LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Alexander Hapfelmeier, Kurt Ulm

Variable selection with Random Forests for missing data

Technical Report Number 137, 2013
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Variable selection with Random Forests for missing data

Alexander Hapfelmeier
Institut für Medizinische Statistik und Epidemiologie,
Technische Universität München,
Ismaninger Str. 22, 81675 Munich, Germany,
Alexander.Hapfelmeier@tum.de

Kurt Ulm
Institut für Medizinische Statistik und Epidemiologie,
Technische Universität München,
Ismaninger Str. 22, 81675 Munich, Germany

January 15, 2013

Abstract

Variable selection has been suggested for Random Forests to improve their efficiency of data prediction and interpretation. However, its basic element, i.e. variable importance measures, can not be computed straightforward when there is missing data. Therefore an extensive simulation study has been conducted to explore possible solutions, i.e. multiple imputation, complete case analysis and a newly suggested importance measure for several missing data generating processes. The ability to distinguish relevant from non-relevant variables has been investigated for these procedures in combination with two popular variable selection methods. Findings and recommendations: Complete case analysis should not be applied as it lead to inaccurate variable selection and models with the worst prediction accuracy. Multiple imputation is a good means to select variables that would be of relevance in fully observed data. It produced the best prediction accuracy. By contrast, the application of the new importance measure causes a selection of variables that reflects the actual data situation, i.e. that takes the occurrence of missing values into account. It's error was only negligible worse compared to imputation.

Keywords: random forests, variable selection, missing data, multiple imputation, surrogates, complete case analysis

1 Introduction

Random forests (Breiman, 2001) are appreciated in many research fields for notable properties like the ability to implicitly deal with missing values and high dimensional data. Moreover, they are able to uncover complex interactions and to identify informative variables (see Cutler et al., 2007; Lunetta et al., 2004, for works highlighting these properties). The latter is achieved by means of variable importance measures which are often used as a basis for variable selection (see Altmann et al., 2010; Archer and Kimes, 2008; Díaz-Uriarte and Alvarez de Andrés, 2006; Genuer et al., 2010; Jiang et al., 2004; Tang et al., 2009; Yang and Gu, 2009; Rodenburg et al., 2008; Sandri and Zuccolotto, 2006; Svetnik et al., 2004, for corresponding approaches). Though it might be a well suited means to distinguish relevant from non-relevant variables its benefit for prediction is still ambiguous. Thus Yang and Gu (2009); Zhou et al. (2010) claim that the predictive power of a forest may improve through variable selection. By contrast, Altmann et al. (2010); Díaz-Uriarte and Alvarez de Andrés (2006); Svetnik et al. (2004) show that it can also be harmful.

A major issue which is in main focus of this work is how to perform variable selection when there is missing data. Existing approaches base on importance measures that can not be computed straightforward in such a case. Therefore possible solutions are investigated in this work: Complete case analysis is a fast and easy way to deal with missing values though it is known to lead to biased inference when the data is not missing completely at random (Schafer and Graham, 2002; Horton and Kleinman, 2007). An alternative approach is given by multiple imputation by chained equations (MICE; van Buuren et al., 2006; White et al., 2011). It enables the simultaneous imputation of multiple variables without the need to specify a joint distribution of the data (see Schafer, 1997, for joint modeling). Furthermore, its superiority to add hoc methods like complete case and single imputation has been shown by many publications (Janssen et al., 2009, 2010). A third solution is given by a new importance measure introduced by (Hapfelmeier et al., 2012). It closely resembles the well known permutation accuracy importance measure and shares most of its appreciated properties. However, due to an essential adaption, it is able to handle missing values without the need to omit or replace

them before analysis.

In this work, two popular variable selection methods, each representing the conceptual classes of performance-based and test-based approaches (cf. section 4 for a detailed definition), are used in combination with the approaches to handle missing values. An extensive simulation study that involves various missing data generating processes is meant to explore their ability to discriminate relevant from non-relevant variables (cf. Hapfelmeier et al., 2012a, for similar studies about the computation of importance measures, upon which variable selection bases, in such situations). In addition the predictive accuracy of resulting models is investigated for a simulated test dataset, too. Both, regression and classification problems are explored.

2 Missing Data

2.1 Missing Data Generating Processes

In early works about statistical inference with missing values Rubin (1976, 1987) specify three processes that cause missingness:

- Missing completely at random (MCAR): $P(R|\mathbf{X}_{\text{comp}}) = P(R)$
- Missing at random (MAR): $P(R|\mathbf{X}_{\text{comp}}) = P(R|\mathbf{X}_{\text{obs}})$
- Missing not at random (MNAR): $P(R|\mathbf{X}_{\text{comp}}) = P(R|\mathbf{X}_{\text{obs}}, \mathbf{X}_{\text{mis}})$

The binary random variable R indicates whether a value is missing. Its probability distribution is given by $P(R)$. $\mathbf{X}_{\text{comp}}$ denotes the complete variable set that consists of observed values $\mathbf{X}_{\text{obs}}$ and missing ones $\mathbf{X}_{\text{mis}}$: $\mathbf{X}_{\text{comp}} = \{\mathbf{X}_{\text{obs}}, \mathbf{X}_{\text{mis}}\}$. Therefore in a MCAR scheme the probability to observe a missing value is independent of the observed and unobserved data. By contrast for MAR this probability is dependent on the observed data. Finally in MNAR the probability depends on unobserved variables or the missing values themselves.

Most of the methods used in this study are known to be sensitive to the missing data generating processes. Thus, complete case analysis can lead to biased inference when the data is MAR or MNAR (Schafer and Graham, 2002; Horton and Kleinman, 2007). Beyond that, even MCAR may induce a systematic bias in Random Forests based on biased split selections (Strobl et al., 2007). MICE is especially qualified for MAR settings. Janssen et al. (2010) also state that it should be preferred to ad hoc methods like complete case analysis even in MNAR situations. Likewise, He et al. (2009) and White et al. (2011) point out that MICE my well handle MNAR schemes as the imputation model becomes more general and includes more variables to make MAR plausible. For these reasons one MCAR, four MAR and one MNAR scheme are investigated in the following simulation study.

2.2 Multivariate Imputation by Chained Equations

Multiple imputation (MI) (Rubin, 1987, 1996) is an attempt to solve the problem that single imputation leads to underestimation of variance (Harel and Zhou, 2007). However, in case of more than one variable with missing values the approach needs to be able to simultaneously impute multiple variables. Therefore, MICE, also known as imputation by fully conditional specification (FCS; van Buuren et al., 2006; van Buuren, 2007; van Buuren and Groothuis-Oudshoorn, 2010; White et al., 2011), cycles through incomplete variables to iteratively update imputed values until convergence. It also enables a flexible specification of predictive models without the need to specify a joint distribution of the data. A repetition of the process based on differing random seeds and initial values produces multiple imputed data sets.

3 Methods

3.1 Random Forests

Recursive partitioning is best described by the example of the CART algorithm (Breiman et al., 1984). Corresponding trees are made up by sequential binary splits of the data that are supposed to produce subsets which are as homogeneous as possible in terms of the outcome. Breiman (1996) further enhanced the method by "bagging" (bootstrap aggregation) and Random Forests (Breiman, 2001; Breiman and Cutler, 2008). For that purpose several trees are fit to bootstrapped or subsampled data (bagging) and splits are performed in random selections of variables (Random Forests). This way a more diverse set of tree models contributes to the joint prediction which leads to an improved performance compared to single trees.

Predictions are found by averaged values or majority votes of each single tree in a Random Forest. Likewise, the so called 'out of bag' (OOB) samples, i.e. observations not used to fit the respective trees, can be used for an unbiased estimate of a Random Forests error, viz. the OOB-error. However, when there are missing values surrogate splits need to be employed. They mimic the initial split of data as they try to archive the same partitioning of complete observations. When several surrogate splits are computed they can be ranked according to their similarity to the initial split. Observations that contain more than a single missing value are processed along this ranking until a decision is found.

The CART and the C4.5 algorithms (Quinlan, 1993), and consequently all Random Forest algorithms based on the same construction principles, are prone to biased variable selection (cf. Breiman et al., 1984; Strobl et al., 2007; White and Liu, 1994; Kim and Loh, 2001; Dobra and Gehrke, 2001; Hothorn et al., 2006). An alternative approach presented by Hothorn et al. (2006)

follows the same rationale as Breiman's original approach yet guarantees unbiased variable selection and variable importance measures when combined with subsampling (as opposed to bootstrap sampling; cf. Strobl et al., 2007). Therefore, it will be used in the following analyses.

3.2 A new variable importance measure for missing data

The permutation accuracy importance measure is a popular means to assess a variables relevance in Random Forests. It is computed by the difference of a trees prediction accuracy before and after random permutation of a predictor variable. If the latter is related to the response and further predictors the accuracy is supposed to drop and the variable is termed to be of relevance. However, the expression 'relevance' is ambiguous. In terms of the original, unconditional importance measure it incorporates informative variables and variables that are (cor-)related to informative ones. By contrast, a conditional version that more closely resembles the behavior of partial correlation or regression coefficients was introduced by Strobl et al. (2008). Both kinds of measures can be of specific value depending on the research question (Nicodemus et al., 2010; Altmann et al., 2010). In this work the unconditional version is preferred for its sensitivity to relations between variables that are supposed to be uncovered.

A general limitation is that the permutation importance measure can not be computed straightforward when there are missing values. It is unclear how to appropriately handle surrogate splits that contribute to the computation of the accuracy in the permutation scheme. A solution to this problem was introduced earlier by a new importance measure (Hapfelmeier et al., 2012). It is closely related to existing methodology, and therefore retains appreciated properties, yet differs in one substantial aspect: The null hypothesis of no relation to the response and other predictors is simulated as observations are randomly send to the daughter nodes when a parent node k is split in a variable X that is of interest. In doing so, the respective probability, e.g. to be sent the left way, is given by the relative frequency $\hat{p}_k$ of observations that initially went the same direction. The algorithm to compute the new importance measure is now given by:

1. Compute the OOB accuracy of a tree.
2. Randomly assign each observation with $\hat{p}_k$ to the child nodes if the parent node k is split in X.
3. Recompute the OOB accuracy of the tree (following step 2).
4. Compute the difference between the original and recomputed OOB accuracy.
5. Repeat step 1 to 4 for each tree and use the average difference over all trees as the overall importance score.

This procedure circumvents the necessity to directly process missing values and solves any problems associated with permutation. It is used in the following simulation studies.

4 Variable selection

Many general ideas about variable selection (see Guyon and Elisseeff, 2003, for an extensive listing) can be rediscovered for Random Forests. Thus, an evaluation of predictive performance is frequently used to determine a best performing model from a set of Random Forests (Díaz-Uriarte and Alvarez de Andrés, 2006; Genuer et al., 2010; Jiang et al., 2004; Svetnik et al., 2004). Therefore, the latter are usually constructed along a sequence of predictor variables that is determined by importance measures. Besides apparent analogies there are also many diversities considering the method(s) used to assess prediction accuracy, the application of sampling methods, the kind of importance measure, the (re-)calculation of variable importances, the sequence of predictor variables, just to name some of them (Hapfelmeier and Ulm, 2013, give a detailed discussion of approaches). As all of these methods incorporate sampling methods in some way they are classified as 'performance-based approaches' in the following.

A very popular representative of this class is given by the approach of Díaz-Uriarte and Alvarez de Andrés (2006). In an initial step it computes the importance measures of variables based on the entire data. In subsequent steps the least important variables are sequentially rejected and the OOB-errors of corresponding Random Forests are recorded. The final model is chosen to be the one with an error within a range of u standard errors to the best performing one. Setting $u = 1$ equals the 'one-standard-error' rule ('1 s.e.' rule) known from works about classification trees (Breiman et al., 1984; Hastie et al., 2009). This approach will be used for variable selection in the following analyses.

A second class of variable selection methods is 'test-based' as it employs a permutation test framework to estimate the significance of variable importances (see Efron and Tibshirani, 1994; Good, 2005, 2000, for further insight in principles). The basic concept is to recompute a Random Forest and its importance measures after a predictor variable was permuted. This procedure is repeated several times to assess the empirical distribution of importance measures under the null-hypothesis of independence between the predictor variable and the response (and the remaining variable space; see Hapfelmeier and Ulm, 2013, for corresponding discussions. Just like the permutation importance measure this procedure supports the identifi-

cation of variables that are (cor-)related to informative variables.). The likelihood of the original importance measures within these empirical distributions can be used to compute p-values. Finally, variables with a p-value beyond a certain threshold, e.g. ≤ 0.05 are selected.

Some approaches like those of Altmann et al. (2010); Rodenburg et al. (2008); Tang et al. (2009); Wang et al. (2010) suggest a simultaneous permutation of entire groups of variables. However, the significance of single importance measures can not be validly determined this way. A solution which is just as fundamental as it might sound simple is to permute a single variable when it comes to the assessment of its significance; this is the second approach used for variable selection in the following analyses. The proper application of such methods has been presented and compared against established variable selection methods earlier (Hapfelmeier and Ulm, 2013). A major advantage of test-based approaches is that they can be used to control for the test-wise error rate (TWER = probability of a null-hypothesis to be falsely rejected). Nevertheless, the application of correction methods like the Bonferroni-Adjustment or the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) can be used to control for the family-wise error rate (FWER) and the false discovery rate (FDR), respectively.

As there are many new and innovative proposals the preceding classification does not claim to cover all of the ongoing developments: For example Sandri and Zuccolotto (2006) suggest the computation of importance measures on a four-dimensional scale. Yang and Gu (2009) and Schwarz et al. (2007) try to handle the high-dimensional data of genome wide association studies (GWAS) as Random Forests are fit to changing subsets of SNPs from which global importances are determined. However, this work focuses on the examination of two representatives of the 'performance-based' and 'test-based' classes, i.e. the approaches of Díaz-Uriarte and Alvarez de Andrés (2006) and Hapfelmeier and Ulm (2013).

5 Simulation study

An extensive simulation study was set up to explore which of complete case analysis, multiple imputation by MICE and the new importance measure is most capable to support variable selection. This quality is compared between two approaches meant to distinguish relevant from non-relevant variables; the latter are defined to be non-informative and not correlated to any informative variables. Two additional investigations will focus the predictive accuracy of Random Forests in a simulated test dataset and the ability of selection methods to control the TWER. Factors like the amount of missing values, correlation schemes, variable strength and different missing data generating processes are of major interest as they potentially influence variable selection. A detailed explanation of the setup is given in the following.

- *Influence of predictor variables*

  The simulated data contained both, a classification and a regression problem. Therefore, a categorical (binary) and a continuous response were created in dependence of six variables with coefficients β:

  $$\beta = (1, 1, 0, 1, 1, 0)^\top.$$

  Repeated values for β make it possible to compare selection frequencies of variables which are, by construction, equally important but show different correlations and contain different amounts of missing values. In addition, the non-influential variables with $\beta = 0$ help to investigate possible undesired effects, serve as a baseline and are used to check for the ability to control the TWER.

- *Data generating models*

  A continuous response was modeled by means of a linear model:

  $$y = \mathbf{x}^\top \beta + \epsilon \text{ with } \epsilon \sim N(0, 1).$$

  The binary response was drawn from a Bernoulli distribution $B(1, \pi)$ where π was assessed by means of a logistic model

  $$\pi = P(Y = 1|\mathbf{X} = \mathbf{x}) = \frac{e^{\mathbf{x}^\top \beta}}{1 + e^{\mathbf{x}^\top \beta}}.$$

  The variable set $\mathbf{X}$ itself contained 100 observations drawn from a multivariate normal distribution with mean vector $\vec{\mu} = 0$ and covariance matrix Σ:

- *Correlation*

  $$\Sigma = \begin{pmatrix} 1 & 0.3 & 0.3 & 0 & 0 & 0 \\ 0.3 & 1 & 0.3 & 0 & 0 & 0 \\ 0.3 & 0.3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

  As the variances of each variable are chosen to be 1, the covariance equals the correlation in this special case. The structure of Σ reveals that there are two blocks of three correlated and three uncorrelated variables.

- *Missing values*

  Several missing data generating processes that follow MCAR, MAR and MNAR schemes were employed. For each, a given fraction $m \in \{0.0, 0.1, 0.2, 0.3\}$ of values is set missing for variables X_2 and X_5. Therefore, the average fraction

Table 1: List of the variables containing missing values and variables determining the probability of missing values.

contains missing values (MCAR, MAR & MNAR)	determines missing values (MAR)	(MNAR)
X_2	X_1	X_2
X_5	X_4	X_5

of observations that contain at least one missing value is $1-(1-\%_{\text{missing}})^{\text{n}_{\text{variables}}} = 1-(1-0.3)^2 = 51\%$ in case of $m = 0.3$. This is an amount not unlikely to be observed in real life data which makes m span a wide range of possible scenarios.

In MAR the probability for missing values can be explained by observed information; e.g. by variables contained in the data. In MNAR, however, it is unknown; e.g. as it may depend on the unobserved values themselves. Accordingly, the probability of a missing value in variables X_2 and X_5 was determined by X_1 and X_4 in MAR and by their own values in MNAR. Table 1 illustrates the corresponding relations.

The schemes for producing missing values are:

- MCAR: Values are randomly replaced by missing values.
- MAR(rank): The probability of a value to be replaced by a missing value rises with the rank the same observation has in the determining variable.
- MAR(median): The probability of a value to be replaced by a missing value is nine times higher for observations whose value in the determining variable is located above the corresponding median.
- MAR(upper): Those observations with the highest values of the determining variable are replaced by missing values.
- MAR(margins): Those observations with the highest and lowest values of the determining variable are replaced by missing values.
- MNAR(upper): The highest values of a variable are set missing.

An independent test dataset of 5000 observations was constructed the same way, though it did not contain missing values, for an evaluation of predictive accuracy. The latter was assessed by the mean squared error (MSE) which equals the misclassification error rate (MER) in classification problems.

In summary, there were 2 variable selection methods and 2 response types investigated for 6 processes to generate and 3 procedures to handle 4 different fractions of missing values. This sums up to as much as 288 simulation settings. Each of them was repeated 1000 times.

5.1 Implementation

The `R` system for statistical computing (R Development Core Team, 2011) was used to perform the simulation study. The package `party` (Hothorn et al., 2008) provides unbiased Random Forests based on conditional inference by the function `cforest()`. Its settings were chosen to fit $ntree = 100$ trees for each forest. Split nodes were determined from $mtry = 3$ randomly selected variables. Depending on the amount of variables left after each selection step the number of surrogate splits was chosen to be maximally $maxsurrogate = 3$. There were no restrictions on the significance of a split ($mincriterion = 0$) and trees were grown until terminal nodes contained less than $minsplit = 20$ observations. Child nodes had to contain at least $minbucket = 7$ observations. MICE is given by the function `mice()` of the package `mice` (van Buuren and Groothuis-Oudshoorn, 2010). It was used to produce five imputed datasets. A normal linear model was applied to impute continuous variables, a logistic regression for binary variables and a polytomous regression for variables with more than two categories; defaultMethod = c("norm", "logreg", "polyreg"). Each variable was part of the imputation models. The fraction of imputed data is approximately $1-(1-m)^3$, $m \in \{0.0, 0.1, 0.2, 0.3\}$.

Variable selection methods were implemented following the descriptions of section 4. For the performance-based approach the rejection steps were limited to one variable a time. In addition, it was empowered to select no variables at all. Therefore the prediction of a null-model, given by the majority vote of classes (for binary outcomes) or the mean outcome (for continuous outcomes) in the training data, was used to compare its MSE against models fit to competing variable sets. The test-based approach made use of 100 permutation runs. According p-values were assessed for one-sided tests as only values on the right margin of the empirical distribution of importance measures (i.e. high values) provide evidence against the null-hypothesis of a non-relevant variable. The significance level was set to 5%.

6 Results

The following discussion presents results for the classification problem. Similar findings for the regression problem are given by Figure 5 in the appendix.

Variable selection frequencies displayed in Figure 1 stress that the test-based approach performs superior to the performance-based approach. The former selects relevant variables, including variable 3 which is non informative, yet correlated to informative variables, more often, independent of the amount of missing values. With reference to the non-relevant variable 6, both approaches control for the TWER. As

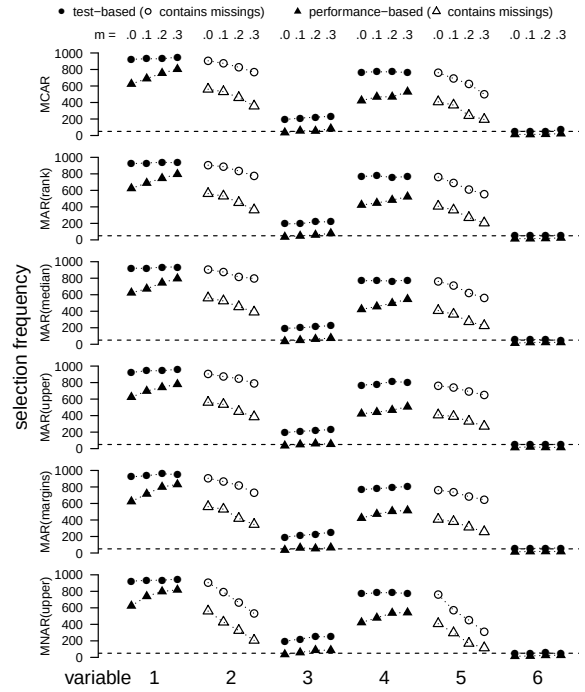


Figure 1: Variable selection frequencies observed for the new importance measure. The horizontal dashed line illustrates a TWER of 5% (m = % of missing values in X_2 and X_5).

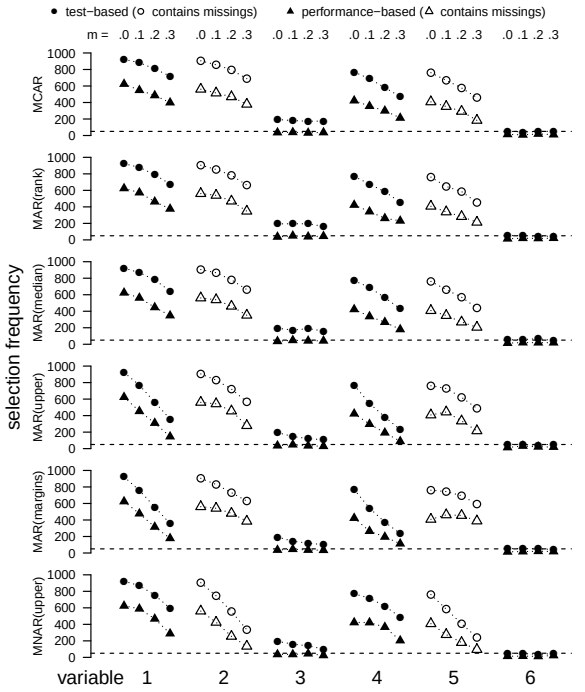


Figure 2: Variable selection frequencies observed for the complete case analysis. The horizontal dashed line illustrates a TWER of 5% (m = % of missing values in X_2 and X_5).

expected, the selection frequencies of variables 2 and 5 drop as they contain a rising amount of missing values. Meanwhile variable 1 and 4 are chosen more frequently by the performance-based approach. This can be seen as the attempt to replace variables with missing information by other predictors (see Hapfelmeier et al., 2012, for corresponding investigations). The same effect can not be observed for the test-based approach which shows higher and rather stable selection frequencies for these fully observed variables. There are minor differences between variables 1 and 4, though they are of the same strength. This is due to the fact that unconditional permutation importance measures, which underly the applied selection methods, rate the relevance of correlated variables higher than for uncorrelated ones (Strobl et al., 2008). In conclusion, there are no apparent differences between the missing data generating processes. The application of the new importance measure for variable selection can be recommended whenever the objective is to describe the data situation at hand; i.e. under consideration of the relevance a variable can take with all its missing values.

In the complete case analysis, illustrated by Figure 2, the performance-based approach is again outperformed by the test-based approach; while both control for the TWER. However, there are some general findings that question the quality of complete case analysis. Thus, selection frequencies of the informative variables 1 and 4 drop with a rising fraction of missing values in variables 2 and 5. One might argue that this is caused by the general loss of information induced by complete case analysis. However, in some cases (e.g. MAR(margins)) this effect is carried to extremes as variables 1 and 4 are even less frequently selected than variables 2 and 5; while the latter are the ones that actually lost part of their information. There is no rational justification for this undesirable property which is present for any missing data generating process. Consequently, complete case analysis is not recommended for application as selection methods might not be capable to detect variables of true relevance.

Results for the application of multiple imputation are given by Figure 3. Again, they reflect the superiority of the test-based approach to the performance-based approach; while both of them control the TWER. Furthermore, imputation leads to rather stable selection frequencies of variables, independent of the amount of missing values. However, a slight decrease can still be observed for variables 2 and 5 as they loose information. This holds for each missing data generating process except for MNAR(upper). It is interesting to note that results for the latter resemble those of Figure 1. Thus, the occurrence of missing values and the associated loss of information seems to directly affect selection frequencies when missing values can not be appropriately imputed. Nevertheless, multiple imputation appears to be a well suited means to select variables according to the relevance they would have if the data was fully observed.

Prediction errors observed for the independent test

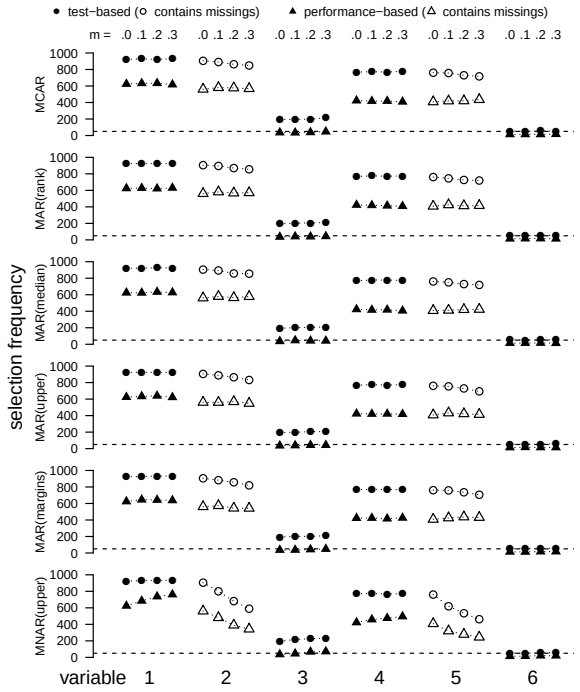


Figure 3: Variable selection frequencies observed for the imputed data. The horizontal dashed line illustrates a TWER of 5% ($m = \%$ of missing values in X_2 and X_5).

sample are displayed by Figure 4. They confirm the superiority of the test-based approach to the performance-based approach in terms of predictive accuracy. This holds independent of the approach to handle missing values, the amount of missing values and the process to generate missing values. The lowest MSE, which is almost stable for any fraction of missing values, was found for models fit to imputed data. Variable selection that bases on the new importance measure produced models that performed only slightly worse. For this procedure the error increased with an increasing number of missing values. This property intensifies for the complete case analysis which clearly produced the worst results for increased fractions of missing values. Similar findings about the predictive accuracy of Random Forests when there is missing data have been published by Rieger et al. (2010); Hapfelmeier et al. (2012b).

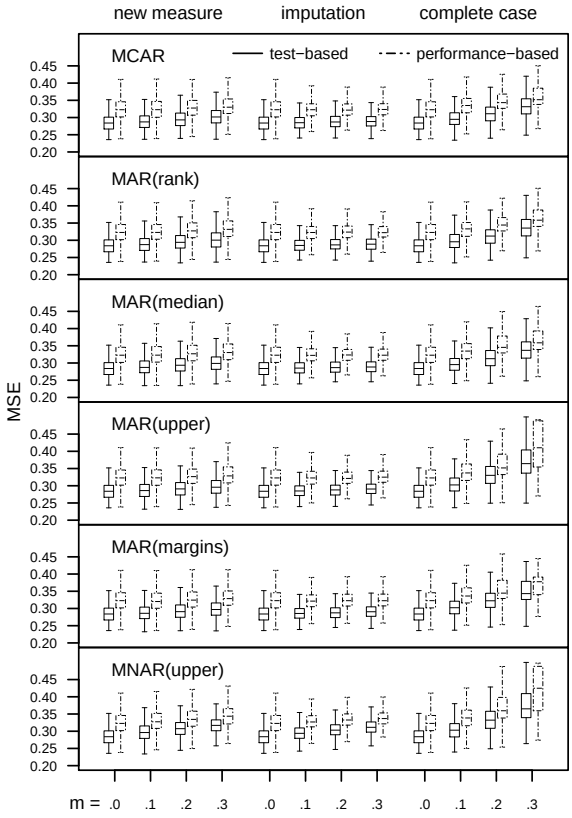


Figure 4: MSE observed for the independent test sample. Outliers are not displayed for clarity ($m = \%$ of missing values in X_2 and X_5).

7 Conclusion

Variable selection with Random Forests is guided by importance measures which are used to rate a variables relevance for prediction. There are several approaches like a new kind of importance measure, complete case analysis and multiple imputation that enables its application when the data contains missing values. An extensive simulation study has been conducted to investigate the ability of such approaches to

discriminate relevant from non-relevant variables under several missing data generating processes. Complete case analysis appeared to provide inaccurate variable selection as the occurrence of missing values inappropriately penalized the selection of informative and fully observed variables. Accordingly, it lead to models that showed the worst prediction accuracies. Selection methods that based on the application of a new importance measure were much more able to reflect the data situation at hand. Thus, fully observed variables were selected constantly and considerably more often than those with missing values. The prediction accuracy of corresponding Random Forests was much higher than for complete case analysis. Multiple imputation also showed constant selection frequencies, that could be called most accurate if the objective was to rate the relevance a variable would have in fully observed data. For any simulation setting and any approach to handle missing values the test-based variable selection method performed superior to the performance-based approach.

There is a clear recommendation for the application of approaches: One should not use complete case analysis because of inaccurate selection properties. Approaches that base on the new kind of importance measure should be used if one is interested in a selection of variables that reflects their relevance under consideration of the given information. By contrast, imputation methods are best used for the selection of variables that would be of relevance in the hypothetical scenario of fully observed data.

References

Altmann, A., L. Tolosi, O. Sander, and T. Lengauer (2010, May). Permutation importance: a corrected feature importance measure. *Bioinformatics 26*(10), 1340–1347.

Archer, K. and R. Kimes (2008, January). Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis 52*(4), 2249–2260.

Benjamini, Y. and Y. Hochberg (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological) 57*(1), 289–300.

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

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

Breiman, L. and A. Cutler (2008). *Random forests*. `http://www.stat.berkeley.edu/users/breiman/RandomForests/cc\_home.htm`. (accessed 09.01.2013).

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984, January). *Classification and Regression Trees* (1 ed.). Chapman & Hall/CRC.

Cutler, D. R., T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler (2007). Random forests for classification in ecology. *Ecology 88*(11), 2783–2792.

Díaz-Uriarte, R. and S. Alvarez de Andrés (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics 7*(1), 3.

Dobra, A. and J. Gehrke (2001). Bias correction in classification tree construction. In C. E. Brodley and A. P. Danyluk (Eds.), *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA*, pp. 90–97. Morgan Kaufmann.

Efron, B. and R. Tibshirani (1994, May). *An Introduction to the Bootstrap (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)* (1 ed.). Chapman and Hall/CRC.

Genuer, R., J.-M. Poggi, and C. Tuleau-Malot (2010). Variable selection using random forests. *Pattern Recognition Letters 31*(14), 2225 – 2236.

Good, P. (2000, January). *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses* (2nd ed.). Springer.

Good, P. (2005). *Introduction to Statistics through Resampling Methods and R/S-Plus*. New York: Wiley-Interscience.

Guyon, I. and A. Elisseeff (2003, March). An introduction to variable and feature selection. *J. Mach. Learn. Res. 3*, 1157–1182.

Hapfelmeier, A., T. Hothorn, and K. Ulm (2012a). Random forest variable importance with missing data. Technical report.

Hapfelmeier, A., T. Hothorn, and K. Ulm (2012b). Recursive partitioning on incomplete data using surrogate decisions and multiple imputation. *Computational Statistics & Data Analysis 56*(6), 1552 – 1565.

Hapfelmeier, A., T. Hothorn, K. Ulm, and C. Strobl (2012). A new variable importance measure for random forests with missing data. *Statistics and Computing*, 1–14.

Hapfelmeier, A. and K. Ulm (2013). A new variable selection approach using random forests. *Computational Statistics & Data Analysis 60*(0), 50 – 69.

Harel, O. and X.-H. Zhou (2007, December). Multiple imputation: review of theory, implementation and software. *Statistics in Medicine 26*(16), 3057–3077.

Hastie, T., R. Tibshirani, and J. Friedman (2009, February). *The Elements of Statistical learning* (Corrected ed.). Springer.

He, Y., A. M. Zaslavsky, M. B. Landrum, D. P. Harrington, and P. Catalano (2009). Multiple imputation in a large-scale complex survey: a practical guide. *Statistical Methods in Medical Research*.

Horton, N. J. and K. P. Kleinman (2007, February). Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models. *The American Statistician 61*(1), 79–90.

Hothorn, T., K. Hornik, C. Strobl, and A. Zeileis (2008). *party: A laboratory for recursive part(y)itioning.* R package version 0.9-9993.

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

Janssen, K. J., A. R. Donders, F. E. Harrell, Y. Vergouwe, Q. Chen, D. E. Grobbee, and K. G. Moons (2010, July). Missing covariate data in medical research: to impute is better than to ignore. *Journal of clinical epidemiology 63*(7), 721–727.

Janssen, K. J., Y. Vergouwe, A. R. Donders, F. E. Harrell, Q. Chen, D. E. Grobbee, and K. G. Moons (2009, May). Dealing with missing predictor values when applying clinical prediction models. *Clinical chemistry 55*(5), 994–1001.

Jiang, H., Y. Deng, H.-S. Chen, L. Tao, Q. Sha, J. Chen, C.-J. Tsai, and S. Zhang (2004). Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics 5*(1), 81.

Kim, H. and W. Loh (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association 96*, 589–604.

Lunetta, K., B. L. Hayward, J. Segal, and P. Van Eerdewegh (2004). Screening large-scale association study data: exploiting interactions using random forests. *BMC Genetics 5*(1).

Nicodemus, K., J. Malley, C. Strobl, and A. Ziegler (2010, February). The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics 11*(1), 110.

Quinlan, J. R. (1993, January). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)* (1 ed.). Morgan Kaufmann.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rieger, A., T. Hothorn, and C. Strobl (2010). Random forests with missing values in the covariates.

Rodenburg, W., A. G. Heidema, J. M. A. Boer, I. M. J. Bovee-Oudenhoven, E. J. M. Feskens, E. C. M. Mariman, and J. Keijer (2008). A framework to identify physiological responses in microarray-based gene expression studies: selection and interpretation of biologically relevant genes. *Physiological Genomics 33*(1), 78–90.

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

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. J. Wiley & Sons, New York.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association 91*(434), 473–489.

Sandri, M. and P. Zuccolotto (2006). Variable selection using random forests. In S. Zani, A. Cerioli, M. Riani, and M. Vichi (Eds.), *Data Analysis, Classification and the Forward Search*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 263–270. Springer Berlin Heidelberg. 10.1007/3-540-35978-8_30.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Chapman & Hall.

Schafer, J. L. and J. W. Graham (2002, June). Missing data: our view of the state of the art. *Psychol Methods 7*(2), 147–177.

Schwarz, D., S. Szymczak, A. Ziegler, and I. Konig (2007). Picking single-nucleotide polymorphisms in forests. *BMC Proceedings 1*(Suppl 1), S59.

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

Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis (2008, July). Conditional variable importance for random forests. *BMC Bioinformatics 9*(1), 307+.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn (2007, January). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics 8*(1), 25+.

Svetnik, V., A. Liaw, C. Tong, and T. Wang (2004). Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. In F. Roli, J. Kittler, and T. Windeatt (Eds.), *Multiple Classifier Systems*, Volume 3077 of *Lecture Notes in Computer Science*, pp. 334–343. Springer Berlin / Heidelberg. 10.1007/978-3-540-25966-4_33.

Tang, R., J. Sinnwell, J. Li, D. Rider, M. de Andrade, and J. Biernacka (2009). Identification of genes and haplotypes that predict rheumatoid arthritis using random forests. *BMC Proceedings 3*(Suppl 7), S68.

van Buuren, S. (2007). Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical Methods in Medical Research 16*(3), 219–242.

van Buuren, S., J. P. L. Brand, C. G. M. Groothuis-Oudshoorn, and D. B. Rubin (2006, December). Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation 76*(12), 1049–1064.

van Buuren, S. and K. Groothuis-Oudshoorn (2010). Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software in press*, 01–68.

Wang, M., X. Chen, and H. Zhang (2010). Maximal conditional chi-square importance in random forests. *Bioinformatics 26*(6), 831–837.

White, A. and W. Liu (1994). Bias in information based measures in decision tree induction. *Machine Learning 15*(3), 321–329.

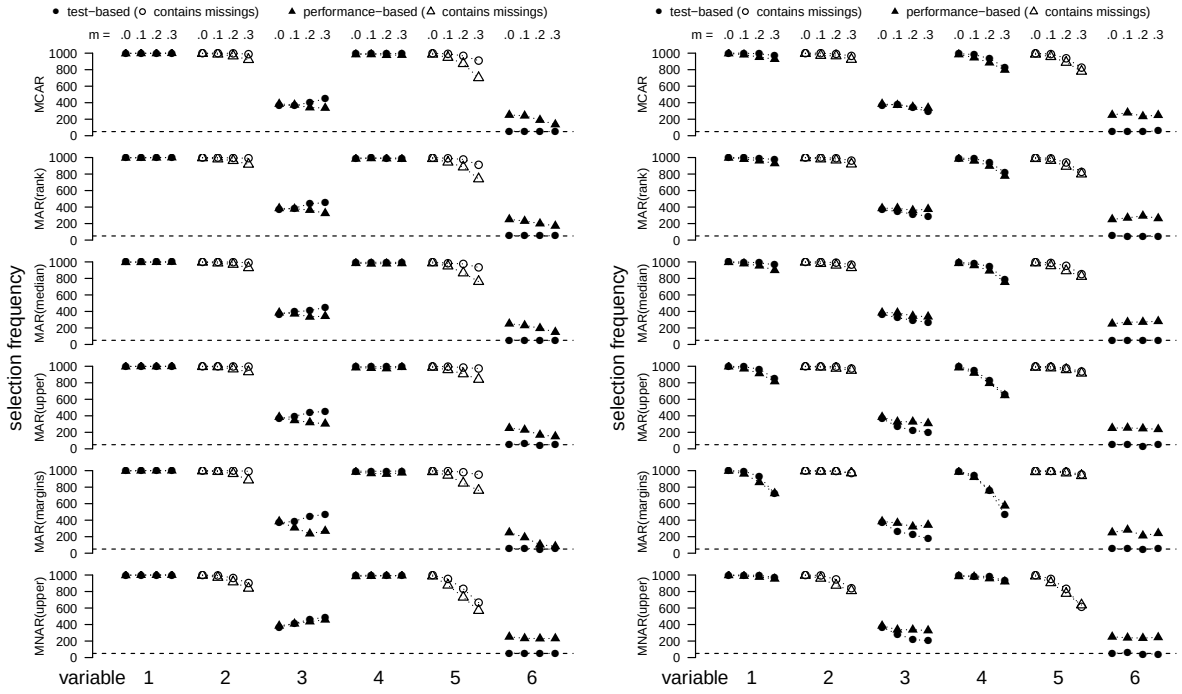
White, I. R., P. Royston, and A. M. Wood (2011). Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine 30*(4), 377–399.

Yang, W. and C. C. Gu (2009). Selection of important variables by statistical learning in genome-wide association analysis. *BMC Proceedings 3*(Suppl 7), S70.

Zhou, Q., W. Hong, L. Luo, and F. Yang (2010). Gene selection using random forest and proximity differences criterion on dna microarray data. *JCIT 5*(6), 161–170.

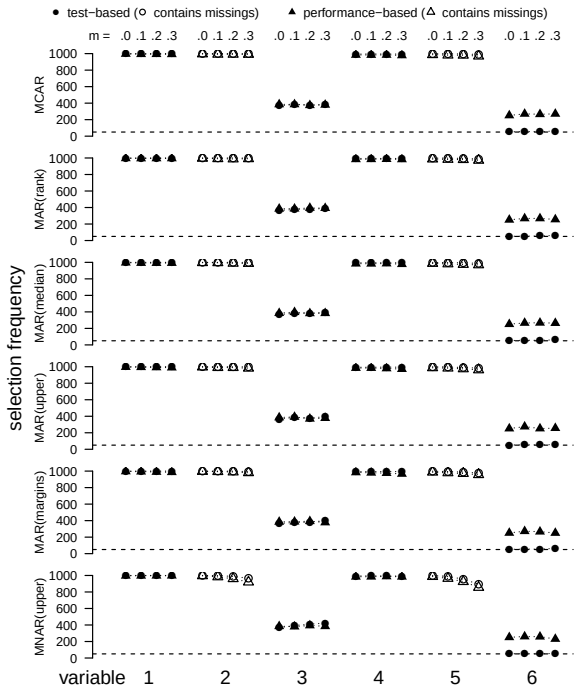
A Supplementary Material

Results for the regression problem are presented in Figure 5. They underline the findings for the classification problem. However, it has to be pointed out that the performance-based variable selection approach has originally been suggested for classification problems. If the 1 s.e. rule was adapted according to a suggestion of Breiman et al. (1984) it could be used for regression problems, too. Yet, in order to stick close to the original definition the 0 s.e. rule was executed here. As a consequence, differences between variable selection approaches were less pronounced. For the performance-based approach this came at the cost of an increased TWER.

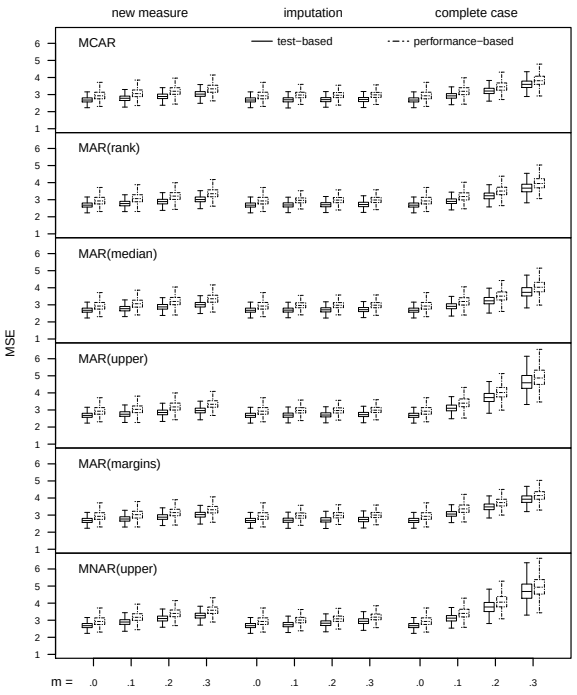


(a) new importance measure

(b) complete case analysis



(c) multiple imputation



(d) MSE

Figure 5: Variable selection frequencies and MSE observed for the regression problem ($m = \%$ of missing values in X_2 and X_5).

B R-Code

```
# load required packages
library("party"); attach(asNamespace("party")); library(mvtnorm); library(mice)

# Function to perform test based variable selection
test.based <- function(Y, X, nperm = 100, ntree = 100, alpha = 0.05) {
  # Y: response, X: predictors, nperm: number of permutations,
  # ntree: number of trees to be fit, alpha: significance level
  mtry <- ceiling(sqrt(ncol(X))) # mtry defaults to the square root of predictors
  dat <- cbind(Y, X); names(dat) <- c("response", paste("V", 1:ncol(X), sep = ""))
  # build an initial forest and record its variable importances
  forest <- cforest(response ~ ., data = dat, controls = cforest_unbiased(mtry = mtry, ntree = ntree,
                    maxsurrogate = min(3, ncol(X) - 1)))
  obs.varimp <- varimp(forest)
  selection <- names(obs.varimp)
  # create a matrix that will contain the variable importances after permutation
  perm.mat <- matrix(NA, ncol = length(selection), nrow = nperm, dimnames = list(1:nperm, selection))
  # each variable is permuted, a new tree is build and the variable importance is recorded
  for (j in selection) {
    perm.dat <- dat
    for (i in 1:nperm) {
      perm.dat[, j] <- sample(perm.dat[, j])
      perm.forest <- cforest(response ~ ., data = perm.dat, controls = cforest_unbiased(mtry = mtry,
                                                ntree = ntree, maxsurrogate = min(3, ncol(X) - 1)))
      perm.mat[i, j] <- varimp(perm.forest)[j]
    }}
  # compute p-values
  p.vals <- sapply(selection, function(x) sum(perm.mat[, x] >= obs.varimp[x]) / nperm)
  # variables with a significant p-value are selected for the final forest
  if (any(p.vals < alpha)) {
   selection <- names(p.vals)[which(p.vals < alpha)]
   mtry <- ceiling(sqrt(length(selection)))
   forest <- cforest(as.formula(paste("response", paste(selection, collapse = " + "), sep = " ~ ")),
                     data = dat, controls = cforest_unbiased(mtry = mtry, ntree = ntree,
                     maxsurrogate = min(3, length(selection) - 1)))
  }
  if (!any(p.vals < alpha)) {selection <- c(); forest <- c()}
  # the out of bag error is computed
  oob.error <- ifelse(length(selection) != 0, mean((as.numeric(as.character(Y)) -
                       as.numeric(as.character(predict(forest, OOB = T))))^2),
                       mean((as.numeric(as.character(Y)) - ifelse(all(Y %in% 0:1),
                       round(mean(as.numeric(as.character(Y)))), mean(Y)))^2))
return(list("selection" = selection, "forest" = forest, "oob.error" = oob.error))
}

# Function to perform performance based variable selection
performance <- function(Y, X, ntree = 100) {
  # Y: response, X: predictors, ntree: number of trees to be fit
  mtry <- ceiling(sqrt(ncol(X))) # mtry defaults to the square root of predictors
  dat <- cbind(Y, X); names(dat) <- c("response", paste("V", 1:ncol(X), sep = ""))
  # build an initial forest and record its variable importances
  forest <- cforest(response ~ ., data = dat, controls = cforest_unbiased(mtry = mtry, ntree = ntree,
                    maxsurrogate = min(3, ncol(X) - 1)))
  selections <- list() # list that contains the names of variables at each size of the forest
  selections[[ncol(X)]] <- names(sort(varimp(forest), decreasing = T))
  errors <- c() # vector of errors at different sizes of the forest
  for (i in ncol(X):1) { # backward elimination of predictors
  mtry <- ceiling(sqrt(i))
  forest <- cforest(as.formula(paste("response", paste(selections[[i]], collapse = " + "), sep = " ~ ")),
                    data = dat, controls = cforest_unbiased(mtry = mtry, ntree = ntree,
                    maxsurrogate = min(3, i - 1)))
  errors[i] <- mean((as.numeric(as.character(Y)) - as.numeric(as.character(predict(forest, OOB = T))))^2)
  if (i > 1) selections[[i - 1]] <- selections[[i]][-i]
  }
  errors <- c(mean((as.numeric(as.character(Y)) - ifelse(all(Y %in% 0:1),
              round(mean(as.numeric(as.character(Y)))), mean(Y)))^2), errors)
  # optimum tree size according to the 1 s.e. or 0 s.e. rule
  optimum.number <- which(errors <= min(errors) + 1 * ifelse(all(Y %in% 0:1),
                          sqrt(min(errors) * (1 - min(errors)) / nrow(X)), 0))[1]
  if (optimum.number == 1) {forest <- c(); selection <- c()}
  if (optimum.number != 1) {
  selection <- selections[[optimum.number - 1]]
  forest <- cforest(as.formula(paste("response", paste(selection, collapse = " + "), sep = " ~ ")),
                    data = dat, controls = cforest_unbiased(mtry = mtry, ntree = ntree,
                    maxsurrogate = min(3, length(selection) - 1)))
  }
  oob.error <- errors[optimum.number] # the out of bag error is computed
  return(list("selection" = selection, "forest" = forest, "oob.error" = oob.error))
}

# function to create the data
create.dat <- function(coefs = c(1,1,0,1,1,0), n = 100, sigma = NULL, regression = F, error = 0.5) {
  if (is.null(sigma)) sigma <- diag(length(coefs))
  if (length(coefs) != nrow(sigma)) stop("dimension of coefs and sigma are not allowed to differ")
  dat <- rmvnorm(n, sigma = sigma)
```

```
  x.beta <- dat %*% coefs
  dat <- as.data.frame(dat)
  if (regression == T) dat$response <- x.beta + rnorm(n, 0, error)
  else dat$response <- as.factor(rbinom(n, 1, exp(x.beta) / (1 + exp(x.beta))))
  return(dat)
  }

# function used for the simulation analysis
myfunc <- function(dat.test, sigma) {
  # dat.test: test data frame, sigma: covariance matrix used to build the training data
  dat.train <- create.dat(sigma = sigma) # training data
  # lists that will contain the variable selections and corresponding errors
  TB <- PB <- lapply(1:6, function(x) array(0, dim = c(6, 4, 3),
                     dimnames = list(paste("V", 1:6, sep = ""), 0:3, c("sur", "cc", "imp"))))
  TB.error <- PB.error <- lapply(1:6, function(x) matrix(0, nrow = 3, ncol = 4,
                                 dimnames = list(c("sur", "cc", "imp"), 0:3)))
  y.test <- as.numeric(dat.test$response)
    for (m in 1:4) { # 4 fractions of missing values
      dat.mis <- lapply(1:6, function(x) dat.train) # 6 missing data generating processes
      if (m != 1) {
      for (k in c("V2", "V5")) {
      ind <- switch(k, "V2" = "V1", "V5" = "V4")
      # induce missing values MCAR, MAR(rank), MAR(median), MAR(upper), MAR(margins), MNAR(upper)
      ind.2 <- list(sample(1:100, (m-1)*.1*100),
                    sample(1:100, (m-1)*.1*100, prob = rank(dat.mis[[2]][,ind]) / 5050),
                    sample(1:100, (m-1)*.1*100, prob = ifelse(dat.mis[[3]][,ind] >=
                                                    median(dat.mis[[3]][,ind]), .9, .1)),
                    dat.mis[[4]][,ind] >= sort(dat.mis[[4]][,ind], decreasing = T)[(m-1)*.1*100],
                    dat.mis[[5]][,ind] >= sort(dat.mis[[5]][,ind], decreasing = T)[(m-1)*.1*100/2] |
                                          dat.mis[[5]][,ind] <= sort(dat.mis[[5]][ ,ind])[(m-1)*.1*100/2],
                    dat.mis[[6]][,k] >= sort(dat.mis[[6]][,k], decreasing = T)[(m-1)*.1*100])
      for (l in 1:6) {is.na(dat.mis[[l]][, k])[ind.2[[l]]] <- TRUE}
      }}
      for (j in 1:6) { # perform variable selection for 6 missing data generating processes
      Tb <- test.based( dat.mis[[j]]$response, dat.mis[[j]][, 1:6])
      Pb <- performance(dat.mis[[j]]$response, dat.mis[[j]][, 1:6])
      y.mis <- dat.mis[[j]]$response
      if (!is.null(Tb$selection)) {
        TB[[j]][Tb$selection, m, 1] <- 1
        TB.error[[j]][1, m] <- mean((y.test - as.numeric(predict(Tb$forest, newdata = dat.test)))^2)}
      else TB.error[[j]][1, m] <- mean((y.test - round(mean(as.numeric(y.mis))))^2)
      if (!is.null(Pb$selection)) {
        PB[[j]][Pb$selection, m, 1] <- 1
        PB.error[[j]][1, m] <- mean((y.test - as.numeric(predict(Pb$forest, newdata = dat.test)))^2)}
      else PB.error[[j]][1, m] <- mean((y.test - round(mean(as.numeric(y.mis))))^2)
      if (m > 1) {
      # perform variable selection with a complete case analysis
      y.mis <- na.omit(dat.mis[[j]])$response
      x.mis <- na.omit(dat.mis[[j]])[, 1:6]
      Tb <-  test.based(y.mis, x.mis)
      Pb <- performance(y.mis, x.mis)
      if (!is.null(Tb$selection)) {
        TB[[j]][Tb$selection, m, 2] <- 1
        TB.error[[j]][2, m] <- mean((y.test - as.numeric(predict(Tb$forest, newdata = dat.test)))^2)}
      else TB.error[[j]][2, m] <- mean((y.test - round(mean(as.numeric(dat.mis[[j]]$response))))^2)
      if (!is.null(Pb$selection)) {
        PB[[j]][Pb$selection, m, 2] <- 1
        PB.error[[j]][2, m] <- mean((y.test - as.numeric(predict(Pb$forest, newdata = dat.test)))^2)}
      else PB.error[[j]][2, m] <- mean((y.test - round(mean(as.numeric(dat.mis[[j]]$response))))^2)
      # perform variable selection with multiple imputation
      imp.dat <- mice(dat.mis[[j]], printFlag = F, defaultMethod = c("norm", "logreg", "polyreg"))
      Tb <- lapply(1:5, function(x) test.based(complete(imp.dat, x)$response, complete(imp.dat, x)[, 1:6]))
      Pb <- lapply(1:5, function(x) performance(complete(imp.dat, x)$response, complete(imp.dat, x)[,1:6]))
      TB[[j]][, m, 3] <- rowSums(sapply(Tb, function(x) table(x$selection)[paste("V", 1:6, sep = "")]),
                                                                                  na.rm = T) / 5
      TB.error[[j]][3, m] <- mean(sapply(Tb, function(x) {
                                 if (!is.null(x$selection)) {
                                 mean((y.test - as.numeric(predict(x$forest, newdata = dat.test)))^2)}
                                 else {mean((y.test - round(mean(as.numeric(dat.mis[[j]]$response))))^2)}}))
      PB[[j]][, m, 3] <- rowSums(sapply(Pb, function(x) table(x$selection)[paste("V", 1:6, sep = "")]),
                                                                                  na.rm = T) / 5
      PB.error[[j]][3, m] <- mean(sapply(Pb, function(x) {
                                 if (!is.null(x$selection)) {
                                 mean((y.test - as.numeric(predict(x$forest, newdata = dat.test)))^2)}
                                 else {mean((y.test - round(mean(as.numeric(dat.mis[[j]]$response))))^2)}}))
      }}}
  for (j in 1:6) { # processes do not differ when there are no missing values
  TB[[j]][, 1, ] <- TB[[1]][, 1, 1]; TB.error[[j]][, 1] <- TB.error[[1]][1, 1]
  PB[[j]][, 1, ] <- PB[[1]][, 1, 1]; PB.error[[j]][, 1] <- PB.error[[1]][1, 1]
  }
return(list(Test.Based = TB, Test.Based.error = TB.error, Perf.Based = PB, Perf.Based.error = PB.error))
}

set.seed(1234) # set a random seed for reproducibility of results
sig <- diag(6); sig[1:3, 1:3] <- 0.3; diag(sig) <- 1 # create covariance matrix
mydat.test <- create.dat(n = 5000, sigma = sig, regression = F) # create the test data
result <- lapply(1:1000, function(x) myfunc(dat.test = mydat.test, sigma = sig)) # run the simulation
```