A. Gretton, K.M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th International Conference on Machine Learning*, pages 823–830, San Francisco, 2007. Morgan Kaufmann Publishers.

C. Strobl and A. Zeileis. Danger: High power!–exploring the statistical properties of a test for random forest variable importance. 2008. URL https://www.methpsy.uzh.ch/publications/reports/strobl_zeileis_compstat.pdf.

C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC bioinformatics*, 8:25, 2007.

C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9:307, 2008.

L. Toloşi and T. Lengauer. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, 27:1986–1994, 2011.

S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113:1228–1242, 2018.

M.N. Wright and A. Ziegler. ranger: a fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software*, 77:1–17, 2017.

Z. Zhou and G. Hooker. Unbiased measurement of feature importance in tree-based methods. *arXiv preprint arXiv:1903.05179*, 2019.

R. Zhu, D. Zeng, and M. R. Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110:1770–1784, 2015.