Analysis of Missing Data with Random Forests

Alexander Hapfelmeier

München 2012

Analysis of Missing Data with Random Forests

Alexander Hapfelmeier

Dissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften

am Institut für Statistik
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

vorgelegt von
Alexander Hapfelmeier

München, den 09.05.2012

Erstgutachter: Prof. Dr. Kurt Ulm
Zweitgutachter: Prof. Dr. Torsten Hothorn
Externer Gutachter: Prof. PhD Adele Cutler
Tag der Disputation: 12.10.2012

To my family

Acknowledgement

I owe my deepest gratitude to Prof. Dr. Kurt Ulm who made this thesis possible and supported me in any possible way. Likewise, I am heartily thankful to Prof. Dr. Torsten Hothorn and Prof. Dr. Carolin Strobl for their supervision and support. I also would like to thank all of my colleagues for their help and constructive discussions.

Contents

List of Figures

List of Tables

Summary

Random Forests are widely used for data prediction and interpretation purposes. They show many appealing characteristics, such as the ability to deal with high dimensional data, complex interactions and correlations. Furthermore, missing values can easily be processed by the built-in procedure of surrogate splits. However, there is only little knowledge about the properties of recursive partitioning in missing data situations. Therefore, extensive simulation studies and empirical evaluations have been conducted to gain deeper insight. In addition, new methods have been developed to enhance methodology and solve current issues of data interpretation, prediction and variable selection.

A variable's relevance in a Random Forest can be assessed by means of importance measures. Unfortunately, existing methods cannot be applied when the data contain missing values. Thus, one of the most appreciated properties of Random Forests – its ability to handle missing values – gets lost for the computation of such measures. This work presents a new approach that is designed to deal with missing values in an intuitive and straightforward way, yet retains widely appreciated qualities of existing methods. Results indicate that it meets sensible requirements and shows good variable ranking properties.

Random Forests provide variable selection that is usually based on importance measures. An extensive review of corresponding literature led to the development of a new approach that is based on a profound theoretical framework and meets important statistical properties. A comparison to another eight popular methods showed that it controls the test-wise and family-wise error rate, provides a higher power to distinguish relevant from non-relevant variables and leads to models located among the best performing ones.

Alternative ways to handle missing values are the application of imputation methods and complete case analysis. Yet it is unknown to what extent these approaches are able to provide sensible variable rankings and meaningful variable selections. Investigations showed that complete case analysis leads to inaccurate variable selection as it may inappropriately penalize the importance of fully observed variables. By contrast, the new importance measure decreases for variables with missing values and therefore causes selections that accurately reflect the information given in actual data situations. Multiple imputation leads to an assessment of a variable's importance and to selection frequencies that would be expected for data that was completely observed. In several performance evaluations the best prediction accuracy emerged from multiple imputation, closely followed by the application of surrogate splits. Complete case analysis clearly performed worst.

Zusammenfassung

Random Forests werden in vielen wissenschaftlichen Bereichen für die Datenanalyse und als Prädiktionsmodell verwendet. Sie besitzen zahlreiche vorteilhafte Fähigkeiten wie mit hochdimensionalen Daten sowie komplexen Interaktions- und Korrelationsstrukturen umgehen zu können. Bislang ist jedoch wenig über ihre Eigenschaften in Datensituationen mit fehlenden Werten bekannt, obgleich sich diese sehr einfach mit Hilfe sogenannter "surrogate splits" behandeln lassen. In dieser Arbeit wurden umfangreiche Simulations und Evaluationsstudien durchgeführt um entsprechende Einsichten zu gewinnen. Neue Verfahren wurden entwickelt um aktuelle Problemstellungen zu lösen.

Durch Wichtigkeitsmaße kann die Relevanz einer Variablen in Random Forests beurteilt werden. Unglücklicherweise lassen sie sich bislang nicht berechnen wenn fehlende Werte in den Daten vorhanden sind. In dieser Arbeit wird durch die Einführung einer neuen Methode eine Lösung für dieses Problem präsentiert. Sie orientiert sich dabei an bereits existierenden Maßen und behält somit beliebte Eigenschaften bei. Es zeigte sich, dass das neue Maß zuvor gestellte Anforderungen erfüllt und gewünschte Eigenschaften aufweist.

Die Möglichkeit der Variablenselektion basierend auf Wichtigkeitsmaßen ist eine zusätzliche Stärke von Random Forests. Eine ausführliche Literaturrecherche führte zu der Idee einer neuen Methode, die basierend of profunden wahrscheinlichkeitstheoretischen Grundlagen wichtige statistische Eigenschaften erfüllt. Im Vergleich mit acht etablierten Algorithmen erwies sie sich als geeignet die vergleichsbezogenen und versuchsbezogenen Irrtumswahrscheinlichkeiten zu kontrollieren und zeigte eine hohe Trennschärfe für die Unterscheidung von relevanten und nicht relevanten Variablen. Hierauf basierende Random Forests erzielten außerdem hohe Vorhersagegüten.

Alternativ lassen sich fehlende Werte durch "complete case" Analysen und Imputation behandeln. Es ist jedoch nicht bekannt inwiefern sich mit diesen Verfahren sinnvolle Wichtigkeitsmaße oder Variablenselektionen berechnen bzw. durchführen lassen. Entsprechende Untersuchungen zeigten, dass complete case Analysen zu inadequaten Selektionen führen, da die Wichtigkeit vollständig beobachteter Variablen fälschlich herabgewertet werden kann. Das neue Wichtigkeitsmaß sinkt dagegen ausschließlich für Variablen mit fehlenden Werten und erzeugt somit Selektionen, die tatsächliche Datensituationen widerspiegeln. Ein Imputationsverfahren führt zu Ergebnissen, die für vollständige Daten zu erwarten gewesen wären. In mehreren Bewertungen wurden für Letzteres auch die besten Vorhersagegüten ermittelt. Die Anwendung von Surrogaten war nur unwesentlich schlechter wobei complete case Analysen deutlich am schlechtesten abschnitten.

Introduction

Recursive partitioning methods, in particular classification and regression trees and Random Forests, are popular approaches in statistical data analysis. They are applied for data prediction and interpretation purposes in many research fields such as social, econometric and clinical science. Among others, there are approaches like the famous CART algorithm introduced by Breiman et al. (1984), the C4.5 algorithm by Quinlan (1993) and conditional inference trees by Hothorn et al. (2006). A detailed listing of application areas and methodological issues, along with discussions about the historical development and state-of-the-art, can be found in Strobl et al. (2009). The popularity of trees is rooted in several appealing characteristics like their easy applicability and interpretability in both, classification and regression problems. Advantages over common approaches like logistic and linear regression are their ability to implicitly deal with missing values, collinearity, nonlinearity and high dimensional data. Random Forests are able to achieve competitive or even superior prediction strengths in comparison to well established approaches (i.e. regression, linear discriminant analysis, support vector machines, neural nets etc.). Moreover, recursive partitioning is able to handle even complex interaction effects – which is a highly valued property e.g. for the analysis of gene-gene relations (Lunetta et al., 2004; Cutler et al., 2007; Tang et al., 2009; Nicodemus et al., 2010).

The main focus of this work is put on the performance and applicability of Random Forests and corresponding features – like variable importance measures and variable selection – in missing data analysis. Therefore, after a short introduction of methodology in chapter 1, the predictive accuracy of Random Forests – and single trees – is explored and compared between the application to data with and without a preliminary imputation of missing values in chapter 2. Several datasets that provide classification and regression problems have been used for simulation studies and empirical evaluations. For the former, missing values were induced into fully observed data while for the latter, data were used that already contained missing values. Multiple imputation produced variable results while the application of surrogates appeared to be a fast and simple way to achieve performances which are only negligibly worse and in many cases even superior.

An important feature of Random Forests is the evaluation of a variable's relevance by means of importance measures. However, existing measures cannot be computed in the presence of missing values. A straightforward application to such data leads to violations of their most basic conceptual principles. A solution to this issue is introduced in chapter 3: A new approach makes the computation of variable importance measures possible even when there are missing data. Its properties are investigated in an extensive simulation study. Results show that a list of sensible, pre-specified requirements are completely fulfilled. An application to two datasets also shows its practicality in real life situations.

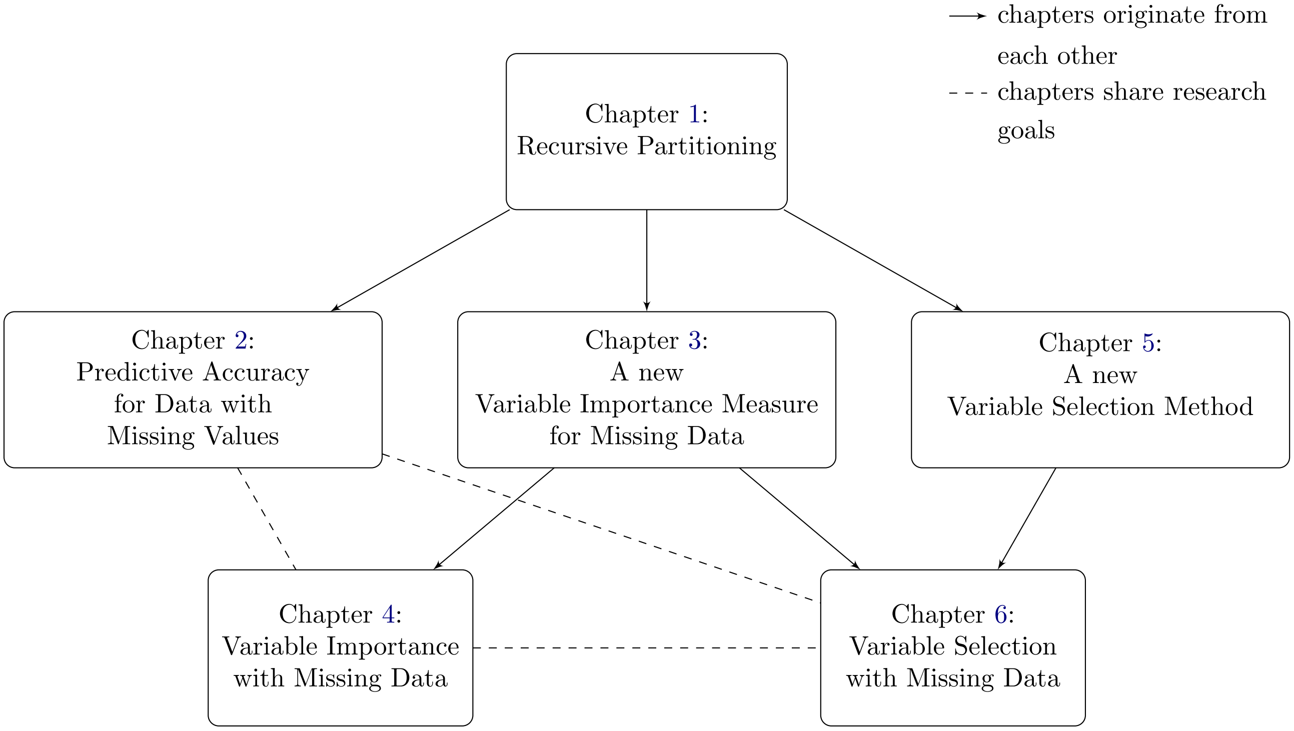
Imputation methods and complete case analysis are two alternative approaches that enable the computation of importance measures in the presence of missing values. However, it is unknown to what extend these approaches are able to provide a reliable estimate of a variable's relevance. Therefore, an extensive simulation study was performed in chapter 4 to investigate this property for a variety of missing data generating processes. Prediction accuracy has been explored in accordance with investigations of chapter 2. Findings suggest that complete case analysis should not be applied as it may inappropriately penalize variables that were completely observed. The new importance measure is much more capable of reflecting decreased information exclusively for variables with missing values and should therefore be used to evaluate actual data situations. By contrast, multiple imputation allows for an estimation of importances one would potentially observe in complete data situations.

Importance measures are often used as a basis for variable selection. Many works (e.g. Tang et al., 2009; Yang and Gu, 2009; Rodenburg et al., 2008; Sandri and Zuccolotto, 2006; Díaz-Uriarte and Alvarez de Andrés, 2006) show that different approaches have been developed to distinguish relevant from non-relevant variables and to improve prediction accuracy. An extensive review of the corresponding literature led to the development of a new approach that is based on a more profound theoretical framework and meets important statistical properties. A comparison to another eight established selection approaches is given in chapter 5. The new proposal is able to outperform these competitors in three simulation studies and four empirical evaluations with regard to discriminatory power and prediction accuracy of resulting models.

Once again, alternatives are given by complete case analysis and imputation methods. Therefore an extensive simulation study has been conducted in chapter 6 to explore the ability of each approach – in combination with the new variable selection method and a popular representative of established approaches – to distinguish relevant from non-relevant variables. In accordance with chapter 2 and chapter 4 the predictive accuracy of resulting models has been investigated, too. Findings suggest that complete case analysis should not be applied as it leads to inaccurate variable selection. Multiple imputation is a good means to select variables that would be of relevance in fully observed data. By contrast, the application of the new importance measure caused a selection of variables that reflects the actual data situation, i.e. that takes the occurrence of missing values into account.

In conclusion, this work presents extensive investigations of Random Forests for the analysis of data with missing values. Important aspects like predictive accuracy, variable importance and variable selection are examined. New methods are introduced and compared with well-known and established approaches.

Scope of Work


chapters originate from each other
chapters share research goals
Chapter 1:
Recursive Partitioning
Chapter 2:
Predictive Accuracy for Data with Missing Values
Chapter 3:
A new Variable Importance Measure for Missing Data
Chapter 5:
A new Variable Selection Method
Chapter 4:
Variable Importance with Missing Data
Chapter 6:
Variable Selection with Missing Data

A short introduction to recursive partitioning is given in chapter 1:

- Construction principles and properties of trees and Random Forests, following the CART algorithm and a conditional inference framework, are presented.
- Easy and comprehensible examples are used for illustration.
- Additional features like the concept of surrogate splits and variable importance measures are discussed.
- A short overview of software used for implementation is given.

An evaluation of the predictive accuracy of trees and Random Forests fit to data with and without imputed missing values is given in chapter 2:

- Related publications with similar research goals are discussed.
- The concept of missing data generating processes is introduced.
- Imputation methods are discussed, in particular a multiple imputation approach.
- Simulation studies and empirical evaluations are used to investigate differences between the application of surrogate splits and multiple imputation.

A new variable importance measure for missing data is introduced in chapter 3:

- Its rationale, definition, computational steps and a postulation of sensible requirements are presented and discussed.
- An investigation of its properties is performed by an extensive simulation study.
- The applicability of the new method is compared to a complete case analysis in empirical evaluations.

The behavior of importance measures for missing data is investigated in chapter 4:

- The ability of the new importance measure, complete case analysis and multiple imputation to produce reliable estimates for a variable's relevance is explored in extensive simulation studies.

A new variable selection method for Random Forests is presented in chapter 5:

- Discussions of existing methods are given based on a broad review of literature.
- A new variable selection approach which is based on the statistical framework of permutation tests is introduced.
- A comparison in terms of prediction accuracy and the power to distinguish relevant from non-relevant variables is performed against eight popular and well-established selection approaches within several simulation studies and empirical evaluations.

Variable selection for missing data is investigated in chapter 6:

- The new importance measure, complete case analysis and multiple imputation are used to investigate the ability of two variable selection methods – i.e. the new approach and a representative of established methods – to discriminate relevant from non-relevant variables in extensive simulation studies.

Finally, a concluding outlook 6.4 refers to future work. It is followed by supplementary material in appendix A and the R-Code of each method and study in appendix B.

Chapter 1

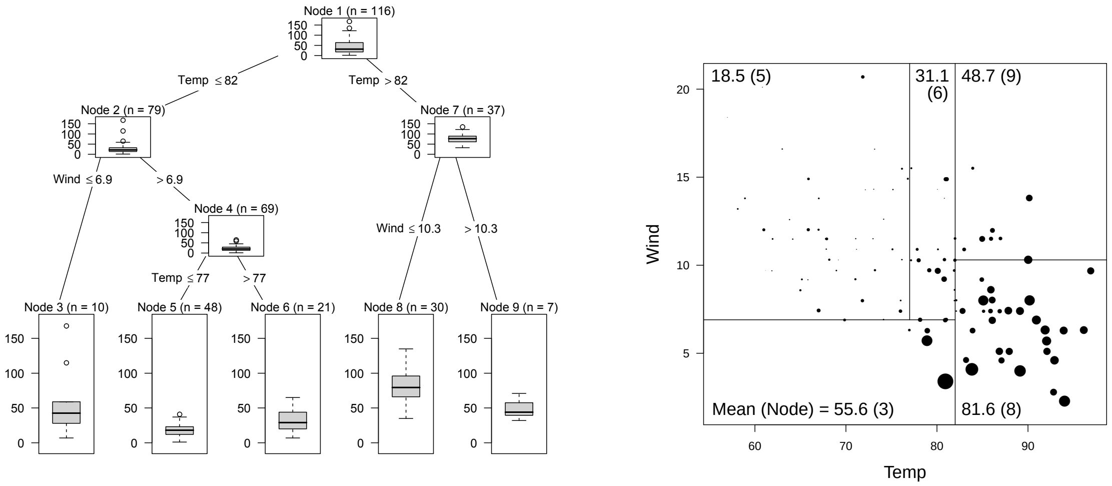
Recursive Partitioning

1.1 Classification and Regression Trees

1.1.1 Rationale

The rationale of recursive partitioning is best described by the example of the CART algorithm (cf. Breiman et al., 1984; Hastie et al., 2009, for details). It constructs trees as it sequentially conducts binary splits of the data in order to produce subsets which, with respect to the outcome, are as homogeneous as possible. An example of a regression tree used to predict ozone concentration in air quality data is given by Figure 1.1a. The data contains daily measurements of the air quality in New York from May to September 1973 and is made available by the `R` software for statistical computing (R Development Core Team, 2011). It consists of 6 variables: Day and month of recording, ozone pollution at Roosevelt Island measured in parts per billion (ppb), solar radiation at Central Park in Langleys (lang), average wind speed in miles per hour (mph) and the maximum daily temperature in degrees Fahrenheit, both of the latter measured at La Guardia Airport (ozone data was originally provided by the New York State Department of Conservation and meteorological data by the National Weather Service). A detailed exploration and analysis of the data can be found in Chambers (1983).

The airquality data originally consists of 153 observations. Though, the outcome ozone contains some missing values which reduces the complete case analysis set to 116 observations. The distribution of the remaining values is displayed by a boxplot in the first node of the regression tree in Figure 1.1a. This node is split into two daughter nodes separating observations with temperature measurements $\leq$ and $>$ 82° into two subsets of size 79 and 37. A comparison of the corresponding boxplots to the one of the parent node reveals that the split was able to create more homogeneous subsets in reference to the outcome. Likewise the heterogeneity between subsets rises with every conducted split. This procedure continues as more split-rules are produced for the variables `Temp` and `Wind`. Finally the diversity of distributions of the outcome in the subsets becomes evident by the final nodes of the tree. Another illustration of the segmentation given in many works for educational reasons (Hastie et al., 2009; Strobl et al., 2009) is displayed by Figure 1.1b. It



(a) (b)

Figure 1.1: (a) Regression tree for the airquality data example. (b) Corresponding segmentation of the variable space. Point size indicates ozone magnitude.

clearly shows the way the variable space is split-up to create more homogeneous subsets. Finally, predictions for new observations can be taken from the conditional distribution of outcomes allocated to these subsets in the training phase of a tree (e.g. mean, relative class frequencies, ...). Therefore new observations are assigned to the final nodes as they are sent down the tree along paths determined by the split-rules. In this example they can even be summed up in simple decision rules as demonstrated by Table 1.1. For example, for a day with a temperature of 70° and a wind speed of 10 the prediction would be 18.5 for ozone.

Temp (t)	Wind (w)	Mean
$82 < t$	$10.3 < w$	48.7
	$w \leq 10.3$	81.6
$t \leq 82$	$w \leq 6.9$	55.6
$77 < t \leq 82$	$6.9 < w$	31.1
$t \leq 77$		18.5

Table 1.1: Split rules and descriptive statistics of the final nodes in the air quality example.

1.1.2 The CART Algorithm

Depending on the response type, different criteria are used to determine the splits of a tree. In a fundamental work of Breiman et al. (1984) several split criteria are suggested.

Popular choices for binary and continuous responses are the Gini Index and the residual sum of squares (RSS), respectively. In the case of a classification tree the former is defined for a given node k by

$$\widehat{G}_k = 2\frac{N_{1k}}{N_k}\frac{N_{2k}}{N_k}$$

with 1 and 2 indicating the response classes and N the number of observations. For example N$_{2k}$ is the number of observations of class 2 in node k. The Gini-Index is used as a measure of node impurity. It takes values between 0 and 1/2 corresponding to pure (only one response class is represented in a node) and maximally impure nodes (both classes are equally represented in a node), respectively. An optimal split is found for the cutpoint of a variable that maximizes the Gini gain of a parent node to its daughter nodes. The Gini gain is defined by the difference of a parent node's Gini-Index to the sum of child nodes Gini-Index, where the latter is weighted by the relative frequency of observations that are sent left (L) or right (R):

$$\widehat{\Delta G}_k = \widehat{G}_k - \left(\frac{N_{Lk}}{N_k}\widehat{G}_{Lk} + \frac{N_{Rk}}{N_R}\widehat{G}_{Rk}\right).$$

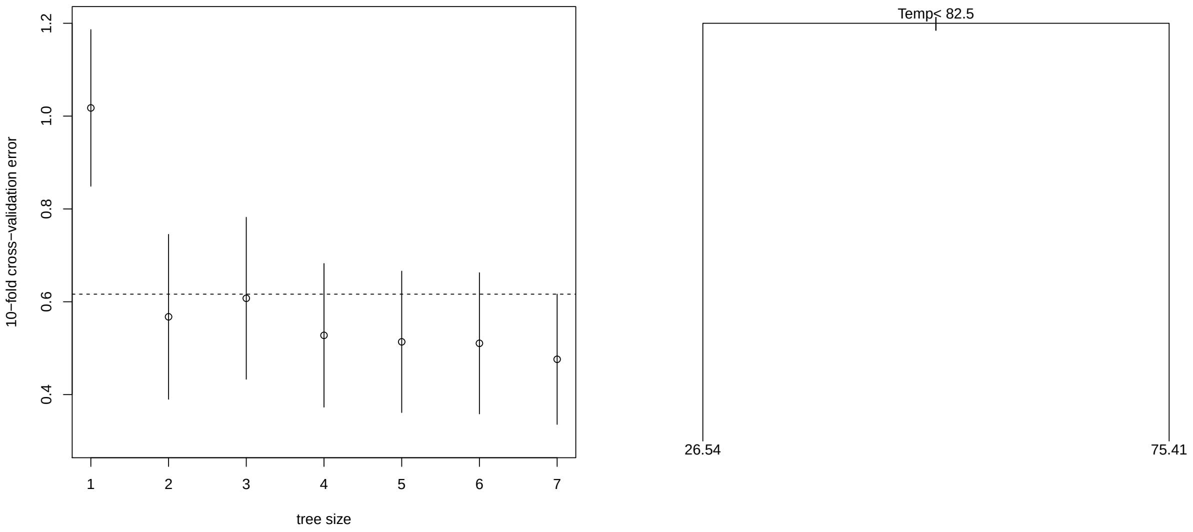
For regression trees the criterion is changed to the maximization of the RSS difference

$$\widehat{\Delta R}_k = \widehat{RSS}_k - \left(\widehat{RSS}_{Lk} + \widehat{RSS}_{Rk}\right).$$

Trees are grown until a certain criterion is reached, e.g.: a limiting number of observations needed in a parent node to allow for further splitting, a minimum size of daughter nodes, complete purity of terminal nodes, or a threshold for the split criterion. Afterwards a tree grown to its full size can be "pruned" back in order to circumvent the issue of overfitting. For this purpose the performance of the tree is evaluated via cross-validation at different growth stages. Finally the smallest tree whose mean performance is within a specified distance of u-times the standard deviation to the best performing tree is chosen. Setting $u = 1$ equals the 'one-standard-error' rule ('1 s.e.' rule). A more detailed description of this approach can be found in Breiman et al. (1984) and Hastie et al. (2009).

A corresponding analysis of the airquality data is shown by Figure 1.2a. The cross validated error observed for different sizes of a CART like regression tree reaches its minimum at seven terminal nodes. However, a tree of size two still provides an error which is within the threshold of one standard deviation to this benchmark (dashed line). According to the 1 s.e. rule it should be chosen as the final model (cf. Figure 1.2b).

Breiman et al. (1984) already stated that the CART algorithm – and other recursive partitioning approaches like the C4.5 method of Quinlan (1993) – favor splits in continuous variables and variables with many categories. Works like those of Lausen et al. (1994) and Hilsenbeck and Clark (1996) have proposed solutions to the related problem of 'optimally selected cutpoints'. Likewise, predictors with many missing values may be preferred if the Gini Index is employed (c.f. Strobl et al., 2007a). This also affects Random Forest algorithms that are based on the same construction principles. To overcome these problems,



(a) (b)

Figure 1.2: (a) Plot of tree sizes against the error (±1 standard error) assessed by 10-fold cross-validation for a CART like regression tree. The horizontal dashed line indicates the threshold of 1 standard error to the minimal error. (b) Tree of size 2.

several unbiased tree algorithms have been suggested (cf. Dobra and Gehrke, 2001; Hothorn et al., 2006; Kim and Loh, 2001; Lausen et al., 1994; Strobl et al., 2007a; White and Liu, 1994).

1.1.3 Conditional Inference Trees

Facing all of these pitfalls Hothorn et al. (2006) introduced the concept of conditional inference trees. In this approach splits are performed in two steps. In a first step the relation of a variable to the response is assessed by permutation tests based on a theoretical conditional inference framework developed by Strasser and Weber (1999). This allows for a fair comparison independent of a predictor's scale. Consequently there is no bias in favor of continuous variables and variables with many categories or many missing values any more. After the strongest relation was found by the minimal p-value of the permutation tests it is checked for significance, optionally with adjustment for multiple testing (one possibility is to use the Bonferroni-Adjustment). Finally, in the second step the best cutpoint for the most significant variable chosen in step one is determined. The growth of a tree stops as soon as there are no further significant relations found. In addition to the advantage of unbiased variable selection, Hothorn et al. (2006) showed that conditional inference trees don't overspend the alpha error and stick closer to the underlying data structure while they produce comparable performance results to CART.

A conditional inference tree for the airquality data is given in Figure 1.3. Note that the p-values of the permutation tests for each split can be read off the nodes.

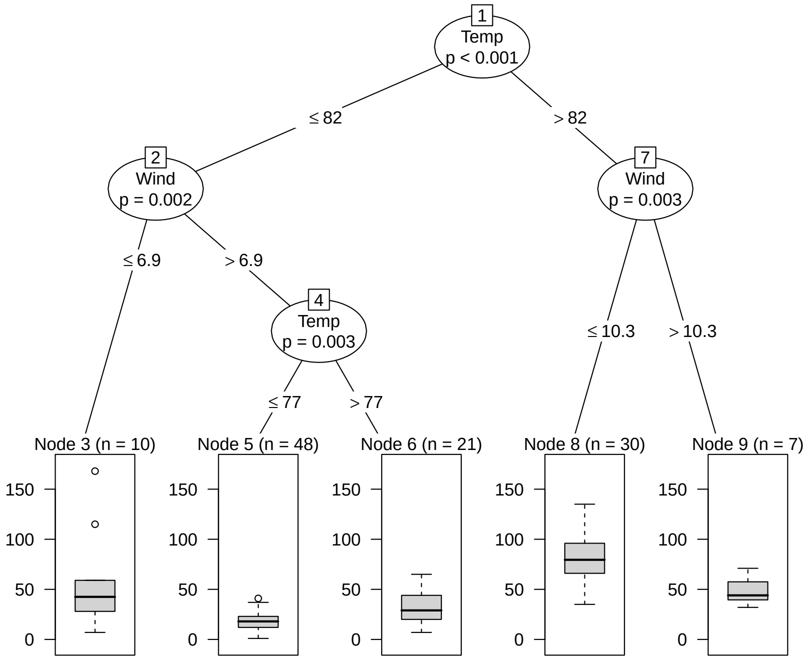


Figure 1.3: Conditional inference regression tree for the airquality data.

The following gives a short summary of the methodology presented in Hothorn et al. (2006):

As already outlined the binary splits in conditional inference trees are assessed in two steps. In the first one it is checked if any variable X_j, $j = 1, \dots, v$ – of the v-dimensional vector $\mathbf{X} = (X_1, \dots, X_v)$ which itself originates from the sample space $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_v$ – is related to the response Y. Therefore $H_0^j : D(Y|X_j) = D(Y)$ is examined. Obviously, checking this hypothesis for several variables induces a multiple test problem which results in a violation of the family-wise error rate (FWR) or false discovery rate (FDR). Several methods like the Bonferroni-Adjustment, Benjamini-Hochberg and Benjamini-Yekutieli procedure have been proposed to control for these errors (cf. Hastie et al., 2009; Bland, 2000; Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001, for details). For the construction of conditional inference trees the Bonferroni-Adjustment may be used: H_0 can be rejected if the corresponding p-value drops below a significance level $\alpha^* = \alpha / n_{\text{tests}}$.

The association between Y and X_j is determined by a linear statistic:

$$\boldsymbol{T}_j(\mathcal{L}_n, \boldsymbol{w}) = vec\left(\sum_{i=1}^{n} w_i g_j(X_{ji}) h(Y_i, (Y_1, ..., Y_n))^\top\right) \in \mathbb{R}^{p_j q}$$

where $g_j : \mathcal{X}_j \to \mathbb{R}^{p_j}$ is a non-random transformation of the variable X_j. The influence function $h : \mathcal{Y} \times \mathcal{Y}^n \to \mathbb{R}^q$ depends on the responses $(Y_1, \dots, Y_n)$ in a permutation symmetric way. A $p_j \times q$ matrix is converted into a $p_j q$ column vector by column-wise combination. The n-dimensional vector $\boldsymbol{w} = (w_1, \dots, w_n)$ contains weights which indicate the correspondence of observations to the nodes. $\mathcal{L}_n = \{(Y_i, X_{1i}, ..., X_{mi}); i = 1, ..., n\}$ denotes the learning set of the data.

Under the null hypothesis H_0^j, by fixing the covariates and by conditioning on all possible permutations $\sigma \in S(\mathcal{L}_n, \boldsymbol{w})$ of the responses, one can derive the conditional expectation $\mu_j \in \mathbb{R}^{p_j q}$ and covariance $\Sigma_j \in \mathbb{R}^{p_j q \times p_j q}$ of $\boldsymbol{T}_j(\mathcal{L}_n, \boldsymbol{w})$ as introduced by Strasser and Weber (1999). By a standardization of the linear test statistic one is able to compute p-values

$$P_j = \mathbb{P}_{H_0^j}(c(\boldsymbol{T}_j(\mathcal{L}_n, \boldsymbol{w}), \mu_j, \Sigma_j) \geq c(\boldsymbol{t}_j, \mu_j, \Sigma_j) | S(\mathcal{L}_n, \boldsymbol{w}))$$

of the conditional test for H_0^j. Depending on whether the standardized statistics are

$$c_{max}(\boldsymbol{t}_j, \mu_j, \Sigma_j) = \underset{k=1,\dots,pq}{max} \left| \frac{(\boldsymbol{t}_j - \mu_j)_k}{\sqrt{(\Sigma_j)_{kk}}} \right| \tag{1.1}$$

or

$$c_{quad}(\boldsymbol{t}_j, \mu_j, \Sigma_j) = (\boldsymbol{t}_j - \mu_j) \Sigma_j^+ (\boldsymbol{t}_j - \mu_j)^\top \tag{1.2}$$

the asymptotic $(n, \boldsymbol{w} \to \infty)$ conditional distributions are normal for (1.1) and χ^2 for (1.2) with degrees of freedom given by the rank of Σ_j. Σ_j^+ is the Moore-Penrose inverse of Σ_j.

Now that one has checked if the null hypothesis can be rejected, a split is performed on the variable X_{j^*} with the lowest p-value. Again a linear test statistic in the permutation test framework helps to find the splitting rule:

$$\boldsymbol{T}_{j^*}^A(\mathcal{L}_n, \boldsymbol{w}) = vec \left(\sum_{i=1}^n w_i \boldsymbol{I}(X_{j^*i} \in A) h(Y_i, (Y_1, ..., Y_n))^\top \right) \in \mathbb{R}^q.$$

Among all possible subsets A of the sample space $\mathcal{X}_{j^*}$ the best split is given by

$$A^* = \underset{A}{argmax}\ c(\boldsymbol{t}_{j^*}^A, \mu_{j^*}^A, \Sigma_{j^*}^A).$$

1.1.4 Surrogate Splits

There are several possibilities to handle missing values. One of them is to stop the throughput of an observation at the node at which the information for the split rule is missing (the prediction is then based on the conditional distribution of the responses that are elements of this node). Another approach makes the missing values simply follow the majority of all observations with observed values (cf. Breiman et al., 1984). However, by far the most popular way to handle missing observations is to use surrogate decisions based on additional variables (cf. Breiman et al., 1984; Hothorn et al., 2006). These splits try to mimic the initial split as they preserve the partitioning of the observations. When several surrogate splits are computed they can be ranked according to their ability to resemble the primary split. An observation that contains several missing values in surrogate variables is processed along this ranking until a decision for a missing value is found. The number of possible surrogate splits is usually determined by the user. Figure 1.4 displays a schematic view of the surrogate split concept for a hypothetical example. Here the first split rule is given by $X_1 \leq x_1$. There are two surrogate splits in X_2 and X_3 which try to mimic this split.

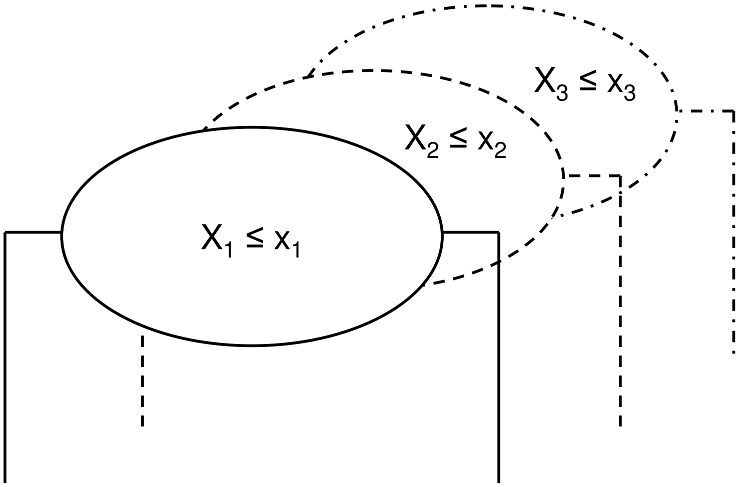


Figure 1.4: Schematic view of the surrogate split conception.

Technically surrogate splits can be found by the exact same procedure used to obtain the primary split (Hothorn et al., 2006). Therefore, the original response vector is replaced by a binary variable which indicates the allocation of observations – the ones that are not missing – to the daughter nodes. A search for the optimal split of variables for this 'new outcome' will provide surrogate splits which mimic the decisions of the primary split as precisely as possible.

An alternative and very general way to handle missing values is to use imputation methods (Schafer and Graham, 2002; Horton and Kleinman, 2007; Harel and Zhou, 2007; Klebanoff and Cole, 2008; Janssen et al., 2009, 2010, for an overview and further reading). However, investigations of Rieger et al. (2010) and Hapfelmeier et al. (2011) have shown that this improves the prediction accuracy of models only to a negligible extent.

1.2 Random Forests

1.2.1 Rationale and Definition

Breiman (1996) used “bagging” (bootstrap aggregation) to enhance the tree methodology. In bagging, several trees are fit to bootstrapped or subsampled data. Averaged values or majority votes of the predictions produced by each single tree are used as predictions. This way, piecewise constant prediction functions – given by a single tree’s hard decision boundaries – are smoothed out. Accordingly, any kind of functional relation, which is potentially not piecewise constant and may be nonlinear or even includes interactions, can be approximated by Random Forests. It can also be shown that the performance improves due to a reduction of the variance of predictions. A simple explanation for the high variability of predictions of single trees is given by their instability. It is a well known fact that small changes in the data can affect the entire tree structure because the sequence of splits and the corresponding relation between decision rules is sensitive to such changes. Researchers working with frequently changing or updated data might already have experienced this issue. Random Forests are based on trees fit to random subsamples of the data and therefore implicitly comprise this variability which results in more stable predictions. Figure 1.5 shows, for a constructed, hypothetic example, how the aggregation of stepwise prediction functions can improve the approximation of the functional relation between the response and its predictors.

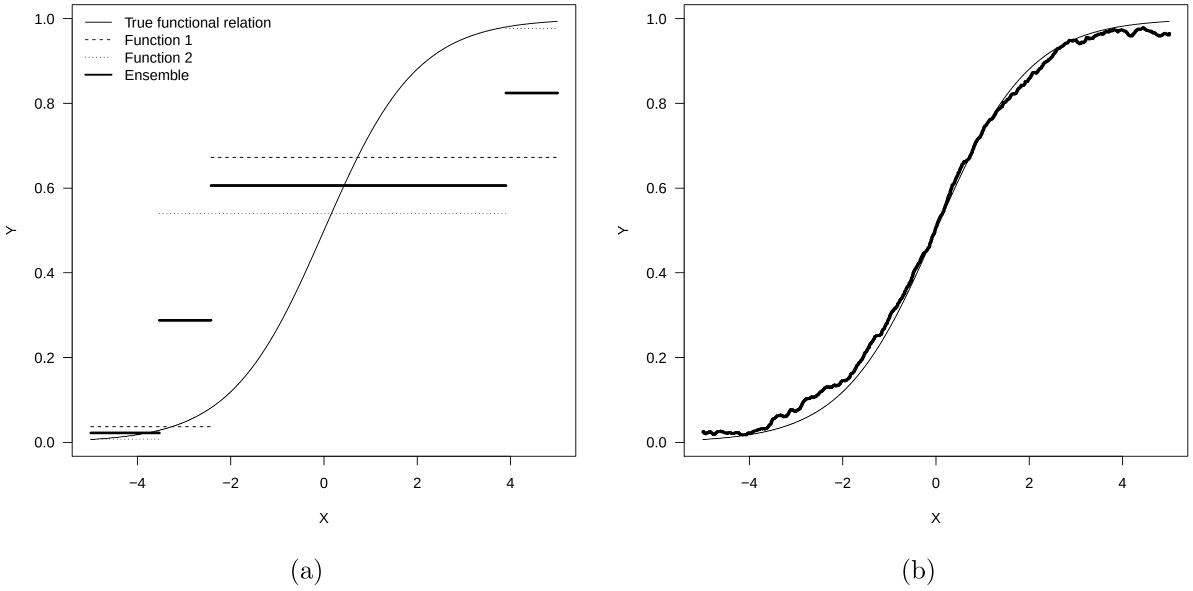


Figure 1.5: Example for the approximation of the functional relation between the outcome Y and the predictor X by (a) an ensemble of two trees and (b) an ensemble of 1000 trees.

As an extension of bagging, Random Forests (cf. Breiman, 2001; Breiman and Cutler, 2008) were introduced: In Random Forests, each split is searched for in a subset of variables. A popular choice is to randomly select the square root of the number of available predictors, as candidates for the split (cf. Díaz-Uriarte and Alvarez de Andrés, 2006). This enables a more diverse set of variables to contribute to the joint prediction of a Random Forest, which results in an improved prediction accuracy. Also, interaction effects between variables that otherwise would have been dominated by stronger predictors might be uncovered. An example of a Random Forest fit to the airquality data is given by Figure 1.6 which highlights the diversity of trees.

The prediction accuracy itself is usually assessed by observations that were not part of the sample used to construct the respective tree (the so called “out of bag” (OOB) observations; cf. Breiman (2001)). Therefore it provides a more realistic estimate of the performance that can be expected for new data (cf., e.g., Boulesteix et al., 2008a; Strobl et al., 2009). Each tree is grown until terminal nodes called leaves are pure or reach a specified minimal size, without any pruning. There is no general advice on how many trees should be used in a Random Forest. Breiman (2001) proves that with a rising number of trees the Random Forest does not overfit but '... produces a limiting value of the generalization error' while the results of Lin and Jeon (2006) indicate that they do overfit when trees are grown too large. Further research of Biau et al. (2008) lead to theorems about the consistency of Random Forest approaches and other averaging rules. Likewise, Genuer (2010) was able to show the superiority in prediction accuracy for a variant of Random Forests, in comparison to single trees, and therefore proved the attendant question of variance reduction in this special case.

The conditional inference approach of Hothorn et al. (2006) can be used to construct Random Forests following the same rationale as Breiman's original approach. Furthermore, it guarantees unbiased variable selection and variable importance measures when combined with subsampling (as opposed to bootstrap sampling; Strobl et al., 2007b). The conditional inference framework is used in the following. An extensive summary of the state-of-the-art can be found in Strobl et al. (2009).

1.2.2 Importance Measures

Random Forests are not solely used to achieve improved prediction accuracy but also for the identification of relevant variables. Variable importance measures enable an assessment of the relevance a variable takes in a Random Forest. In addition, importance measures are often used as a basis for variable selection. The latter issue will be the topic of chapter 5, which introduces a new variable selection approach. However, a publication of Nicodemus et al. (2010) clarifies that the properties of importance measures are still not fully understood and need to be object of further investigation. Not surprisingly there are new and promising approaches for the computation of importance measures and corresponding variable selection methods. The work of Sandri and Zuccolotto (2006), Altmann et al. (2010), Wang et al. (2010) and Zhou et al. (2010) shows that the development of new

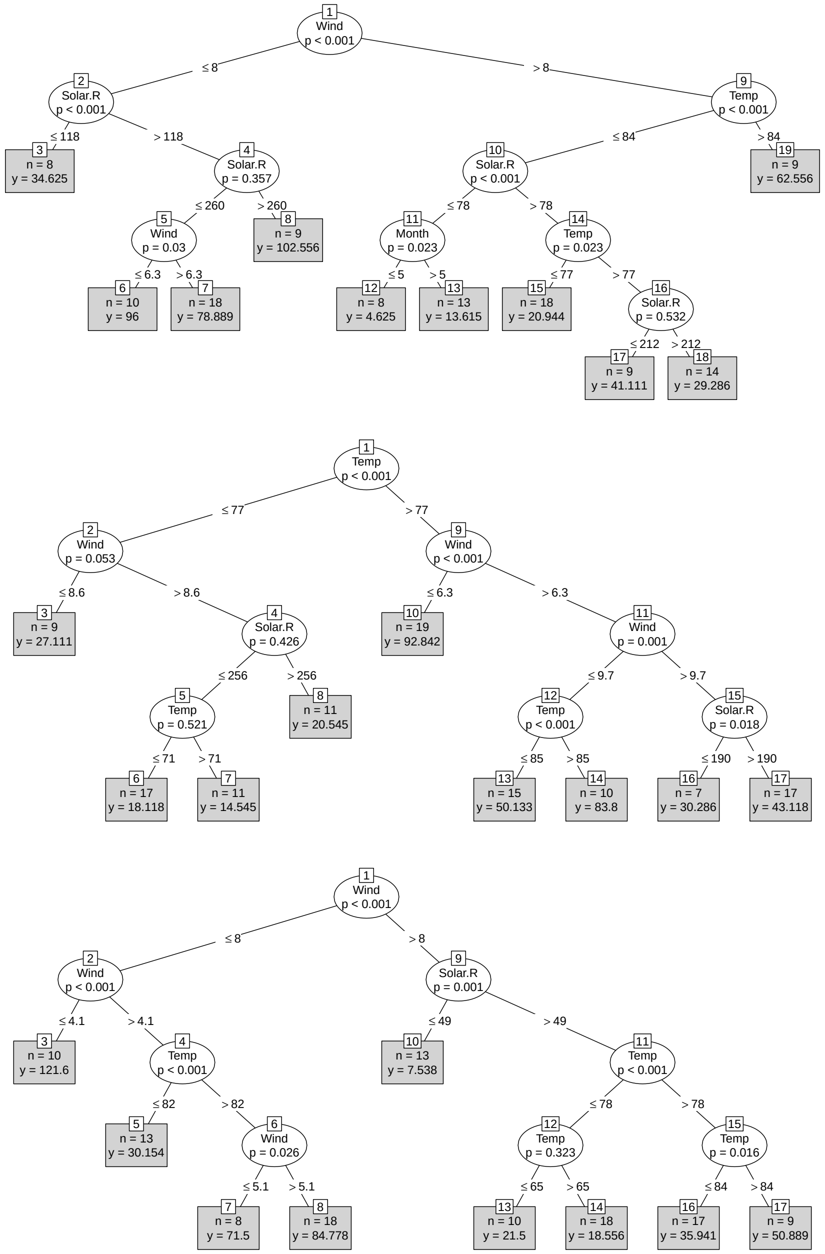


Figure 1.6: Example of three trees of a Random Forest fit to the airquality data.

importance measures is an ongoing process. The most popular approaches to determine variable importances in Random Forests are presented in the following.

Count

A very simple way to determine a variable's importance is to count the number of times it is chosen for splits in a Random Forest. Advantages of this approach are its easy and fast realization. Moreover it's a well-known and established procedure to evaluate the importance of a variable by assessment of its selection frequency when variable selection is applied to several bootstrap samples of the data. Examples for linear, logistic or Cox regression are given by Sauerbrei (1999), Sauerbrei et al. (2007) and Austin and Tu (2004). In the field of microarray data analysis Qiu et al. (2006) published further interesting examples. However, this popular approach comes along with some evident disadvantages: A count rates each split in the same way independent of its position in a tree and its discriminatory power. Therefore it will not be investigated any further in this work.

An example for the selections frequencies of predictors in the airquality data is given in Figure 1.7 for a Random Forest that consists of 500 trees. The corresponding `count()` function written to count the number of times a variable is chosen to represent a split in a Random Forest is given in the appendix B.

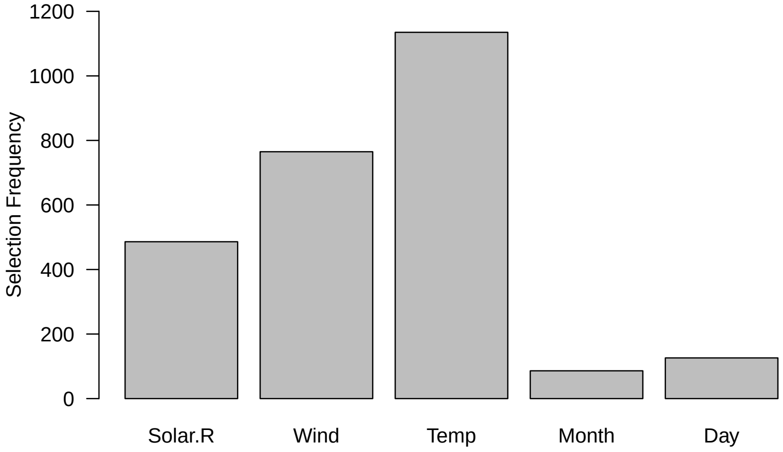


Figure 1.7: Selection frequencies of predictors in the airquality data example for a Random Forest consisting of 500 trees. Note that frequencies may well exceed the number of trees as predictors can be chosen multiple times for splits of a tree.

Gini importance

The Gini importance, that is available in many Random Forest implementations, accumulates the Gini gain over all splits and trees of a Random Forest to evaluate the discriminatory power of a variable (Hastie et al., 2009). A severe disadvantage of this measure is that all tree and Random Forest algorithms based on the Gini splitting criterion are prone to biased variable selection (Strobl et al., 2007a; Hothorn et al., 2006). Recent results also indicate that it has undesirable variable ranking properties, especially when dealing with unbalanced category frequencies (Nicodemus, 2011). Furthermore it is only applicable to classification problems. For these reasons the Gini importance is not considered any further in this work.

Permutation Accuracy Importance

The most popular and most advanced variable importance measure for Random Forests is the permutation accuracy importance. One of its advantages is its broad applicability and unbiasedness (when used in combination with subsampling as shown by Strobl et al., 2007c). The permutation importance is assessed by a comparison of the prediction accuracy, in terms of correct classification rate or mean squared error (MSE), of a tree before and after random permutation of a predictor variable X_j. By permutation the original association with the response is destroyed and the accuracy is supposed to drop for a relevant predictor. More precisely, this procedure, which clearly emerges from the framework of permutation tests (further insight in basic principles is given by several works like those of Good, 2000; Efron and Tibshirani, 1994), is meant to cancel any association between X_j and the response Y and therefore simulates the null hypothesis

$$H_0 : Y \perp X_j.$$

When the accuracies – before and after permutation – are almost equal there is no evidence against H_0. Consequently, the importance of X_j is termed to be low as its permutation did not show any remarkable influence. By contrast, if the prediction accuracy drops substantially X_j is considered to be of relevance. The average difference across all trees provides the final importance score. Large values of the permutation importance indicate a strong association between the predictor variable and the response. Values around zero (or even small negative values, cf. Strobl et al., 2009) indicate that a predictor is of no value for the prediction of the response. However, considering the structure of a tree, generated by sequential binary splits in different variables, it becomes evident that the importance measure is not only sensitive to relations between the predictor variable X_j and the outcome Y but also to relations between X_j and the remaining variable space $\mathbf{Z} = \mathbf{X} \setminus X_j$. Thus, simply permuting X_j actually checks for deviations from the null hypothesis

$$H_0 : Y, \mathbf{Z} \perp X_j \text{ which equals } H_0 : Y \perp X_j \wedge \mathbf{Z} \perp X_j. \tag{1.3}$$

Consequently, X_j can also achieve a high importance because of its relation to $\mathbf{Z}$ and not only to Y. Therefore Strobl et al. (2008) introduced a conditional version that more closely resembles the behavior of partial correlation or regression coefficients. For the computation of the importance measure they suggest to permute X_j in dependence of $\mathbf{Z}$. Now the null hypothesis can be expressed as

$$H_0 : (X_j \perp Y)|\mathbf{Z}.$$

The discussions given in chapter 5 show that both kinds of measures, conditional and unconditional, can be of specific value depending on the research question (cf. Nicodemus et al., 2010; Altmann et al., 2010). For example in large-scale screening studies like genome wide association studies the identification of correlated markers by unconditional importance measures is a desirable property for uncovering physical proximities and causal variants. A similar argumentation holds for microarray studies. Many recent publications indicate that this measure is still of vast popularity and appreciated for its unconditional properties (i.e. for its sensitivity to (cor-)relations between variables). By contrast conditional importance measures can help to differentiate influential predictors from correlated, non-influential ones.

The variable importance itself is given by

$$VI(X_j) = \frac{\sum_{t=1}^{n_{\text{tree}}} VI^{(t)}(X_j)}{n_{\text{tree}}} \tag{1.4}$$

while

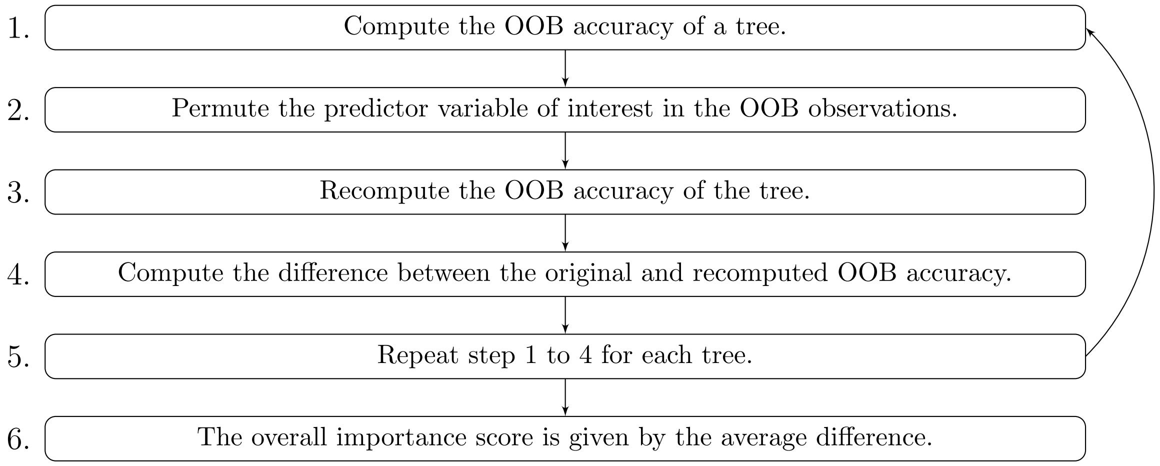
$$VI^{(t)}(X_j) = \frac{\sum_{i \in \bar{\mathcal{B}}^{(t)}} I(y_i = y_i^{(t)})}{|\bar{\mathcal{B}}^{(t)}|} - \frac{\sum_{i \in \bar{\mathcal{B}}^{(t)}} I(y_i = y_{i,\pi_j|\mathbf{Z}}^{(t)})}{|\bar{\mathcal{B}}^{(t)}|} \tag{1.5}$$

for categorical variables and

$$VI^{(t)}(X_j) = \frac{\sum_{i \in \bar{\mathcal{B}}^{(t)}} (y_i - y_{i,\pi_j|\mathbf{Z}}^{(t)})^2}{|\bar{\mathcal{B}}^{(t)}|} - \frac{\sum_{i \in \bar{\mathcal{B}}^{(t)}} (y_i - y_i^{(t)})^2}{|\bar{\mathcal{B}}^{(t)}|} \tag{1.6}$$

for metric outcomes.

$\bar{\mathcal{B}}^{(t)}$ indicates that in the computation of the permutation importance, the assessment of the prediction accuracy – in terms of correct classification or mean squared error – is usually based on observations that were not part of the sample used to fit the respective tree (the so called “out of bag” (OOB) observations). This way, the OOB permutation importance provides a more reliable, less biased estimate of the importance a variable may have, independent of the respective training samples. The index π_j denotes the permutation of the vector X_j. Equations (1.4), (1.5) and (1.6), make up the computational steps of the permutation accuracy importance measure and can be summarized in a short schematic view:



1. Compute the OOB accuracy of a tree.
2. Permute the predictor variable of interest in the OOB observations.
3. Recompute the OOB accuracy of the tree.
4. Compute the difference between the original and recomputed OOB accuracy.
5. Repeat step 1 to 4 for each tree.
6. The overall importance score is given by the average difference.

It can be read off equation (1.6) that the importance measure is based on the computation of the mean squared error (MSE) before and after permutation. Equation (1.5) inherents the accuracy which equals 1 - error rate = 1 - MSE for binary responses. This interpretation in terms of MSE is commonly used; as retraced in Nicodemus et al. (2010) for instance. Thus both measures are based on a statistic that is solely and directly derived from the response and the variable space. Good (2000) denotes the selection of an appropriate statistic as step three in his listing of 'five steps to a permutation test'. The quality of such a statistic is supposed to be its ability to discriminate between the null-hypothesis and the alternative. The MSE clearly satisfies this quality. Further examples of suitable statistics for permutation tests are given by Good (2005).

An example of permutation importance measures for the airquality data is given by Figure 1.8. As the original measure is not applicable to missing data these results emerge from a complete case analysis. The next chapter introduces a new kind of importance measure which is able to deal with missing data and circumvents dangers associated with complete case analysis.

1.3 Statistical Software

All analyses and computations presented in this work were performed with the `R` system for statistical computing (R Development Core Team, 2011, version 2.14.1). It provides several freely available implementations of recursive partitioning: The CART algorithm is given by the function `rpart()` of the package `rpart` (Therneau et al., 2011, version 3.1-52) and another implementation by `tree()` which is part of the `tree` package (Ripley, 2011, version 1.0-29). In this work `rpart()` is used to represent the CART algorithm. Conditional inference trees and an implementation of conditional inference based Random Forests – called by `ctree()` and `cforest()` – are both included in the `party` package (Hothorn et al., 2008, version 1.0-0). Unfortunately the CART related implementation of Random Forests given by the function `randomForest()` in the package `randomForest` (Liaw and Wiener, 2002, version 4.6-6) does not support the fitting of Random Forests

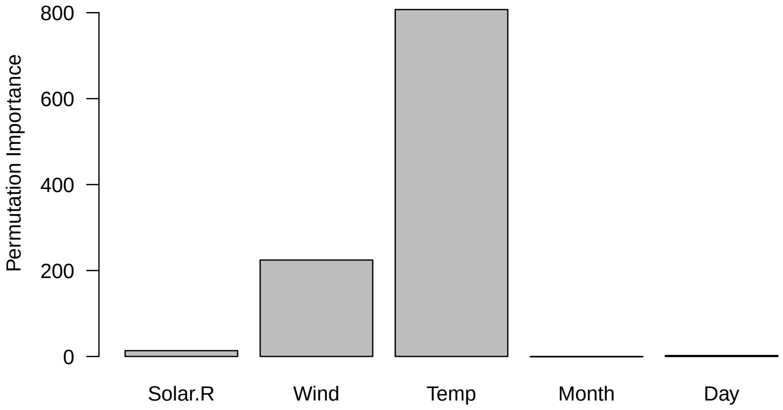


Figure 1.8: Permutation accuracy importance for a complete case analysis of the airquality data.

to incomplete data. However, as the occurrence of missing values is in main focus of the investigations of this work and even more importantly: as it has been discussed in this chapter that the algorithm is prone to biased variable selection it is not used any further. Thus, Random Forests are executed by the function `cforest()`. A short summary of functions used in this work is given in Table 1.2.

In this chapter the default settings for each function were used to perform the exemplary analyses. In the following chapters specific settings used for the analyses will be listed separately.

Method	Model	Function	Package	Used
CART	tree	`tree()`	`tree`	
		`rpart()`	`rpart`	✓
	Random Forest	`randomForest()`	`randomForest`	
conditional	tree	`ctree()`	`party`	✓
inference	Random Forest	`cforest()`		✓

Table 1.2: Summary of functions used to perform recursive partitioning.

Chapter 2

Predictive Accuracy for Data with Missing Values

2.1 Research Motivation and Contribution

The occurrence of missing values is a major problem in statistical data analysis. All scientific fields and data of all kinds and size are touched by this problem. A popular approach to handle missing values is the application of imputation methods (see Schafer and Graham, 2002; Horton and Kleinman, 2007, for a summary of approaches). There is a number of ad-hoc solutions – e.g. available case and complete case analysis as well as single imputation by mean, hot-deck, conditional mean and predictive distribution substitution – which can lead to a loss of power, biased inference, underestimation of variability and distorted relationships between variables (cf. Groenwold et al., 2012, for corresponding discussions about the proper analysis of missing outcome data in randomized trials and observational studies). A more promising approach of rising popularity is multiple imputation by chained equations (MICE) also known as imputation by full conditional specification (FCS) (van Buuren et al., 2006; White et al., 2011). It allows for the imputation of multivariate data without the need to specify a joint distribution of predictor variables. Furthermore, its superiority to ad hoc and single imputation methods has been shown by many publications (e.g. Janssen et al., 2009, 2010). Alternatives to imputation are given by methods with built-in procedures to handle missing values. This includes recursive partitioning by classification and regression trees as well as Random Forests.

However there is only a few publications that compare the two approaches. Two reference publications that investigate performance differences are given by Feelders (1999) and Farhangfar et al. (2008). Unfortunately, they lack generalizability as investigations are restricted to classification tasks, categorical data and special simulation schemes. A third related paper that focuses on Random Forests is given by Rieger et al. (2010). It is based on much more extensive simulation studies that involve different missing data generating processes (section 2.3.1) for classification and regression problems.

The goal of this chapter is to compare the predictive accuracy of CART, conditional inference trees and Random Forests when surrogates and multiple imputation are used to handle missing values. Comparative analyses for various datasets and different simulation settings are designed to improve and extend the investigations of related publications. Both classification and regression problems are examined. Findings show that multiple imputation produces ambiguous performance results for both simulation studies and empirical evaluations. By contrast, the use of surrogates is a fast and simple way to achieve performances which are often only negligibly worse and in some cases even superior. The investigations and findings of this chapter have been published in Hapfelmeier et al. (2011).

2.2 Discussion of Related Publications

- Feelders (1999) favors the application of imputation methods. This conclusion is based on the investigation of two classification problems. The `rpart()` routine implemented in `S` (Becker, 1984), which closely resembles the CART algorithm proposed by Breiman et al. (1984) was applied. Procedures were compared by an assessment of the misclassification error rate (MER) which equals the fraction of wrong predictions in the case of a binary outcome.

  One of the examined datasets is the Pima Indians Diabetes Data Set (section 2.4.2). The MER of a tree that used surrogate splits was 30.6%. Single imputations based on EM-estimates were repeated by ten independent draws and achieved an averaged MER of 26.8%; Little and Rubin (2002) clearly show that the variability of estimates is likely to be underestimated by single imputation. Thus comparisons and tests within each of the repetitions might be invalid. In a second experiment a multiple imputation approach was applied ten times. The averaged MER equals 25.2%. To back up the observed differences an exact binomial test was computed for each repetition. In the first experiment there were 6 of 10 and in the second experiment there were 9 of 10 p-values below 0.05. Nevertheless a test for the comparison of two proportions like the McNemar-Test would have been more appropriate. In addition only the training data contained incomplete observations.

  The second data is the waveform recognition data originally used by Breiman et al. (1984). Missing values were introduced completely at random in the training data in fractions between 10% and 45%. The imputation was performed by an LDA model based on EM-estimates. The MER of trees was assessed in two experiments which differed by the application of single imputation and multiple imputation. For the former the MER of a tree fit to imputed data was between 29.2% and 30.6%, seemingly unrelated to the fraction of missing values. Trees that used surrogate splits produced MER values between 29.8% and 34.3%. Results were similar for multiple imputation. The MER of trees with imputation lay between 25.5% and 26.1%. With surrogates the MER increased from 28.9% to 35.6%. Differences became more and more pronounced with high fractions of missing values. However, 45% missing values in each variable is rather rare in real life data. A data set of only 5 variables would already include $1 - (1 - 0.45)^5 = 95.0\%$ incomplete observations if the locations of missing values are statistically independent. Likewise an equal spread of missing data is rather artificial.

- Farhangfar et al. (2008) published a profound comparison of various classification methods applied to data with missing values. Several single imputation methods and a multiple imputation approach by polytomous regression using the MICE algorithm were explored. Classification models were support vector machines (SVM), k-nearest neighbors (kNN), C4.5 (a decision tree algorithm introduced by Quinlan, 1993), among others. Missing values were induced into 15 completely observed datasets which exclusively consisted of qualitative variables. Results showed that the application of MICE, compared to other imputation methods, leads to superior results in most cases. For none of the data sets the C4.5 method benefited from imputation. By contrast, the latter even led to worse MER values in some cases. The performance of C4.5 was also independent of the amount of missing values. Like Feelders (1999) the authors restricted the occurrence of missing values to the training data. The problem of too many missing values equally spread among the variables was present, too. Up to 50% of observations per variable were set missing.

- In an extensive simulation study Rieger et al. (2010) concluded that the application of a k-nearest neighbors (kNN) imputation approach did not improve the performance of conditional Random Forests. Classification and regression problems with three different correlation structures and seven schemes to generate missing values were investigated. These studies were repeated for high-dimensional settings with additional noise variables and for two scenarios that differed by the introduction of missing values in the training and test data or solely in the training data. The fraction of missing values was not varied and chosen to be two times 20% and one time 10% in three variables. The comparison of approaches was based on prediction accuracy measured by binomial log-Likelihood and mean squared error (MSE). Results showed no clear advantage of imputation. Despite elaborate simulation settings the authors point out that results may not be generalizable due to specific choices of parameters. However, this publication does not incorporate trees, uses a single imputation method and does not vary fractions of missing values.

Feelders (1999) showed increased MER for an increasing number of missing values when single trees are used with surrogate splits. Meanwhile the MER of trees based on imputation almost did not change. Differences between methods were rather weak for lower fractions of missing values which are more likely to be observed in real life data. Farhangfar et al. (2008) found no improvement for C4.5 Trees with imputed data. They even claim a harmful effect of imputation in this case. Pitfalls and drawbacks of the former two publications are unrealistic simulation schemes, invalid test procedures, the application of biased imputation and tree building methods and the limited generalizability due to the predominant examination of nonstandard polytomous data and classification tasks only.

By contrast the work of Rieger et al. (2010) resolves many of these issues as it presents an extensive simulation study for classification and regression problems. The authors conclude that a k-nearest neighbor imputation approach was not able to improve the performance of Random Forests.

2.3 Missing Data

2.3.1 Missing Data Generating Processes

In an early work Rubin (1976) specifies the issue of correct statistical inference from data containing missing values. A key instrument is the declaration of the process that causes the missingness. Based on these considerations many strategies for inference and elaborate definitions of the subject have been developed. An extensive summary can be found in Schafer and Graham (2002). In general, three types of missingness are distinguished:

- 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 status of missingness (yes = 1/no = 0) is indicated by a binary random variable R and its probability distribution $P(R)$. The letter R, that was adopted from the original notation, may emerge from the fact that Rubin (1987) originally was dealing with 'R'esponse rates in surveys. The complete variable space $\mathbf{X}_{\text{comp}}$ is made up of observed $\mathbf{X}_{\text{obs}}$ and missing $\mathbf{X}_{\text{mis}}$ parts; $\mathbf{X}_{\text{comp}} = \{\mathbf{X}_{\text{obs}}, \mathbf{X}_{\text{mis}}\}$. Therefore MCAR indicates that the probability of observing a missing value is independent of the observed and unobserved data. By contrast for MAR this probability is dependent on the observed values (but not on the missing values themselves). Finally in MNAR the probability depends on unobserved information or the missing values themselves. An example for the latter is a study in which subjects with extreme values for an outcome systematically drop out while there is no information in the data that could help to explain this discontinuation.

Farhangfar et al. (2008) outline that in practice the MCAR scheme is assumed for most imputation methods. He et al. (2009) and White et al. (2011) point out that the MICE algorithm is also capable of dealing with MAR schemes as the imputation model becomes more general and includes more variables. In this situation it becomes more probable that missing values can be explained by observed data. The latter property is especially valuable for data that already contain missing data. In such settings it is not clear which scheme really holds. Similar statements can be found in Janssen et al. (2010) which claims that even a false assumption of MAR under MNAR has minor impact on results in many realistic cases. The performance of Random Forests under several MAR schemes was investigated by Rieger et al. (2010). These authors compared the use of surrogates against a single imputation method. In extensive simulation studies they were able to show that results did not differ between MCAR and MAR. For all these reasons the introduction of missing data is done in a MCAR scheme in the following simulation studies.

2.3.2 Multivariate Imputation by Chained Equations

Using MICE, imputation is performed by flexible specifications of predictive models for each variable. There is no need to determine any joint distributions of the data. Cycling through incomplete variables iteratively updates imputations until convergence. Repeating the procedure several times leads to multiple imputed data sets. A short summary of theory and appealing properties is given in the following.

Multiple Imputation

A simple and popular approach to handle missing data is the application of multiple imputation (MI) as introduced by Rubin (1987, 1996). Little and Rubin (2002) point out that an apparent advantage of this approach is its ability to make standard complete-data methods applicable to incomplete data. Therefore the user is able to stick to his preferred method of analysis. There is no necessity to switch to one he is not used to, he does not understand or is known to be less powerful.

In general any measure of interest Q (e.g. parameter estimates $\widehat{\theta}$ or response predictions $\widehat{\mathrm{y}}$) is assessed by the average

$$\overline{\mathrm{Q}}_{\mathrm{E}} = \frac{1}{\mathrm{E}} \sum_{\mathrm{e}=1}^{\mathrm{E}} \widehat{\mathrm{Q}}_{\mathrm{e}}$$

using E estimates $\widehat{\mathrm{Q}}_{\mathrm{e}}$ derived from the imputed complete data sets. The total variability of the estimate is given by

$$\mathrm{T}_{\mathrm{E}} = \overline{\mathrm{W}}_{\mathrm{E}} + \frac{\mathrm{E}+1}{\mathrm{E}} \mathrm{B}_{\mathrm{E}}$$

where

$$\overline{\mathrm{W}}_{\mathrm{E}} = \frac{1}{\mathrm{E}} \sum_{\mathrm{e}=1}^{\mathrm{E}} \widehat{\mathrm{W}}_{\mathrm{e}} \quad \text{and} \quad \mathrm{B}_{\mathrm{E}} = \frac{1}{\mathrm{E}-1} \sum_{\mathrm{e}=1}^{\mathrm{E}} (\widehat{\mathrm{Q}}_{\mathrm{e}} - \overline{\mathrm{Q}}_{\mathrm{E}})^2$$

are the average of the within-imputation variances $\widehat{\mathrm{W}}_{\mathrm{e}}$ and the between-imputation variance, respectively. Of course the essential preceding step is the creation of E imputed data sets. If imputation was only done once, like in single imputation, the imputed values would be treated like they were known. This can lead to a severe underestimation of the variance, 'which affects confidence intervals and statistical tests' as stated by Harel and Zhou (2007). However, it is not sufficient to simply create more than 1 dataset by drawing from the conditional distribution $\mathrm{P}(\mathbf{X}_{\mathrm{mis}}|\mathbf{X}_{\mathrm{obs}}, \widehat{\theta})$. The uncertainty inherent in the estimate $\widehat{\theta}$ itself has to be incorporated, too. The posterior predictive distribution of $\mathbf{X}_{\mathrm{mis}}$ is

$$\mathrm{P}(\mathbf{X}_{\mathrm{mis}}|\mathbf{X}_{\mathrm{obs}}) = \int \mathrm{P}(\mathbf{X}_{\mathrm{mis}}|\mathbf{X}_{\mathrm{obs}}, \theta)\mathrm{P}(\theta|\mathbf{X}_{\mathrm{obs}})\mathrm{d}\theta$$

with

$$\mathrm{P}(\theta|\mathbf{X}_{\mathrm{obs}}) \propto \mathrm{P}(\theta) \int \mathrm{P}(\mathbf{X}_{\mathrm{obs}}, \mathbf{X}_{\mathrm{mis}}|\theta)\mathrm{d}\mathbf{X}_{\mathrm{mis}}$$

denoting the observed-data posterior distribution of θ. A proper multiple imputation approach is supposed to first draw E estimates $\theta^{(1)}, ..., \theta^{(\mathrm{E})}$ from $\mathrm{P}(\theta|\mathbf{X}_{\mathrm{obs}})$. These are subsequently used in the conditional distributions $\mathrm{P}(\mathbf{X}_{\mathrm{mis}}|\mathbf{X}_{\mathrm{obs}}; \widehat{\theta}^{(\mathrm{e})})$, $\mathrm{e} = 1, ..., \mathrm{E}$.

An example of this procedure was taken from Rubin (1987) and White et al. (2011) to illustrate the case of parameter estimates $\widehat{\theta} = \widehat{\beta}$ for a linear regression model. Drawing from its conditional distribution does not consider the uncertainty about the maximum likelihood estimate $\widehat{\beta}$. Thus $\widehat{\beta}$ needs to be drawn from its posterior distribution $\mathrm{P}(\beta|\mathbf{X}_{\mathrm{obs}})$, too. Under the assumption of ignorable nonresponse the estimation is based on the observed data only. In a first step a linear regression is fit to the observed data which gives

$$\widehat{\beta}_{\mathrm{obs}} = \underbrace{\left(\mathbf{x}_{\mathrm{obs}}^{\top}\mathbf{x}_{\mathrm{obs}}\right)^{-1}}_{\mathbf{V}} \mathbf{x}_{\mathrm{obs}}^{\top}\mathrm{y}_{\mathrm{obs}}$$

and

$$\widehat{\sigma}^2_{\mathrm{obs}} = \left(\left[(\mathrm{y}_{\mathrm{obs}} - \mathbf{x}_{\mathrm{obs}}\widehat{\beta}_{\mathrm{obs}})^2\right]^{\top} \mathbf{1}\right) / (\mathrm{n}_{\mathrm{obs}} - \mathrm{n}_{\mathrm{par}}).$$

$\mathrm{n}_{\mathrm{par}}$ is the dimension of β and $\mathrm{n}_{\mathrm{obs}}$ the number of observed values. When the prior distribution on *log* σ is proportional to a constant it can be shown that $\sigma^2/\widehat{\sigma}^2$ follows an inverted χ^2 distribution on $\mathrm{n} - 1$ degrees of freedom. Based on these considerations the imputation starts with the computation of

$$\sigma^2_* = \widehat{\sigma}^2_{\mathrm{obs}}(\mathrm{n}_{\mathrm{obs}} - \mathrm{n}_{\mathrm{par}})/\mathrm{g}$$

where g is a random number drawn from $\chi^2_{\mathrm{n}_{\mathrm{obs}} - \mathrm{n}_{\mathrm{par}}}$. Using the estimates σ^2_* and $\widehat{\beta}_{\mathrm{obs}}$, one is able to compute

$$\beta_* = \widehat{\beta}_{\mathrm{obs}} + \frac{\sigma_*}{\widehat{\sigma}_{\mathrm{obs}}}[\mathbf{V}]^{1/2}\mathrm{z}$$

where z is a vector that contains $\mathrm{n}_{\mathrm{par}}$ independent draws from a standard normal distribution. $[\mathbf{V}]^{1/2}$ is the Cholesky decomposition of $\mathbf{V}$. Finally,

$$\mathrm{y}_{\mathrm{mis}} = \mathbf{x}_{\mathrm{obs}}\beta_* + \mathrm{z}_*\sigma_*.$$

Again z_* is a vector of n_{mis} independent random draws from a standard normal distribution. This procedure is repeated to generate several imputed data sets.

MICE

Whenever there is more than one variable with missing values the imputation approach needs to be adapted. There are mainly two approaches for missing data imputation in this case. Joint modeling (JM) approaches, as presented by Schafer (1997), are not discussed in detail here. Still it is worth mentioning that imputations are directly drawn from the parametric multivariate density $\mathrm{P}(\mathbf{X}_{\mathrm{mis}}, \mathbf{X}_{\mathrm{obs}}, \mathrm{R}|\theta)$ by this approach. Appropriate methods exist for the multivariate normal, log-linear and general location model. A more practical approach which makes it possible to bypass the specification of a joint distribution is MICE also known as imputation by fully conditional specification (FCS) (cf. van Buuren, 2007; White et al., 2011). Although it lacks profound theory van Buuren et al. (2006) showed in simulation studies that MICE produces reasonable imputations and coverages of statistics of concern. FCS, using linear regression, even equals JM under the multivariate normal joint distribution, given specific regularity conditions. The same holds for some special cases of the log linear model. According to van Buuren and Groothuis-Oudshoorn (2010) FCS is an attempt to obtain a posterior distribution of θ by chained equations. These authors state that starting with the imputation of missing values by random samples of the observed values the tth iteration of the chained equations is

$$\begin{aligned}
\theta_1^{\mathrm{t}} &\sim \mathrm{P}(\theta_1|\mathrm{X}_{1,\mathrm{obs}}, \mathrm{X}_2^{\mathrm{t}-1}, ..., \mathrm{X}_{\mathrm{v}}^{\mathrm{t}-1}),\\
\mathrm{X}_{1,\mathrm{mis}}^{t} &\sim \mathrm{P}(\mathrm{X}_1|\mathrm{X}_{1,\mathrm{obs}}, \mathrm{X}_2^{\mathrm{t}-1}, ..., \mathrm{X}_{\mathrm{v}}^{\mathrm{t}-1}, \theta_1^{\mathrm{t}}),\\
&\vdots\\
\theta_{\mathrm{v}}^{\mathrm{t}} &\sim \mathrm{P}(\theta_{\mathrm{v}}|\mathrm{X}_{\mathrm{v},\mathrm{obs}}, \mathrm{X}_1^{\mathrm{t}}, ..., \mathrm{X}_{\mathrm{v}-1}^{\mathrm{t}}),\\
\mathrm{X}_{\mathrm{v},\mathrm{mis}}^{\mathrm{t}} &\sim \mathrm{P}(\mathrm{X}_{\mathrm{v}}|\mathrm{X}_{\mathrm{v},\mathrm{obs}}, \mathrm{X}_1^{\mathrm{t}}, ..., \mathrm{X}_{\mathrm{v}-1}^{\mathrm{t}}, \theta_{\mathrm{v}}^{\mathrm{t}}),
\end{aligned}$$

where $\mathrm{X}_{\mathrm{j}}^{\mathrm{t}}$ is the jth imputed variable at iteration t. It is easy to see how turns are taken within the iterative steps to infer θ and $\mathbf{X}_{\mathrm{mis}}$. After the convergence of the algorithm it is possible to draw $\widehat{\theta}$ from its posterior and to use it to obtain $\widehat{\mathbf{X}}_{\mathrm{mis}}$. Several imputed datasets are produced by repeating the procedure with different starting values. A practical advantage of MICE are the many possibilities to model $\mathrm{P}(\mathrm{X}_{\mathrm{j}}|\mathrm{X}_{\mathrm{j},\mathrm{obs}}, \mathbf{X}_{-\mathrm{j}}, \theta_{\mathrm{j}})$. A profound discussion can be found in van Buuren and Groothuis-Oudshoorn (2010) and White et al. (2011). MICE is especially suitable in MAR settings although Janssen et al. (2010) state that it should also be preferred to ad hoc methods like complete case analysis even in MNAR situations. In a review paper of epidemiologic literature Klebanoff and Cole (2008) conclude that MICE is still of minor popularity. Despite its outstanding benefits compared to simpler ad hoc methods it seems like researchers feel uncomfortable to use it. Thus they give recommendations about the proper publication of multiple imputation methods to increase their popularity. These are followed in this chapter and outlined in section 2.4.3.

To date there are still proposals for further developments. For example Burgette and Reiter (2010) claim that complex distributions as well as interactions and nonlinear relations might better be fit using CART as imputation model within the MICE algorithm. They are able to present promising results in a simulation study and an application to real life data. Likewise, Stekhoven and Bühlmann (2011) present an iterative approach that uses Random Forests to impute missing values and show its superior performance in empirical evaluations; especially when there are complex interactions and nonlinear relations. Templ et al. (2011) emphasize that many imputation approaches '...assume that the data originate from a multivariate normal distribution...'. They suggest an iterative robust model-based imputation procedure to deal with data that deviate from this assumption (e.g. data that contain outliers or originate from skewed or multimodal distributions). Accordingly, future studies should take the diversity of MICE approaches into account.

2.4 Studies

In order to explore several kinds of data which display a random sample of real life situations with a wide range of attributes there were no constrictive exclusion criteria applied for their selection. There are 12 datasets; half of them were supposed to contain regression and half of them classification problems. Four datasets without any missing values were used for a simulation study and the remaining eight datasets, that contained missing values in advance, were used for an application study. There were no restrictions about the number of observations or variables and the amount of missing values. The datasets were included without any prior knowledge of these characteristics or any presumptions about potential findings. Therefore a broad set of data emerging from different scientific fields was used.

There are two kinds of studies. The first one is meant to retrace and extend the simulations discussed in section 2.2. The introduction of missing values was varied for a deeper insight to effects caused by special schemes. The second study is based on empirical evaluations of data that already contain missing values. It is supposed to provide a more reliable assessment of potential benefits of imputation without the need of artificial specifications for the simulation settings. To stick close to existing publications, the performance assessment is given by the mean squared error (MSE). This measure equals the misclassification error rate (MER) for binary responses.

Evaluation is done by Monte-Carlo Cross-Validation (MCCV; see Boulesteix et al., 2008b). Analyses are performed for two cases that differ by the decision to use imputation through MICE or surrogates. It is well known that for valid evaluations the test data need to be isolated from the training data in every aspect. This can only be achieved if two separate imputation models are fit to each of these datasets. Otherwise it is believed that the MSE estimation would be positively biased as an imputation model fit to the training data and applied to the test data would transfer information. One could not call the observations of the test data 'unseen' to the predictive model any more. Consequently, separate imputation models were fit to the training and test dataset. Of course the response is also not allowed to be part of the imputation model for the test data. It is considered unknown until the comparison of predicted and real outcomes for evaluation purposes.

2.4.1 Simulation Studies

Analysis settings

Data were MCAR in the simulation studies. Thus each value had the same probability to be missing independent of any observed or unobserved information. Four datasets were chosen for the simulation studies as they were fully observed. Two of them are used for the classification of a binary response. Another two are used for regression. Missing values were artificially introduced in a procedure which is close to the one of Feelders (1999) and Farhangfar et al. (2008). However, the introduction of missing data was not restricted to the training set but extended to the test set, too. Fractions of missing values are 0% (benchmark), 10%, 20%, 30% and 40%. The procedure was repeated 1000 times using Monte-Carlo Cross-Validation (MCCV). In each iteration a random sample of 80% was used for the training of a model while the remaining 20% served as test set. This also facilitated the separation of imputation models for the training and test data.

Part of the criticism in section 2.2 was about the huge amount of missing values which is beyond the fractions found in most of real life data; the number of observations that contain at least one missing value is $1 - (1 - \%_{\text{missing}})^{\text{n}_{\text{variables}}}$ which, on average, already makes $1 - (1 - 0.40)^5 = 92.2\%$ incomplete observations for a dataset that contains only 5 variables with 40% missing values. To stick closer to real life situations the simulation was repeated with a randomly chosen third of the variables partly set missing in each MCCV step. The corresponding R-Code is given in the appendix B.2.1.

Data

A summary of the data is given by Table 2.1 and the following listing:

- **Haberman's Survival Data** contains data about the 5 year survival of patients after breast cancer surgery. It was originally used for investigations of log-linear models by Haberman (1976). The corresponding study was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital. There are 306 observations in 3 independent variables namely age, year of operation and number of positive axillary nodes. The survival status of a patient was used as the outcome of a classification problem.

- The **Heart Disease Data** was collected at four clinical institutions. These are the Cleveland Clinic Foundation, the Hungarian Institute of Cardiology, Budapest, the V.A. Medical Center, Long Beach, CA and the University Hospital, Zurich, Switzerland. It contains information about the incidence of heart disease along with the assessment of a patients age, gender, chest pain type, resting blood pressure, serum cholestoral in mg/dl, a fasting blood sugar measurement (> 120 mg/dl), resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels (0–3) colored by flourosopy and thallium scan status information. These socioeconomic data and clinical outcomes were used to classify the binary outcome, i.e. heart disease. The data contain 270 observations and 13 independent variables.

- The **Swiss Fertility and Socioeconomic Indicators Data** contains a standardized fertility measure and socio-economic indicators. It was originally used for regression analysis by Mosteller and Tukey (1977). Features are percentages of males involved in agriculture, draftees receiving highest mark on army examination, draftees with education beyond primary school, catholic population and the infant mortality within the first year of life. Data were gathered in 47 French-speaking provinces of Switzerland at about 1888; this makes 47 observations in 5 independent variables. The regression aims at the explanation of the standardized fertility measure in each province.

- The **Infant Birth Weight Data** was collected at the Baystate Medical Center, Springfield, Mass during the year 1986. It contains physical measures and information about the health condition of women giving birth. Venables and Ripley (2003) originally used it for a classification of the binary outcome, low and high birth weight of newborns. Here, by contrast, a regression was performed to predict a child's birth weight in grams by the means of a mother's health status and history. The latter is given by a mother's age in years, weight in pounds at last menstrual period, race, smoking status during pregnancy, number of previous premature labors, history of hypertension, presence of uterine irritability and the number of physician visits during the first trimester. The data contain 189 observations and 9 independent variables.

It has to be pointed out that for the Heart Disease Data and the Infant Birth Weight Data the number of predictor variables was reduced to 12 and 8, respectively. This was due to computational issues with the MICE implementation, which might be seen as another disadvantage of this approach, but only touches the analyses of this chapter.

Data	Obs.	ind. Var.
H. Survival	306	3
Heart	270	13
Fertility	47	5
Birthweight	189	9

Table 2.1: Count of observations and independent variables for datasets used in the simulation studies.

2.4.2 Empirical Evaluation

Analysis settings

Although simulation studies might be helpful to investigate theoretical properties there are some deficiencies in the simulation schemes that clearly limit generalizability of results. Corresponding statements have been discussed in section 2.2. Thus, the main focus of this chapter is put on empirical evaluations of data that originally contain missing values. A total of eight datasets has been used to explore four classification and four regression problems. Again, each data was split in 1000 MCCV runs into 80% training and 20% test observations to estimate a method's performance in terms of MSE. The corresponding R-Code is given in the appendix B.2.2.

Data

The eight datasets summarized in Table 2.2 are:

- The **Hepatitis Dataset** contains information about 155 patients that suffered from hepatitis and of whom 32 died. A total of 19 independent variables is available for the classification problem to predict a patient's survival. These variables include demographic data like sex and age, information about drug intake like steroids and antivirals and further clinical factors. One missing value was observed in 4 variables, 5 missing values in 4 variables, and 4, 6, 10, 11, 16 and 29 missing values in one variable, respectively. Therefore missing values were present in 14 out of 18 variables. The fraction of missing values per variable ranges from 0.6% to 18.7%. In total 43 (27.7%) observations contain at least one missing value.

- The **Mammographic Mass Data** is made up of several features extracted from breast cancer screenings. The latter are performed by physicians, especially radiologists, who try to determine the severity (benign or malign) of a suspicious lesion. In the recent past efforts have been made to solve the classification problem by machine learning approaches. The resulting systems are called CAD (Computer Aided Decision/Detection) systems. The data were originally used by Elter et al. (2007) for the evaluation of such systems (c.f. Hapfelmeier and Horsch, 2011, for corresponding evaluation studies). Analyses were performed to describe the severity status of a lesion. The data also contain information about the four independent variables age, shape, margin and density of mass lesions observed in 961 women. Age contains 5 missing values while shape, margin and density are missing 31, 48 and 76 times, respectively. The corresponding fractions of missing values are 0.5%, 3.2%, 5.0% and 7.9%. Overall 130 (13.5%) observations contain at least one missing value.

- The **Pima Indians Diabetes Dataset** was also used for the comparison of trees with and without imputation by Feelders (1999). It contains information about the diabetes disease of 768 pima indian women which are at least 21 years old. In addition to age, the number of pregnancies, plasma glucose concentration, diastolic blood pressure, triceps skin fold thickness, 2-Hour serum insulin, BMI and diabetes pedigree function were recorded. This makes 8 independent variables used for the classification problem to determine whether a women shows signs of diabetes according to the WHO definition.

At first glance these data do not seem to contain any missing values. However, the missing values are actually "hidden" behind many zero values that are biologically implausible or impossible. Pearson (2006) calls this situation "disguised missing data" and gives a profound discussion about its occurrence in the Pima Indians Diabetes Data Set. According to his description, there are five variables that contain missing data. The total numbers of missing values in these variables are 5, 35, 227, 374 and 11, which equals fractions of 0.7%, 4.6%, 29.6%, 48.7% and 1.4%. Overall 376 (49.0%) observations contain at least one missing value.

- The **Ozone Level Detection Dataset** was collected from 1998 to 2004 at the Houston, Galveston and Brazoria area and contains information about geographic measures and ozone levels. The classification problem is to distinguish days of high and low ozone concentration based on information about wind speed, temperature, solar radiation etc. In total there are 2534 observations in 73 measured features. Each of the variables contains between 2 (0.08%) and 300 (11.8%) missing values. For the entire data 687 (27.1%) observations contained at least one missing value.

- The **Airquality Dataset** contains daily measurements of the air quality in New York from May to September 1973. There are four variables that were of interest for the planned analyses. These are the ozone pollution in parts per billion (ppb), the solar radiation in Langleys (lang), the average wind speed in miles per hour (mph) and the maximum daily temperature in degrees Fahrenheit (degrees F). The ozone data were originally provided by the New York State Department of Conservation and the meteorological data by the National Weather Service. A more detailed explanation of the data can be found in Chambers (1983). All of the metric variables were used to predict the temperature in a regression problem. For 153 days the ozone pollution contains 37 (24.2%) missing values while the solar radiation contains 7 (4.6%). There are 42 (27.5%) observations that have at least one missing value.

- The **El Nino Dataset** was gained from the Tropical Atmosphere Ocean (TAO) array of the international Tropical Ocean Global Atmosphere (TOGA) program. TAO is an assemblage of ca. 70 moored buoys that record oceanographic and surface meteorological variables in the equatorial pacific. The present data contain information about four independent variables, i.e. the sea surface temperature, air temperature, humidity as well as zonal and meridional wind speeds. The regression problem was to predict the sea surface temperature from the remaining variables. There are 733 observations of which 78 (10.6%) and 91 (12.4%) are missing for the air temperature and the humidity, respectively. For the entire data 168 (22.9%) observations contained at least one missing value.

- The **CHAIN Project Data** contains information from a longitudinal cohort study of HIV infected people living in New York City by 1994. It was originally used by Messeri et al. (2003) for the assessment of HIV treatment effects on survival. For the planned analyses there were 508 observations of seven variables. These are the log of self reported viral load level, age at time of interview, family annual income, a continuous scale of physical health, the CD4 count, a binary measure of poor mental health and an indicator for the intake of HAART. The regression problem was to explain the continuous scale of physical health. There were 155 missing values in the self reported viral load level, 14 in the family annual income and 39 in the CD4 count. This equals fractions of 30.5%, 2.8% and 7.7%. At least one missing value in any variable was observed for 173 (34.1%) observations.

- The **Mammal Sleep Data** contain features of 62 species ranging from mice over opposums and elephants to man. It was originally used by Allison and Cicchetti (1976) to examine relations between sleep, ecological influences and constitutional characteristics. The observed sleep features include information about duration and depth of sleep phases as well as the occurrence of dreams. Constitution is given by measures like body weight and brain weight. The safety of sleep is assessed by scaling for overall danger, danger for being hunted, sleep exposure, gestation time etc. One of the main findings in the original paper was a negative correlation between slow-wave sleep and body size. In alignment with these investigations the data was used in a regression analysis for the prediction of body weight. The data contains 9 independent variables and 62 observations. There are 20 (32.3%) observations which are not completely observed for all variables. It is interesting to note that Allison and Cicchetti (1976) had originaly chosen a complete case analysis as they found the incomplete data to be "... not suitable for the multivariate analyses ...". There are five variables containing 4 (3 times), 12 and 14 (6.5% (3 times), 19.4% and 22.6%) missing values.

Most of the data is provided by the open source UCI Machine Learning Repository (Frank and Asuncion, 2010). However, the Fertility and Airquality Data were taken directly from the `R` routine. Likewise, the Birthweight, CHAIN Project and Sleep Data are part of the `R` packages `MASS`, `MI` and `VIM`, respectively.

2.4.3 Implementation

All analyses were performed with the `R` software for statistical computing (R Development Core Team, 2011, version 2.14.1). The CART algorithm is provided by the function `rpart()` which is part of the equally named package `rpart` (Therneau et al., 2011, version 3.1-52). It is opposed to conditional inference trees called by `ctree()` which is part of the `party` package (Hothorn et al., 2008, version 1.0-0). This package also includes the function `cforest()` which is used for the implementation of Random Forests. Unfortunately the function `randomForest()` in the package `randomForest` (Liaw and Wiener, 2002, version 4.6-6) does not support the fitting of Random Forests to incomplete data. Thus,

	Observations			Variables		
Data	#	≥ 1 missing		#	missing	per Var.
Hepatitis	155	43	(27.7%)	19	14	(0.6% – 18.7%)
Mammo	961	130	(13.5%)	4	4	(0.5% – 7.9%)
Pima	768	376	(49.0%)	8	5	(0.7% – 48.7%)
Ozone	2534	687	(27.1%)	73	73	(0.1% – 11.8%)
Airquality	153	42	(27.5%)	4	2	(4.6% – 24.2%)
El Nino	733	168	(22.9%)	4	2	(10.6% – 12.4%)
CHAIN	508	173	(34.1%)	7	3	(2.8% – 30.5%)
Sleep	62	20	(32.3%)	9	5	(6.5% – 22.6%)

Table 2.2: Characteristics of datasets used for the empirical evaluation. The number of independent variables and observations is given in addition to absolute and relative frequencies of missing values among them.

this biased, CART based version could not be used for comparison matters. Multivariate Imputation by Chained Equations was done by the `mice()` function of the `mice` package (van Buuren and Groothuis-Oudshoorn, 2010, version 2.11).

The number of trees in Random Forests was set to be ntree = 500. Each split was chosen from mtry = min(5, variables available) randomly selected variables. Trees and Random Forests use maxsurrogate = min(3, variables available) surrogate splits. MICE produces five imputed datasets. A normal linear model was used for the imputation of continuous variables, logistic regression for binary variables and a polytomous regression for variables with more than two categories: defaultMethod = c(”norm”, ”logreg”, ”polyreg”). Concerning the training data each variable contributed to the imputation models. In the test data the response was excluded from these models. The fraction of imputed values and number of variables used for imputation can be read off Table 2.1 and 2.2.

2.5 Results

2.5.1 Simulation Studies

The following contains discussions for each of the four investigated datasets. A corresponding graphical representation is given by Figure 2.1. A summary of observed MSE values is presented by Table 2.3 and an even more elaborate listing is given in the appendix A.1.1.

- **Haberman’s Survival Data** is used to predict the 5 year survival of patients after breast cancer surgery. Random Forests, `ctree` and `rpart` perform comparably. They are able to preserve the benchmark MSE (obtained for 0% missing values) independent of the procedure to handle missing values. The relative improvement by MICE instead of surrogates ranges from -2% to 5% and gets even less pronounced (-2% to 1%) when only one third of variables contains missing values.

- By means of the **Heart Disease Data** it is assessed how well the presence of heart disease can be predicted. In terms of prediction accuracy, Random Forests outperforms both single tree methods, while `rpart` produces slightly superior results than `ctree`. An increased number of missing values makes error rates rise especially when they are introduced in each variable. The relative improvements by MICE shrink from a range of 0% to 22% to a range of -5% to 2% when only one third of variables contain missing values.

- The **Swiss Fertility and Socioeconomic Indicators Data** is used to examine whether a continuous fertility measure can be explained by socio-economic indicators. All three methods produce comparable results. Although in some instances, one is able to produce results which are close to the benchmark when surrogates are used it is obvious that MICE results exceed this level. A slight rise in differences between methods can be observed for an increased number of missing values. The mean relative improvement due to imputation is between 1% and 19%. When there are missing values in only one third of the variables this relative improvement ranges from -2% to 4%.

- The **Infant Birth Weight Data** is used to predict a child's birth weight in grams. Random Forests clearly outperform its competitors while `ctree` and `rpart` perform comparably. The difference between imputation and surrogates does not increase with an increased number of missing values. In some cases MICE makes the performance exceed the benchmark. The improvements by imputation drop from 2%–7% to 0%–4% when the number of variables that contain missing values is restricted to one third.

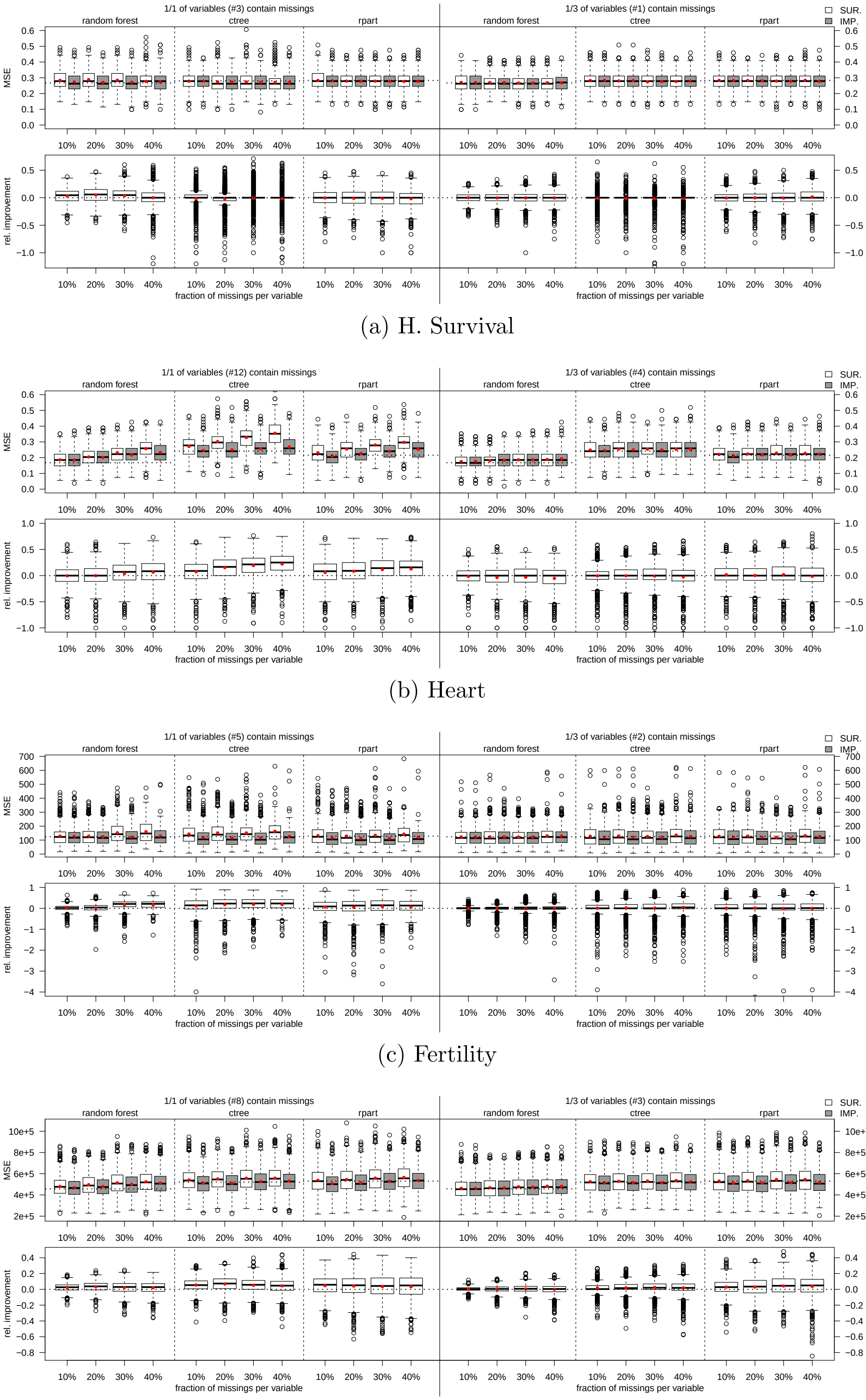
Referring to single trees it has to be stressed that a comparison between the application of MICE and surrogates is not quite fair. MICE produces multiple datasets that vary in the imputed values. To each of them a tree is fit which may consequently differ from each other. Their average or majority decision for an observation is used for prediction. Several works (e.g. Bühlmann and Yu, 2002; Breiman, 1996) show that ensemble approaches perform superior to single trees. This fact becomes apparent for the Swiss Fertility Data as MICE is able to even exceed the benchmark obtained for 0% missing values. By contrast Random Forests are an ensemble method themselves which makes them less prone to this effect. Therefore, one might find them even more suitable for a fair comparison. Anyhow as MICE is very popular this multiple imputation approach is still preferred to single imputation in order to reflect use-oriented results.

2.5.2 Empirical Evaluation

Results obtained for the eight investigated datasets which originally contain missing values are displayed by Figure 2.2. The relative improvement of MICE compared to the application of surrogates can be read off Table 2.4. An extensive listing of results can be found in the appendix A.1.2.

				missing values			
			missing	0%	10%–40%	10%–40%	
Type	Data	Model	variables	benchmark	Surrogates	MICE	rel. imp.
Classi- fication	H. Survival	forest	3	0.27	0.28 – 0.29	0.27	0% – 5%
			1		0.27	0.27	0%
		ctree	3	0.28	0.27 – 0.28	0.27 – 0.28	-2% – -1%
			1		0.27 – 0.28	0.28	-2% – -1%
		rpart	3	0.28	0.28	0.28	-2% – 0%
			1		0.28	0.28	-1% – 0%
	Heart*	forest	12	0.17	0.19 – 0.26	0.18 – 0.23	0% – 7%
			4		0.18 – 0.19	0.18 – 0.19	-5% – -2%
		ctree	12	0.24	0.27 – 0.35	0.24 – 0.27	7% – 22%
			4		0.25	0.25	-2% – 0%
		rpart	12	0.22	0.23 – 0.30	0.21 – 0.25	6% – 13%
			4		0.22 – 0.23	0.21 – 0.22	-1% – 2%
Regres- sion	Fertility	forest	5	124	129 – 160	123 – 129	1% – 17%
			2		123 – 131	122 – 129	-2% – 0%
		ctree	5	126	144 – 164	116 – 126	12% – 19%
			2		126 – 136	119 – 125	1% – 4%
		rpart	5	128	129 – 143	113 – 121	5% – 9%
			2		121 – 132	115 – 125	-1% – 2%
	Birthweight*	forest	8	46e+4	48e+4 – 52e+4	47e+4 – 51e+4	2% – 3%
			3		46e+4 – 48e+4	46e+4 – 48e+4	0% – 1%
		ctree	8	52e+4	54e+4 – 56e+4	51e+4 – 53e+4	4% – 7%
			3		52e+4 – 54e+4	51e+4 – 52e+4	2%
		rpart	8	53e+4	54e+4 – 56e+4	51e+4 – 54e+4	3% – 5%
			3		53e+4 – 55e+4	51e+4 – 52e+4	3% – 4%

Table 2.3: Summary of mean MSE values and mean relative improvements (rel. imp. = $\frac{\mathrm{MSE}_{\mathrm{Sur.}} - \mathrm{MSE}_{\mathrm{MICE}}}{\mathrm{MSE}_{\mathrm{Sur.}}}$) obtained by multiple imputation and surrogates. Please note that the mean relative improvement is given by the mean of improvements across simulation runs. It can not simply be computed by the mean MSE values used in the formula given here (as the mean of ratios does not equal the ratio of means). *The predictor sets of 13 and 9 variables were reduced to 12 and 8 variables due to computational issues with the MICE implementation.



(a) H. Survival

(b) Heart

(c) Fertility

(d) Birthweight

Figure 2.1: MSE of the simulation studies. Left and right columns show results when all or one third of variables had missing values. White and grey boxes correspond to surrogates and imputation. Solid points represent mean values. Relative improvements are shown beneath. Horizontal dashed lines give the mean benchmark MSE for 0% missing values.

A close look at the observed MSE values reveals that Random Forest performs best while `ctree` and `rpart` are comparable. The relative improvement for Random Forests lies within -41% and 4%. For `ctree` these values are between -16% and 15%. For `rpart` values range from -65% to 36%.

	Data	forest	ctree	rpart
Classification	Hepatitis	-1%	-4%	1%
	Mammo	4%	3%	0%
	Pima	0%	1%	0%
	Ozone	-39%	10%	0%
Regression	Airquality	4%	15%	15%
	El Nino	-41%	-16%	36%
	CHAIN	1%	2%	4%
	Sleep	1%	0%	-65%

Table 2.4: Summary of the mean relative improvement (rel. imp. = $\frac{\text{MSE}_{\text{Sur.}} - \text{MSE}_{\text{MICE}}}{\text{MSE}_{\text{Sur.}}}$).

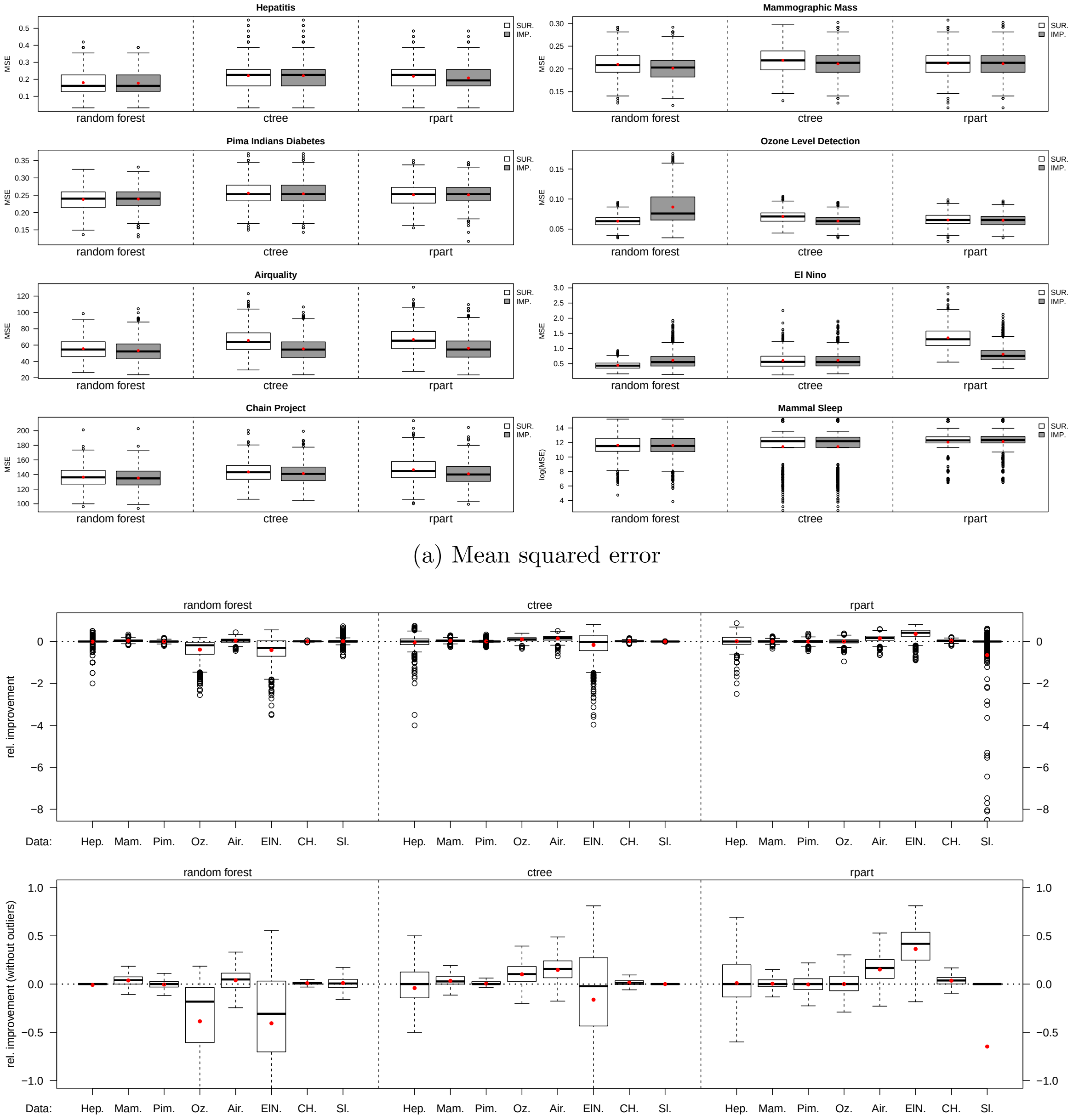
2.6 Discussion and Conclusion

In the simulation studies `rpart` and `ctree` alternately beat each other in performance. Similar results were already found by Hothorn et al. (2006) in their work introducing conditional inference trees. Still one may tend to use `ctree` as `rpart` is known to be biased towards the selection of variables with many possible cutpoints and many missing values. In fulfillment of expectations, Random Forests do not show inferior results compared to single tree approaches. Therefore Random Forests are recommended for application when the main focus is put on prediction.

Simulation Studies

Independent of the statistical model, the underlying dataset and the fraction of missing values, it is found that results are affected by the proportion of variables that contain missing values. If there are missing values in all of them, the relative improvement by an application of MICE instead of surrogates ranges from 0% to 17% for Random Forests. For `ctree` it is between -2% and 22%, and for `rpart` it lies between -2% and 13%. If only one third of the variables contain missing values the improvement diminishs. Now it ranges from -5% to 1% for Random Forests, -2% to 4% for `ctree` and -1% to 4% for `rpart`. These results show that on one hand MICE tends to be beneficial when there are many missing values in many variables. On the other hand it loses this advantage when the number of missing values is limited. In such cases it may even produce inferior results.

In combination with the considerations and findings about the simulation setting in sections 2.2 and 2.5.1 this raises strong doubt about the usefulness of such comparisons



(a) Mean squared error

(b) Relative improvement

Figure 2.2: Boxplots of MSE values are given by Figure 2.2a. White and gray boxes correspond to surrogates and imputation, respectively. Figure 2.2b shows the relative improvement of multiple imputation compared to surrogates. Solid points represent mean values.

for real life situations. A simulation pattern that equally spreads missing values across the entire data in much too high fractions is extremely artificial. There is a strong need to extend simulation to a wider range of patterns that are closer to those found in real life data. A first big step into this direction has already been taken by Rieger et al. (2010) who use lower fractions of missing values and additionally take MAR settings into account. Similar investigations can also be found in chapter 4 and 6 of this work. However eligible structures are difficult to define and it is easier to investigate data that already include missing values.

Application Studies

The potential improvement by the application of imputation instead of surrogates lies within -41% and 4% for Random Forests. Results were equaly ambigious for tree methods although the benefit of MICE was slightly more pronounced. To some extend this might be affected by the property of MICE to implicitly produce ensembles of trees. The relative improvement reaches from -16% to 15% for `ctree` and -65% to 36% for `rpart`. Independent of the prediction method used, MICE produced inferior results in some cases, indicating that imputation may also decrease prediction accuracy.

General

Recursive partitioning by trees is still the method of choice if one is interested in clear decision rules. Nevertheless, the conducted studies confirmed the superiority of Random Forests in terms of prediction strength. There was no convincing improvement by the application of MICE in combination with Random Forests. In terms of prediction accuracy Random Forests seem to be capable to handle missing values by surrogates almost as well as by imputation. A slightly more distinct benefit was found for single tree procedures though it was rather negligible in many cases. For all methods and studies the application of MICE also produced inferior results in some cases. Furthermore the extra effort of imputation should not be underestimated. For example one might decide to create five imputed datasets which results in a fit of five models. If these are subsequently applied to each of another five imputed datasets there are 25 predictions to be made. In total this makes 30 computational steps (5 times fitting + 25 times prediction). With surrogates it takes only one fitting and one prediction step. Generally, the number of computational steps is $\mathrm{n_{fit}} + \mathrm{n_{fit}} * \mathrm{n_{test}}$ while $\mathrm{n_{fit}}$ is the desired number of imputed datasets for the fitting and $\mathrm{n_{test}}$ for the application of models. In addition, when multiple imputation is used during the fitting process, single trees lose their ability to provide simple decision rules which is often one of the main reasons for their application.

Results for the application of imputation or surrogates in both the training and test steps have been presented in this chapter. Actually, when the fitting and the application of a statistical method is done by two different researchers these habits could also mix. Some might not be used to imputation methods and will not apply them. Others could have experienced good results with MICE which makes them use it whenever possible. Likewise, it has often been claimed that one positive aspect of imputation is that the imputed data can be passed to third party analysts. Therefore all analyses have also been conducted by imputation of the training set while the test set was not touched and vice versa. There were two interesting findings: Firstly, the average MSE values of both cases lay between those obtained for the imputation of both datasets or none of them. Secondly, there was also no consistent benefit or harm observed for the imputation of one set instead of the other.

Conclusion

Results indicate that the theoretical properties of the investigated recursive partitioning methods could be retraced in the simulation and application studies. Thus, Random Forests showed the best or at least not inferior performances. The CART algorithm and conditional inference trees implemented by the functions `rpart` and `ctree` performed equally well.

The simulation based on four datasets showed no clear improvement of results by the application of multiple imputation versus surrogates. A potential benefit is highly dependent on the composition of the simulation setting. MICE may even produce inferior results when missing values are limited in number and are not arbitrarily spread across the entire data. Thus the generalizability of simulation results is limited. A broader application to diverse simulation schemes is needed for further insight.

In order to be close to practical scenarios empirical evaluations of another eight datasets were performed. The benefit of imputation in terms of prediction accuracy was found to be ambiguous. For Random Forests the relative reduction was rather negligible in six datasets ranging from -1% to 4%. In another two datasets it even showed extremely harmful effects; i.e. -41% and -39%. Similar results were found for single tree methods though the benefit was slightly more pronounced. Refering to `ctree`, it reached from -16% to 15% while it was between -65% and 36% for `rpart`.

Due to reasons like lacking familiarity or additional work and time that needs to be spent for multiple imputation, a practitioner might not be willing to use it in combination with recursive partitioning methods. The application of surrogates instead is fast, simple, works in any data situation and leads to only negligibly worse (and in some cases even superior) results. These statements are based on the analysis of as much as four simulation settings and eight empirical evaluations. Although they cover a huge range of missing value patterns, variable scales, data dimensions and research fields, the presented results need to prove generalizability in further studies.

Chapter 3

A new Variable Importance Measure for Missing Data

3.1 Research Motivation and Contribution

A highly valued feature of Random Forests, which is in the main focus of this chapter, is the assessment of a variable's relevance by means of importance measures. Unfortunately, there is no suggestion on how to compute such measures in the presence of missing values. Existing methods can not be used as there are evident violations against their most basic principles if applied straightforward. Hence one of the most appreciated properties of Random Forests, namely its ability to handle missing values, gets lost for the computation of importance measures. This chapter presents a solution to this pitfall by the introduction of a new variable importance measure that:

- retains the widely appreciated qualities of existing variable importance measures;
- is applicable to any kind of data, whether it does or does not contain missing values;
- is robust against different kinds of missing data generating processes;
- shows good variable ranking properties;
- meets sensible requirements;
- incorporates the full information provided by the entire data (i.e. without any need for restrictions like complete case analysis);
- can deal with missing values in an intuitive and straightforward way.

The properties of the new method are investigated in an extensive simulation study. Two data evaluations show the practicability of the new method in real life situations. They also indicate that the new approach may provide a more sensible variable ranking than the widespread complete case approach. The new proposal and corresponding investigations of this chapter have been accepted for publication (Hapfelmeier et al., 2012b).

3.2 New Proposal

3.2.1 Definition

A new approach is suggested here in order to provide a straightforward and intuitive way of dealing with missing values in the computation of a Random Forest variable importance measure. The construction of the new measure closely sticks to existing methodology while it deviates from the original permutation accuracy importance (section 1.2.2) only in one – yet one substantial – step.

In the original permutation accuracy importance, the OOB values of a predictor variable of interested are randomly permuted to simulate the null-hypothesis $H_0 : Y, \mathbf{Z} \perp X_j$ (cf. equation (1.3)). This mechanism destroys the relation of the predictor variable to the response and the remaining variable space. If the OOB accuracy drops substantially as a result, the variable is termed to be of relevance. However, it is not clear how to proceed in the presence of missing values. In particular, it is not clear how conclusions about the importance of a variable can be drawn from the permutation approach when surrogate splits are used for the computation of the OOB accuracy but are not part of the permutation scheme. A simple extension of the permutation scheme to cover the surrogate variables does not solve this issue but leads to additional undesirable effects; e.g. the importance of all variables that are involved in the permutation would somehow be admixed. For this problem, a simple but efficient solution is suggested in the following.

The main idea of the new proposal is the following: Instead of permuting the OOB values of a variable (that may be missing), the corresponding observations are randomly allocated to one of the child nodes if the split of their parent node is conducted in the variable of interest. This procedure detaches any decision from the raw values of the variable, and therefore circumvents any problems associated with the occurrence of missing values and the application of surrogate splits for the computation of the OOB accuracy.

The rest of the computation procedure, however, is not affected by this “trick”: In the first step of the computation one proceeds as normal by recording the OOB accuracy of a tree (using all surrogate splits, which can be considered as an implicit imputation of the missing values). In a second step, the OOB accuracy is again recomputed by randomly assigning OOB observations that were originally split in the variable of interest to the corresponding child nodes.

Formally introducing a binary random variable D, that indicates the decision for one of the child nodes, the probability of being sent to the left ($D = 0$) or to the right ($D = 1$) child node respectively is given by $P_k(D = 0)$ and $P_k(D = 1) = 1 - P_k(D = 0)$ for a node k. The random allocation of the OOB observations, just like the random permutation of the OOB values of a predictor variable X_j itself in the original permutation importance, mimics the null hypothesis that the assignment of observations to nodes does not depend on this particular predictor variable any more. Under the null-hypothesis, the probability to end up in a specific child node of node k is

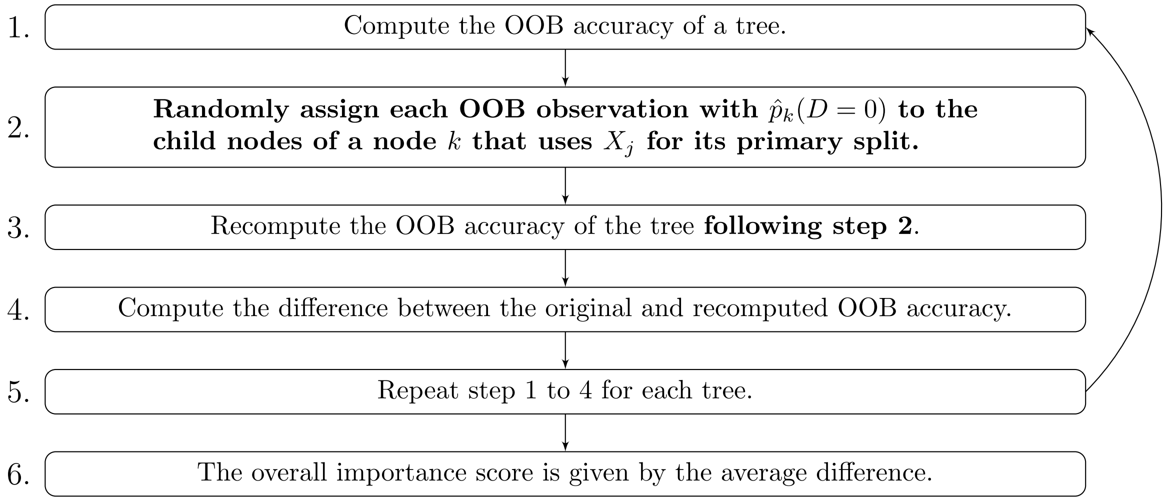
$$P_k(D|X_j) = P_k(D).$$

Therefore it does not matter whether values of X_j are missing or not, as it is not used for the decision of how to further process an observation.

For the practical computation of the prediction accuracy, the probability $P_k(D = 0)$ is replaced by its empirical estimator, the relative frequency:

$$\hat{p}_k(D = 0) = n_{k,\text{left}}/n_k$$

where $n_{k,\text{left}}$ and n_k are the number of observations that were originally sent to the left child node and were present in parent node k, respectively. In contrast to the original permutation importance measure presented in section 1.2.2, the computation of the new measure consists of the following steps, highlighting the essential difference in step 2:



1. Compute the OOB accuracy of a tree.
2. **Randomly assign each OOB observation with $\hat{p}_k(D = 0)$ to the child nodes of a node k that uses X_j for its primary split.**
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.
6. The overall importance score is given by the average difference.

A corresponding implementation is given by the function `varimp()` in the `R` package `party` since version 1.0-0.

3.2.2 Requirements

The characteristics of the proposed importance measure needed to be explored while it also had to follow some requirements. Therefore several properties were supposed to be close to those of the original permutation importance measure like the fact that correlated variables obtain higher importances than uncorrelated ones in an unconditional assessment. Others were still to be investigated like the effect of different correlation structures (that determine the quality and amount of surrogate variables), schemes of missingness, the amount of missing values and so forth.

In order to rate the performance of the newly suggested approach a list of requirements that should be met by a sensible variable importance measure designed for dealing with missing values was formulated:

- (R1) When there are no missing values, the measure should provide the same variable importance ranking as the original permutation importance measure.

- (R2) The importance of a variable is not supposed to artificially increase, but to decrease with an increasing amount of missing values (because the variable loses information, cf. Strobl et al., 2007a).

- (R3) Like the original permutation importance (that is a marginal importance in the sense of Strobl et al., 2008), the importance of a variable is supposed to increase with an increasing correlation to other influential predictor variables.

- (R4) The importance ranking of variables not containing missing values is supposed to stay unaffected by the amount of missing values in other variables. This is only required within groups of equally correlated variables as differences in correlation directly affect variable importances and therefore may well change the ranking.

- (R5) The importance of influential variables is supposed to be higher than the importance of non-influential variables. This should hold for equally correlated variables with equal amounts of missing values – considering that both facts influence the importance of a variable. For example a non-influential variable which does not contain missing values and is correlated with influential variables can achieve a higher importance than an isolated influential variable containing missing values (cf. Strobl et al., 2008). In any case, the lowest importance should always be assigned to non-influential variables that are uncorrelated with any other influential variables.

In order to investigate these requirements and shed light on further characteristics of the newly suggested approach, an extensive simulation study – as described in the following sections – was set up.

3.3 Studies

3.3.1 Simulation Studies

There are several factors that need to be varied in the simulation setting, in particular the amount of missing values, correlation strength, the number of correlated variables (termed block size in the following), variable influence and different missing data generating processes. A detailed explanation of the setup is given in the following.

- *Influence of predictor variables*

  The proposed importance measure is supposed to be applicable in both classification and regression problems. Thus, the simulated data contained a categorical (binary) and a continuous response. In both cases the coefficients β, that were used to produce 20 variables in the data generating model described below, are:

$$\beta = (4, 4, 3, 4, 3, 4, 3, 4, 3, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0)^\top$$

The idea of this setup was that a repeated choice of the same values for β enables a direct comparison of importances of variables which are, by construction, equally influential. The different β values are crossed with the other experimental factors to allow an evaluation of differences and provide reference values for each setting. In addition, the non-influential variables with $\beta = 0$ help to investigate possible undesired effects and serve as a baseline.

- *Data generating models*

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

  $$y = \mathbf{x}^\top \beta + \epsilon \text{ with } \epsilon \sim N(0, .5).$$

  The binary response was drawn from a Bernoulli distribution $B(1, \pi)$ while 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 contains 100 observations drawn from a multivariate normal distribution with mean vector $\vec{\mu} = 0$ and covariance matrix Σ:

- *Correlation*

  $$\Sigma = \begin{pmatrix}
1 & r & r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
r & 1 & r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
r & r & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 1 & r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & r & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & r & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & r & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & r & r & r & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r & 1 & r & r & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r & r & 1 & r & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r & r & r & 1 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

  As the variances of each variable are chosen to be 1, the covariance equals the correlation in this special case. The strength of correlation was varied by setting r to 0, .3, .6 and .9. The structure of the 20×20 dimensional covariance matrix Σ reveals that there are four blocks of correlated variables of various sizes, each of them consisting of 3, 2, 2 and 4 variables, respectively. Thus it was possible to investigate the effect that the strength and extent of the correlation had on the importance measure.

- *Missing values*

  In analogy to the simulation setting of Rieger et al. (2010), who investigate the performance of Random Forests on missing data, several MCAR and MAR schemes were implemented to induce missing values. In addition, a further MNAR setting was investigated. In each scheme, a given fraction m of observed values is replaced by missing values for selected variables. As the amount of missing values is of major concern in the simulation experiments, m takes the values 0%, 10%, 20% and 30%.

  In a MAR setting, the probability for missing values in a variable depends on the values of another variable. In the MNAR scheme this probability is determined by a variables own values. Accordingly, each variable containing missing values has to be linked to at least one other variable or itself. Table 3.1 lists the corresponding relations.

contains missing values	determines missing values	
(MCAR, MAR & MNAR)	(MAR)	(MNAR)
X_2	X_3	X_2
X_4	X_5	X_4
X_8	X_9	X_8
X_{10}	X_{11}	X_{10}
X_{12}	X_{13}	X_{12}
X_{14}	X_{15}	X_{14}

Table 3.1: List of variables that contain missing values and variables that determine the probability of a missing value.

The schemes to produce 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 9 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.

The findings of Little and Rubin (2002) showed that usual sample estimates, for example in linear regression, stay unaffected by the MCAR scheme. However, Strobl

> et al. (2007a) outlined that in classification and regression trees even MCAR may induce a systematic bias, that may be carried forward to Random Forests based on biased split selections. Therefore, in the following simulation study, one MCAR, four MAR and one MNAR process to generate missing values are investigated to shed light on the sensitivity of the proposed method to these schemes.

A schematic illustration of β summarizes all factors varied in the simulation design below. Correlated blocks of variables are enumerated by roman figures and separated by '|'. Bold figures indicate variables that contain missing values:

$$\beta = (\underbrace{4, \mathbf{4}, 3}_{\text{I}} | \underbrace{\mathbf{4}, 3}_{\text{II}} | \underbrace{4, 3}_{\text{III}} | \underbrace{\mathbf{4}, 3, \mathbf{0}, 0}_{\text{IV}} | \underbrace{\mathbf{2}}_{\text{V}} | \underbrace{2}_{\text{VI}} | \underbrace{\mathbf{0}}_{\text{VII}} | \underbrace{0}_{\text{VIII}} | \underbrace{0}_{\text{IX}} | \underbrace{0}_{\text{X}} | \underbrace{0}_{\text{XI}} | \underbrace{0}_{\text{XII}} | \underbrace{0}_{\text{XIII}})^\top$$

In summary, there are 2 response types, 6 missing value generating processes, 4 fractions of missing values and 4 correlation strengths, summing up to as much as 192 different simulation settings. Variable importances were recorded by repeating each setting 1000 times. The corresponding `R`-Code is given in the appendix B.3.1.

3.3.2 Empirical Evaluation

In addition to the extensive simulation study, two well known data sets were used to show the applicability of the new approach in real life situations. These are The Pima Indians Diabetes Dataset and the Mammal Sleep Data as presented in section 2.4.2. Both were chosen to provide a varying number of missing values in several variables. The total number of variables equals 8 and 9 to allow for an easy and clear comparison of importance measures.

Besides the examination of the newly suggested approach, the original permutation importance measure was applied, too. For the latter, the still popular complete case analysis approach was used, for which observations that contain missing values are entirely omitted before the Random Forest is fit. Finally the ranking of variable importances within each approach was compared and discussed. The corresponding `R`-Code is given in the appendix B.3.2.

3.3.3 Implementation

The analyses of this chapter were again performed with the `R` system for statistical computing (R Development Core Team, 2011, version 2.14.1). The function `cforest()` of the `party` package (Hothorn et al., 2008, version 1.0-0) was used as an unbiased implementation of Random Forests. The settings for the simulation studies were chosen to result in a computation of $ntree = 50$ trees and $maxsurrogate = 3$ surrogate splits in each node. The number of randomly selected variables serving as candidates for splits was set to be $mtry = 8$. Sticking to the default setting $mincriterion = 0$ there were no restrictions concerning the significance of a split. Trees were grown until terminal nodes contained less than *minsplit* = 20 observations while not allowing splits that led to less than *minbucket* = 7 observations in a child node. As the number of complete observations becomes extremely low in the complete case analysis of the additional simulation study these parameters were set to *minsplit* = 2 and *minbucket* = 1 in this case. The examination of two empirical evaluations was based on Random Forests that consisted of *ntree* = 5000 trees in order to produce stable variable importance rankings. The number of variables chosen for splits was set to *mtry* = 3 considering that the data contain only 8 and 9 variables, respectively. The number of surrogate splits and observations required in terminal nodes and parent nodes was the same as in the simulation studies. An implementation of the new importance measure is provided by the function `varimp` since version 1.0-0 of the `party` package. A function that counts selection frequencies, i.e. the number of times a variable is chosen for splits in a Random Forest, is given in appendix B.

Genuer et al. (2008) have conducted elaborate studies to investigate the effect of the number of observations, *ntree* and *mtry* on the computation of importance measures. They found that the stability of estimation improved with a rising amount of observations and trees (*ntree*). However, the rankings of importance measures – which are in main focus of this work – remained almost untouched. This property is also supported by the fact that simulation studies are repeated 1000 times; aiming at an averaged assessment of rankings. It is a common choice to make *mtry* equal the square root of the number of predictors (cf. Díaz-Uriarte and Alvarez de Andrés, 2006; Chen et al., 2011). Again, Genuer et al. (2008) found this value and even higher values for *mtry* to be convenient for the identification of relevant variables by means of importance measures. Therefore, all of the parameter settings of the simulation studies are in accordance with these considerations.

3.4 Results

3.4.1 Simulation Studies

The following extensive investigations are based on the regression analysis in the MAR(rank) scheme. Due to the study design, each requirement can be explored by the presentation of results for specific sets of variables. However, it has to be pointed out that non-influential variables were only partly presented as they all gave the same results and did not show any unexpected effects. Thus, except for requirement (R1), variables 16 to 20 are omitted from any presentation. A discussion about the reproducibility of findings in the investigated classification problem and further processes that generate missing values is given in the end of this section.

Requirement (R1) is satisfied for all of the investigated variables and correlation strength (Figure 3.1). The newly suggested approach and the original permutation importance measure even approximately equal each other when there are no missing values (m = 0%). Deviations of single assessments are due to the inherent variability of importance measures. Therefore, results are also presented as median importance across 1000 simulation runs to stress the average equality.

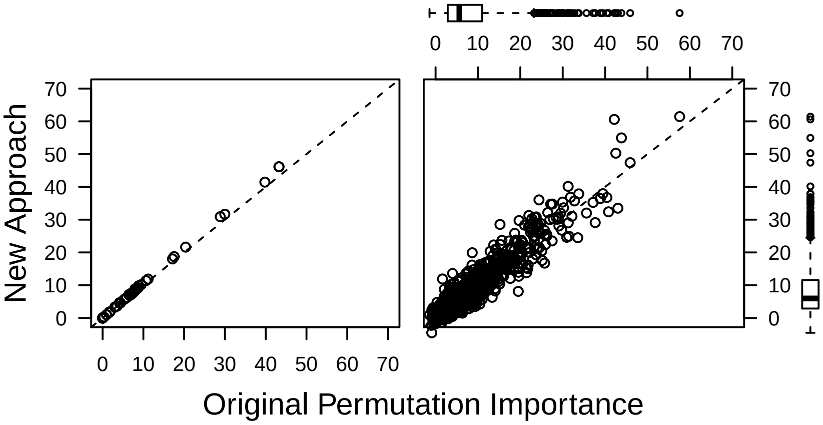


Figure 3.1: Comparison of the new approach and the original permutation importance. Left: Median importance measures across 1000 simulation runs for all variables and correlations when there is no missing data (m = 0%). Right: Distribution of values observed in 1000 simulation runs for the example of variable 5 (r = 0.6).

Requirement (R2) is met as the importance of variables decreases the more missing values they contain (Figure 3.2). This holds for all variables and correlation strengths (Figure A.1 in appendix A.2).

Requirement (R3) holds as correlations with influential variables induce higher importances (Figure 3.3). This is true for all variables and fractions of missing values (Figure A.1 in appendix A.2; A comparison of blocks I and II shows that block size is another factor that affects variable importance. However, non-influential variables, given in block IV, do not contribute to this effect.).

The effects of correlation and missing values appear to be interacting (see block I in Figure 3.4): Although all variable importances rise with a rising strength of correlation, the importance of variable 2 drops in relation to the variables of the same block when the

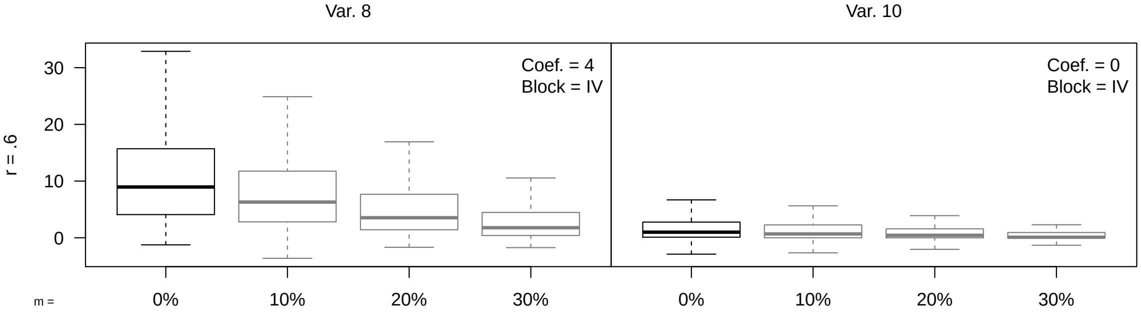


Figure 3.2: Variable importance of variables 8 and 10 for a correlation of r = .6 and m = 0%, 10%, 20%, 30% missing values. Boxplots of variables with missing values are colored grey. Outliers are omitted from illustration for clarity.

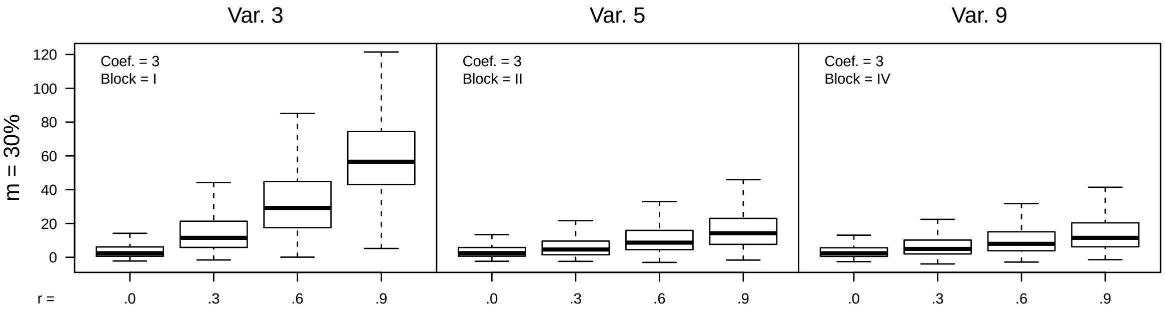


Figure 3.3: Variable importances of variables 3, 5 and 9 for correlations of r = 0, .3, .6, .9. Outliers are omitted from illustration for clarity.

amount of missing values increases. An investigation of selection frequencies – i.e. the number of times a variable is chosen for splits in a Random Forest (displayed as horizontal lines) – reveals that it is replaced by other variables in the tree building process. This effect follows a simple rule: the more similar the information of variables becomes due to an increased correlation, and the more information a variable is lacking because of missing values, the more often it will be replaced by others.

Requirement (R4) is satisfied as the ranking of fully observed variables from the same block stays unaffected by the amount of missing values in other variables (Figure 3.4). Note that between blocks the variable rankings may change: The importance of variable 5 increases as it is able to replace variable 4 that contains missing values. It rises above variable 7 with the same (and for strong correlations and many missing values even above variable 6 with a higher) influence on the response. Another question emerging from the fact that variables may replace others in a tree is if this also holds for isolated blocks that are not correlated with any variables that contain missing values. Figure 3.5 shows that this is almost not the case as selection frequencies and variable importances stay on a certain level for block III compared to block II). This finding even partly extends (R4) which demands stable rankings for fully observed variables only within blocks, not across blocks.

Requirement (R5) is met as the importance of influential variables is higher than for non-influential variables (Figure 3.6). This holds for variables with and without missing values, but not necessarily for comparisons between the two cases. Importances of influential variables may drop below those of non-influential ones if the former contain missing values and the latter are part of a correlated block with influential variables. An example is given by block IV: Variable 8 shows a higher importance than variable 10 (both containing missing values) and variable 9 shows a higher importance than variable 11 (both without missing values). However, the importance of the influential variable 8 drops below that of the non-influential variable 11, as the former contains missing values and the latter is correlated to variable 9. The importance of variable 11 even rises above that of influential variables contained in other blocks (e.g. variable 13). However, the lowest importance

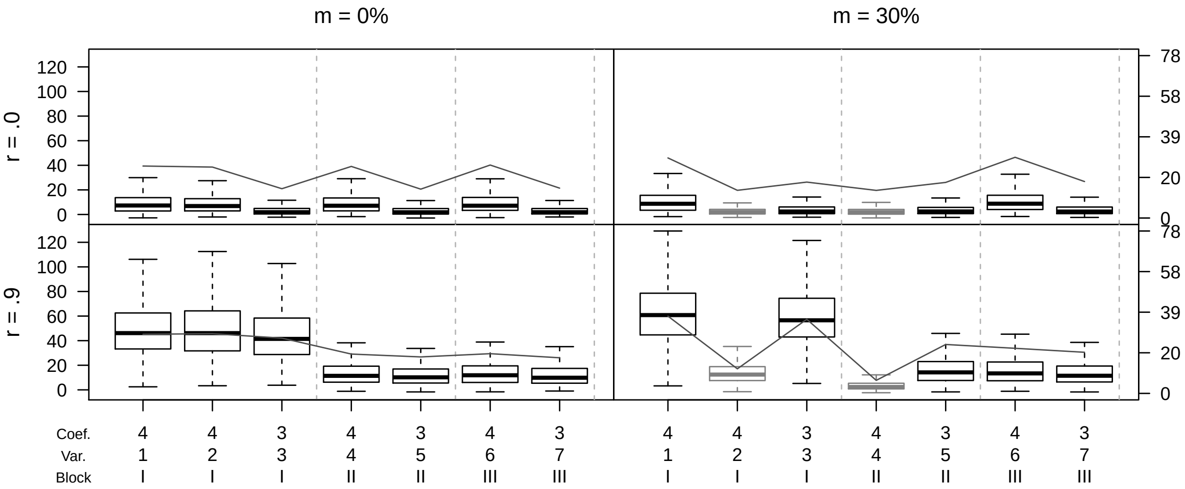


Figure 3.4: Variable importances (left axis) of variables 1–7 (Block I, II, III) for correlations of r = 0, .9 and fractions of missing values m = 0%, 30%. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

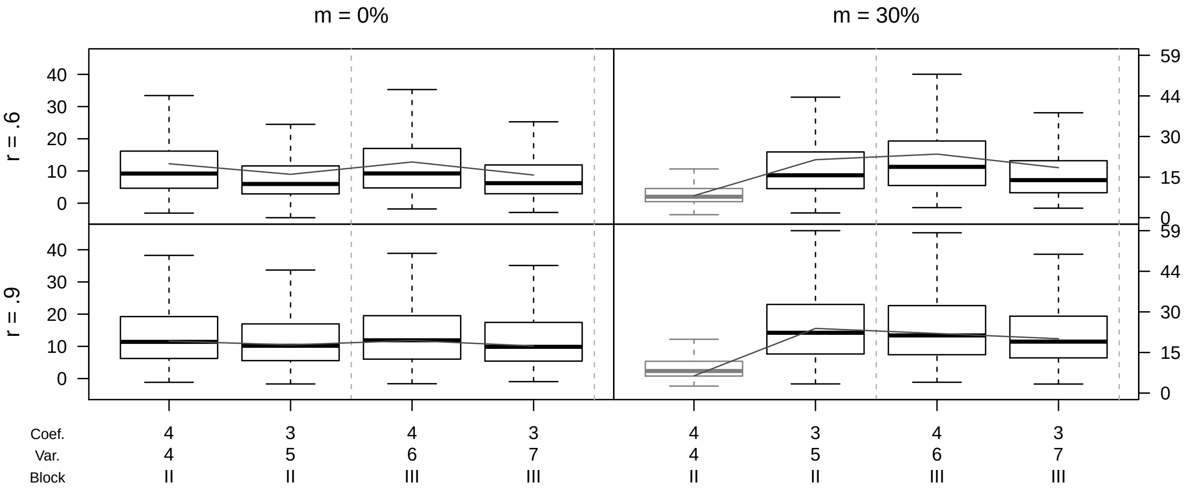


Figure 3.5: Variable importances (left axis) of variables 4–7 (Blocks II, III) for correlations of r = .6, .9 and fractions of missing values m = 0%, 30%. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

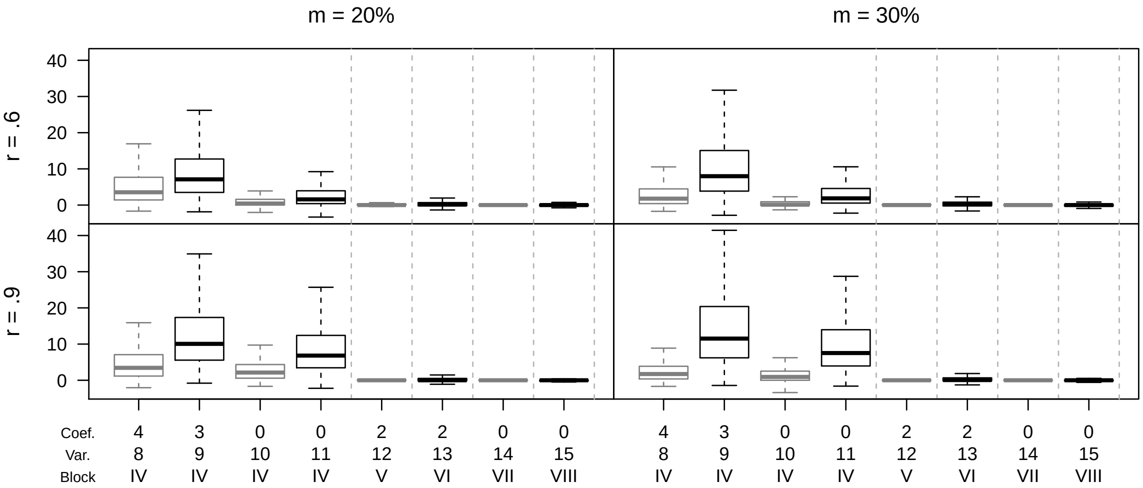


Figure 3.6: Variable importances of block IV–VIII for correlations of r = .6, .9 and fractions of missing values m = 20%, 30%. Boxplots of variables that contain missing values are colored grey. Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

should always be assigned to non-influential variables that are uncorrelated with any other influential variables. This claim is supported by the examples of variable 14 and 15.

In conclusion, factors like the occurrence of missing values, the number and influence of correlated variables as well as the correlation strength, can positively affect the importance of variables. However, these are properties to be expected from a marginal variable importance measure when dealing with variables that lose information due to missing values, yet are correlated to other variables that can "take over for them".

Results of the entire simulation setting for the regression problem are displayed in Figure 3.7 for a broad overview. The results show the same properties that were already pointed out for the specific settings above. An additional comparison of results for all schemes used to produce missing values (MCAR, MAR(rank), MAR(median), MAR(upper), MAR(margins) and MNAR(upper)) is given by Figure 3.8. For the purpose of clarity it concentrates on the case of r = .6 and m = 20%. None of the schemes shows any noticeable differences; all findings of the previous analyses in the MAR(rank) setting can be retraced in each scheme. In conclusion, the proposed approach shows the same properties for a wide range of MCAR, MAR and MNAR missing data schemes. Likewise, results for the classification problem show the same properties (Figure A.3 in the appendix A.2) and thus are not discussed in detail to omit redundancy.

3.4.2 Empirical Evaluation

Figure 3.9 shows the results observed for the two data evaluations. Both datasets were analyzed in two ways: By applying the new approach to the full data set and by applying the

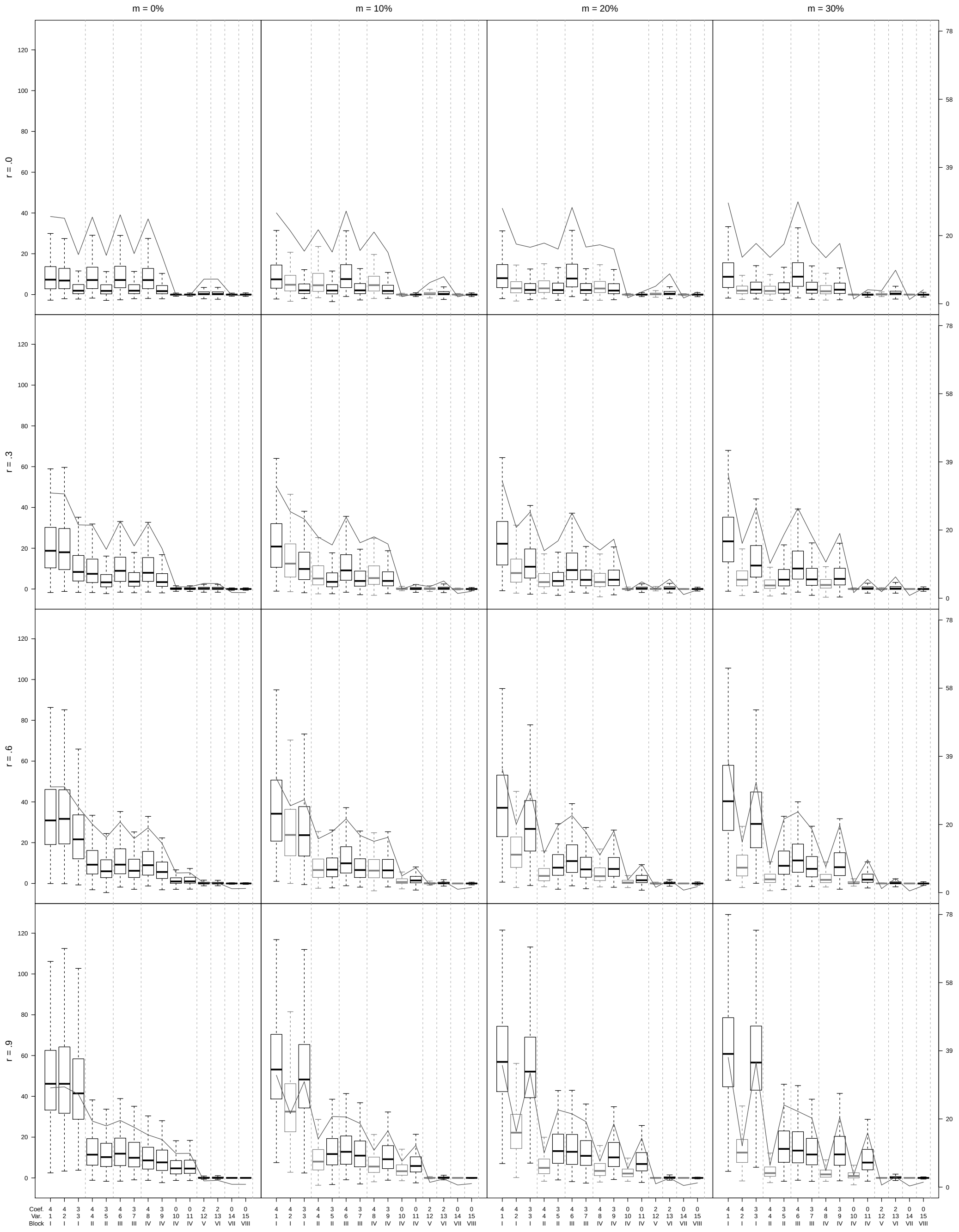
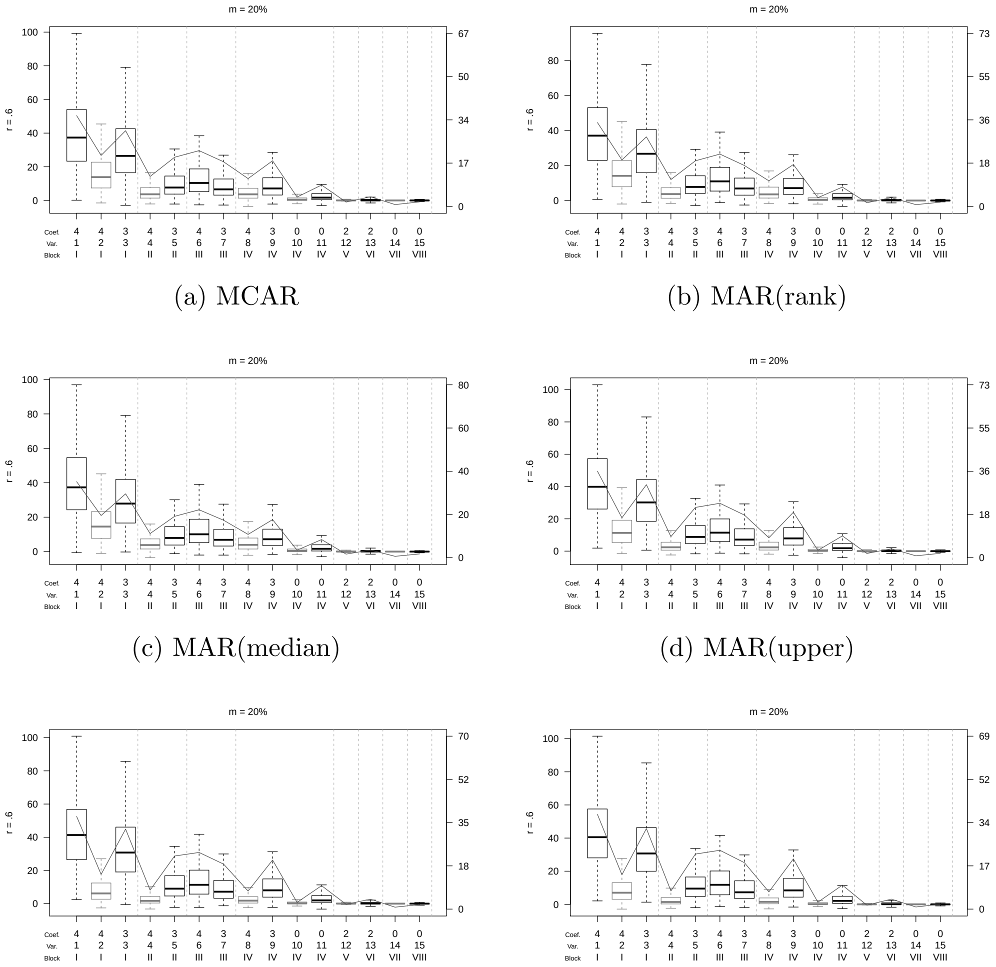


Figure 3.7: Variable importances (left axis) of block I–VIII for correlations of r = 0, .3, .6, .9 and fractions of missing values m = 0%, 10%, 20%, 30% in the MAR(rank) setting of the regression problem. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.



(a) MCAR

(b) MAR(rank)

(c) MAR(median)

(d) MAR(upper)

(e) MAR(margins)

(f) MNAR(upper)

Figure 3.8: Variable importances (left axis) of variables 1–15 for a correlation of r = .6 and a fraction of missing values m = 20% in the regression problem. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

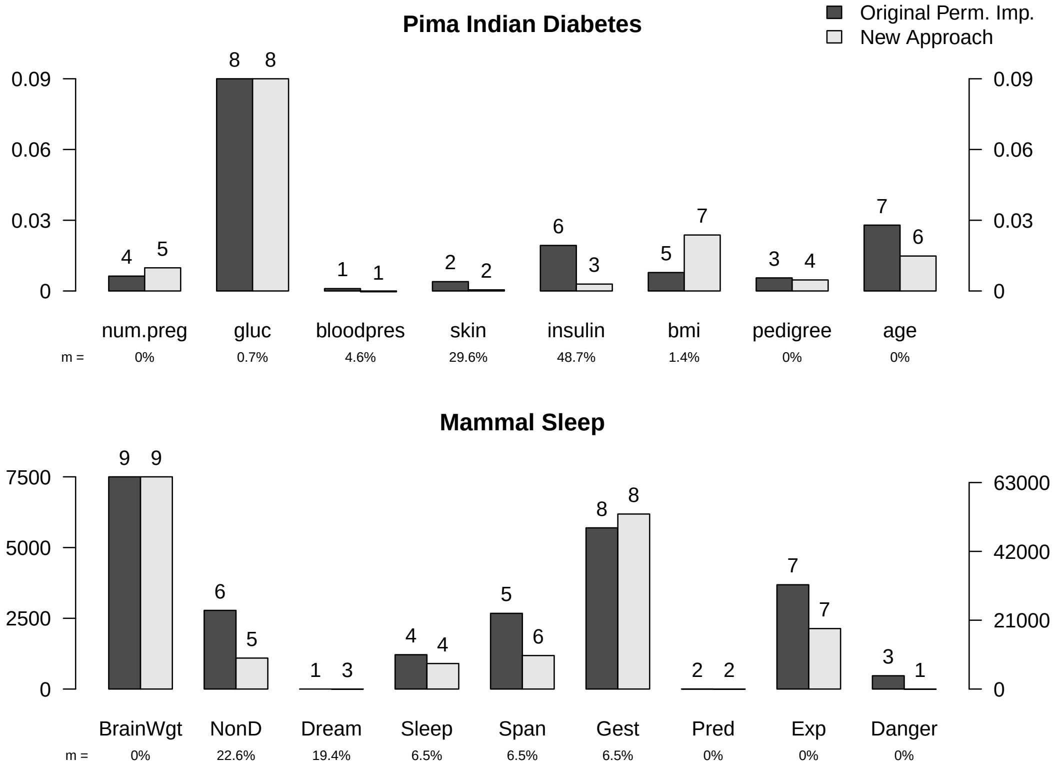


Figure 3.9: Variable Importances for the Pima Indians and the Mammal Sleep Data. Black and grey bars correspond to the original permutation importance measure (complete case analysis; left axis) and the new approach (right axis), respectively. Heights of bars were matched in relation to the maximally observed importance of each approach. Figures above bars indicate ranks within methods. The fraction of missing values per variable is given by m.

original permutation importance measure in a complete case analysis, where observations with missing values are excluded from analysis.

In the Pima Indians Diabetes Data the ranking of predictor variables shows severe differences between methods: The new approach assigns a higher importance to the variables BMI, number of pregnancies and the diabetes pedigree function. It downgrades the variables age and 2-Hour serum insulin. The strongest and weakest variables, however, plasma glucose concentration, diastolic blood pressure and triceps skin fold thickness are ranked equally. Similar findings can be observed for the Mammal Sleep Data. The variables slow wave sleep (‘NonD’ = ‘nondreaming’), dreaming sleep, maximum life span and the overall danger index are ranked differently.

A plausible reason for these differences is that complete case analysis can induce a bias when observations are not MCAR. This is well-known, yet complete case analysis is still frequently applied in practice. Results show that differences do not directly (or solely) depend on whether or not a variable contains missing values. Complete case analysis can modify the entire importance ranking just because information is omitted when observations are excluded from the analysis.

To illustrate one possible scenario that can lead to a change in the variable ranking when complete case analysis is applied to data that is not MCAR, another small simulation was conducted. Given a pair of binary variables (U, V) the response Y follows the distribution:

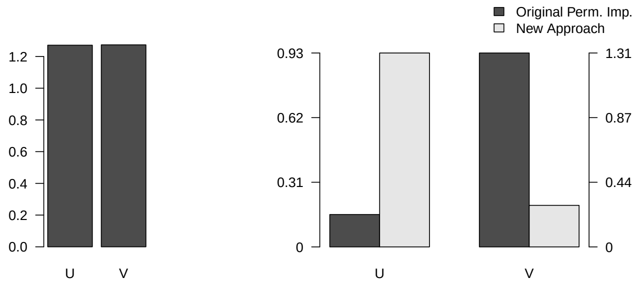
$$Y \sim \begin{cases} N(2,1) & \text{if } (u,v) = (1,0) \\ N(0,1) & \text{if } (u,v) = (0,0) \text{ or } (u,v) = (1,1) \\ N(-2,1) & \text{if } (u,v) = (0,1) \end{cases}$$

The relative frequencies of class 0 and class 1 in U and V are 80% and 20%, respectively. They are not correlated. Missing values are induced into V dependent on the highest values of Y which resembles the MAR(upper) scheme. To produce stable results the simulation is based on 5,000 observations and Random Forests growing 5,000 trees. According to our expectation Figure 3.10a displays the same importance for both variables when there are no missing values in the data. In Figure 3.10b a fraction of 30% of V is set missing. The new approach is able to incorporate the entire data into the computation of the importance measures and assigns a reduced importance to V while U remains of high relevance. This finding again meets our expectations. In a complete case analysis however, U suffers the loss of its explanatory power although it does not contain any missing values at all. It is not even correlated to V. The explanation of this effect is quite simple: The highest values of Y which cause the missing values in V are most frequently related to $u = 1$. Deleting these observations in a complete case analysis makes U mainly consist of class 0. As a consequence it loses its discriminatory power. This example demonstrates how a complete case analysis can distort the ranking of variable importances when the missingness scheme is not MCAR. The new approach follows a much more sensible way of producing importance measures in any situation. Corresponding, more elaborate investigations of this issue are given in the following chapters 4 and 6.

3.5 Discussion and Conclusion

In summary, the simulation results have shown that all requirements that were previously formulated were fulfilled by the newly suggested importance measure for different types of MCAR, MAR and MNAR missing data. Most importantly: In the absence of missing values, both the original permutation importance measure and the newly suggested approach produce similar results. The importance of variables containing missing values does not artificially increase but decreases with the number of missing values and the respective decrease of information. Moreover, in the presence of correlation, the measure shows all properties that are to be expected from a marginal variable importance measure.

A particularly interesting effect is that with regard to the variable selection frequencies, variables with increasing numbers of missing values are increasingly replaced by fully



(a) No missing values (b) 30% missing values in V

Figure 3.10: Variable importances of U and V computed by the original permutation importance measure (black bars; left axes) and the new approach (grey bars; right axis). Case (a) is based on the entire data without any missing values. In case (b) 30% of variable V are set missing. A complete case analysis is used to compute the original permutation importance (black) while the new approach is able to process the entire data (grey).

observed variables that are correlated with them: the complete variables “take over” for those with missing values within a group of correlated ones. Similar findings about the ’...competition of correlated variables for selection into a tree...’ have already been outlined by Nicodemus et al. (2010). In this sense, the effects of correlation and missing values are interacting. This is an intuitive property, since both affect the amount of information a variable retains.

What is important to note here is that, besides effects of the correlation on the permutation importance that were already pointed out by Strobl et al. (2008), in the presence of missing values the correlation is also linked to the quality of surrogate variables. The exact role that surrogate variables play for the variable importance is still ambiguous: On one hand they help to reconstitute missing information, but on the other hand they also compete for the selection in the tree. However, the selection frequencies displayed in the results indicate that the latter effect is stronger.

Besides the findings for the simulation analysis the new approach also appears well suited to deal with missing values in the evaluation study: There were some profound differences between the variable ranking suggested by the new approach and a complete case analysis. As the latter is known to produce biased results in many situations (e.g., Janssen et al., 2009, 2010) this strongly indicates that the omission of observations with missing values has induced artifacts because the values were not missing at random. Results of corresponding simulation studies support this claim.

The advantage of the new approach proposed in this work is that it incorporates the full information provided by the data. Moreover, it reconstitutes one of the most appreciated properties of recursive partitioning methods, namely their ability to deal with missing values. The rationale of the approach is not to undo the influence missing values have on the information carried by a variable, but to reflect the remaining information that the variable retains with the respective values missing. Accordingly, the resulting importance rankings do not only depend on the amount of missing values but also on the quality and availability of surrogate variables.

Chapter 4

Variable Importance with Missing Data

4.1 Research Motivation and Contribution

A new variable importance measure that is able to deal with missing values has been introduced in chapter 3. However, there are also alternative solutions like imputation methods and complete case analysis that enable the computation of the permutation importance measure in such cases. Therefore an extensive simulation study, that involves various missing data generating processes, is conducted in this chapter to explore and compare the ability of

- complete case analysis,
- multiple imputation and
- the new importance measure

to produce reliable estimates of a variable's relevance. Both, regression and classification problems are explored. In addition, the predictive accuracy of Random Forests that are based on each of these approaches is investigated for a simulated test dataset. The latter issue has already been explored in chapter 2: comparisons of models fit with and without imputation of missing values showed only negligible differences. By contrast, the following study focuses on the assessment of a variables importance measure. As a result the ability to produce reliable estimates differs between approaches. Findings and recommendations: Complete case analysis should not be applied as it may inappropriately penalize variables that were completely observed. The new importance measure is much more capable of reflecting decreased information exclusively for variables with missing values and should therefore be used to evaluate actual data situations. By contrast, multiple imputation allows for an estimation of importances one would potentially observe in complete data situations. The investigations of this chapter have been published as a technical report in Hapfelmeier et al. (2012a).

4.2 Simulation Studies

There are several factors of potential influence that needed to be explored in the simulation study. Therefore the amount of missing values, correlation schemes, variable strength and different processes to generate missing values were of particular interest. The setup closely resembles the one given in section 3.3.1. However, there are essential differences as listed in the following:

- *Influence of predictor variables*

  The simulated data again 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:

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

  Repeated values for β make it possible to compare importances of variables which are, by construction, equally influential but show different correlations and different fractions of missing values. In addition, the non-influential variables with $\beta = 0$ help to investigate possible undesired effects and serve as a baseline.

- *Correlation*

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

  As the variances of each variable are chosen to be 1, the covariance equals the correlation in this special case. The structure of the covariance matrix Σ reveals that there is a block of four correlated variables and two uncorrelated ones.

- *Missing values*

  Several MCAR, MAR and MNAR processes to create missing values were implemented. For each scheme, a given fraction $m \in \{0.0, 0.1, 0.2, 0.3\}$ of values is set missing for the variables X_2, X_4 and X_5 (cf. Table 4.1). The number of observations that contain at least one missing value is given by $1 - (1 - \%_{\text{missing}})^{\text{n}_{\text{variables}}}$. Thus, a dataset that contains three variables with 30% missing values includes $1 - (1 - 0.3)^3 = 65.7\%$ incomplete observations on average. This seems to be a rather huge amount rarely seen for real life data. Nevertheless, this way m comprises a wide range of possible scenarios. The schemes to produce missing values were MCAR, MAR(rank), MAR(median), MAR(upper), MAR(margins) and MNAR(upper).

contains missing values (MCAR, MAR & MNAR)	determines missing values (MAR)	 (MNAR)
X_2	X_1	X_2
X_4	X_2	X_4
X_5	X_6	X_5

Table 4.1: List of variables that contain missing values and determine the probability of missing values.

- *Validation*

  An independent test dataset served the purpose to evaluate the predictive accuracy of a Random Forest. It was created the same way as the training data though it contained 5000 observations and was completely observed. The accuracy was assessed by the mean squared error (MSE) which equals the misclassification error rate (MER) in classification problems.

- *Implementation*

  The implementation of the simulation was similar to the one described in section 3.3.3: The function `cforest()` of the `party` package (Hothorn et al., 2008, version 1.0-0) was used with settings $ntree = 50$, $mtry = 3$, $maxsurrogate = 3$, $mincriterion = 0$, $minsplit = 20$ and $minbucket = 7$. MICE was again applied by the function `mice()` of the `mice` package (van Buuren and Groothuis-Oudshoorn, 2010, version 2.11) to produce five imputed datasets with defaultMethod = c(”norm”, ”logreg”, ”polyreg”). Corresponding `R`-Code is given in section B.4.

In summary, there were 2 response types investigated for 6 processes to generate and 3 procedures to handle 4 different fractions of missing values. This sums up to 144 simulation settings. The simulation was repeated 1000 times.

4.3 Results

The following investigations are based on the classification analysis. Results for the regression problem are presented as supplementary material in section A.3 (Figure A.4) as they showed similar properties.

A general finding which holds for each analysis accentuates the well-known fact that unconditional permutation importance measures rate the relevance of correlated variables higher than that of uncorrelated ones with the same coefficients (Strobl et al., 2008). This becomes evident by the example of variables 1, 2 and 5. Although they are of equal strength the latter is assigned a lower relevance as it is uncorrelated to any other predictor; in some research fields this effect is appreciated to uncover relations and interactions among variables (cf. Nicodemus et al., 2010; Altmann et al., 2010). Also, there were no artificial effects observed for the non-influential variable 6 in any analysis setting.

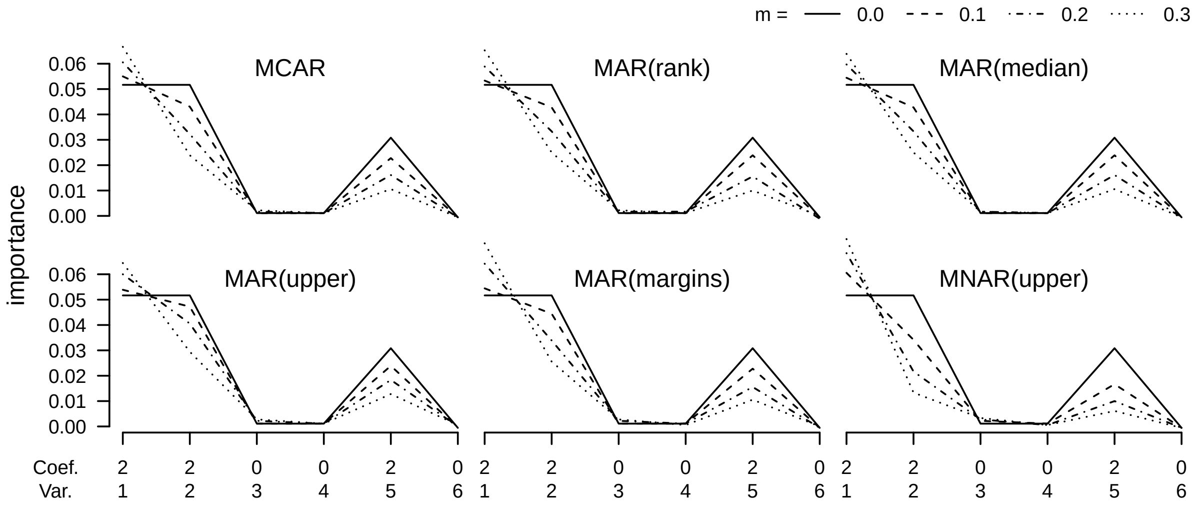


Figure 4.1: Median variable importance observed for the new importance measure in the classification problem (m = % of missing values in X_2, X_4 and X_5).

Findings for the new variable importance measure which is able to implicitly deal with missing values are displayed by Figure 4.1. According to expectations, the importance of variables 2, 4 and 5 decreased as they contained a rising amount of missing values. It is interesting to note that meanwhile the importance of variable 1 rose, although it does not seem to be directly affected. However, the findings of chapter 3 showed that this gain of relevance is justified: variables that are correlated and therefore provide similar information replace each other in a Random Forest when some of the information gets lost due to missing values. Accordingly, variable 1 takes over for variable 2 which results in an increased selection frequency of variable 1 in the tree building process. In conclusion, this approach is allowed to be affected by the occurrence of missing values as it mirrors the situation at hand, i.e. the relevance a variable takes in a Random Forest considering of the information it actually provides. The new importance measure appeared to be well suited for any of the missing data generating processes as results did not differ substantially.

Results for the complete case analysis – given by Figure 4.2 – showed undesired effects. A rising amount of missing values lead to a decreased importance of the completely observed variable 1. This might partly be explained by the general loss of information as some observations are discarded from analysis. However, the importance of variable 1 is not supposed to drop below that of variable 2 which is of equal strength yet contains the missing values. Unfortunately, this latter effect can be observed for every missing data generating process, except for MNAR(upper). It is most pronounced for MAR(upper) and MAR(margins). There is no rational justification for this property as variable 1 actually retains its information while other variables lose it. A proper evaluation of a variable's relevance is supposed to reflect this fact. Considering this vulnerability of complete case analysis to different missing data generating processes it should not be used for the assessment of importance measures when there are missing data.

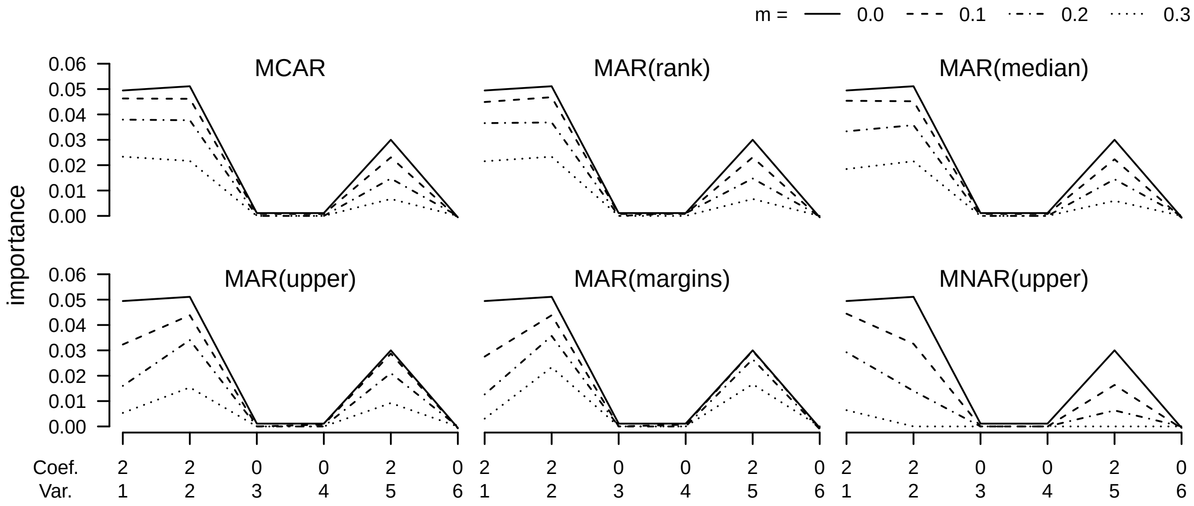


Figure 4.2: Median variable importance observed for the complete case analysis in the classification problem (m = % of missing values in X_2, X_4 and X_5).

An examination of Figure 4.3 reveals that multiple imputation, with as few as five imputed data sets, is a convenient way to maintain and recover the importance of variables that would have been observed if there were no missing data at all. This equally held for variables that contained missing values and those which were completely observed; none of their importances was arbitrarily decreased or increased. Even the importance of variable 5, which is only related to the outcome and therefore is associated with a rather weak imputation model, remained unaffected by the amount of missing values. The example of variable 4 shows that the imputation of non-influential variables did not induce artificial importances. All missing data generating processes showed these advantageous properties, except for the MNAR(upper) setting.

The prediction error produced by each approach for the independent test sample is displayed in Figure 4.4. For multiple imputation the prediction accuracy only slightly decreases with a rising amount of missing values. This effect is more pronounced for Random Forests that use surrogate splits; though there are only minor differences to multiple imputation (Rieger et al., 2010; Hapfelmeier et al., 2011, for similar findings). Complete case analysis appears to be much worse and leads to very high errors with a rising fraction of missing values. Missing data generating processes give comparable results within each approach. However, there is one exception for the MNAR setting that always causes the worst results. A corresponding evaluation of the regression problem is given as supplementary material in section A.3 (Figure A.5).

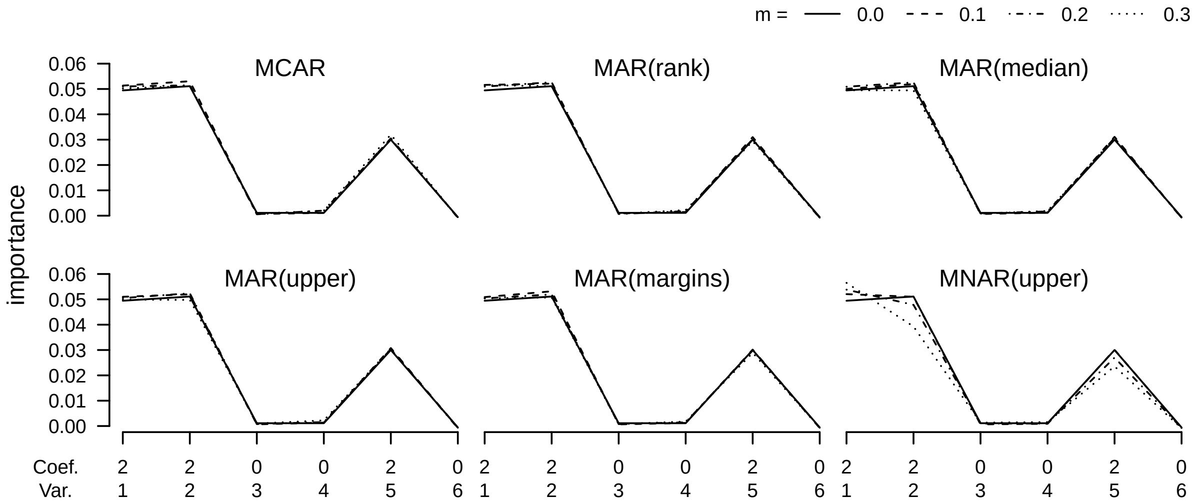


Figure 4.3: Median variable importance observed for the imputed data in the classification problem (m = % of missing values in X_2, X_4 and X_5).

4.4 Conclusion

The ability of a new importance measure, complete case analysis and a multiple imputation approach to produce reasonable estimates for a variable's importance in Random Forests has been investigated for data that contain missing values. Therefore, an extensive simulation study that employed several MCAR, MAR and MNAR processes to generate missing values has been conducted. There are some clear recommendations for application: Inappropriate results have been found for the complete case analysis in the MAR settings; it penalized the importance of variables that were completely observed in an arbitrary way. As a consequence, the sequence of importances was not able to reflect the true relevance of variables any more. This approach is not recommended for application. By contrast the new importance measure was able to express the loss of information exclusively for variables that contained missing values. Therefore, it should be used to describe the relevance of a variable under consideration of its actual information. In some cases one might prefer to investigate the relevance a variable would have taken if there had been no missing values. Multiple imputation appeared to serve this purpose very well except for the MNAR setting. An additional evaluation of prediction accuracy revealed that Random Forests based on multiple imputed data were mostly unaffected by the occurrence of missing values. Results were only slightly worse when surrogate splits were used to process missing values. Complete case analysis lead to models with the lowest prediction strength.

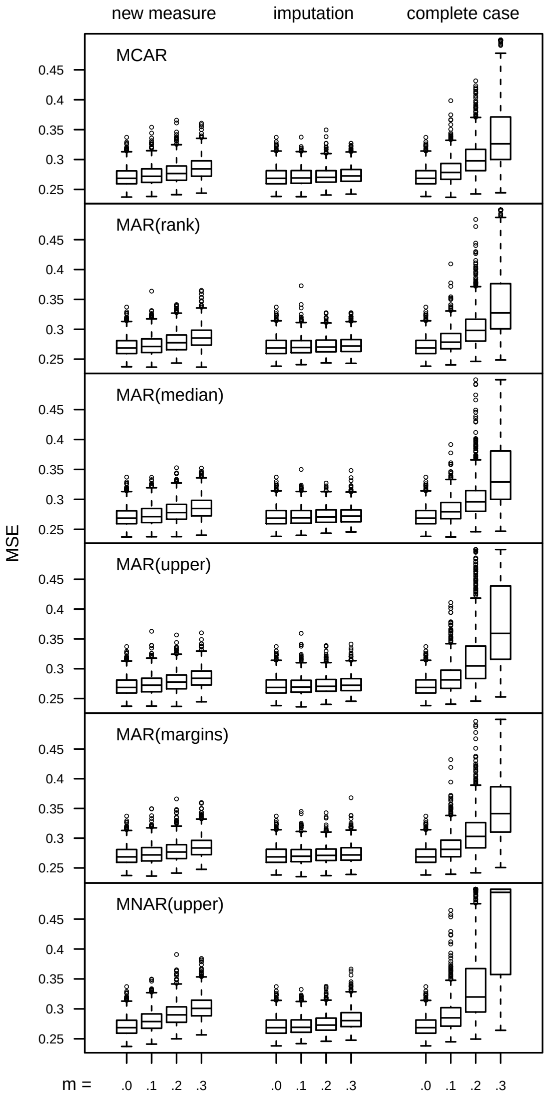


Figure 4.4: MSE observed for the classification problem ($m = \%$ of missing values in X_2, X_4 and X_5).

Chapter 5

A new Variable Selection Method

5.1 Research Motivation and Contribution

The application of variable selection mainly pursues two objectives: an improved prediction accuracy and the identification of relevant variables. Several approaches for variable selection have been proposed to combine and intensify these qualities (e.g. Díaz-Uriarte and Alvarez de Andrés, 2006; Rodenburg et al., 2008; Sandri and Zuccolotto, 2006; Tang et al., 2009; Yang and Gu, 2009); a summary of various selection methods and related publications is provided by Archer and Kimes (2008). However, the definition of the expression 'relevance' is strongly influenced by the research field the analytical question emerges from. For example, in the field of genome-wide association studies (GWAS) or microarray analysis, variable selection is used to uncover noticeable expression levels, influential genes and genes which are functionally related to the latter. Following this definition, variables which themselves are informative and variables which are non-informative but correlated to informative ones are termed to be of relevance. The identification of such variables is also supported by the application of variable importance measures that are sensitive to (cor-)relations between variables. For instance, measures like the permutation accuracy importance are well suited to detect relations and interactions among predictors (Nicodemus et al., 2010; Altmann et al., 2010); therefore it will be used by the new variable selection method and throughout the following investigations.

Alternatively, researchers might intend to solely aim at the identification of informative variables. The application of conditional importance measures could prove beneficial in such a case, and should be the subject of future research. In conclusion, it is still controversially discussed whether correlation should contribute to the importance and relevance of a predictor (section 1.2.2). However, recent proposals of importance measures (cf. Strobl et al., 2008) have provided the means to investigate data in either way and researchers are free to decide which kind of definition is more suitable for their respective research question.

Yang and Gu (2009); Zhou et al. (2010) show that the predictive power of a Random Forest may benefit from variable selection. Others like Altmann et al. (2010); Díaz-Uriarte and Alvarez de Andrés (2006); Svetnik et al. (2004) claim that the opposite might be true and that any approach should at least maintain a certain prediction accuracy when variables are rejected.

An extensive review of the corresponding literature showed that existing variable selection approaches are closely related to each other, yet differ in essential conceptual aspects. It also led to the development of a new variable selection approach which is based on the well known and established statistical framework of permutation tests. A comparison with another eight popular variable selection methods in three extensive simulation studies and four empirical evaluations indicated that the new approach is able to outperform its competitors as it:

- can be used to control the test-wise and family-wise error rate;
- provides a higher power to distinguish relevant from non-relevant variables, based on a clear, application-dependent, definition of 'relevance';
- achieves the highest fraction of relevant variables among selected variables;
- leads to models that belong to the very best performing ones;
- is equally applicable to regression and classification problems.

In contrast to the popular stepwise variable selection for regression models this work focuses on the detection of all relevant variables – even though they may be highly correlated. Also, the information of such variables is not necessarily unique and might just as well be omitted for prediction purposes. However, there is no need to create a sparse prediction model in order to prevent overfitting. Random Forests implicitly deal with this issue as they are fitted to random subsets of the data and perform splits in random subsets of the variable space. Quite the contrary, variable selection might even harm their performance when too much information is left out. Therefore, the main focus is put on the selection of all variables which are of relevance and to simultaneously improve or at least maintain the prediction accuracy that can be achieved without any selection. Both of these goals were explored for a new approach which stands on solid theoretical grounds. Note that it can generally be applied to any kind of Random Forest algorithm without any restrictions. The new variable selection method and corresponding investigations of this chapter have been published in (Hapfelmeier and Ulm, 2012).

5.2 Variable Selection

Various variable selection approaches have been proposed for application in different research fields. A work of Guyon and Elisseeff (2003) provides an overview of general findings and methodologies. Some of these ideas, like the permutation of variables or the application of cross-validation, are summarized in the following discussion about variable selection with Random Forests.

An investigation of the corresponding literature makes clear that the basic element of variable selection is given by importance measures. They are used by all approaches – which therefore can be allocated to the family of embedded variable selection methods (cf. Abeel et al., 2010; Guyon and Elisseeff, 2003), to guide the decision of whether a variable should be included in or rejected from the model. However, there are also many differences concerning the number of rejection or inclusion steps, the fraction of variables rejected per step, the (re-)calculation of variable importances, the kind of importance measure, the method to assess prediction accuracy, the application of sampling methods, forward or backward selection and the stopping criterion.

A very prominent difference can be used to distinguish two major classes. One is to repeatedly fit models to the data in order to determine the best performing one in terms of prediction accuracy. Related methods are henceforth called 'performance-based approaches'. A second kind applies a permutation test framework to estimate the significance of variable importances. These methods are henceforth termed 'test-based approaches'. The following sections will further discuss and explain the most popular representatives which are also used in the simulation and application studies.

5.2.1 Performance-based Approaches

Performance-based approaches are popular and widely used in many research fields. Although there is some diversity, most of the existing methods only differ in minor aspects while they share the same methodological scheme. The following listing presents and discusses well-established representatives of these methods:

- Svetnik et al. (2004) produce several orderings of variables by the computation of importance measures on each training set of a 5-fold cross-validation. The error of the corresponding test sets is recorded while the number of variables used to build a Random Forest is halved along the orderings. The procedure is repeated 20 times to 'smooth out' the variability and the minimum averaged error is used to determine the optimal number of variables. The final model is now fit to this optimal number of most relevant variables, i.e. with the highest importance measures, in the entire dataset.

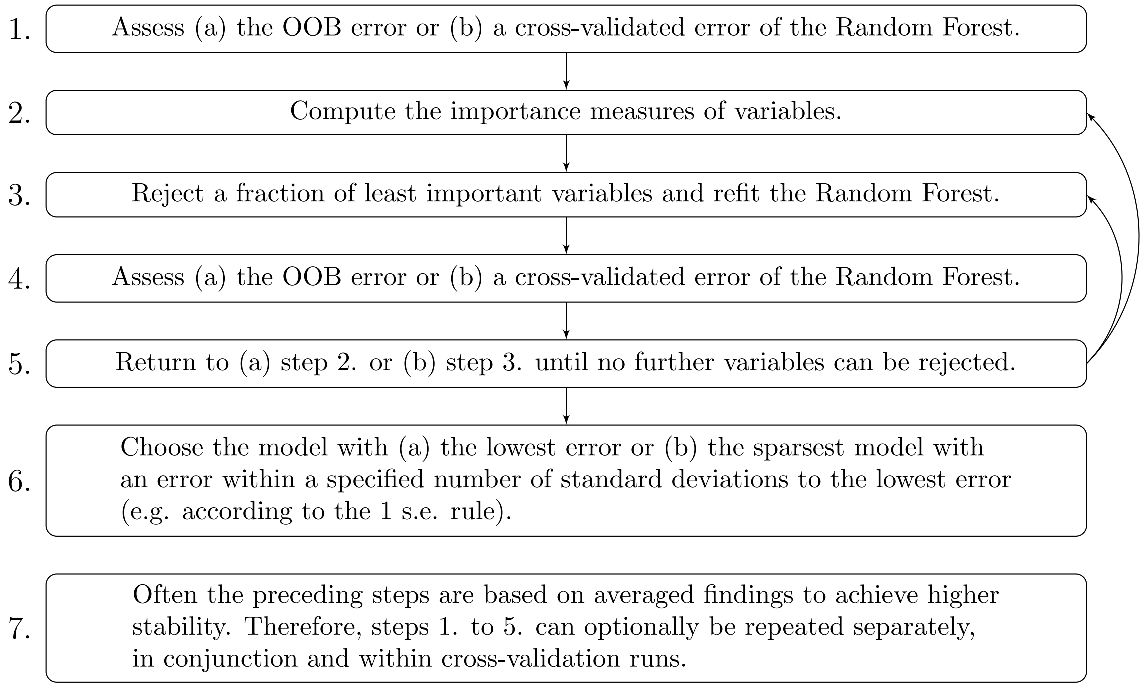
  Svetnik et al. (2004) repeatedly emphasize that there are two dangers that lead to overfitting. The first one is given by the recalculation of importance measures after each variable rejection step. In a simulation study this 'recursive' approach is shown to be inferior to the 'non-recursive' approach which computes the variable importances only once using the entire variable set. The second risk arises when the OOB error is used for evaluation purposes instead of cross-validation. The explanation is that the observed OOB errors are related to the sequence of variable importances as the latter were computed on the entire data or on the OOB observations. Consequently the independence of the model fit and evaluation is violated.

- Jiang et al. (2004) introduce a method in which they claim to combine the unsupervised 'gene shaving' approach (cf. Hastie et al., 2000) and the supervised Random Forests. They follow a very similar concept to Svetnik et al. (2004) though they do not repeat the algorithm several times. There are three major differences: Firstly they act against the warnings about overfitting as they use the OOB error. Secondly, they recalculate the variable importances after each rejection step. And thirdly, two data sets are actually used to determine the minimal error. A Random Forest is fit to each of the datasets and the OOB error is recorded. An application of the Random Forests to the datasets they were not fit to produces another, independent assessment of the prediction error. The aggregation of the OOB errors and the prediction errors leads to the final decision of the optimal number of variables. The incorporation of independent prediction errors, instead of OOB errors only, might result in a variable selection which is more robust to overfitting.

- Díaz-Uriarte and Alvarez de Andrés (2006) present an approach which is again very similar to the ones suggested by Jiang et al. (2004) and Svetnik et al. (2004). It uses the OOB error and computes variable importances only once. The best model is chosen to be the smallest one with an error rate within u standard errors of the best performing model. Setting $u = 1$ equals the 'one-standard-error' rule ('1 s.e.' rule) known from works about classification trees (cf. Breiman et al., 1984; Hastie et al., 2009). The authors are well aware of the fact that the OOB error might lead to overfitting. Therefore, the error is investigated in an additional simulation study by the .632+ estimator (cf. Efron and Tibshirani, 1997). By contrast to the findings of Svetnik et al. (2004) no overfitting was detected in this analysis. In a concluding remark the authors state that their approach returns small sets of uncorrelated variables which are able to retain predictive performance.

- Genuer et al. (2010b) repeatedly (e.g. 50 times) fit Random Forests to the data and record the resulting variable importances. The averaged values are used to produce an ordering of variables. Along this sequence Random Forests are repeatedly fit to an increasing number of variables (e.g. another 50 times). An application of the 1 s.e. rule to the average of the observed OOB errors leads to the determination of the best model. The authors call this first step the "variable selection procedure for interpretation" meant to "find important variables highly related to the response for interpretation purpose". In a second step they propose another method, the "variable selection procedure for prediction", which further reduces the selected variable set and is supposed to achieve a better prediction accuracy. Therefore a threshold is computed by averaging the error decreases observed at each inclusion of a variable after step one. This threshold is supposed to reflect the "average variation obtained by adding noisy variables". Now, again following the initial sequence a variable is only selected if the error gain of the corresponding Random Forests exceeds the threshold. Interesting applications of this approach exist in the fields of brain state decoding or malaria infectiousness (cf. Genuer et al., 2011; Genuer et al., 2010a).

A summary of the corresponding computational steps is given by the following pseudo-code. The possibility to abstract all methods in a single pseudo-code highlights the similarity of approaches:



1. Assess (a) the OOB error or (b) a cross-validated error of the Random Forest.
2. Compute the importance measures of variables.
3. Reject a fraction of least important variables and refit the Random Forest.
4. Assess (a) the OOB error or (b) a cross-validated error of the Random Forest.
5. Return to (a) step 2. or (b) step 3. until no further variables can be rejected.
6. Choose the model with (a) the lowest error or (b) the sparsest model with an error within a specified number of standard deviations to the lowest error (e.g. according to the 1 s.e. rule).
7. Often the preceding steps are based on averaged findings to achieve higher stability. Therefore, steps 1. to 5. can optionally be repeated separately, in conjunction and within cross-validation runs.

An example of application is given by Zhou et al. (2010) who use the method suggested by Svetnik et al. (2004). However, they follow a recommendation of Díaz-Uriarte and Alvarez de Andrés (2006) as they investigate a different kind of variable importance measure based on proximity. Another interesting application is given by the work of Chehata et al. (2009). The authors use the method of Díaz-Uriarte and Alvarez de Andrés (2006) to select multi-echo and full-waveform lidar features for the classification of urban scenes.

There are many new and innovative proposals of alternative methods. These are briefly presented here though they are not further investigated as they are not commonly used or represent special applications in specific research fields. Still it is worthwhile to list them in order to highlight the variety of ongoing developments: Sandri and Zuccolotto (2006) suggest the computation of four different kinds of importance measures for each variable. The distance to a corresponding four-dimensional centroid determined by the importance vectors of all variables is meant to reflect the relevance of a variable. Noise variables are supposed to cluster together and therefore have a strong impact on the position of the centroid. Thus, distant observations which exceed a specified threshold, are meant to represent relevant variables. Yang and Gu (2009) propose a method in the field of GWAS which is able to deal with thousands of single-nucleotide polymorphisms (SNPs). Therefore, Random Forests are fit to changing subsets of SNPs and global importances are determined. This procedure was shown to outperform Random Forests fit to the entire data in terms of power to detect relevant SNPs. A similar approach is presented by Schwarz et al.

(2007) who also suggest to fit Random Forests to subsets of variables. Afterwards averaged importances are used to determine a ranking of variables. Along this ranking, the Random Forest that generates a local minimum for the OOB error is selected. Another very simple but widely applicable approach is proposed by Strobl et al. (2009). They suggest to assess the random variability of non-informative variables by the range of observed, negative importance measures. Only variables with a positive importance that exceeds this range are termed informative and supposed to be subject of further exploratory investigations.

5.2.2 Test-based Approaches

Altmann et al. (2010) present a method that uses a permutation test framework to produce unbiased importance measures (cf. Strobl et al., 2007b). In addition, the approach offers the opportunity to perform variable selection: In a first step the importance measures of variables are recorded. In a second step the response variable is permuted several times. Each time a new Random Forest is fit to the data which now contains the permuted response vector. The corresponding importance measures are used to determine their empirical distribution under the condition that the relation between predictors and outcome is destroyed by permutation. In combination with the original importance measures assessed in the initial step, a p-value can be assigned to each variable. Variable selection can now be performed by the rejection of variables with a p-value above a certain threshold, e.g. > 0.05. The authors also claim that this approach distinguishes informative from non-informative variables, improves prediction accuracy and selects correlated variables in groups which is a major advantage to identify functionally related genes in microarray experiments.

An almost identical approach has already been introduced earlier by Rodenburg et al. (2008) whereas these authors directly aim at the introduction of a variable selection approach. They repeat the procedure several times and combine the selected variables in a final set. Another related work of Wang et al. (2010) is based on a different kind of importance measure called the ‘maximal conditional chi-square importance’ to identify relevant SNPs in GWAS. Following the same research goal Tang et al. (2009) simultaneously permute entire sets of SNPs which belong to the same gene. This method is closely related to the approach of Altmann et al. (2010) as permuting the response vector equals a permutation of the largest group of variables available, which is the entire set.

Due to their similarities the approaches can again be summarized by pseudo-code:

```
1. Compute any kind of importance measure using the original data.
2. Permute groups of variables (up to all variables available) several times
   to assess the empirical distributions of importance measures when the
   relation to the response – and the remaining variables – is destroyed.
3. Assess the p-value for each variable by means of the empirical
   distributions and the original importance measures.
4. Select the variables with a p-value below a certain threshold, e.g. $\leq 0.05$.
```

The concept of significance tests for variable importance measures has already been introduced by Breiman and Cutler (2008). However Strobl and Zeileis (2008) revealed that the power of such tests depends on the number of trees in the Random Forest. Consequently they state that due to "alarming statistical properties ... any statement of significance made with this test is nullified". Likewise, for the approaches presented in this section, conclusions about the significance of importance measures have to be drawn with care. It is highly ambiguous what kind of hypothesis is investigated by the corresponding tests. An exploration of properties on the grounds of a permutation test framework is given in the following section. It reveals apparent shortcomings which lead to the introduction of a new approach that supports an improved interpretability and is based on sound theory.

5.2.3 New Method

The following section contains a discussion about an adequate permutation test for variable importance measures. There are obvious similarities to section 1.2.2 as the construction of the permutation accuracy importance is based on the same framework. By contrast, permutation is used for hypothesis testing here. This demands for a new detailed exploration of its statistical properties and the effects of an improper application.

Rationale

To identify the most relevant and informative variables from a set of candidates one may pose the question: "Which variables are related to the outcome?" A formulation in mathematical terms transforms this question into a null-hypothesis of independence between the response Y and a variable X_j, $(j = 1, \dots, v)$:

$$H_0 : Y \perp X_j,$$

where X_j comes from the v-dimensional vector $\mathbf{X} = (X_1, \dots, X_v)$. A variable which violates this hypothesis is termed to be relevant. Whether information against it is present in the data can be assessed in a permutation test framework (see Efron and Tibshirani, 1994;

Good, 2005, 2000, for further insight in basic principles). Therefore X_j is permuted to destroy its relation to Y. The data created this way are used to refit the Random Forest and to recalculate the variable importance. Values observed across several repetitions of the permutation are used to reflect the empirical distribution of importance measures under the null-hypothesis. Finally, the location and likelihood of the original importance measure within this distribution can be used to judge its concordance with the null-hypothesis and to produce a p-value.

However, it is well known that the permutation of X_j does not only destroy its relation to Y but also to the remaining variable space $\mathbf{Z} = \mathbf{X} \setminus X_j$ (cf. Strobl et al., 2008). Therefore a non-informative variable can still show an increased variable importance if it is related to an informative one. The correct question which truly fits to the permutation scheme now extends to: "Which variables are related to some other or to the outcome?" A reformulation of the corresponding hypothesis is

$$H_0 : Y, \mathbf{Z} \perp X_j. \tag{5.1}$$

In many research fields like the analysis of GWAS or microarray data this is a desirable property as relations among variables are of major interest; by contrast, this hypothesis might not always agree with the research question of interest, e.g. when correlation should not be a relevant factor for variable selection (cf. discussions of section 5.1). An example emerging from the field of ecology is given by Cutler et al. (2007).

Based on these considerations it is unclear which information is provided by the permutation scheme proposed by Altmann et al. (2010) and Tang et al. (2009). These authors suggest to simultaneously permute multidimensional vectors (i.e. groups) of variables $\mathbf{X}^* \subseteq \mathbf{X}$ which destroys the relation of each of these variables to the outcome and also to variables $\mathbf{Z}^* = \mathbf{X} \setminus \mathbf{X}^*$ which themselves are not part of the permutation scheme. The resulting empirical distributions do not reflect H_0 (5.1) but a null-hypothesis which, for all variables $\mathbf{X}^*$ within the mutual permutation, postulates independence of the outcome Y and the variables $\mathbf{Z}^*$:

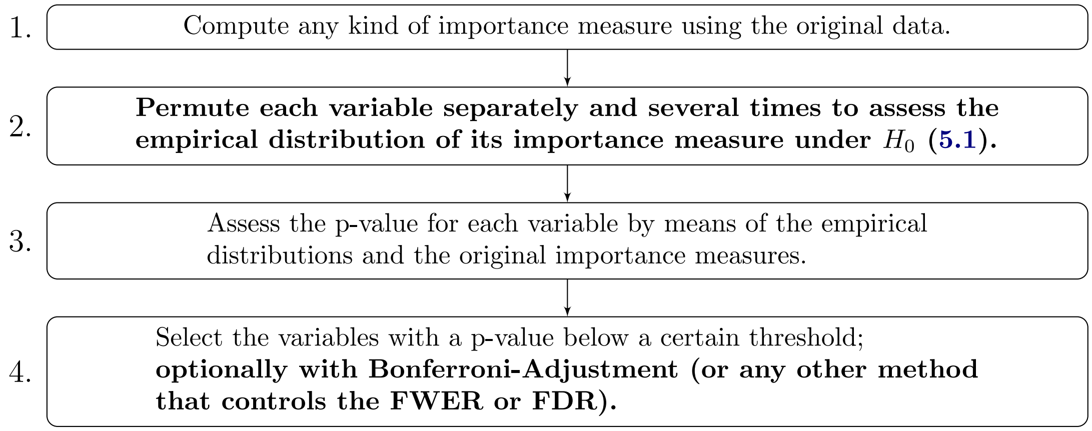
$$H_0^* : Y, \mathbf{Z}^* \perp \mathbf{X}^*. \tag{5.2}$$

As a consequence evaluations of the original importance measures within the empirical distributions under H_0^* (5.2) can not be used to draw conclusions about the significance of a single variable X_j.

Another point of concern is that the significance of each variable is tested independently. This way only the test-wise error rate (TWER = probability of a null-hypothesis to be falsely rejected) can be controlled at a threshold α (e.g. $\alpha = 0.05$). However, the procedure obviously leads to a multiple testing problem. Therefore it is advisable to apply a correction method that helps to control the family-wise error rate (FWER). A simple but effective method is given by the Bonferroni-Adjustment which is suggested for the new approach. Hereby the probability of at least one false rejection among a set of true null-hypotheses is bounded by α (Hastie et al., 2009). Likewise any correction method to handle the multiple testing problem can be applied. This also includes methods which control the false discovery rate (FDR) instead of the FWER (e.g. the Benjamini-Hochberg or Benjamini-Yekutieli procedures; cf. Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001, for details).

Implementation

Following these considerations we introduce a method which rates the relation between a variable and the outcome and other variables, i.e. its relevance according to H_0 (5.1), on a sound theoretical permutation test framework. The new approach differs in slight but essential adaptions from the test-based approaches (differences are highlighted):



1. Compute any kind of importance measure using the original data.
2. **Permute each variable separately and several times to assess the empirical distribution of its importance measure under H_0 (5.1).**
3. Assess the p-value for each variable by means of the empirical distributions and the original importance measures.
4. Select the variables with a p-value below a certain threshold; **optionally with Bonferroni-Adjustment (or any other method that controls the FWER or FDR).**

This way, as a single variable X_j is permuted, it is assured that the distribution of importance measures is simulated under the null-hypothesis of interest, namely H_0 (5.1). A permutation of the response Y, as a special case for the simultaneous permutation of several variables when $\mathbf{X}^* = \mathbf{X}$, follows H_0^* (5.2) which can not validly reflect the significance of single variables X_j.

In the following simulation studies (section 5.4.1 and 5.4.1) approaches basing on H_0^* (5.2) only achieve an inferior power to detect relevant variables. A closer examination of results led to a simple explanation: A permutation of the response Y destroys any relation to the entire variable space $\mathbf{X}$. As a consequence, Random Forests are forced to use any of such non-informative variables to build a model and will frequently allocate an importance score of a certain size to them. By contrast, Random Forests are still able to select informative variables if only a single variable of interest is permuted under H_0 (5.1). Accordingly, this permuted variable will not be able to contribute to the Random Forest and will therefore only achieve very low importance scores, instead. For this reason, the resulting upper percentiles, which represent the critical values of the corresponding permutation tests, of the empirical distribution obtained under H_0 (5.1) are probably lower than the ones under H_0^* (5.2). This results in a higher power to detect relevant variables for the new approach. A short yet explicit example of these properties is given by Figure 5.1 for the relevant variable X_1 in the case of a regression problem: $y = x_1 + x_2 + x_3 + x_4 + e$, $e \sim N(0, 0.5)$, $x_j \overset{iid}{\sim} N(0, 1)$.

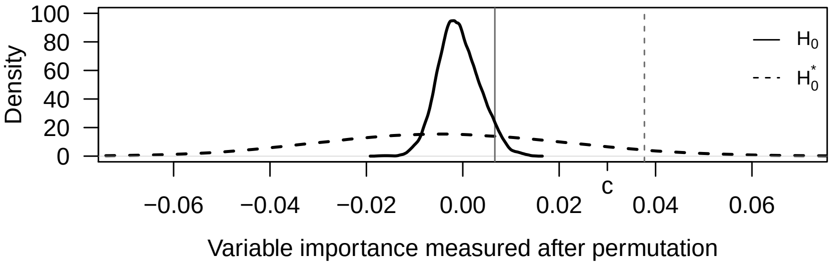


Figure 5.1: Kernel density estimation of importance measures obtained for X_1 after permutation following H_0 and H_0^*. Vertical lines indicate the respective 95% percentiles, i.e. the critical values of one-sided permutation tests on 5% significance levels. For instance: An initial importance taking a value of c would lead to a significant test result under H_0 and a non-significant result under H_0^*.

The necessity of a re-computation of Random Forests and variable importances after each permutation of a predictor variable makes this approach computationally demanding. Yet, after an expensive and even more time-consuming data collection phase e.g. of microarray data, the main focus should be put on a meaningful selection and not on the expenditure of time. In addition, Altmann et al. (2010) already outlined that the entire process might be parallelized on multiple cores of a system to significantly reduce computation time.

5.3 Studies

The new method is compared with the approaches introduced in section 5.2 in terms of prediction accuracy and even more importantly, the ability to distinguish relevant from non-relevant variables. Three simulation studies were conducted. The first two are used to explore theoretical properties while the third one represents an application to an artificial dataset. In addition, the performance of each method was assessed by four empirical evaluations.

5.3.1 Simulation Studies

The first simulation study (Study I) explores the TWER and FWER of the approaches. For both, an error is defined to be the selection of a non-relevant variable, which is non-informative and not correlated to any informative one. By definition, the new approaches are to control the TWER and FWER at a specified level (e.g. $\alpha \leq 0.05$).

The second simulation setting (Study II) is meant to shed light on the power of the approaches to identify relevant variables and to distinguish them from non-relevant ones. According to H_0 (5.1) there are two aspects that might affect this ability and that need to be checked for: the predictive strength of a variable and the correlation between variables.

A third simulation setting (Study III) represents the more specific case of an application to a simulated, artificial dataset. It includes a broad assemblage of relevant and non-relevant variables, in total there are 20, with differing correlation schemes. This way the properties of each method can be examined in an extensive but known lineup of settings. This time, next to selection frequencies, focus was also put on prediction accuracy. As it might not always improve when variable selection is applied to Random Forests (Díaz-Uriarte and Alvarez de Andrés, 2006) a baseline is given by the performance of a Random Forest using the entire variable set.

The variable selection methods are investigated in classification and regression problems. Therefore a categorical (binary) and a continuous outcome are created for Studies I, II and III. The continuous outcome is modeled by means of a linear model

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

while values of the binary outcome follow a Bernoulli distribution $B(1, \pi)$. The parameter π was modeled 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 contains 100 observations drawn from a multivariate normal distribution with mean vector $\vec{\mu} = 0$ and covariance matrix Σ. As the variances of each variable are chosen to be 1, the covariance equals the correlation in this special case.

For Study I,

$$\Sigma_I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \beta_I = (0, 0, 0, 0, 0)^\top.$$

This makes five non-relevant (non-informative and uncorrelated) variables. A selection of these variables is rated as an error with reference to the TWER and FWER. The uncorrelated structure of variables assures the stochastic independence of permutation tests conducted by the new approaches. This way their ability to control the TWER and FWER can be validly investigated.

In Study II,

$$\Sigma_{II} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & .7 & 0 & 0 \\ 0 & 0 & 1 & 0 & .7 & 0 \\ 0 & .7 & 0 & 1 & 0 & 0 \\ 0 & 0 & .7 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \beta_{II} = (s, s, s, 1, 0, 0)^\top.$$

With $s \in \{0, 0.1, 0.2, ..., 1\}$ one is able to investigate how an increasing strength of a predictive variable affects selection frequencies. In combination with Σ_{II} this effect can be observed for variables that are uncorrelated, correlated with an informative variable and correlated with a non-informative variable.

For study III,

$$\Sigma_{III} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 1 & .7 & .7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & .7 & 1 & .7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & .7 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & .7 & .7 & .7 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & .7 & .7 & .7 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & .7 & 1 & .7 & .7 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & .7 & .7 & 1 & .7 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & .7 & .7 & .7 & 1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1
\end{pmatrix} \quad \text{and}$$

$$\beta_{III} = (3, 2, 1, 3, 2, 1, 3, 2, 1, 0, \cdots, 0)^\top.$$

The structure of Σ_{III} reveals that the 20 variables are assembled in correlated blocks of different sizes; while it equals the identity matrix for variables $14 - 20$. The first three informative variables are not correlated at all. They are compared to a triplet of informative variables which is of the same strength but is correlated. The next block of variables is identical but contains two additional non-informative – yet, due to correlation, relevant – variables. These are followed by two non-relevant variables which are correlated to each other and another seven, non-informative and uncorrelated variables. The combination of different prediction strength and correlation schemes is supposed to answer questions about the influence of these factors on selection frequencies.

Variable selection was repeated in 5000 simulation runs for study I and 1000 simulation runs for study II and III. In each run and study, the data were made up of 100 observations. For validation purposes an independent dataset consisting of 5000 observations was created once in study III. It was used to measure the prediction strength of a Random Forest by the MSE which equals the misclassification error rate (MER) in classification problems. The corresponding R-Code is given in the appendix B.5.2.

5.3.2 Empirical Evaluation

Analysis settings

Empirical evaluations of four datasets (two regression and two classification problems) have been conducted to investigate the performance of each method. Observed selection frequencies can only be used to explore differences between methods but not to evaluate their concordance with the real relevance of a variable because it is unknown. The performance is again measured by means of the MSE. Bootstrap sampling makes it possible to draw conclusions about the predictive strength of a modeling strategy. Therefore a number of observations, which equals the initial size of the data, is repeatedly drawn with replacement. This way approximately $1 - (1 - 1/n)^n \approx 1 - e^{-1} = 63.2\%$ of observations make up the training set while the remaining observations are used as test set. The .632 estimator suggested by Efron (1983) provides a less biased estimate of the MSE compared to sampling methods like cross-validation or resubstitution (cf. Boulesteix et al., 2008a; Hastie et al., 2009). It is defined as a weighted sum of the overoptimistic resubstitution estimate $\overline{mse}$ and the too pessimistic leave-one-out bootstrap $\widehat{MSE}^{(1)}$ estimate:

$$\widehat{MSE}^{(.632)} = 0.368 \cdot \overline{mse} + 0.632 \cdot \widehat{MSE}^{(1)}.$$

Another appealing property of the bootstrap is that it is commonly used to evaluate a variable's importance by the number of times it is selected within several bootstrap samples. Examples for linear, logistic or Cox regression can be found in publications of Austin and Tu (2004); Sauerbrei et al. (2007); Sauerbrei (1999). Further examples for microarray data are given by Qiu et al. (2006).

In summary the bootstrap method is used for two obvious reasons: Firstly to produce an estimate of low bias for the MSE and secondly to evaluate the selection frequencies of variables. Therefore 1000 bootstrap samples were repeatedly drawn for each data set. Variable selection was additionally conducted on the entire data to produce a selection that is based on the entire information. A baseline assessment of the MSE is again given by the performance of a Random Forest without any kind of variable selection. The corresponding R-Code is given in the appendix B.5.3.

Data

Four well known datasets, two regression and two classification problems, with differing numbers of variables and observations were chosen for the application studies. These are the Infant Birth Weight Data and Heart Disease Data already introduced in section 2.4.1. Two other datasets have been added:

- The **Boston Housing Data** was originally used by Harrison and Rubinfeld (1978) to assess the willingness of citizens to pay for clean air. One of their objectives was to model the median value of owner-occupied homes by regression analysis. Here, this task is solved by an application of Random Forests using demographic and economic factors (e.g. crime rate, proportion of land over 25,000 sq. ft., proportion of non-retail business acres per town, a Charles River dummy variable, nitric oxides concentration, average number of rooms per dwelling, proportion of owner-occupied units built prior to 1940, weighted distances to five Boston employment centres, index of accessibility to radial highways, full-value property-tax rate, pupil-teacher ratio, a factor for the proportion of blacks in the town and the fraction of lower status of the population). The data consists of 506 observations and 13 independent variables.

- The **Parkinson's Disease Detection Dataset** contains voice recordings of several healthy people and people suffering from Parkinson's disease. Little et al. (2007) originally used it to investigate new feature extraction and speech analysis methods for general voice disorders. The distinction of healthy and diseased people creates a classification problem which is meant to be solved by the aid of several biomedical voice measurements. In total there are 22 independent variables measured on 195 observations.

A summary of data characteristics is given in table 5.1. Except for the Infant Birth Weight Data which was taken from the `R` package `MASS`, the data are provided by the open source UCI Machine Learning Repository (Frank and Asuncion, 2010).

	Data	# obs.	# var.
regression	Birthweight	189	9
	B. Housing	506	13
classification	Heart	270	13
	Parkinson	195	22

Table 5.1: Characteristics of data used in the application studies.

5.3.3 Implementation

Analyses of this chapter were again performed with the `R` system for statistical computing (R Development Core Team, 2011, version 2.14.1). The computation of unbiased Random Forests based on a conditional inference framework is provided by the function `cforest()` which is part of the package `party` (Hothorn et al., 2008, version 1.0-0). Each Random Forest contained *ntree* = 100 trees. The number of randomly selected candidate variables for splits `ntry` was chosen to be the square root of available variables, following the recommendation of Díaz-Uriarte and Alvarez de Andrés (2006). Sticking to the default setting *mincriterion* = 0, there were no restrictions concerning the significance of a split. Trees were grown until terminal nodes contained less than *minsplit* = 20 observations while child nodes had to contain at least *minbucket* = 7 observations.

5.3.4 Settings of Variable Selection Approaches

The variable selection methods presented in section 5.2 were implemented as R-functions (corresponding codes are given in the appendix B.5.1). Although the instructions of authors were closely followed, a minor adjustment was made for the performance-based approaches. The rejection of a certain fraction of variables was reduced to one variable in each step in order to investigate a finer grid. Another adaption, which deviates from the original definitions but is felt to be a major improvement, is that performance-based approaches were empowered to select no variables at all. Therefore the prediction of such a null-model is simply given by the majority vote of classes (for binary outcomes) or the mean outcome (for continuous outcomes). The resulting MSE is compared to the performance of Random Forests at different variable selection stages. Within the algorithm it technically represents one of the MSE values that can be chosen to be optimal e.g. according to the 1 s.e. rule.

For test-based approaches tests were conducted in a one-sided manner as only values on the right margin of the empirical distribution of importance measures (i.e. high values as opposed to low or negative values) provide evidence against the null-hypothesis of a non-relevant variable.

The performance-based approaches have originally been constructed for classification problems only. Consequently, the discussion and presentation of results focus on such data. Technically, they are applicable to regression problems, too. However, in such cases only the 0 s.e. rule is executed to stick close to the original definitions. In principle, and this is subject to further research, Breiman et al. (1984) already suggested an adaption of the 1 s.e. rule for regression analysis. In addition there have been proposals of advanced cross-validation to select optimal models in linear regression analysis (Shao, 1993; Zhang, 1993). If adapted to the performance-based variable selection approaches, such methods could prove beneficial to reduce type-I errors, i.e. lessen the selection of non-relevant variables. Likewise, the s.e. rule could serve as a kind of tuning parameter for this purpose. Furthermore, the test-based approaches can easily be modified to potentially control for the TWER and FWER in the same manner as the new proposal.

A summary of all variable selection methods and specific settings is given in Table 5.2. It also contains a list of labels which will be used to name the methods in the following. The algorithms themselves are presented in section 5.2.

5.4 Results

5.4.1 Simulation Studies

Study I

Study I was designed to explore the TWER and FWER of the variable selection approaches for non-relevant variables. Table 5.3 shows that the approaches NAP, NAP.B and ALT are able to control the TWER of 5%. NAP.B meets the expectations and even controls the FWER due to the application of the Bonferroni-Adjustment. Although other approaches

Publication	Label	Settings
Svetnik et al. (2004)	SVT	- 5-fold CV - 20 repetitions
Jiang et al. (2004)	J.0	- 0 s.e. rule*
	J.1	- 1 s.e. rule*
Díaz-Uriarte and Alvarez de Andrés (2006)	D.0	- 0 s.e. rule
	D.1	- 1 s.e. rule
Genuer et al. (2010b)	G.i	- model for interpretation
	G.p	- model for prediction
Altmann et al. (2010)	ALT	- 400 permutation runs - $\alpha = 0.05$**
New Approach	NAP	- 400 permutation runs - $\alpha = 0.05$**
	NAP.B	- 400 permutation runs - $\alpha = 0.05/n_{\text{tests}}$**,***
–	All	- all variables are used

*only the OOB error is computed
**one-sided test
***Bonferroni-adjusted significance level

Table 5.2: Summary of variable selection methods, labels and specific settings.

have originally not been constructed to control for the TWER or FWER they are judged the same way (see section 5.3.4 for a discussion of possible modifications). Thus, J.0, D.0 and SVT produce TWER which are far beyond the threshold of 5%. Others like G.i, G.p, D.1, and J.1 at least show a less pronounced violation of this restriction. Accordingly, approaches are allocated to three classes to facilitate the presentation and comparison of results in the following. Class I is made up by approaches which control the TWER or FWER – not surprisingly these are the test-based approaches. Class II contains approaches which showed an TWER of moderate extent. Finally class III consists of approaches which showed a severe TWER.

For the regression problem NAP, NAP.B and ALT controlled the TWER and FWER the same way as for the classification problem (cf. Table A.3 in appendix A.4.1). The error rates of their competitors were again too high, with one exception for G.p, while differences between methods were less pronounced. Once again it becomes evident that they were not constructed for the purpose of controlling neither the TWER nor the FWER.

Another interesting finding is given for the approaches using the 0 s.e. rule. Both produce very high TWER and are therefore members of class III. The explanation is quite simple: As long as the prediction accuracy does not improve by omitting variables, the full variable set, or at least a subset that contains non-relevant variables, may provide the best performing model. Considering this fact it is advisable to apply the 1 s.e. rule which choses the smallest set of variables among models of comparable performance.

	TWER					FWER	Class
	var. 1	var. 2	var. 3	var. 4	var. 5		
NAP.B	1.0%	0.8%	0.7%	1.2%	0.9%	4.4%	I
NAP	5.1%	5.0%	4.9%	5.7%	5.3%	22.7%	I
ALT	5.3%	4.7%	5.0%	5.2%	5.2%	23.3%	I
G.p	9.2%	9.1%	8.6%	9.9%	9.4%	40.6%	II
D.1	13.9%	14.8%	14.4%	15.1%	14.9%	49.7%	II
J.1	14.1%	14.8%	14.8%	15.2%	15.1%	54.0%	II
G.i	18.7%	19.1%	18.4%	20.0%	19.7%	64.2%	II
J.0	29.1%	28.5%	28.7%	29.8%	29.9%	79.7%	III
D.0	29.1%	29.3%	29.7%	29.5%	30.6%	75.7%	III
SVT	29.7%	29.6%	29.1%	29.8%	30.2%	58.9%	III

Table 5.3: TWER and FWER for 5000 simulation runs of the classification problem in study I. Approaches are ranked and allocated to classes I, II and III according to their mean TWER.

Study II

Study II was designed to investigate the power of the approaches to discriminate relevant from non-relevant variables. Results are shown by Figure 5.2 which displays the observed selection frequencies against a rising strength s of the predictor variables. For a clear presentation, classes are subsumed by gray areas which range from the lowest to highest values observed for the corresponding approaches. The evident discrepancy of classes confirms the definition found in study I. A more detailed view and discussion of each approach is given in the appendix A.4.1.

An examination of selection frequencies for the non-relevant variable 6, and variables 1, 3, and 5 when $s = 0$, makes it clear that approaches of class III heavily exceed the TWER of 5% (which is indicated by a horizontal line in the plots). Accordingly, it is not surprising that they also show a high power to detect the relevant variables 1, 2, 3, 4 and 5 (when $s > 0$). However, this comes at the cost of a high TWER. It is also interesting to note that for some of the approaches, the selection frequency of the relevant variable 4 even drops with a rising strength of variables 1, 2 and 3. The new approaches NAP and NAP.B show competitive and in some cases even superior results though they control the TWER, or even the FWER in the case of NAP.B.

Results for class II (+ ALT) indicate that these approaches are much more capable to control the TWER. However, in the majority of cases they are outperformed by NAP and NAP.B which show a much higher power to select the relevant variables. The new approaches also seem to be the only ones which are able to detect variable 5, which is non-informative yet relevant as it is correlated to the informative variable 3. They also show a constant selection frequency for variable 4 – independent of the amount and strength of informative variables.

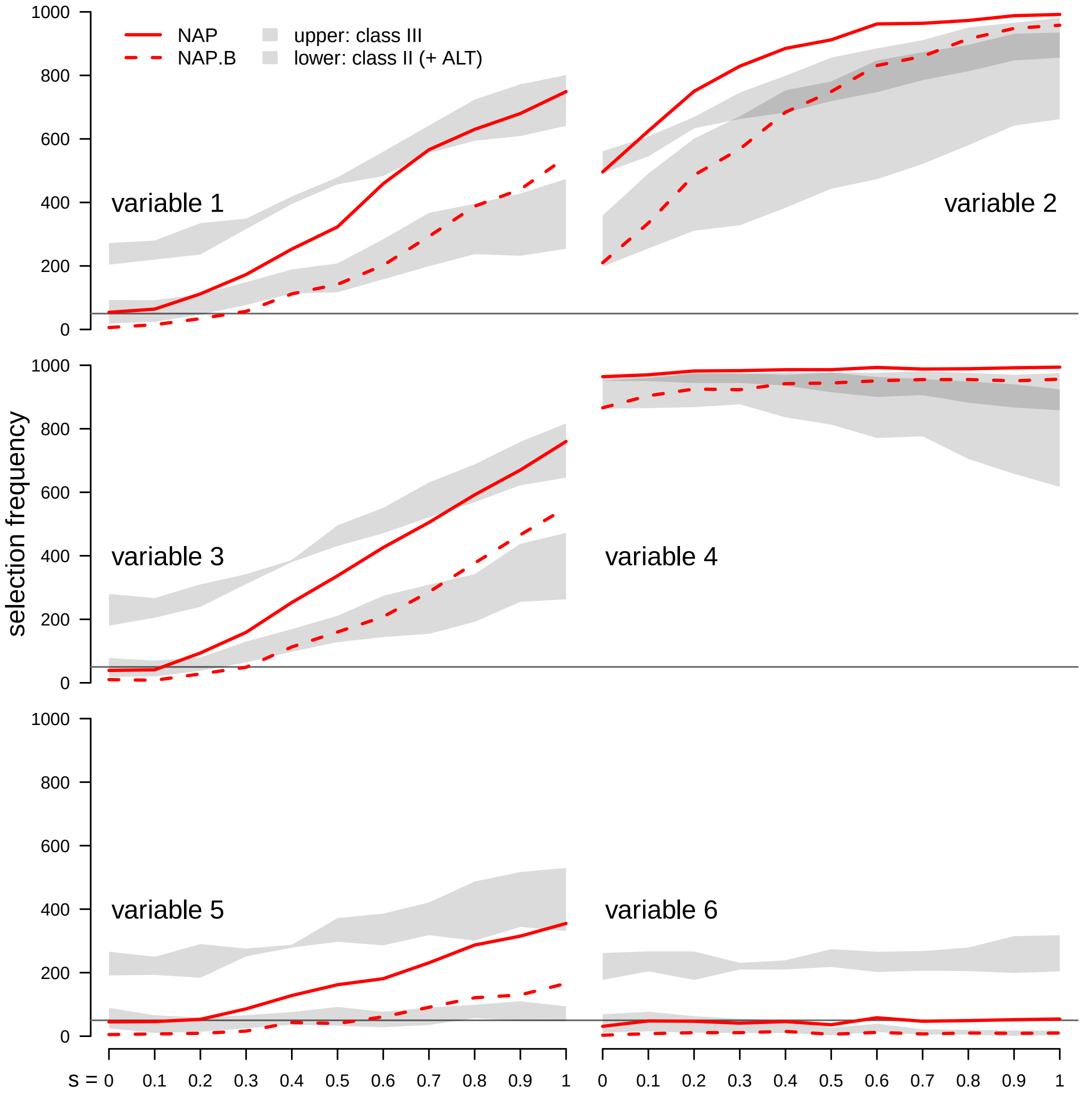


Figure 5.2: Plot of selection frequencies in 1000 simulation runs of the classification problem of study II. Results are shown in dependence of predictor strength as determined by $\beta_{II} = (s, s, s, 1, 0, 0)^\top$. There are no relations except for pairwise correlations ($r = 0.7$) between variable 2 and 4 as well as 3 and 5. Class III and II (+ ALT) are represented by gray areas which range from the lowest to highest values observed for the corresponding approaches. The horizontal line represents a 5% TWER. [Relevant Variables: var. 4 & var. 1, 2, 3, 5 for $s > 0$; Informative Variables: var. 4 & var. 1, 2, 3 for $s > 0$]

Results for the regression problem show a clear superiority of the new approaches which is even more demonstrative than in the classification problem (cf. Figure A.7 in the appendix A.4.1). NAP and NAP.B once again outperform their competitors in terms of discriminatory power.

Study III

Figure 5.3 displays results for variables 1–15 of study III (the non-relevant variables 16–20 were omitted from the illustration for redundancy). The presentation is again grouped by class and a more detailed discussion of each approach is given in appendix A.4.1. As a general finding, it is interesting to note that variables of correlated blocks (var. 4–11) show higher selection frequencies than uncorrelated variables (var. 1–3), even though the former might be non-informative (var. 10–11). This is due to the sensitivity of unconditional importance measures to relations between variables (cf. Strobl et al., 2008).

An investigation of selection frequencies for variables 12-15 makes clear that approaches of class III again select non-relevant variables far too often – confirming the results of studies I and II. Despite this increased TWER they are not able to outperform NAP and in most cases even the opposite is true when it comes to the identification of the relevant variables 1–11. Even NAP.B, which by definition is much more conservative, is able to keep up with these approaches in some cases. All methods of class II (+ ALT) produce a TWER of about 5% and therefore also agree with the findings of Study II. A comparison reveals a clear superiority of NAP and NAP.B in terms of the ability to identify relevant variables.

It is also interesting to note that a precise interpretation of equation (5.1) suggests that even a relation between non-informative variables could theoretically provide evidence against H_0. However, this is not a grave issue as in practice the computation of importance measures underlying the hypothesis tests is only affected by relations of variables to the response, whether they are direct or achieved via correlation. Therefore, NAP and NAP.B did not select the correlated and non-informative variables 12 and 13. This is a desirable property as it perfectly matches the definition of non-relevance (cf. section 5.1) which demands that a relevant variable is either informative itself or correlated to an informative one.

Once again, results demonstrate that the new approaches show a high power to distinguish relevant from non-relevant variables among approaches which either tend to select any kind of variable (class III), not matter their relevance, or are too weak to detect relevant variables (class II).

Boxplots of the observed MSE values show that among all variable selection approaches, NAP performs best while NAP.B is in the sixth position. However, it has to be pointed out that all results up to and including NAP.B are on a comparable level. Results also underline the well known fact that the prediction accuracy of Random Forests does not necessarily benefit from variable selection: the Random Forest which was build without any selection performs best.

In a perfect scenario all of the selected variables are relevant while the non-relevant ones are rejected. An additional illustration of mean selection frequencies investigates this property. The second best performing approach SVT selected 10.7 variables on average. Among those, 9 variables were of relevance. This makes a difference of 1.7 (16%) selected but non-relevant variables. Results for D.0 and J.0 are even worse. NAP only selects 9.2 variables but as much as 8.8 relevant ones. This makes a difference of 0.4 (4%) while the absolute amount of selected relevant variables is almost as high as for SVT (8.8 vs. 9). Thus for the selection of an equal number of relevant variables NAP produces less false positive detections. These values are even lower for NAP.B and ALT (0.1 and 0; 2% and 0%), although the absolute amount of detected relevant variables is lower (6.5 and 6.7). In conclusion NAP and NAP.B show a high efficiency as the fraction of relevant variables among a set of selected variables is high. NAP is also more effective as the absolute amount of detected relevant variables is high, too. Meanwhile they are able to produce a comparable MSE while other approaches are less efficient and less effective.

In the regression problem the new approaches NAP and NAP.B outperform their competitors even more clearly (see Figure A.9 in appendix A.4.1). They show the highest power to detect relevant variables, produce the lowest median MSE (even lower than for a Random Forest without any variable selection) and show effectivenesses and efficiencies that range among the highest.

5.4.2 Empirical Evaluation

A summary of results found for the empirical evaluation is given by Table 5.4. It reveals that the new variable selection approaches, compared to eight competitors, are consistently ranked among the best performing ones. In some instances they are even able to beat the performance of a Random Forest that uses no variable selection. It is also remarkable that methods of class II and III show alternating performance rankings while they can clearly be differentiated from class I which achieves better results throughout.

The number of selected variables for each classification and regression problem differs between methods. A comparison of performances makes clear that an increased or decreased set of variables is not necessarily associated with prediction accuracy. Likewise, it is not possible to rate the quality or correctness of selections as the true relevance of variables is unknown for real data (section 5.4.1 for similar evaluations). However, the number of times a variable is selected across all bootstrap runs may at least be taken as an indicator for its importance and the stability of its selection (cf. Table A.4 in appendix A.4.2). Considering the achieved performances, NAP and NAP.B seem to be well suited to select variables which are of predictive relevance.

5.5 Discussion and Conclusion

An extensive review of literature about variable selection using Random Forests led to the proposal of a new approach. It was basically invented within a permutation test framework to meet important theoretical properties. In addition, three simulation studies showed further appealing properties: Firstly, the new approach makes it possible to control

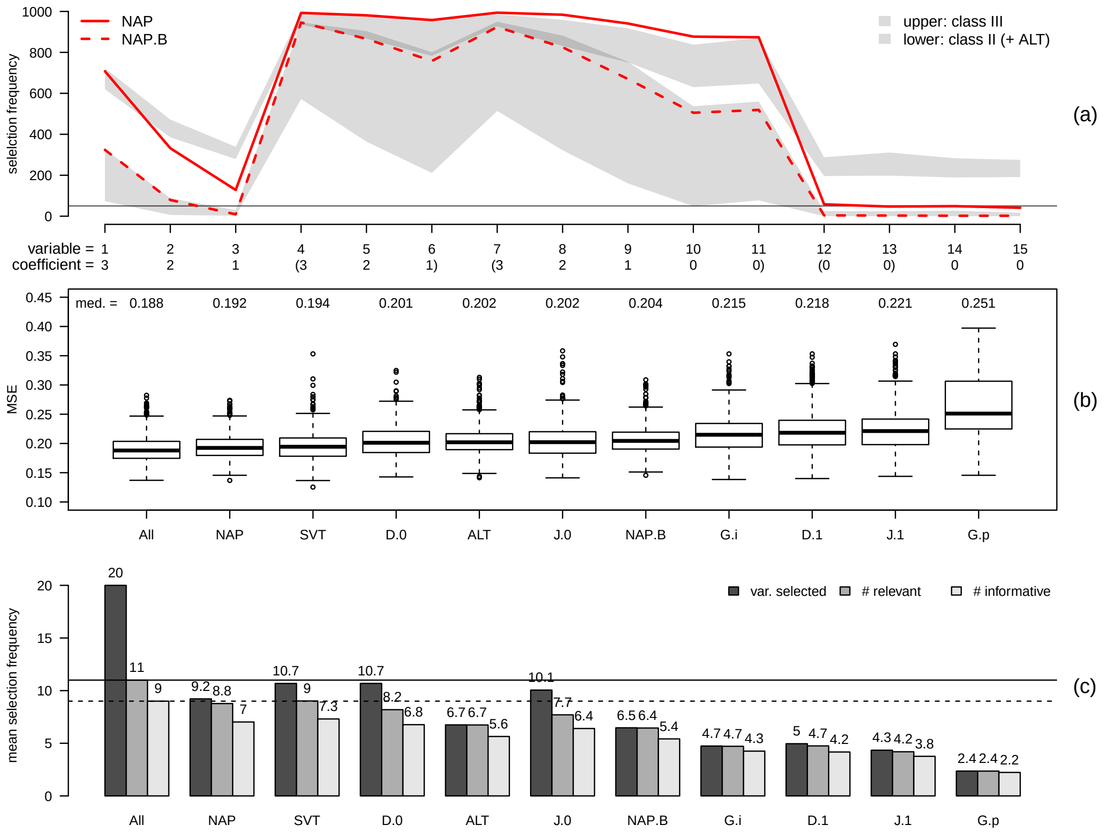


Figure 5.3: (a) Selection frequencies of variables 1–15 in 1000 simulation runs of the classification problem of study III. Class III and II (+ ALT) are represented by gray areas which range from the lowest to highest values observed for the corresponding approaches. The horizontal line represents a 5% TWER. Brackets indicate correlated variables. (b) Boxplots of the observed MSE aranged in increasing order of median values. (c) Mean selection frequency of variables and an additional information about the amount of relevant and informative variables among them. Reference values are indicated by horizontal lines. [Relevant Variables: var. 1 – 11; Informative Variables: var. 1 – 9]

Rank	1	2	3	4	5	6	7	8	9	10	11
Birthweight	**NAP.B**	**NAP**	ALT	G.p	G.i	SVT	All	D.0	J.0		
class	I	I	I	II	II	III		III	III		
$\widehat{MSE}^{(1)}$	0.208	0.213	0.209	0.209	0.209	0.210	0.227	0.315	0.309		
var. selected	5.80	7.25	4.99	1.00	1.22	1.09	9.00	5.18	5.01		
$\widehat{mse}$	0.166	0.169	0.192	0.203	0.203	0.203	0.191	0.272	0.319		
var. selected	5	5	3	1	1	1	9	5	5		
$\widehat{MSE}^{(.632)}$	0.193	0.197	0.203	0.207	0.207	0.207	0.214	0.299	0.312		
B. Housing	ALT	**NAP.B**	**NAP**	All	SVT	G.i	G.p	D.0	J.0		
class	I	I	I		III	II	II	III	III		
$\widehat{MSE}^{(1)}$	16.31	16.61	16.70	16.79	20.22	20.05	20.10	27.30	26.71		
var. selected	9.39	12.56	12.83	13.00	2.78	2.63	2.09	9.01	8.81		
$\widehat{mse}$	10.78	10.33	10.91	11.63	11.86	14.94	14.95	15.79	18.15		
var. selected	7	13	13	13	3	2	2	7	10		
$\widehat{MSE}^{(.632)}$	14.27	14.30	14.57	14.89	17.14	18.17	18.20	23.07	23.56		
Heart	All	**NAP**	**NAP.B**	ALT	SVT	D.0	G.i	J.1	D.1	J.0	G.p
class		I	I	I	III	III	II	II	II	III	II
$\widehat{MSE}^{(1)}$	0.173	0.175	0.177	0.180	0.175	0.177	0.181	0.181	0.182	0.177	0.248
var. selected	13.00	11.44	9.94	7.54	10.42	9.79	5.19	5.78	5.99	9.53	1.99
$\widehat{mse}$	0.100	0.118	0.122	0.126	0.137	0.137	0.137	0.137	0.137	0.170	0.237
var. selected	13	10	8	7	11	3	3	3	3	4	1
$\widehat{MSE}^{(.632)}$	0.146	0.154	0.157	0.160	0.161	0.162	0.165	0.165	0.165	0.174	0.244
Parkinson	**NAP**	All	**NAP.B**	ALT	J.0	D.0	SVT	J.1	G.i	D.1	G.p
class	I		I	I	III	III	III	II	II	II	II
$\widehat{MSE}^{(1)}$	0.152	0.153	0.152	0.152	0.150	0.151	0.155	0.152	0.153	0.155	0.162
var. selected	19.16	22.00	13.54	9.83	11.27	12.74	11.26	4.92	5.53	5.77	1.94
$\widehat{mse}$	0.092	0.097	0.103	0.108	0.113	0.113	0.108	0.133	0.133	0.133	0.133
var. selected	14	22	10	5	19	14	14	1	1	1	1
$\widehat{MSE}^{(.632)}$	0.130	0.133	0.134	0.136	0.136	0.137	0.138	0.145	0.146	0.147	0.152

Table 5.4: Bootstrap-, resubstitution- and .632 estimators for the MSE of the investigated methods arranged in order of decreasing performance (assessed by $\widehat{MSE}^{(.632)}$) for each dataset. In addition, the average number of selected variables in the bootstrap runs and for a single fit on the entire dataset are given.

the TWER and FWER. Secondly, it showed a higher power to distinguish relevant from non-relevant variables compared to common approaches. This finding was also confirmed in a simulated data application. Thirdly, it achieved the highest ratio of relevant to selected variables. Corresponding Random Forests produced MSE values which were comparable to the best performing models. Within an application to four datasets the two versions of the new approach always ranked among the best three (out of eleven) performing approaches in terms of MSE. Moreover, it is equally applicable to regression and classification problems.

Despite the clear superiority of the new approach which was observed in this work, the benefit of its application, in terms of prediction accuracy and especially for the selection of relevant variables, has to be further investigated. Additional simulation studies and empirical evaluations, possibly emerging from research fields in which these approaches are commonly used (e.g. microarray analysis, GWAS), are needed for further insight. Investigations should be intensified to further explore the effects of correlation strength, block size, interactions, type of importance measure, definition of ‘relevance’ and kind of alpha adjustment. Analyses also need to be extended to high-dimensional data.

Chapter 6

Variable Selection with Missing Data

6.1 Research Motivation and Contribution

Variable selection has been suggested for Random Forests to enhance data prediction and interpretation (cf. chapter 5). However, its basic element, i.e. variable importance measures, can not be computed in a straightforward manner when there are missing data (chapter 3). Possible solutions that still enable variable selection despite the occurrence of missing values are

- complete case analysis,
- multiple imputation and
- the new importance measure.

These have been used in combination with two variable selection methods:

- NAP and
- D.1,

representing the conceptual classes of test-based and performance-based approaches, respectively (section 5.3.4). An extensive simulation study that involves various missing data generating processes is conducted to explore their ability to discriminate relevant from non-relevant variables. In addition, the predictive accuracy of resulting models is investigated for a simulated test dataset (see chapter 2 and 4 for similar studies). Both regression and classification problems are explored. Findings and recommendations: Complete case analysis should not be applied as it leads 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. Its error was only negligibly worse compared to imputation.

6.2 Simulation Studies

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 of supporting variable selection. Therefore the NAP and D.1 approaches, as representatives of test-based and performance-based methods, have been chosen to investigate their quality to distinguish relevant from non-relevant variables. Two additional investigations will focus on 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. The setup of the simulation study resembles the ones given in chapter 3 and 4. Essential differences are presented 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.

- *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 the covariance matrix 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 in variables X_2 and X_5 (cf. Table 6.1). Therefore, the average percentage of observations that contain at least one missing value is $1 - (1 - \%_{\text{missing}})^{\text{n}_{\text{variables}}} =$

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

Table 6.1: List of variables that contain missing values and determine the probability of missing values.

$1-(1-0.3)^2 = 51\%$ 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. The schemes to produce missing values were MCAR, MAR(rank), MAR(median), MAR(upper), MAR(margins) and MNAR(upper).

- *Validation*

  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.

- *Implementation*

  The implementation of the simulation almost equals the one of chapter 4 except for the fact that Random Forests contained more trees: $ntree = 100$. Corresponding R-Code is given in section B.6.

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 288 simulation settings. Each of them was repeated 1000 times.

6.3 Results

The following discussion presents results for the classification problem. Similar findings for the regression problem are given as supplementary material in section A.5 (cf. Figure A.10)

Variable selection frequencies displayed in Figure 6.1 stress that the test-based approach performs better than 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 expected, the selection frequencies of variables 2 and 5 drop as they contain a rising amount of missing values. Meanwhile variables 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 chapter 3). The same effect can not be observed for the test-based

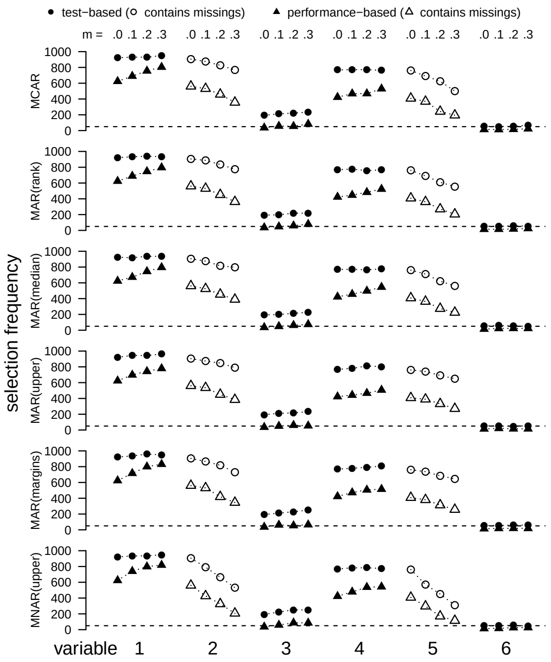


Figure 6.1: Variable selection frequencies observed for the new importance measure. The horizontal dashed line illustrates a TWER of 5% (m = % of missing values in X_2 and X_5).

approach which already 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 of its missing values.

In the complete case analysis (illustrated by Figure 6.2) 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 of detecting variables of true relevance.

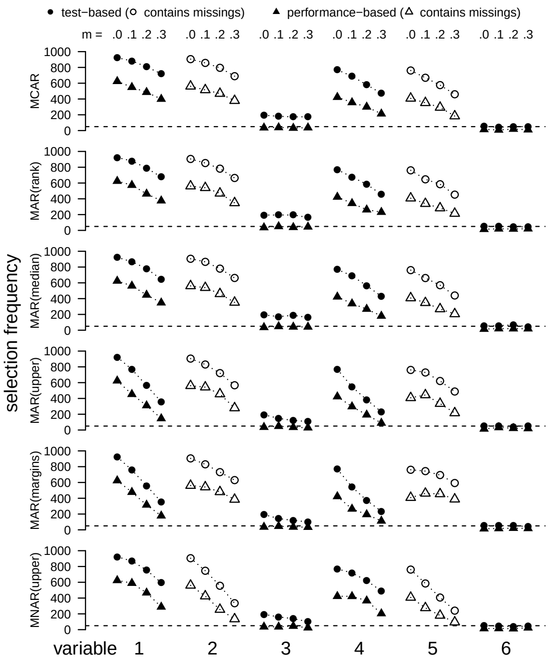


Figure 6.2: Variable selection frequencies observed for the complete case analysis. The horizontal dashed line illustrates a TWER of 5% ($m = \%$ of missing values in X_2 and X_5).

Results for the application of multiple imputation are given by Figure 6.3. 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 lose 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 6.1. 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 of selecting variables according to the relevance they would have if the data were fully observed.

Prediction errors observed for the independent test sample are displayed by Figure 6.4. 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 is based on the new importance measure produced models that performed only slightly worse (see chapter 2 and 4 for corresponding findings). 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

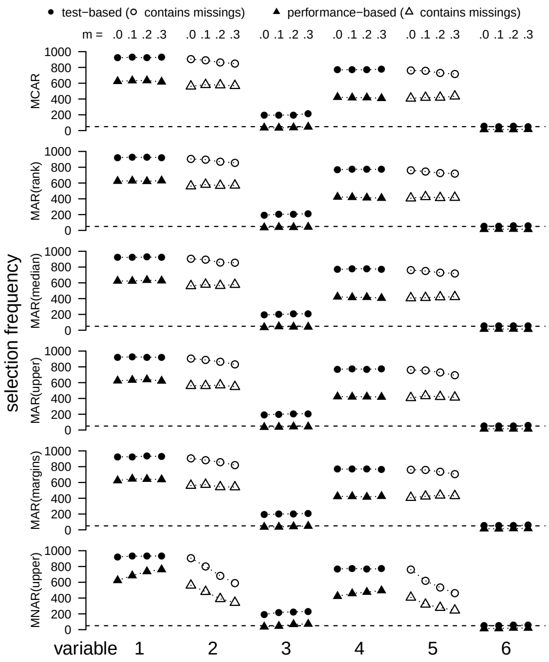


Figure 6.3: Variable selection frequencies observed for the imputed data. The horizontal dashed line illustrates a TWER of 5% (m = % of missing values in X_2 and X_5).

Random Forests when there are missing data have been published by Rieger et al. (2010) and Hapfelmeier et al. (2011).

6.4 Conclusion

Variable selection with Random Forests is guided by importance measures which are used to rate a variable's relevance for prediction. There are several approaches like a new kind of importance measure, complete case analysis and multiple imputation, that enable its application when the data contain 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 led to models that showed the worst prediction accuracies. Selection methods 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 consistently and considerably more often than those with missing values. The prediction accuracy of the corresponding Random Forests was much higher than for the complete case analysis. Multiple imputation also showed consistent 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

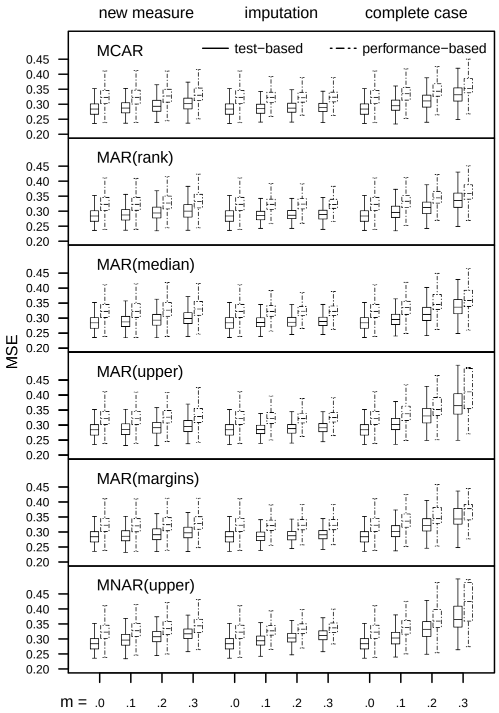


Figure 6.4: MSE observed for the independent test sample. Outliers are not displayed for clarity ($m = \%$ of missing values in X_2 and X_5).

any simulation setting and any approach to handle missing values, the test-based variable selection method performed better than 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.

Outlook

The assessment of predictive accuracy has been performed for the newly introduced variable importance, complete case analysis and multiple imputation in chapter 2, 4 and 6. However, investigations need to be expanded:

- A concluding discussion in section 2.3.2 already presents some recent and ongoing developments of imputation methods. This diversity of approaches should be taken into account for future research.

- Despite elaborate simulation settings and extensive empirical evaluations the findings need to prove generalisability in further investigations.

A new variable importance measure for Random Forests with missing data is proposed in chapter 3. Although it meets all postulated requirements and shows some appealing characteristics the corresponding discussions have shown that there is still room for further developments and investigations:

- A conditional variable importance measure that can handle missing data is to be developed. Strobl et al. (2008) introduced a conditional version of the permutation accuracy importance measure that more closely resembles the behavior of partial correlation or regression coefficients. Yet, non-conditional importance measures, to which the one proposed in chapter 3 belongs to, are often appreciated for their sensitivity to correlations and hence for their ability to uncover relations between variables (see chapter 5 for further discussions of this matter). However, in some research fields a conditional assessment might be preferred which raises the task of enhancing the current method to a measure that, besides its property to handle missing values, is conditional.

Chapter 5 presents an extensive investigation of a new variable selection procedure for Random Forests. Despite its appealing properties and power to distinguish relevant from non-relevant variables, future research and further developments are proposed to enhance its applicability:

- As the permutation steps of the proposed test-based variable selection algorithm are independent from each other they can be parallelized on multiple cores of a system to significantly reduce computation time (see Altmann et al., 2010, for an earlier suggestion of this procedure). In future work this benefit can be checked for and realized by an according implementation.

- In section 5.3.4 it has been pointed out that the performance-based approaches have originally been suggested for classification problems only. Thus, corresponding definitions of the 1 s.e. rule are not applicable to regression problems. However, Breiman et al. (1984) already suggested an adaption of the 1 s.e. rule which will be implemented for further performance evaluations in regression analysis.

- The application of variable selection with Random Forests is particularly popular in the field of microarray data analysis (Díaz-Uriarte and Alvarez de Andrés, 2006; Jiang et al., 2004; Rodenburg et al., 2008; Zhou et al., 2010). Many of the presented methods were originally designed for this special data case which is characterized by a low ratio of observations to predictors. Accordingly, the properties and competitiveness of the new approach should be reassessed in this special research field.

- There are several aspects like correlation strength, block size, interactions, type of importance measure, definition of 'relevance' and kind of alpha adjustment that should be the object of further, intensified investigations as to examine their effect on the performance of variable selection approaches.

- Despite a possible parallelization of the new algorithm, the computation time is expected to be exceptionally high in the case of thousands or tens of thousands of predictors. For the analysis of microarray data, this is a well known problem faced by many methods. It becomes even more challenging as more and more genetic information is made available with ongoing research, e.g. in GWAS. A proposed solution is to reduce the information transfered to the new approach by a preliminary selection step. Therefore a suggestion of Strobl et al. (2009) is followed: all variables with an importance which lies within the random variability of non-relevant variables (determined by the range of negative importance measures) are rejected. This way the amount of variables passed to the new approach is supposed to be significantly reduced. Especially in the case of microarray data, which can contain lots of redundant information, the computational time is expected to decrease substantially due to this very fast and easy preliminary selection step. Corresponding experience has already been published in Yahya et al. (2011).

- The new variable selection approaches proposed in this work are based on an unconditional importance measure (Strobl et al., 2008, for a definition of conditional and unconditional measures). For this reason non-informative variables which are related to informative ones are found to be of relevance and possibly selected by the approaches. However, this might also be seen as an undesired property. Future research is meant to explore variable selection that is based on a conditional importance measure. In this manner, the question of whether the selection can be restricted to informative variables is meant to be answered, too.

Appendix A

Supplementary Material

A.1 Chapter 2

A.1.1 Simulation Studies

Table A.1 contains an extensive listing of results for the simulation studies presented in section 2.5.

Table A.1: Summary of mean MSE values, mean absolute (abs. imp. $= \text{MSE}_{\text{Sur.}} - \text{MSE}_{\text{MICE}}$) and mean relative improvement (rel. imp. $= \frac{\text{MSE}_{\text{Sur.}} - \text{MSE}_{\text{MICE}}}{\text{MSE}_{\text{Sur.}}}$) obtained by using multiple imputation and surrogates. Missing values were induced completely at random into data that was originally fully observed. Two imputation schemes are distinguished. For one of them all variables and for another one only one third of variables is partly set missing. Please note that the mean relative improvement is given by the mean of improvements across simulation runs. It can not simply be computed by using the mean MSE values in the formula given here (as the mean of ratios does not equal the ratio of means).

			missing		Surrogates		MICE		abs. imp.		rel. imp.	
	Data	Type	# Var.	% Values	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Classification	H. Survival	forest		0%	0.27	0.05	0.27	0.05				
			3	10%	0.28	0.05	0.27	0.05	0.01	0.03	0.03	0.12
				20%	0.29	0.05	0.27	0.05	0.02	0.04	0.05	0.14
				30%	0.28	0.05	0.27	0.06	0.01	0.04	0.04	0.15
				40%	0.28	0.06	0.27	0.05	0.00	0.05	0.00	0.18
			1	10%	0.27	0.05	0.27	0.05	0.00	0.02	0.00	0.08
				20%	0.27	0.05	0.27	0.05	0.00	0.03	0.00	0.10
				30%	0.27	0.05	0.27	0.05	0.00	0.03	0.00	0.11
				40%	0.27	0.05	0.27	0.05	0.00	0.03	0.00	0.12
		ctree		0%	0.28	0.05	0.28	0.05				
			3	10%	0.28	0.06	0.28	0.05	0.00	0.04	-0.01	0.18
				20%	0.27	0.06	0.27	0.05	0.00	0.05	-0.02	0.18
				30%	0.27	0.06	0.27	0.06	0.00	0.05	-0.01	0.16
				40%	0.27	0.06	0.27	0.05	0.00	0.05	-0.02	0.19
			1	10%	0.28	0.05	0.28	0.05	0.00	0.03	-0.01	0.10
				20%	0.28	0.05	0.28	0.05	0.00	0.03	-0.01	0.10
				30%	0.27	0.05	0.28	0.05	0.00	0.03	-0.02	0.11
				40%	0.28	0.05	0.28	0.05	0.00	0.03	-0.01	0.11
		rpart		0%	0.28	0.05	0.28	0.05				
			3	10%	0.28	0.06	0.28	0.05	0.00	0.04	0.00	0.15
				20%	0.28	0.05	0.28	0.05	0.00	0.04	-0.01	0.16
				30%	0.28	0.05	0.28	0.06	0.00	0.04	-0.01	0.17
				40%	0.28	0.05	0.28	0.05	0.00	0.04	-0.02	0.17
			1	10%	0.28	0.05	0.28	0.05	0.00	0.03	0.00	0.12
				20%	0.28	0.05	0.28	0.05	0.00	0.04	0.00	0.13

				30%	0.28	0.05	0.28	0.05	0.00	0.04	-0.01	0.15
				40%	0.28	0.05	0.28	0.05	0.01	0.04	0.01	0.15
	Heart	forest		0%	0.17	0.05	0.17	0.05				
			12	10%	0.19	0.05	0.18	0.05	0.00	0.03	0.00	0.19
				20%	0.20	0.05	0.20	0.05	0.00	0.04	0.00	0.22
				30%	0.23	0.06	0.22	0.06	0.01	0.05	0.04	0.23
				40%	0.26	0.06	0.23	0.06	0.02	0.06	0.07	0.24
			4	10%	0.18	0.05	0.18	0.05	0.00	0.03	-0.02	0.17
				20%	0.18	0.05	0.18	0.05	0.00	0.03	-0.04	0.22
				30%	0.19	0.05	0.19	0.05	0.00	0.04	-0.03	0.24
				40%	0.19	0.05	0.19	0.05	0.00	0.04	-0.05	0.25
		ctree		0%	0.24	0.06	0.24	0.06				
			12	10%	0.27	0.06	0.24	0.06	0.03	0.06	0.07	0.23
				20%	0.30	0.06	0.25	0.06	0.05	0.07	0.15	0.22
				30%	0.33	0.07	0.26	0.06	0.07	0.07	0.19	0.22
				40%	0.35	0.07	0.27	0.06	0.09	0.08	0.22	0.22
			4	10%	0.25	0.06	0.25	0.06	0.00	0.04	0.00	0.19
				20%	0.25	0.06	0.25	0.06	0.00	0.05	0.00	0.23
				30%	0.25	0.06	0.25	0.06	0.01	0.05	-0.01	0.25
				40%	0.25	0.06	0.25	0.06	0.00	0.06	-0.02	0.25
		rpart		0%	0.22	0.06	0.22	0.06				
			12	10%	0.23	0.06	0.21	0.06	0.02	0.05	0.06	0.26
				20%	0.25	0.06	0.23	0.06	0.03	0.06	0.08	0.25
				30%	0.28	0.06	0.24	0.06	0.04	0.07	0.12	0.24
				40%	0.30	0.06	0.25	0.06	0.04	0.07	0.13	0.24
			4	10%	0.22	0.06	0.21	0.06	0.01	0.04	0.02	0.20
				20%	0.22	0.06	0.22	0.06	0.01	0.05	0.00	0.25
				30%	0.23	0.06	0.22	0.06	0.01	0.05	0.02	0.25
				40%	0.23	0.06	0.22	0.06	0.00	0.06	-0.01	0.27
Regression	Fertility	forest		0%	123.88	59.39	123.88	59.39				
			5	10%	128.72	60.77	125.91	60.10	2.81	16.36	0.01	0.15
				20%	128.74	62.13	124.19	60.00	4.55	22.13	0.01	0.20
				30%	152.36	74.58	122.91	61.76	29.44	30.91	0.17	0.23
				40%	160.17	80.29	129.33	69.28	30.84	34.39	0.16	0.24
			2	10%	122.58	58.97	122.49	58.95	0.09	10.96	0.00	0.11
				20%	123.29	64.43	123.25	62.17	0.03	16.67	-0.02	0.17
				30%	122.71	58.84	122.31	58.52	0.40	23.08	-0.02	0.24
				40%	130.65	65.59	129.37	63.44	1.28	27.08	-0.02	0.25
		ctree		0%	125.50	72.64	125.50	72.64				
			5	10%	143.59	71.67	118.73	66.97	24.85	48.13	0.12	0.41
				20%	151.47	73.84	116.20	63.06	35.27	52.43	0.18	0.37
				30%	151.47	72.91	116.20	61.87	35.27	49.96	0.19	0.34
				40%	163.68	82.91	125.79	72.49	37.89	52.84	0.18	0.35
			2	10%	127.61	69.47	121.11	69.15	6.49	32.18	0.01	0.38
				20%	129.85	70.76	121.92	70.21	7.93	34.76	0.02	0.35
				30%	126.32	65.43	119.44	65.11	6.88	37.05	0.01	0.38
				40%	135.95	72.54	125.19	70.03	10.77	42.25	0.04	0.35
		rpart		0%	128.08	67.24	128.08	67.24				
			5	10%	131.89	67.47	117.00	64.73	14.89	47.04	0.06	0.37
				20%	129.09	67.36	113.72	64.43	15.37	50.17	0.05	0.45
				30%	134.35	70.86	113.24	60.85	21.11	50.21	0.09	0.41
				40%	143.20	82.27	120.70	71.32	22.50	51.87	0.06	0.43
			2	10%	127.62	64.68	120.97	67.30	6.65	35.78	0.01	0.54
				20%	129.00	66.68	119.53	64.42	9.46	40.08	0.02	0.45
				30%	120.61	60.19	114.85	60.16	5.76	39.03	-0.01	0.44
				40%	131.61	65.26	124.85	66.33	6.76	42.09	0.00	0.45
	Birthweight	forest		0%	457232	95500	457232	95500				
			8	10%	480114	98110	468505	96912	11610	22990	0.02	0.05
				20%	496568	99652	479233	96640	17335	31472	0.03	0.07
				30%	513674	106290	500251	103400	13423	38018	0.02	0.08
				40%	523973	106809	511407	102216	12565	39898	0.02	0.08
			3	10%	461966	97094	460541	97183	1426	12439	0.00	0.03
				20%	468606	97316	466447	95369	2160	20850	0.00	0.05
				30%	477325	100949	473842	99294	3483	24978	0.01	0.05
				40%	482337	98226	481618	97479	719	29519	0.00	0.06
		ctree		0%	521135	102026	521135	102026				
			8	10%	541681	106881	510502	101726	31179	44007	0.05	0.08
				20%	550504	108030	511162	103410	39343	54991	0.07	0.10
				30%	557568	113516	526071	107761	31497	55300	0.05	0.10
				40%	558431	112739	530738	106455	27692	58479	0.04	0.10
			3	10%	524473	105341	514840	104686	9634	32771	0.02	0.06
				20%	528372	104142	516334	102509	12038	38736	0.02	0.07
				30%	531124	104639	516274	102125	14850	43622	0.02	0.08
				40%	535775	105253	523234	103125	12541	49098	0.02	0.09
		rpart		0%	530248	117622	530248	117622				
			8	10%	537710	117599	507510	106833	30200	67410	0.05	0.13
				20%	545339	119787	514642	104013	30697	80091	0.04	0.15
				30%	558719	124351	530830	111282	27889	84234	0.04	0.15
				40%	564695	122945	536551	107390	28144	86001	0.03	0.15
			3	10%	529680	115061	511419	108543	18261	54492	0.03	0.10
				20%	533993	120503	512754	109923	21239	66603	0.03	0.12
				30%	544904	123811	516096	112904	28808	72635	0.04	0.13
				40%	545331	120782	518021	110371	27310	76905	0.04	0.15

A.1.2 Empirical Evaluation

Table A.2 gives an extensive listing of results for the application studies of section 2.5.

			Surrogates		MICE		abs. imp.		rel. imp.	
	Data	Type	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Classi-fication	Hepatitis	forest	0.18	0.07	0.18	0.06	0.00	0.02	-0.01	0.20
		ctree	0.22	0.07	0.22	0.07	0.00	0.06	-0.04	0.35
		rpart	0.22	0.07	0.21	0.07	0.01	0.06	0.01	0.30
	Mammo	forest	0.21	0.03	0.20	0.03	0.01	0.01	0.04	0.06
		ctree	0.22	0.03	0.21	0.03	0.01	0.02	0.03	0.07
		rpart	0.21	0.03	0.21	0.03	0.00	0.01	0.00	0.07
	Pima	forest	0.24	0.03	0.24	0.03	0	0.01	0.00	0.05
		ctree	0.26	0.03	0.25	0.03	0	0.02	0.01	0.06
		rpart	0.25	0.03	0.25	0.03	0	0.02	0.00	0.10
	Ozone	forest	0.06	0.01	0.09	0.03	-0.02	0.03	-0.39	0.49
		ctree	0.07	0.01	0.06	0.01	0.01	0.01	0.10	0.12
		rpart	0.07	0.01	0.06	0.01	0.00	0.01	0.00	0.12
Regres-sion	Airquality	forest	55.52	12.73	53.19	13.17	2.34	6.23	0.04	0.11
		ctree	65.65	15.42	55.37	13.57	10.28	9.98	0.15	0.14
		rpart	66.80	14.95	56.16	14.26	10.64	10.81	0.15	0.15
	El Nino	forest	0.45	0.13	0.61	0.27	-0.16	0.25	-0.41	0.58
		ctree	0.61	0.26	0.62	0.27	-0.01	0.33	-0.16	0.63
		rpart	1.35	0.36	0.81	0.27	0.54	0.40	0.36	0.25
	CHAIN	forest	136.43	14.11	135.24	14.27	1.19	2.10	0.01	0.02
		ctree	143.56	14.00	141.35	14.15	2.21	4.79	0.02	0.03
		rpart	146.55	16.76	141.00	15.20	5.55	8.18	0.04	0.05
	Sleep	forest	486943.26	1032618.19	485402.09	1032126.12	1541.17	7874.01	0.01	0.12
		ctree	517357.24	958954.63	517330.69	958864.23	26.55	1928.41	0.00	0.00
		rpart	548506.76	946491.05	554341.03	946373.04	-5834.27	39446.44	-0.65	7.06

Table A.2: Summary of mean MSE values, mean absolute (abs. imp. $= \text{MSE}_{\text{Sur.}} - \text{MSE}_{\text{MICE}}$) and mean relative improvement (rel. imp. $= \frac{\text{MSE}_{\text{Sur.}} - \text{MSE}_{\text{MICE}}}{\text{MSE}_{\text{Sur.}}}$) obtained by using multiple imputation and surrogates within 1000 MCCV runs for the data that originally includes missing values. Please note that the mean relative improvement is given by the mean of improvements across simulation runs. It can not simply be computed by using the mean MSE values in the formula given here (as the mean of ratios does not equal the ratio of means).

A.2 Chapter 3

Figure A.1 displays the importances of all variables that contain missing values (i.e., variables 2, 4, 8, 10, 12 and 14) for different fractions of missing values and correlation strength. Requirement (R2), claiming that the importance should not increase but decrease with an increasing amount of missing values, is met for all combinations of influential, non-influential, correlated and uncorrelated variables. The apparent difference of importance measures between variable 2 and variable 4 – though both of them have a coefficient of the same magnitude – is due to their correspondence to blocks of different size (cf. discussions about requirement (R3))

An investigation of Figure A.2 reveals that requirement (R3), that correlations with influential variables induce higher importance values, is also met. This claim holds for all combinations of influential and non-influential variables with different fractions of missing values (represented by blocks I, II and IV). However, the effect is less pronounced for variables 4 and 8. Also, for variables which do not contain missing values (e.g. variables 1, 3, 5, 9 and 11) the importance is rising with a rising number of missing values in

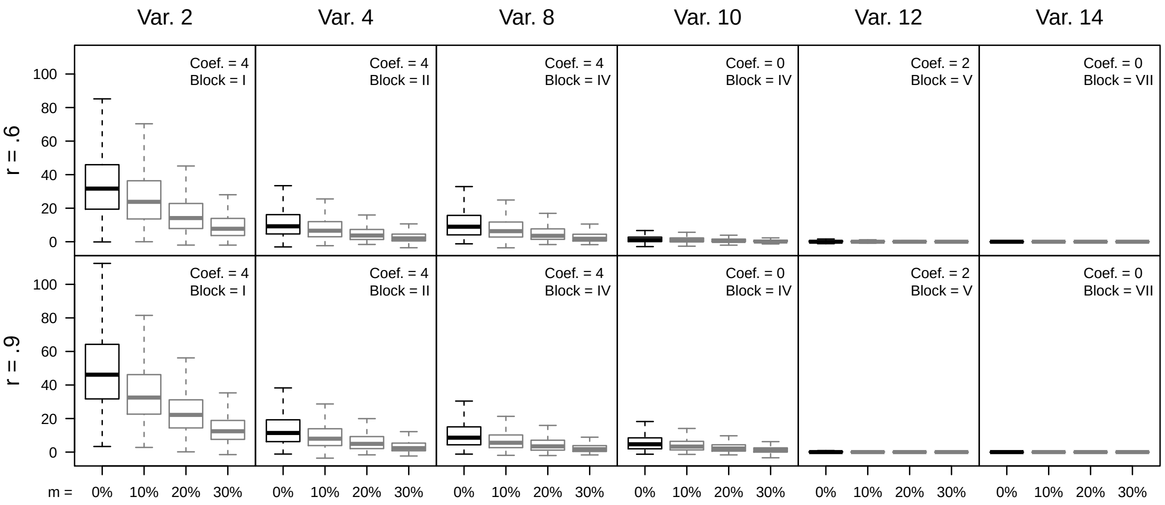


Figure A.1: Variable importance of variables 2, 4, 8, 10, 12, 14 – that contain missing values when m > 0% – for correlations r = .6, .9 and fractions of missing values m = 0%, 10%, 20%, 30%. Boxplots of variables that contain missing values are colored grey. Outliers are omitted from illustration for clarity.

other variables. Both effects occur because with a rising correlation and a rising number of missing values variables are replaced by others (cf. section 3.4.1 for corresponding discussions).

Block sizes – i.e., the number of variables that are correlated with each other – is another factor that strongly influences importance measures. A comparison of blocks I to II clearly shows that an increasing block size makes importances rise. However, this statement is only true for blocks containing influential variables. Non-influential variables do not induce this effect, as can be seen by the example of block IV compared to block II. This investigation also shows that the importance of non-influential variables benefits from the correlation to influential variables while the reverse is not true.

Figure A.3 presents results for the simulated classification problem and therefore corresponds to Figure 3.7 in section 3.4.1.

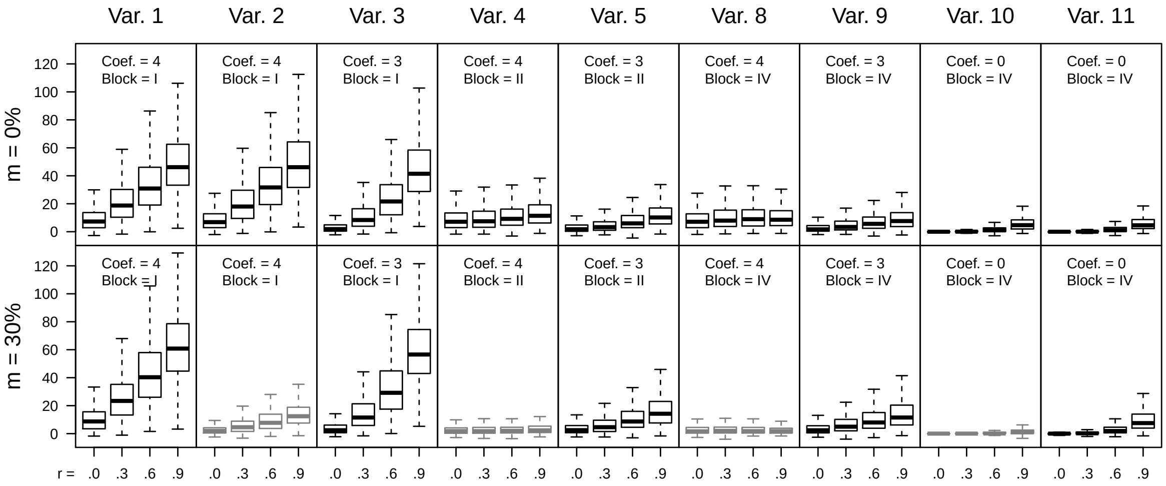


Figure A.2: Variable importances of variables 1–5, 8–11 (Blocks I, II, IV) and correlations r = 0, .3, .6, .9 for fractions of missing values m = 0%, 30%. Boxplots corresponding to variables with missing values are colored grey. Outliers were omited from illustration for clarity.

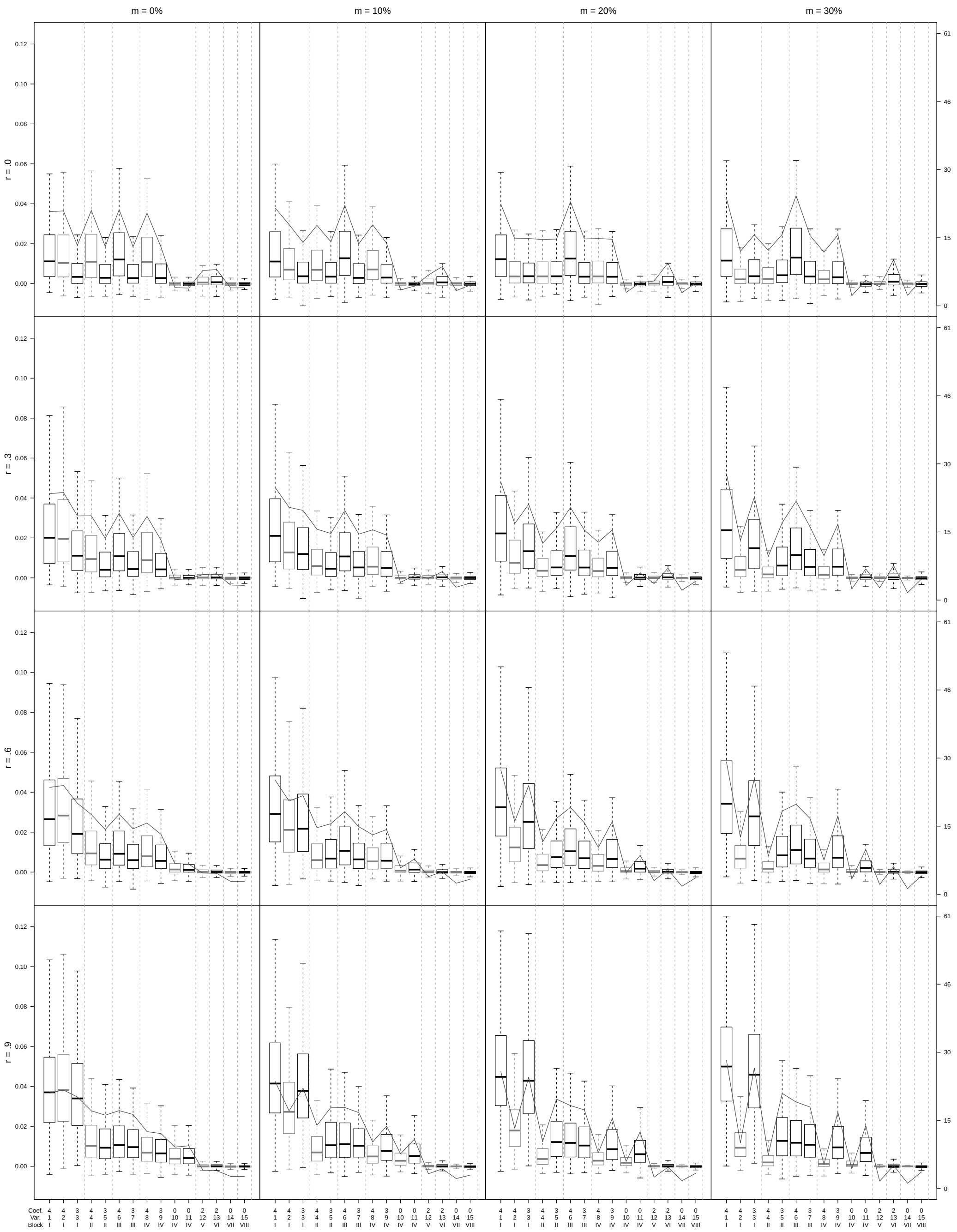
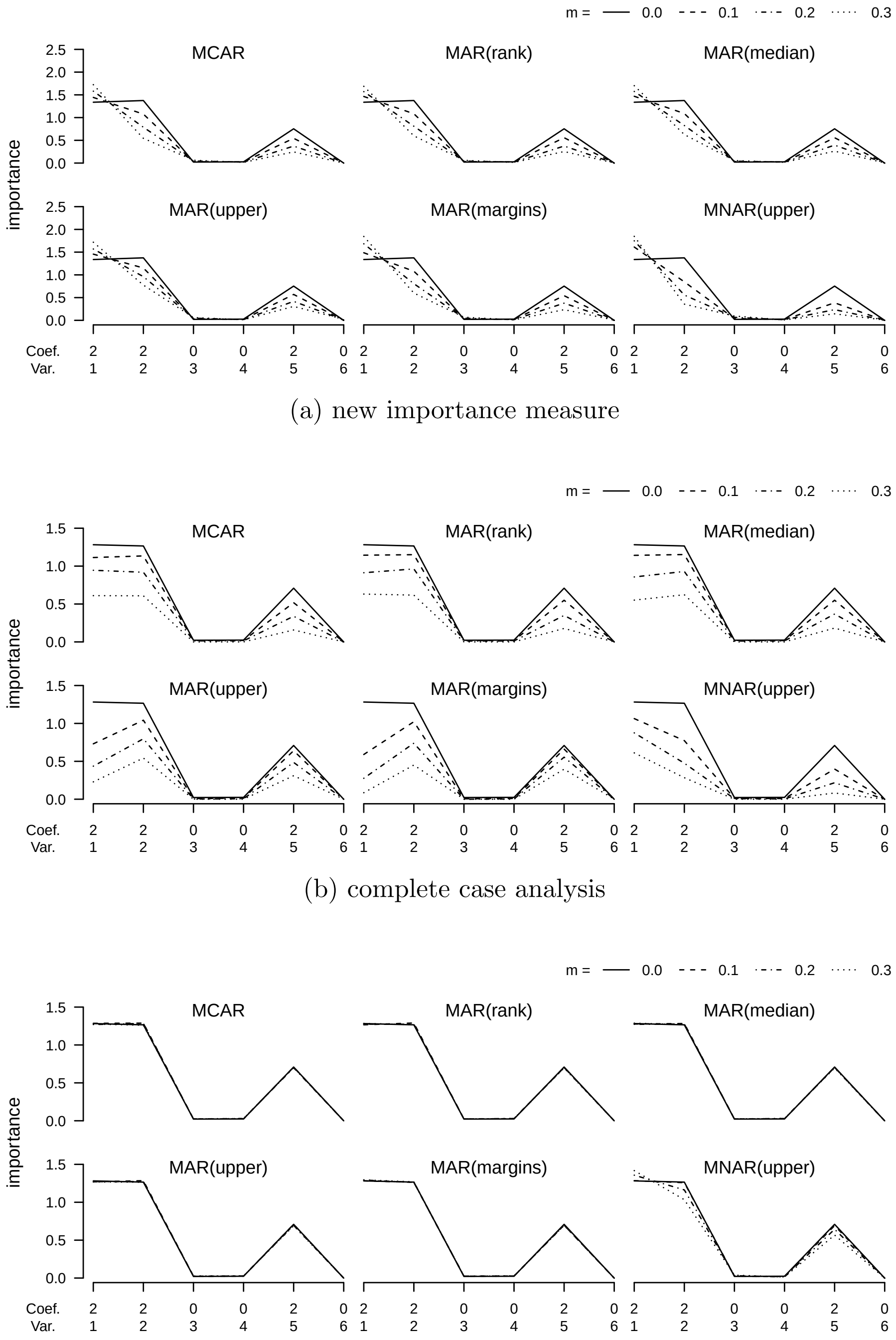


Figure A.3: Variable importances (left axis) of block I–VIII and correlations r = 0, .3, .6, .9 for fractions of missing values m = 0%, 10%, 20%, 30% in the MAR(rank) setting and classification problem. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

A.3 Chapter 4

Figure A.4 displays median importance measures observed for the regression problem.



(a) new importance measure

(b) complete case analysis

(c) multiple imputation

Figure A.4: Median variable importance observed for the regression problem (m = % of missing values in X_2, X_4 and X_5).

Figure A.5 displays the evaluation of prediction error for the regression problem.

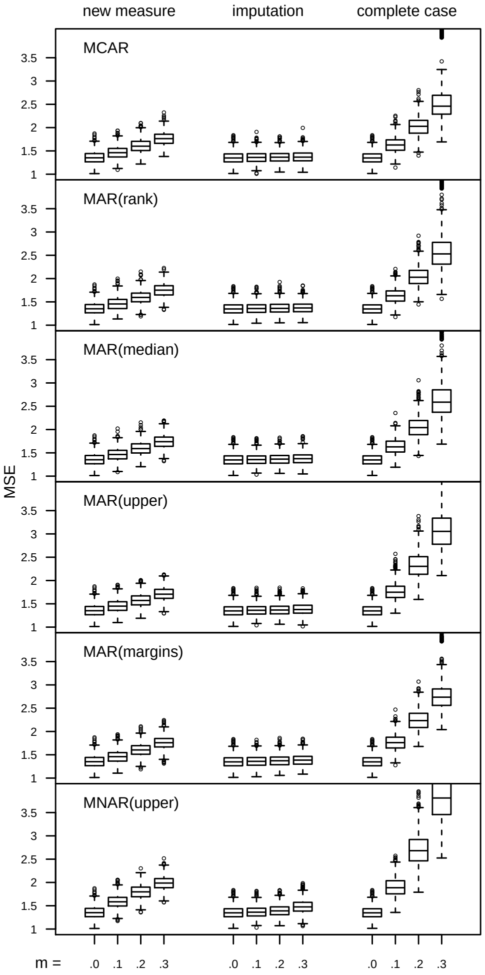


Figure A.5: MSE observed for the regression problem ($m = \%$ of missing values in X_2, X_4 and X_5).

A.4 Chapter 5

A.4.1 Simulation Studies

Study I

Results for the regression problem of Study I are given in Table A.3. The TWER and FWER are controlled by NAP, NAP.B and ALT the same way as in the classification problem (cf. section 5.4.1). The error rates of methods allocated to classes II and III are again too high – except for G.p. Differences between classes are less pronounced as in the classification problem.

	TWER					FWER	Class
	var. 1	var. 2	var. 3	var. 4	var. 5		
NAP.B	1.0%	0.8%	1.0%	1.1%	1.1%	4.8%	I
NAP	5.2%	4.7%	5.5%	5.4%	4.8%	22.6%	I
ALT	4.8%	5.1%	5.4%	5.2%	5.1%	23.4%	I
G.p	1.2%	1.5%	2.0%	1.4%	1.6%	6.2%	II
D.1	13.6%	13.6%	14.6%	13.7%	13.6%	27.9%	II
J.1	13.7%	13.9%	14.1%	14.4%	13.9%	29.3%	II
G.i	10.2%	9.9%	10.5%	10.7%	10.5%	25.0%	II
J.0	13.7%	13.9%	14.1%	14.4%	13.9%	29.3%	III
D.0	13.6%	13.6%	14.6%	13.7%	13.6%	27.9%	III
SVT	20.0%	19.9%	20.5%	20.1%	20.0%	25.3%	III

Table A.3: TWER and FWER for 5000 simulation runs of the regression problem in study I. Approaches are ranked and allocated to classes according to the results of the classification problem.

Study II

The following contains – in addition to the findings and conclusions already presented in section 5.4.1 – an extensive discussion of the performance of single approaches in Study II. For an improved clarity of presentation, Figure A.6 consists of two columns separating approaches which produce low to moderate TWER (class I and II) from those which showed an extremely increased TWER (class III). The new approaches are compared against each of them.

The left column of Figure A.6 displays results for J.0, D.0 and SVT. The horizontal gray lines indicate a selection frequency of 50 which is expected to be reached when the TWER of a non-relevant variable equals 5%. An examination of results for variable 6 makes clear that J.0, D.0 and SVT select this non-relevant variable far too often. The same holds for variable 1, 3 and 5 which are also of no relevance when $s = 0$. Consequently it is not surprising that these approaches – besides this tendency to select non-relevant variables – also show a high power to detect relevant ones. Thus, variables 1, 2 and 3 for $s > 0$ as well as variable 4 and variable 5 (which is correlated to the informative variable 3) are frequently selected by these approaches. SVT slightly outperforms J.0 and D.0 as it shows fewer false discoveries and a higher power as s increases. It is also interesting to note that the selection frequency of the relevant variable 4 even drops for J.0 and D.0 with a rising relevance of variables 1, 2 and 3 ($s > 0$). In summary the approaches seem to be capable to detect relevant variables with a high power. However, this is not a big achievement as it comes at the cost of a high TWER. More surprisingly the new approaches NAP and NAP.B show competitive and in some cases even superior results though they control the TWER – or even the FWER (which holds for NAP.B). The latter property is shown by selection frequencies of variable 6 as well as variables 1, 3 and 5 when $s = 0$. In terms of informative variables NAP is even able to outperform its competitors (c.f. variable 2 and 4) while it is not inferior in any other case. Even the by definition more conservative NAP.B shows competitive results when s increases.

The right column of Figure A.6 displays results for J.1, D.1, ALT, G.i and G.p. An investigation of selection frequencies of variable 6 as well as variables 1, 3 and 5 when $s = 0$ makes clear that the approaches – except for a slight violation by G.i – do not exceed the threshold TWER (gray line). In regard of the relevant variables 1, 2, 3, 4 and 5 ($s > 0$) the approaches NAP, NAP.B, ALT and G.i clearly perform best – though there is always one of the latter two that shows inferior results for some of the variables. NAP.B is only slightly inferior to NAP. It often outperforms or compares to ALT and G.i though it is expected to be more conservative (as it is the only approach that controls the FWER). NAP – and NAP.B for larger values of s – are the only approaches which are able to detect variable 5 which is of relevance as it is correlated to the informative variable 3. J.1, D.1 and G.p show inferior results throughout. For these approaches and G.i the selection frequency of the informative variable 4 even decreases with a rising strength of variables 1, 2 and 3 ($s > 0$); while it stays constant for approaches of class I (i.e. NAP, NAP.B and ALT).

NAP shows the highest power and clearly outperforms ALT – which is of major interest as both methods only slightly differ, yet in a substantial aspect of the permutation scheme. A discussion in section 5.2.3 already outlined that this is because the empirical distribution of importances of relevant variables has higher upper percentiles under H_0^* (5.2) – followed by ALT – than under H_0 (5.1) – followed by NAP. Therefore NAP is able to reject the null-hypothesis for relevant variables, that initially have an importance score of a certain magnitude, more often.

Results for the regression problem are similar, yet they stress the superiority of the new approaches even stronger (cf. Figure A.7). NAP and NAP.B clearly outperform their competitors as they show the highest discriminatory power in this case.

Study III

In relation to section 5.4.1 a much more detailed illustration of the results for each approach in study III is given by Figure A.8. A comparison to the approaches G.i, G.p, J.1, D.1 and ALT reveals a clear superiority of NAP and NAP.B in terms of the ability to identify relevant variables. In analogy to the findings of Study II the only approach which is able to keep up – now in a setting of a simulated data set – is ALT. However, it is outperformed by NAP – which it resembles the most (cf. section 5.2.3 for a discussion about the reasons behind this superiority of NAP) – and only compares to NAP.B though the latter is – by definition – much more conservative as it uses the Bonferroni-Adjustment. A comparison to J.0, D.0 and SVT reveals that these approaches again produce error rates far beyond the 5% level. Still they are not able to clearly outperform NAP and NAP.B, especially for higher values of s.

The superiority of the new approaches becomes even more evident in the regression problem (cf. Figure A.9): they show the highest power to detect relevant variables, the

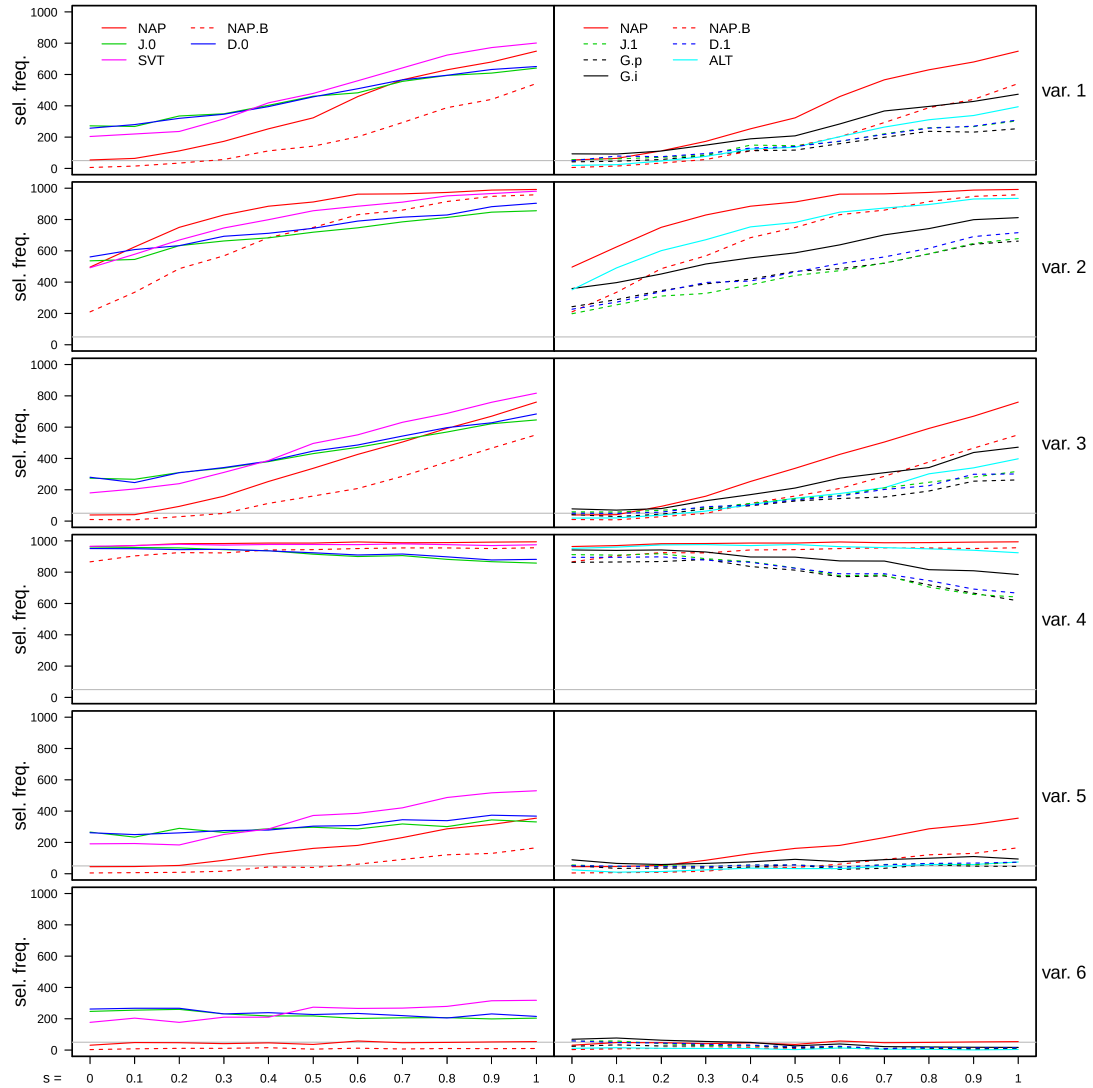


Figure A.6: Plot of selection frequencies in 1000 simulation runs of the classification problem of study II. Results are shown in dependence of predictor strength as determined by $\beta_{II} = (s, s, s, 1, 0, 0)^\top$. There are no relations except for pairwise correlations ($r = 0.7$) between variable 2 and 4 as well as 3 and 5. Comparisons of the new approaches to class III and II (+ ALT) are given in the left and right column, respectively. The horizontal line represents a 5% TWER. [Relevant Variables: var. 4 & var. 1 – 3, 5 for $s > 0$; Informative Variables: var. 4 & var. 1 – 3 for $s > 0$]

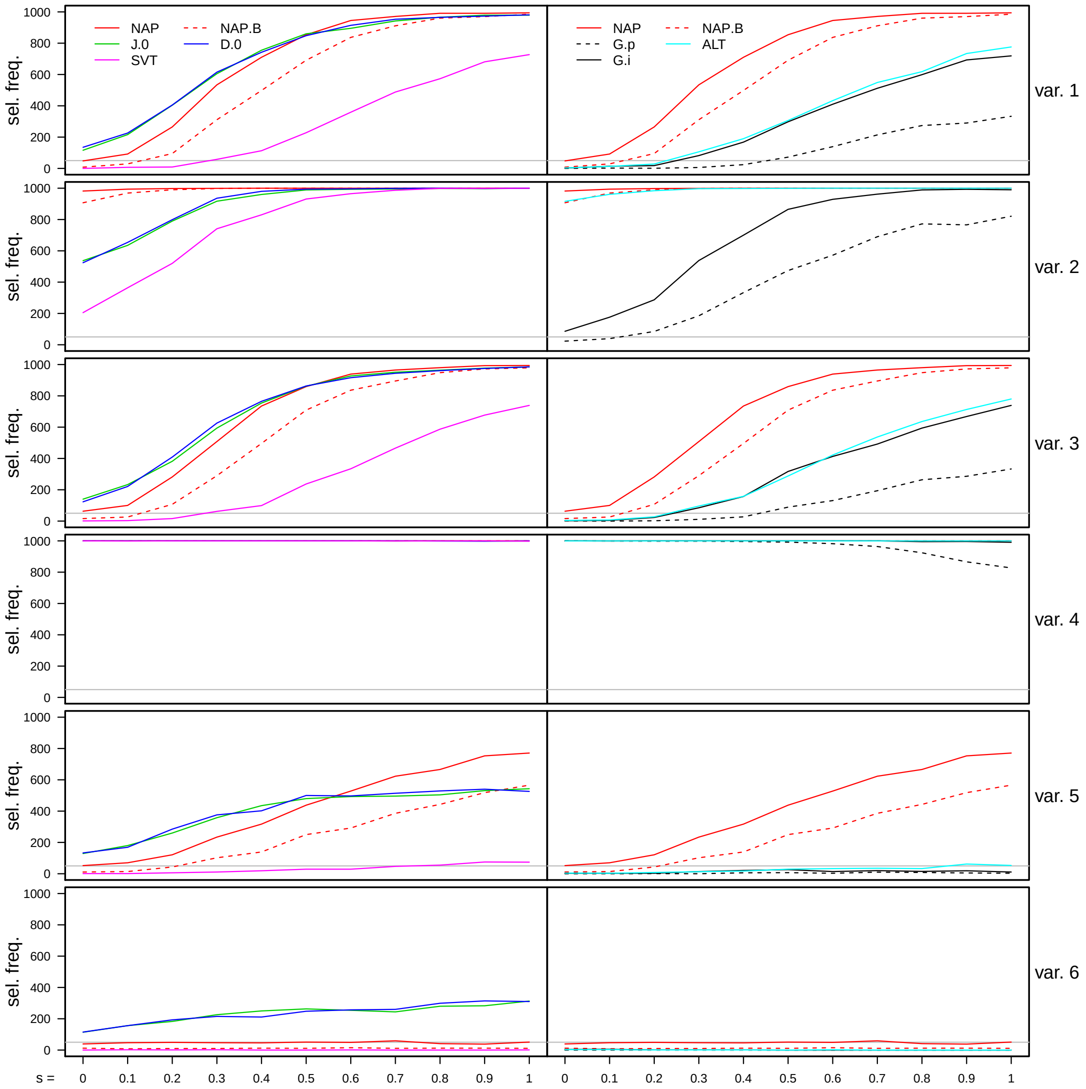


Figure A.7: Plot of selection frequencies in 1000 simulation runs of the regression problem of study II. Results are shown in dependence of predictor strength as determined by $\beta_{II} = (s, s, s, 1, 0, 0)^\top$. There are no relations except for pairwise correlations ($r = 0.7$) between variable 2 and 4 as well as 3 and 5. Comparisons of the new approaches to class III and II (+ ALT) are given in the left and right column, respectively. The horizontal line represents a 5% TWER. [Relevant Variables: var. 4 & var. $1-3$, 5 for $s > 0$; Informative Variables: var. 4 & var. $1-3$ for $s > 0$]

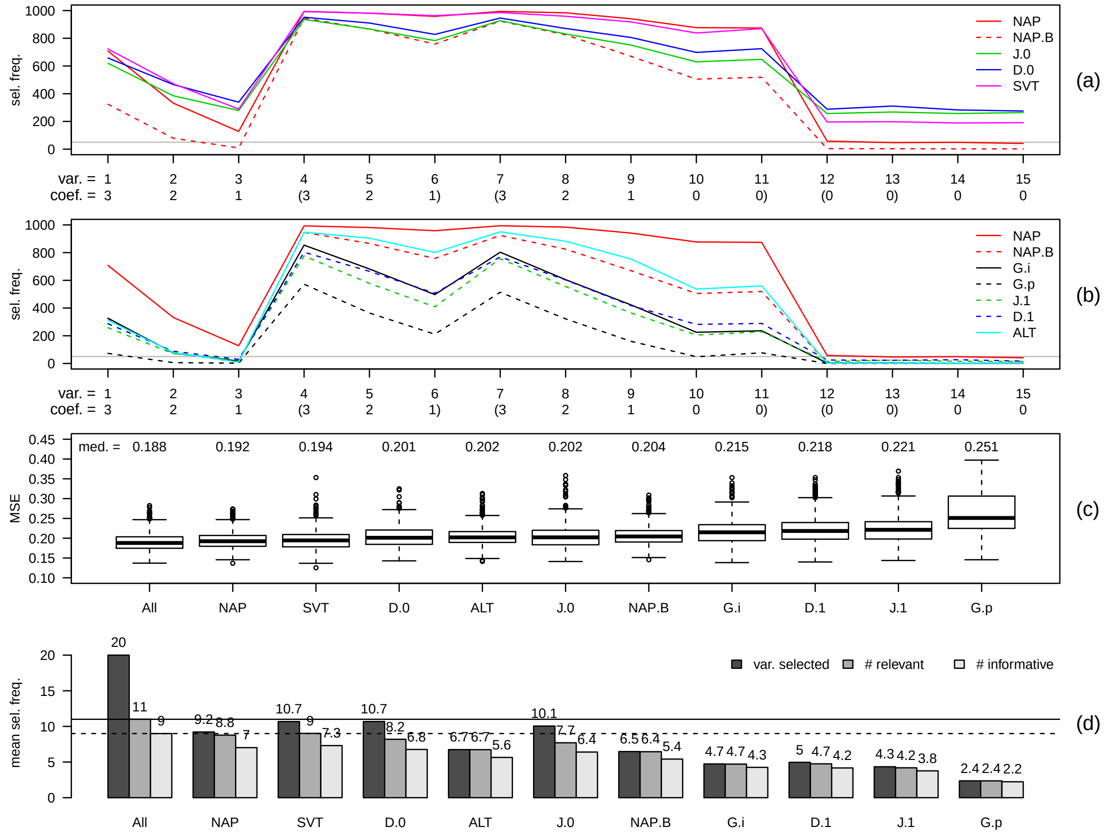


Figure A.8: Selection frequencies of variables 1-15 in 1000 simulation runs of the classification problem of study III. The new approaches are compared to class III (a) and class II (+ ALT) (b). The horizontal grey line represents a 5% TWER. Brackets indicate correlated variables. (c) Boxplots of the observed MSE aranged in increasing order of median values. (d) Mean selection frequency of variables and an additional information about the amount of relevant and informative variables among them. Reference values are indicated by horizontal lines. [Relevant Variables: var. 1 – 11; Informative Variables: var. 1 – 9]

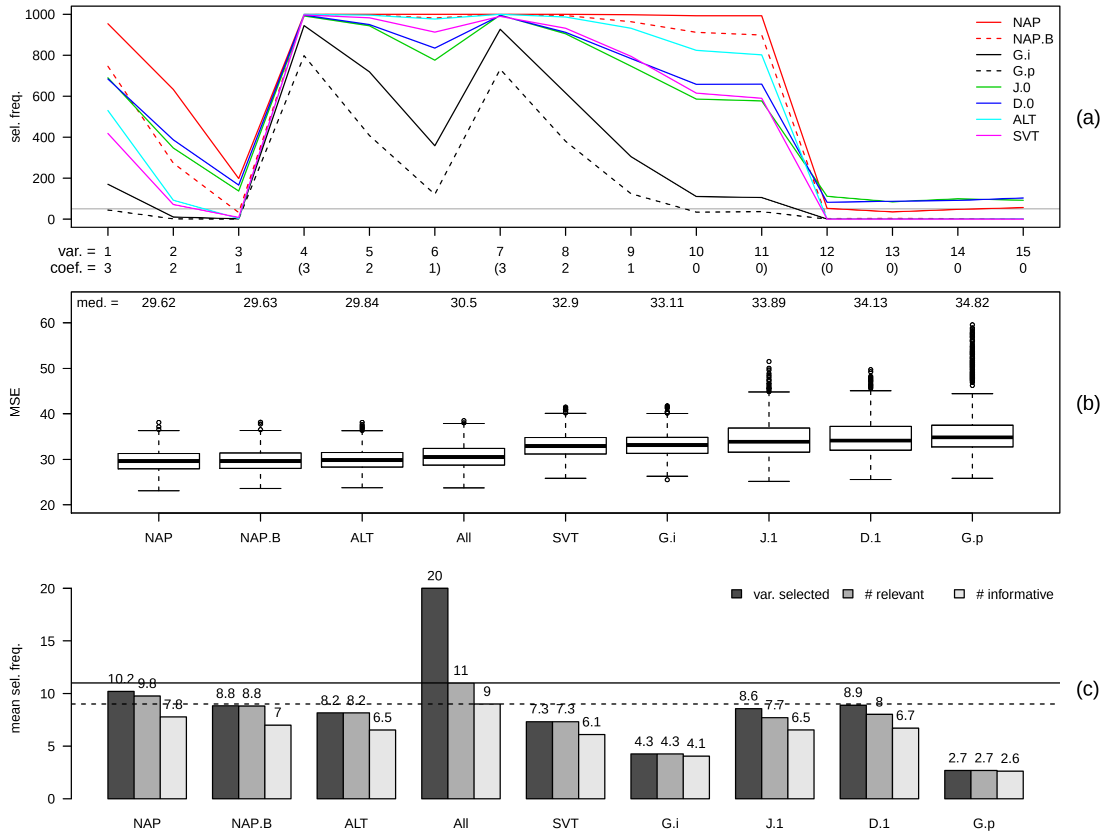


Figure A.9: Selection frequencies of variables 1-15 in 1000 simulation runs of the regression problem of study III. The new approaches are compared to class III (a) and class II (+ ALT) (b). The horizontal grey line represents a 5% TWER. Brackets indicate correlated variables. (c) Boxplots of the observed MSE aranged in increasing order of median values. (d) Mean selection frequency of variables and an additional information about the amount of relevant and informative variables among them. Reference values are indicated by horizontal lines. [Relevant Variables: var. 1 – 11; Informative Variables: var. 1 – 9]

lowest median MSE and the highest effectiveness while the efficiencies are among the highest, too.

A.4.2 Empirical Evaluation

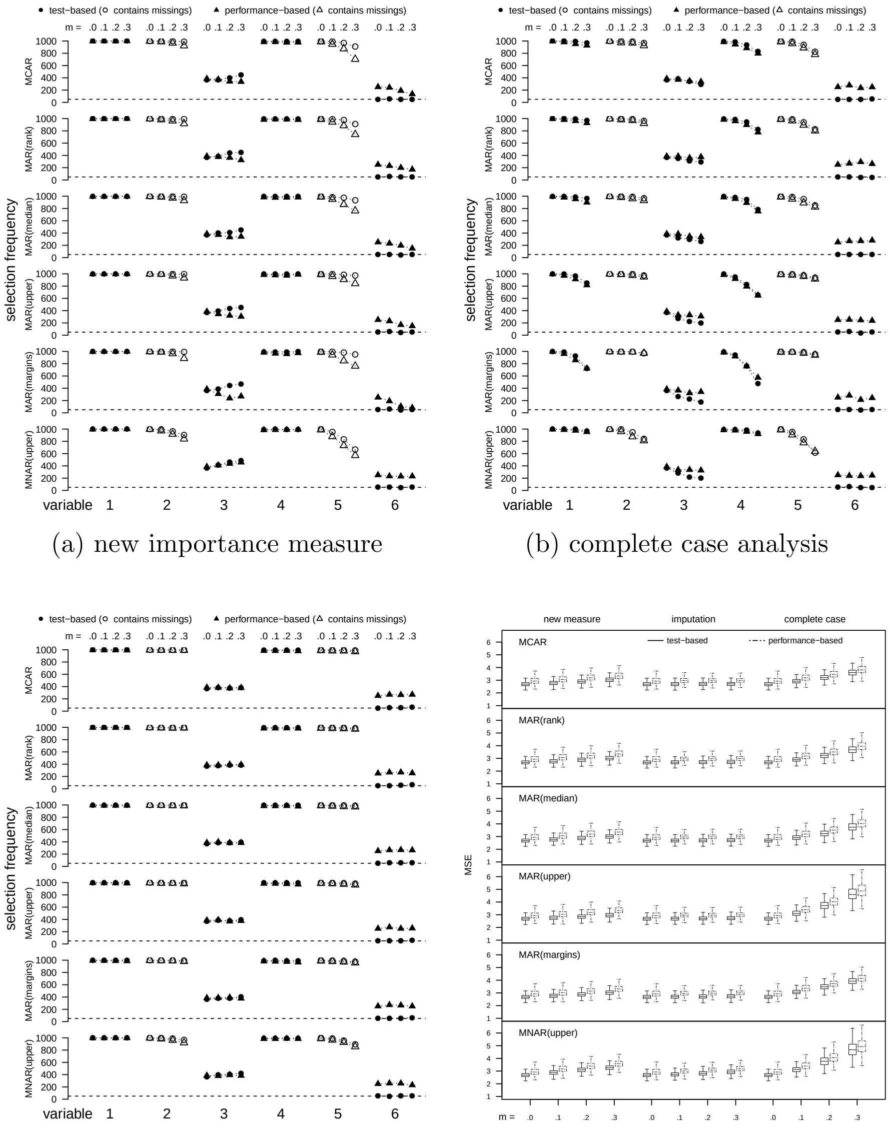
In addition to the findings of section 5.4.2 Table A.4 contains the selection frequencies of variables which were observed for the empirical evaluation within 1000 bootstrap runs.

	Variable	NAP	NAP.B	G.i	G.p	J.0	J.1	D.0	D.1	ALT	SVT
Birthweight	low	1000	1000	1000	1000	1000		1000		1000	1000
	age	907	553	7	0	658		612		255	3
	lwt	990	922	214	1	920		924		783	72
	race	967	861	0	0	791		826		759	10
	smoke	987	917	0	0	808		839		836	5
	ptl	576	323	0	0	47		98		274	0
	ht	353	208	0	0	26		34		175	0
	ui	975	883	0	0	654		731		875	1
	ftv	498	137	0	0	102		112		30	0
B. Housing	CRIM	1000	1000	135	14	979		957		999	107
	ZN	969	860	0	0	65		105		73	0
	INDUS	1000	1000	69	1	813		910		996	224
	CHAS	859	698	0	0	43		50		169	0
	NOX	1000	1000	226	57	958		970		997	92
	RM	1000	1000	1000	999	1000		1000		1000	1000
	AGE	1000	1000	0	0	636		657		648	0
	DIS	1000	1000	1	0	757		745		699	2
	RAD	1000	1000	0	0	405		411		515	0
	TAX	1000	1000	5	0	862		853		997	29
	PTRATIO	1000	1000	191	18	909		955		1000	326
	B	1000	1000	0	0	385		397		298	0
	LSTAT	1000	1000	1000	1000	1000		1000		1000	1000
Heart	age	917	724	83	5	628	161	672	182	253	761
	sex	993	932	233	25	768	306	825	358	647	882
	cp	1000	999	985	448	991	944	995	954	994	999
	trestbps	765	435	50	1	486	96	501	95	62	586
	chol	761	452	68	2	544	128	547	107	83	663
	fbs	296	90	1	0	183	12	201	13	1	175
	restecg	725	423	50	3	525	121	515	122	97	582
	thalach	1000	997	464	119	878	531	902	587	905	958
	exang	995	977	361	59	849	476	875	504	843	932
	oldpeak	999	996	609	192	908	682	948	708	959	972
	slope	984	917	310	44	777	366	810	391	701	907
	ca	1000	1000	991	450	998	984	998	985	1000	1000
	thal	1000	1000	982	643	996	977	997	987	1000	999
Parkinson	MDVP.Fo.Hz.	995	950	681	147	863	577	887	558	922	845
	MDVP.Fhi.Hz.	962	708	241	15	634	277	629	222	451	506
	MDVP.Flo.Hz.	966	745	212	29	565	159	657	242	554	585
	MDVP.Jitter...	572	98	0	0	140	3	206	15	9	187
	MDVP.Jitter.Abs.	940	653	55	5	401	59	529	113	369	408
	MDVP.RAP	595	167	5	0	177	15	242	25	28	212
	MDVP.PPQ	599	138	1	0	153	6	208	24	13	207
	Jitter.DDP	590	183	11	0	187	17	248	33	33	231
	MDVP.Shimmer	991	897	353	34	649	242	752	322	759	645
	MDVP.Shimmer.dB.	923	590	80	0	390	73	513	131	376	436
	Shimmer.APQ3	965	702	121	2	540	160	617	189	402	484
	Shimmer.APQ5	964	754	188	11	498	129	643	236	534	536
	MDVP.APQ	988	907	326	33	594	189	744	334	670	661
	Shimmer.DDA	973	738	153	9	592	194	636	202	452	522
	NHR	455	86	5	0	128	4	155	15	4	146
	HNR	931	598	112	15	493	125	567	160	244	454
	RPDE	888	464	63	4	463	158	508	129	183	393
	DFA	847	378	67	4	399	77	435	108	224	336
	spread1	1000	1000	957	880	992	982	977	916	1000	994
	spread2	995	980	632	140	810	439	879	595	951	861
	D2	955	759	349	62	623	253	688	325	617	576
	PPE	1000	1000	898	541	936	760	967	857	1000	988

Table A.4: Summary of variable selection frequencies observed for 1000 bootstrap runs of the empirical evaluation.

A.5 Chapter 6

Results for the regression problem are presented in Figure A.10. They underline the findings for the classification problem. The 0 s.e. rule was executed for the performance-based approach. As a consequence, its TWER and selection frequencies are increased.



(a) new importance measure

(b) complete case analysis

(c) multiple imputation

(d) MSE

Figure A.10: Variable selection frequencies and MSE observed for the regression problem ($m = \%$ of missing values in X_2 and X_5).

Appendix B

R-Code

All of this works computations are operable by the following Code using `R` (version 2.14.1). General functions used throughout the studies are:

```
# Function to count selection frequencies of variables in Random Forests
count <- function(forest, inames = NULL) {
  # forest: object of class "RandomForest" created by the function cforest()
  # inames: names of variables to be assessed (defaults to NULL, using all variables)
  if (is.null(inames) && extends(class(forest), "RandomForest"))
  inames <- names(forest@data@get("input"))
  resultvec <- rep(0, length(inames))
  myfunc <- function(x, inames, resultvec) {
    names(x) <- c("nodeID", "weights", "criterion", "terminal", "psplit",
                                "ssplits", "prediction", "left", "right")
    names(x$criterion) <- c("statistic", "criterion", "maxcriterion")
    if (!x$terminal) {
     resultvec[x$psplit[[1]]] <- resultvec[x$psplit[[1]]] + 1
     resultvec <- myfunc(x$left, inames = inames, resultvec = resultvec)
     resultvec <- myfunc(x$right, inames = inames, resultvec = resultvec)
     }
    return(resultvec)
    }
  for (k in 1:length(forest@ensemble)) {
  resultvec <- myfunc(forest@ensemble[[k]], inames, resultvec)
  }
names(resultvec) <- inames
return(resultvec)
}
environment(count) <- environment(varimp)

# Function to simulate data
create.dat <- function(coefs = c(0, 0, 0, 0, 0), n = 100,
                       sigma = NULL, regression = T, error = 1) {
  # coefs: coefficients; n: number of observations; sigma: covariance matrix
  # regression: TRUE produces a regression problem, FALSE a classification problem
  # error: error added to outcome in case of a regression problem
  if (is.null(sigma)) sigma <- diag(length(coefs)) # initialize sigma
  if (length(coefs) != nrow(sigma)) stop("dimension of coefs and sigma differs")
  dat <- rmvnorm(n, sigma = sigma) # create covariates
  x.beta <- dat %*% coefs
  dat <- as.data.frame(dat)
  if (regression == T) dat$response <- x.beta + rnorm(n, 0, error) # create the response
  else dat$response <- as.factor(rbinom(n, 1, exp(x.beta) / (1 + exp(x.beta))))
  return(dat)
}

# Function to induce missing values into a data set
with.missings <- function(data, mis.var, ind.var, m) {
 # data: data meant to contain missing values
 # mis.var: names of variables to be partly set missing
 # ind.var: variables that induce the missing values
 # m * 10: fraction of missing values in %
 X <- lapply(1:6, function(x) data) # for 6 missing data generating processes
 n <- nrow(data)
 for (k in mis.var) {
  ind <- ind.var[mis.var == k]
  # induce missing values MCAR
  is.na(X[[1]][,k])[sample(1:n, m * .1 * n)] <- TRUE
  # induce missing values MAR(rank)
  is.na(X[[2]][,k])[sample(1:n, m * .1 * n, prob = rank(X[[2]][,ind]) / sum(1:n))] <- TRUE
  # induce missing values MAR(median)
  is.na(X[[3]][,k])[sample(1:n, m * .1 * n, prob = ifelse(X[[3]][,
                                            ind] >= median(X[[3]][,ind]), .9, .1))] <- TRUE
  # induce missing values MAR(upper)
  is.na(X[[4]][,k])[X[[4]][,ind] >=  sort(X[[4]][,ind], T)[m * .1 * n]] <- TRUE
  # induce missing values MAR(margins)
  is.na(X[[5]][,k])[X[[5]][,ind] >= sort(X[[5]][,ind], T)[m * .1 * n / 2] |
                    X[[5]][,ind] <= sort(X[[5]][,ind])[m * .1 * n / 2]] <- TRUE
  # induce missing values MNAR(upper)
  is.na(X[[6]][,k])[X[[6]][,k] >= sort(X[[6]][,k], T)[m * .1 * n]] <- TRUE
  }
  return(X)
}
```

B.1 Chapter 1

The following code gives a short summary of how to create and display trees and Random Forests. Further functionalities are given by manuals of the packages `party` and `rpart`. Also, steps to create figures and views can be taken from any R-Code textbook and are not presented here.

```
set.seed(290875) # set a random seed for reproducibility of results

# load the airquality data and reject observations with missing response
airq <- subset(airquality, !is.na(Ozone))
```

```
library(party) # load required package ‘party’ version 1.0-0
airct <- ctree(Ozone ~ ., data = airq, # create a tree with 3 surrogates
               controls = ctree_control(maxsurrogate = 3))
airct # display it
plot(airct) # and plot it

library(rpart) # load required package ‘rpart’ version 3.1-52
aircart <- rpart(Ozone ~ ., data = airq) # create a CART like tree
printcp(aircart) # assess the cross-validated error for different tree sizes
aircart.pruned <- prune(aircart, cp = 0.08) # prune the tree to the optimal size
plot(aircart.pruned); text(aircart.pruned) # plot the tree

aircf <- cforest(Ozone ~ ., data = airq) # create a Random Forest
count(aircf) # apply the count() function

airq2 <- na.omit(airquality) # create a complete case version of the data
aircf2 <- cforest(Ozone ~ ., data = airq2) # and a corresponding Random Forest
# compute the permutation importance measure and its conditional version
varimp(aircf2, pre1.0_0 = T); varimp(aircf2, conditional = T)
```

B.2 Chapter 2

B.2.1 Simulation Studies

R-Code used for the simulation studies by the example of Haberman’s Survival Data (other datasets were processed the same way):

```
# load required packages
library(party) # version 1.0-0
library(mice)  # version 2.11
library(rpart) # version 3.1-52

# define the predictors used in the imputation models
predmat <- matrix(1, ncol = 4, nrow = 4); diag(predmat) <- 0; predmat[, 4] <- 0

repetitions <- 1000 # there were 1000 MCCV runs
# create matrices that take the observed MSE values for Random Forests, conditional
# inference trees (=tree) and CART (=oldtree). "with" = imputation was used
mse.without.forest  <- mse.with.forest  <- mse.without.tree <- mse.with.tree <-
mse.without.oldtree <- mse.with.oldtree <- matrix(NA, nrow = 4, ncol = repetitions,
                                   dimnames = list(seq(.1, .4, .1), 1:repetitions))

# load the data
dat <- read.table("your_local_directory/haberman.txt", sep = ",")
names(dat) <- c("age", "year", "nodes", "surv")
dat$surv <- as.factor(dat$surv)

for (j in 1:4) { # there are four fractions of missing values
for (i in 1:repetitions) { # and 1000 repetitions
  misdat <- dat
  miss <- sapply(1:3, function(x) sample(1:nrow(misdat), round(nrow(misdat)* 0.1 * j)))
  for (k in 1:3) {is.na(misdat[miss[, k], k]) <- T}
  ### when only 1/3 of variables should contain missing values, use instead:
  #miss <- sample(1:nrow(misdat), round(nrow(misdat)* 0.1 * j))
  #is.na(misdat[miss, sample(1:3, 1)]) <- T
  # samples are drawn to build the training and test data
  samp <- sample(1:nrow(misdat), round(nrow(misdat) * .8), replace = F)
  dat.train <- misdat[samp, ]; dat.test <- misdat[-samp, ]
  # use MICE to impute the data
  test.mi <- mice(dat.test,   printFlag = F, defaultMethod = c("norm", "logreg", "polyreg"),
                                                                  predictorMatrix = predmat)
  train.mi <- mice(dat.train, printFlag = F, defaultMethod = c("norm", "logreg", "polyreg"))
  # compute the MSE values without imputation
  forest <- cforest(surv ~ ., data = dat.train, controls =
                         cforest_unbiased(maxsurrogate = min(3, ncol(dat) - 2)))
  mse.without.forest[j, i] <- mean((as.numeric(dat.test$surv) -
                             as.numeric(Predict(forest, newdata = dat.test)))^2)
  tree <- ctree(surv ~ ., data = dat.train, controls =
                            ctree_control(maxsurrogate = min(3, ncol(dat) - 2)))
  mse.without.tree[j, i] <- mean((as.numeric(dat.test$surv) -
                               as.numeric(predict(tree, newdata = dat.test)))^2)
  oldtree <- rpart(surv ~ ., data = dat.train)
  mse.without.oldtree[j, i] <- mean((as.numeric(dat.test$surv) -
                      round(predict(oldtree, newdata = dat.test)[, "2"]) - 1)^2)
  # compute the MSE values with imputation
  mse.with.forest[j, i] <-  mean((as.numeric(dat.test$surv) - round(rowMeans(sapply(1:5,
                                 function(x) {mi.forest <- cforest(surv ~ .,
                                 data = complete(train.mi, action = x), controls =
                                 cforest_unbiased(maxsurrogate = min(3, ncol(dat) - 2)));
                                 return(rowMeans(sapply(1:5, function(y)
                                 as.numeric(Predict(mi.forest, newdata = complete(test.mi,
                                 action = y))))))}))))^2)
  mse.with.tree[j, i] <- mean((as.numeric(dat.test$surv) - round(rowMeans(sapply(1:5,
                                 function(x) {mi.forest <- ctree(surv ~ ., data =
                                 complete(train.mi, action = x), controls =
                                 ctree_control(maxsurrogate = min(3, ncol(dat) - 2)));
                                 return(rowMeans(sapply(1:5, function(y)
                                 as.numeric(predict(mi.forest, newdata =
                                 complete(test.mi, action = y))))))}))))^2)
  mse.with.oldtree[j, i] <-  mean((as.numeric(dat.test$surv) - round(rowMeans(sapply(1:5,
                                 function(x) {mi.forest <- rpart(surv ~ ., data =
                                 complete(train.mi, action = x));
                                 return(rowMeans(sapply(1:5, function(y)
                                 round(predict(mi.forest, newdata = complete(test.mi,
                                 action = y))[, "2"]))))}))) - 1)^2)
  }
}
```

B.2.2 Empirical Evaluation

R-Code used for the application studies by the example of the Hepatitis Dataset (other datasets were processed the same way):

```
# load required packages
library(party) # version 1.0-0
library(mice)  # version 2.11
library(rpart) # version 3.1-52

# define the predictors used in the imputation models
predmat <- matrix(1, ncol = 20, nrow = 20); diag(predmat) <- 0; predmat[, 1] <- 0

repetitions <- 1000 # there were 1000 MCCV runs
# create matrices that take the observed MSE values for Random Forests, conditional
# inference trees (=tree) and CART (=oldtree). "with" means that imputation was used
mse.ohne.forest <- mse.mit.forest <- mse.ohne.tree <- mse.mit.tree <-
mse.ohne.oldtree <- mse.mit.oldtree <- rep(NA, repetitions)

# load the data
dat <- read.table("your_local_directory/hepatitis.txt", sep = ",", na.strings = "?")
names(dat) <- c("status", "age", "sex", "steroid", "antivirals", "fatigue",
                "malaise", "anorexia", "liver", "livfirm", "spleen",
                "spiders", "ascites", "varices", "bilirubin", "alk", "sgot",
                "albumin", "protime", "histology")
dat$status <- as.factor(dat$status)

for (i in 1:repetitions) { # there are 1000 repetitions
  # samples are drawn to build the training and test data
  samp <- sample(1:nrow(dat), round(nrow(dat) * .8), replace = F)
  dat.train <- dat[samp, ];  dat.test <- dat[-samp, ]
  # use MICE to impute the data
  train.mi <- mice(dat.train, printFlag = F, defaultMethod = c("norm", "logreg", "polyreg"))
  test.mi <-  mice(dat.test,  printFlag = F, defaultMethod = c("norm", "logreg", "polyreg"),
                                                                  predictorMatrix = predmat)
  # compute the MSE values without imputation
  mse.ohne.forest[i] <- mean((as.numeric(dat.test$status)   -  as.numeric(Predict(cforest(
                             status ~ ., data = dat.train, controls =
                             cforest_unbiased(maxsurrogate = 3)), newdata = dat.test)))^2)
  tree <- ctree(status ~ ., data = dat.train, controls = ctree_control(maxsurrogate  = 3))
  mse.ohne.tree[i] <- mean((as.numeric(dat.test$status) - as.numeric(predict(tree,
                                                                  newdata = dat.test)))^2)
  oldtree <- rpart(status ~ ., data = dat.train)
  mse.ohne.oldtree[i] <- mean((as.numeric(dat.test$status) - round(predict(oldtree,
                                                       newdata = dat.test)[, "2"]) - 1)^2)
  # compute the MSE values with imputation
  mse.mit.forest[i] <-  mean((as.numeric(dat.test$status) - round(rowMeans(sapply(1:5,
                              function(x) {mi.forest <- cforest(status ~ ., data =
                              complete(train.mi, action = x), controls =
                              cforest_unbiased(maxsurrogate = 3)); return(rowMeans(
                              sapply(1:5, function(y) as.numeric(Predict(mi.forest,
                              newdata = complete(test.mi, action = y))))))}))))^2)
  mse.mit.tree[i] <- mean((as.numeric(dat.test$status) - round(rowMeans(sapply(1:5,
                              function(x) {mi.forest <- ctree(status ~ ., data =
                              complete(train.mi, action = x), controls =
                              ctree_control(maxsurrogate = 3));    return(rowMeans(
                              sapply(1:5, function(y) as.numeric(predict(mi.forest,
                              newdata =  complete(test.mi, action = y))))))})))))^2)
  mse.mit.oldtree[i] <-  mean((as.numeric(dat.test$status) - round(rowMeans(sapply(
                              1:5, function(x) {mi.forest <- rpart(status ~ .,  data =
                              complete(train.mi, action = x)); return(rowMeans(sapply(
                              1:5, function(y) round(predict(mi.forest, newdata =
                              complete(test.mi, action = y))[, "2"]))))}))) - 1)^2)
}
```

B.3 Chapter 3

B.3.1 Simulation Studies

R-Code used for the simulation study:

```
# load required packages
library("party"); attach(asNamespace("party")) # version 1.0-0
library(mvtnorm) # version 0.9-9992
library(mice) # version 2.11

# create a list of covariance matrices for each correlation strength
sig <- lapply(1:4, function(x) {r <- c(0,.3,.6,.9)[x]; y <- diag(20);
                y[1:3, 1:3] <- r; y[4:5, 4:5] <- r; y[6:7, 6:7] <- r;
                        y[8:11, 8:11] <- r; diag(y) <- 1; return(y)})

# create lists that contain arrays used to collect the results of 20 variables for 4
# fractions of missing values, 4 correlation strength and 6 missing data generating
# processes in 1000 simulation runs. The common importance measure (‘old’), the new
# approach (‘new’) and selection frequencies (‘count’) are recorded for the
# regression (‘reg’) and classification (‘clas’) problem.
reg.old <- clas.old <- reg.new <- clas.new <- reg.count <- clas.count <-
lapply(1:6, function(y) lapply(1:4, function(x) array(dim = c(1000, 20, 4))))

set.seed(1234) # set a random seed for reproducibility of results
# 1000 simulation runs start here
for (i in 1:1000) {
  for (r in 1:4) { # there are 4 correlation strength
    dat <- as.data.frame(rmvnorm(100, sigma = sig[[r]])) # create the data
    x.beta <- with(dat, 4 * V1 + 4 * V2 + 3 * V3 + 4 * V4 + 3 * V5  +
                        4 * V6 + 3 * V7 + 4 * V8 + 3 * V9 + 2 * V12 + 2 * V13
    dat$y.reg  <- x.beta + rnorm(100, 0, .5)
    dat$y.clas <- rbinom(100, 1, exp(x.beta) / (1 + exp(x.beta)))
    for (m in 1:4) { # there are 4 fractions of missing values
      # the data is replicated for each of 6 missing data generating process
      dat.mis <- with.missings(dat, c("V2", "V4", "V8", "V10", "V12", "V14"),
                                    c("V3", "V5", "V9", "V11", "V13", "V15"), m - 1)
      for (j in 1:6) { # compute results for 6 missing data generating processes
      #  create random forests and compute importances and selection frequencies
      forest.reg  <- cforest(as.formula(paste("y.reg",  paste("V", 1:20, sep = "",
                             collapse = " + "), sep = " ~ ")), data = dat.mis[[j]],
                             controls = cforest_unbiased(mtry = 8, ntree = 50,
                             maxsurrogate = 3))
      forest.clas <- cforest(as.formula(paste("y.clas", paste("V", 1:20,  sep = "",
                             collapse = " + "), sep = " ~ ")), data = dat.mis[[j]],
                             controls = cforest_unbiased(mtry = 8, ntree = 50,
                             maxsurrogate = 3))
      reg.new[[j]][[r]][i, , m]    <- varimp(forest.reg)
      clas.new[[j]][[r]][i, , m]   <- varimp(forest.clas)
      reg.count[[j]][[r]][i, , m]  <- count(forest.reg)
      clas.count[[j]][[r]][i, , m] <- count(forest.clas)
      if (m == 1) {
      reg.old[[j]][[r]][i, , m]  <- varimp(forest.reg,  pre1.0_0 = TRUE)
      clas.old[[j]][[r]][i, , m] <- varimp(forest.clas, pre1.0_0 = TRUE)
      }}}
    }
}
```

B.3.2 Empirical Evaluation

R-Code used for the empirical evaluation:

```
# load required package
library("party"); attach(asNamespace("party")) # version 1.0-0

# load the pima indians diabetes data as provided by the UCI repository
pima <- read.table("your_local_directory/pima_data.txt",  sep = ",")
names(pima) <- c("num.preg", "gluc", "bloodpres", "skin", "insulin",
                               "bmi", "pedigree", "age", "diabetes")
pima$diabetes <- as.factor(pima$diabetes)  # prepare the data
is.na(pima$gluc[pima$gluc == 0]) <- T      # and define the missing values
is.na(pima$bloodpres[pima$bloodpres == 0]) <- T
is.na(pima$skin[pima$skin == 0]) <- T; is.na(pima$bmi[pima$bmi == 0]) <- T
is.na(pima$insulin[pima$insulin == 0]) <- T

# load the mammal sleep data
slep <- read.table("your_local_directory/sleep_data.txt", header = T)

set.seed(5) # set a random seed for reproducibility of results
# create forests of 5000 trees for the pima data and the sleep data
forest.pima.new <- cforest(diabetes ~ ., controls = cforest_unbiased(mtry = 3,
                                 maxsurrogate = 3, ntree = 5000), data = pima)
forest.sleep.new <- cforest(BodyWgt ~ ., controls = cforest_unbiased(mtry = 3,
                                 maxsurrogate = 3, ntree = 5000), data = slep)
# create a forest of 5000 trees for a complete case versions of the data
forest.pima.old <- cforest(diabetes ~ ., controls = cforest_unbiased(mtry = 3,
                        maxsurrogate = 3, ntree = 5000), data = na.omit(pima))
forest.sleep.old <- cforest(BodyWgt ~ ., controls = cforest_unbiased(mtry = 3,
                        maxsurrogate = 3, ntree = 5000), data = na.omit(slep))

# compute the corresponding variable importances by the new approach
pima.new  <- varimp(forest.pima.new); sleep.new <- varimp(forest.sleep.new)
pima.old  <- varimp(forest.pima.old, pre1.0_0 = T)
sleep.old <- varimp(forest.sleep.old, pre1.0_0 = T)

# Additional simulation study

# create the response Y and predictors U and V
U <- sample(rep(1:0, times = c(1000, 4000)))
V <- sample(rep(0:1, times = c(4000, 1000)))
Y <- rep(NA, 5000)
Y[U == 1] <- rnorm(sum(U == 1), 2, 1); Y[V == 1] <- rnorm(sum(V == 1), -2, 1)
Y[U == 0 & V == 0] <- rnorm(sum(U == 0 & V == 0), 0, 1)
Y[U == 1 & V == 1] <- rnorm(sum(U == 1 & V == 1), 0, 1)
# induce missing values (MAR(upper)) in V
V2 <- V; is.na(V2)[which(Y >= quantile(Y, .7))] <- TRUE

# fit a Random Forest to the complete data
fullfor <- cforest(Y ~ U + V, controls = cforest_unbiased(mtry = 2, ntree = 5000))
fullimp <- varimp(fullfor, pre1.0_0 = T); fullimp   # compute variable importances
# recompute the Random Forest in a complete case analysis
misfor <- cforest(Y ~ ., data = na.omit(as.data.frame(cbind(Y, U, V2))),
                    controls = cforest_unbiased(mtry = 2, ntree = 5000))
misimp <- varimp(misfor, pre1.0_0 = T); misimp # recompute the variable importance
# recompute the Random Forest using the entire data
misfor2 <- cforest(Y ~ U + V2, controls = cforest_unbiased(mtry = 2,
                                    ntree = 5000, maxsurrogate = 1))
newimp <- varimp(misfor2); newimp # apply the new approach
```

B.4 Chapter 4

R-Code used for the simulation studies:

```
# load required packages
library("party"); attach(asNamespace("party")) # version 1.0-0
library(mvtnorm) # version 0.9-9992
library(mice)    # version 2.11

# create covariance matrix
sig <- matrix(c(  1, 0.3, 0.3, 0.3, 0, 0,
                0.3,   1, 0.3, 0.3, 0, 0,
                0.3, 0.3,   1, 0.3, 0, 0,
                0.3, 0.3, 0.3,   1, 0, 0,
                  0,   0,   0,   0, 1, 0,
                  0,   0,   0,   0, 0, 1), ncol = 6, byrow = T)

# create a list of arrays to collect results for 6 variables, 4 fractions of
# missing values and 6 missing data generating processes in 1000 simulation runs
```

```
mylist <- lapply(1:6, function(x) array(dim = c(1000, 6, 4)))

# create lists which contain importance measures for the new approach (‘new’),
# complete case analysis (‘cc’) and multiple imputation (‘imp’) in a
# classification (‘.c’) and regression problem (‘.r’)
new.c <- cc.c <- imp.c <- new.r <- cc.r <- imp.r <-  mylist
# create arrays that contain the corresponding MSE values
new.err.c <- cc.err.c <- imp.err.c <-
new.err.r <- cc.err.r <- imp.err.r <- array(NA, c(1000, 4, 6))

set.seed(1234) # set a random seed for reproducibility of results

# create the test data
test.dat <- as.data.frame(rmvnorm(5000, sigma = sig))
x.beta <- 1 * test.dat$V1 + 1 * test.dat$V2 + 1 * test.dat$V5
test.dat$y.reg <- x.beta + rnorm(5000, 0, .5)
test.dat$y.clas <- as.factor(rbinom(5000, 1, exp(x.beta) / (1 + exp(x.beta))))

# 1000 simulation runs start here
for (i in 1:1000) {
dat <- as.data.frame(rmvnorm(100, sigma = sig)) # simulate the data
x.beta <- 1 * dat$V1 + 1 * dat$V2 + 1 * dat$V5
dat$y.reg <- x.beta + rnorm(100, 0, .5)
dat$y.clas <- as.factor(rbinom(100, 1, exp(x.beta) / (1 + exp(x.beta))))
for (m in 1:4) { # there are 4 fractions of missing values
dat.mis <- with.missings(dat, c("V2", "V4", "V5"), c("V1", "V3", "V6"), m - 1)
for (j in 1:6) { # there are 6 missing data generating processes
# create Random Forests without imputation
for.reg  <- cforest(y.reg ~ V1 + V2 + V3 + V4 + V5 + V6, data = dat.mis[[j]],
                  controls = cforest_unbiased(mtry = 3, ntree = 50, maxsurrogate = 3))
for.clas <- cforest(y.clas ~ V1 + V2 + V3 + V4 + V5 + V6, data = dat.mis[[j]],
                  controls = cforest_unbiased(mtry = 3, ntree = 50, maxsurrogate = 3))
# compute variable importances and MSE
new.c[[j]][i, , m] <- varimp(for.clas)
new.r[[j]][i, , m] <- varimp(for.reg)
new.err.c[i, m, j] <- mean((as.numeric(test.dat$y.clas) -
                                 as.numeric(predict(for.clas, newdata = test.dat)))^2)
new.err.r[i, m, j] <- mean((test.dat$y.reg -  predict(for.reg, newdata = test.dat))^2)
# create Random Forests in a complete case analysis
cc.reg  <- cforest(y.reg  ~ V1 + V2 + V3 + V4 + V5 + V6, data = na.omit(dat.mis[[j]]),
                  controls = cforest_unbiased(mtry = 3, ntree = 50, maxsurrogate = 3))
cc.clas <- cforest(y.clas ~ V1 + V2 + V3 + V4 + V5 + V6, data = na.omit(dat.mis[[j]]),
                  controls = cforest_unbiased(mtry = 3, ntree = 50, maxsurrogate = 3))
# compute variable importances and MSE
cc.c[[j]][i, , m] <- varimp(cc.clas, pre1.0_0 = T)
cc.r[[j]][i, , m] <- varimp(cc.reg,  pre1.0_0 = T)
cc.err.c[i, m, j] <- mean((as.numeric(test.dat$y.clas) -
                               as.numeric(predict(cc.clas, newdata = test.dat)))^2)
cc.err.r[i, m, j] <- mean((test.dat$y.reg - predict(cc.reg, newdata = test.dat))^2)
# create Random Forests with imputation when there is missing data
if (m > 1) {
```

```
imp.dat <- mice(dat.mis[[j]], printFlag = F,
                                  defaultMethod = c("norm", "logreg", "polyreg"))
imp.reg  <- lapply(1:5, function(x) cforest(y.reg  ~ V1 + V2 + V3 + V4 + V5 + V6,
                       data = complete(imp.dat, action = x), controls =
                       cforest_unbiased(mtry = 3, ntree = 50, maxsurrogate = 3)))
imp.clas <- lapply(1:5, function(x) cforest(y.clas ~ V1 + V2 + V3 + V4 + V5 + V6,
                       data = complete(imp.dat, action = x), controls =
                       cforest_unbiased(mtry = 3, ntree = 50, maxsurrogate = 3)))
# compute variable importances and MSE
imp.r[[j]][i, , m] <- rowMeans(sapply(imp.reg,  function(x) varimp(x, pre1.0_0 = T)))
imp.c[[j]][i, , m] <- rowMeans(sapply(imp.clas, function(x) varimp(x, pre1.0_0 = T)))
imp.err.c[i, m, j] <- mean(sapply(imp.clas, function(x)
                           mean((as.numeric(test.dat$y.clas) -
                           as.numeric(predict(x, newdata = test.dat)))^2)))
imp.err.r[i, m, j] <- mean(sapply(imp.reg,  function(x)
                           mean((test.dat$y.reg - predict(x, newdata = test.dat))^2)))}
if (m == 1) {
imp.r[[j]][i, , m] <- cc.r[[j]][i, , m]
imp.c[[j]][i, , m] <- cc.c[[j]][i, , m]
imp.err.c[i, m, j] <- cc.err.c[i, m, j]
imp.err.r[i, m, j] <- cc.err.r[i, m, j]
}}}
}
```

B.5 Chapter 5

B.5.1 Functions to perform Variable Selection

R-Code used to implement the variable selection methods of section 5.3.4:

```
# load required packages
library("party"); attach(asNamespace("party")) # version 1.0-0

####################################
### The NAP and NAP.b approaches ###
####################################
NAP <- function(Y, X, nperm = 100, ntree = 50, alpha = 0.05) {
  # Y: response vector
  # X: matrix or data frame containing the predictors
  # nperm: number of permutations
  # ntree: number of trees contained in a Random Forest
  # alpha: alpha level for permutation tests
  # RETURNS: selected variables, a corresponding Random Forest and the
  #          OOB-error with and without Bonferroni-Adjustment
  mtry <- ceiling(sqrt(ncol(X))) # automatically set mtry to sqrt(p)
  dat <- cbind(Y, X) # create the data
  names(dat) <- c("response", paste("V", 1:ncol(X), sep = ""))
  forest <- cforest(response ~ ., data = dat, # fit a Random Forest
                      controls = cforest_unbiased(mtry = mtry, ntree = ntree))
  obs.varimp <- varimp(forest, pre1.0_0 = T) # compute initial importances
  selection <- names(obs.varimp)
  # create a matrix that contains the variable importances after permutation
  perm.mat <- matrix(NA, ncol = length(selection), nrow = nperm,
                                              dimnames = list(1:nperm, selection))
  for (j in selection) { # repeat the computational steps for each variable
    perm.dat <- dat # perm.dat will be the data after permutation
    for (i in 1:nperm) { # do nperm permutations
      perm.dat[, j] <- sample(perm.dat[, j]) # permute each variable
      perm.forest <- cforest(response ~ ., data = perm.dat, # recompute the forest
                        controls = cforest_unbiased(mtry = mtry, ntree = ntree))
      perm.mat[i, j] <- varimp(perm.forest, pre1.0_0 = T)[j]}} # recompute importances
  p.vals <- sapply(selection, function(x) sum(perm.mat[, x] # compute p-values
                                                      >= obs.varimp[x]) / nperm)
  p.vals.bonf <- p.vals * length(p.vals) # p-values with Bonferroni-Adjustment
  if (any(p.vals < alpha)) { # keep significant variables
   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))}
  if (any(p.vals.bonf < alpha)) { # keep significant variables (Bonferroni)
   selection.bonf <- names(p.vals.bonf)[which(p.vals.bonf < alpha)]
   mtry <- ceiling(sqrt(length(selection.bonf)))
   forest.bonf <- cforest(as.formula(paste("response", paste(selection.bonf,
                                   collapse = " + "), sep = " ~ ")), data = dat,
                       controls = cforest_unbiased(mtry = mtry, ntree = ntree))}
  if (!any(p.vals < alpha)) { # if there are not significant variables
   selection <- c(); forest <- c()}
  if (!any(p.vals.bonf < alpha)) { # if there are not significant variables
   selection.bonf <- c(); forest.bonf <- c()}
  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))
  oob.error.bonf <- ifelse(length(selection.bonf) != 0,
                    mean((as.numeric(as.character(Y)) -
                    as.numeric(as.character(predict(forest.bonf, 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,
              "selection.bonf" = selection.bonf, "forest.bonf" = forest.bonf,
              "oob.error.bonf" = oob.error.bonf))
}

########################
### The ALT approach ###
########################
ALT <- function(Y, X, nperm = 100, ntree = 50, alpha = 0.05) {
  # Y: response vector
  # X: matrix or data frame containing the predictors
  # nperm: number of permutations
  # ntree: number of trees contained in a Random Forest
  # alpha: alpha level for permutation tests
  # RETURNS: selected variables, a corresponding Random Forest and OOB-error
  mtry <- ceiling(sqrt(ncol(X))) # automatically set mtry to sqrt(p)
  dat <- cbind(Y, X) # create the data
  names(dat) <- c("response", paste("V", 1:ncol(X), sep = ""))
  forest <- cforest(response ~ ., data = dat, # fit a forest
                        controls = cforest_unbiased(mtry = mtry, ntree = ntree))
  obs.varimp <- varimp(forest, pre1.0_0 = T) # compute initial importances
  selection <- names(obs.varimp)
  # create a matrix that contains the variable importances after permutation
  perm.mat <- matrix(NA, ncol = length(selection), nrow = nperm,
                                            dimnames = list(1:nperm, selection))
  perm.dat <- dat # perm.dat will be the data after permutation
  for (i in 1:nperm) { # do nperm permutations
    perm.dat[, "response"] <- sample(perm.dat[, "response"]) #  permute the response
    perm.forest <- cforest(response ~ ., data = perm.dat,    # recompute the forests
                           controls = cforest_unbiased(mtry = mtry, ntree = ntree))
    perm.mat[i, ] <- varimp(perm.forest, pre1.0_0 = T)} # recompute variable importances
  p.vals <- sapply(selection, function(x) sum(perm.mat[, x] # compute p-values
                                                      >= obs.varimp[x]) / nperm)
  if (any(p.vals < alpha)) { # keep significant variables
   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))}
  if (!any(p.vals < alpha)) { # if there are not significant variables
   selection <- c(); forest <- c()}
  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))
}

############################################
### The J.0, J.1, D.0 and D.1 approaches ###
############################################
Diaz <- function(Y, X, recompute = F, ntree = 3000) {
  # Y: response vector
  # X: matrix or data frame containing the predictors
  # recompute: should the variable importances be recomputed after each
  #            regection step? TRUE produces J.0 and J.1
  # ntree: number of trees contained in a Random Forest
  # RETURNS: selected variables, a corresponding Random Forest and OOB-error
  #          for the 0 s.e. and 1 s.e. rule
  mtry <- ceiling(sqrt(ncol(X))) # automatically set mtry to sqrt(p)
  dat <- cbind(Y, X) # create the data
  names(dat) <- c("response", paste("V", 1:ncol(X), sep = ""))
  forest <- cforest(response ~ ., data = dat, # fit a forest
                        controls = cforest_unbiased(mtry = mtry, ntree = ntree))
  selections <- list() # a list that contains the sequence of selected variables
  selections[[ncol(X)]] <- names(sort(varimp(forest, pre1.0_0 = T), decreasing = T))
  errors <- c()
  for (i in ncol(X):1) { # take backward rejection steps
  mtry <- ceiling(sqrt(i)) # set mtry to sqrt() of remaining variables
  forest <- cforest(as.formula(paste("response", paste(selections[[i]],
                    collapse = " + "), sep = " ~ ")), data = dat, # fit forest
                    controls = cforest_unbiased(mtry = mtry, ntree = ntree))
  errors[i] <- mean((as.numeric(as.character(Y)) - # compute the OOB-error
                     as.numeric(as.character(predict(forest, OOB = T))))^2)
  # define the next set of variables
  if (recompute == F & i > 1) selections[[i - 1]] <- selections[[i]][-i]
  if (recompute == T & i > 1) selections[[i - 1]] <- names(sort(varimp(forest,
                                                      pre1.0_0 = T), decreasing = T))[-i]}
  # compute the error expected when no predictor is used at all
  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)
  # define the number of variables determined by the 0 s.e. and 1 s.e. rule
  optimum.number.0se <- which.min(errors)
  optimum.number.1se <- which(errors <= min(errors) + 1 * ifelse(all(Y %in% 0:1),
                          sqrt(min(errors) * (1 - min(errors)) / nrow(X)), 0))[1]
  # compute the corresponding Random Forests and OOB-errors
  if (optimum.number.0se == 1) {forest.0se <- c(); selection.0se <- c()}
  if (optimum.number.1se == 1) {forest.1se <- c(); selection.1se <- c()}
  if (optimum.number.0se != 1) {
  selection.0se <- selections[[optimum.number.0se - 1]]
  forest.0se <- cforest(as.formula(paste("response", paste(selection.0se,
                        collapse = " + "), sep = " ~ ")), data = dat,
                        controls = cforest_unbiased(mtry = mtry, ntree = ntree))}
  if (optimum.number.1se != 1) {
  selection.1se <- selections[[optimum.number.1se - 1]]
  forest.1se <- cforest(as.formula(paste("response", paste(selection.1se,
                        collapse = " + "), sep = " ~ ")), data = dat,
                        controls = cforest_unbiased(mtry = mtry, ntree = ntree))}
  oob.error.0se <- errors[optimum.number.0se]
  oob.error.1se <- errors[optimum.number.1se]
  return(list("selection.0se" = selection.0se, "forest.0se" = forest.0se,
              "oob.error.0se" = oob.error.0se, "selection.1se" = selection.1se,
              "forest.1se" = forest.1se, "oob.error.1se" = oob.error.1se))
}

########################
### The SVT approach ###
########################
SVT <- function(Y, X, ntree = 50, folds = 5, repetitions = 20) {
  # Y: response vector
  # X: matrix or data frame containing the predictors
  # ntree: number of trees contained in a Random Forest
  # folds: determines ’folds’-fold cross validation
  # repetitions: the results of ’repetitons’ repetitons should be aggregated
  # RETURNS: selected variables, a corresponding Random Forest and OOB-error
  mtry <- ceiling(sqrt(ncol(X))) # automatically set mtry to sqrt(p)
  dat <- cbind(Y, X) # create the data
  names(dat) <- c("response", paste("V", 1:ncol(X), sep = ""))
  forest <- cforest(response ~ ., data = dat, # fit a Random Forest
                        controls = cforest_unbiased(mtry = mtry, ntree = ntree))
  final.imps <- names(sort(varimp(forest, pre1.0_0 = T), decreasing = T)) # final sequence
  errors <- array(NA, dim = c(repetitions, ncol(X) + 1, folds))
  for (x in 1:repetitions) { # repeatedly produce results of several...
    samps <- sample(rep(1:folds, length = nrow(X)))
    for (k in 1:folds) { # ...crossvalidations
      train <- dat[samps != k, ]; test <- dat[samps == k, ] # train and test data
      forest <- cforest(response ~ ., data = train, # fit a Random Forest
                        controls = cforest_unbiased(mtry = mtry, ntree = ntree))
      selection <- names(sort(varimp(forest, pre1.0_0 = T), decreasing = T))
      for (i in ncol(X):1) { # do backward rejection steps
        mtry <- min(mtry, ceiling(sqrt(i)))
        forest <- cforest(as.formula(paste("response", paste(selection[1:i],
                          collapse = " + "), sep = " ~ ")), data = train,
                          controls = cforest_unbiased(mtry = mtry, ntree = ntree))
        errors[x, i + 1, k] <- mean((as.numeric(as.character(test$response)) -
                   as.numeric(as.character(predict(forest, newdata = test))))^2)}
      errors[x, 1, k] <- mean((as.numeric(as.character(test$response)) -
                     ifelse(all(Y %in% 0:1), round(mean(as.numeric(
                     as.character(train$response)))), mean(train$response)))^2)}}
  mean.errors <- sapply(1:(ncol(X) + 1), function(x) mean(errors[, x, ]))
  optimum.number <- which.min(mean.errors)   # optimal number of variables
  if (optimum.number == 1) { # determine the final forest, selection and OBB-error
   forest <- c(); selection <- c()}
  if (optimum.number != 1) {
  selection <- final.imps[1:(optimum.number - 1)]
  forest <- cforest(as.formula(paste("response", paste(selection,
                    collapse = " + "), sep = " ~ ")), data = dat,
                    controls = cforest_unbiased(mtry = mtry, ntree = ntree))}
  error <- mean.errors[optimum.number]
  return(list("selection" = selection, "forest" = forest, "error" = error))
}

##################################
### The G.i and G.p approaches ###
##################################
Gen <- function(Y, X, ntree = 50, se.rule = 1, repetitions = 50) {
  # Y: response vector
  # X: matrix or data frame containing the predictors
  # ntree: number of trees contained in a Random Forest
  # se.rule: kind of s.e. rule used; e.g. = 1 equals the 1 s.e. rule
  # repetitions: the results of 'repetitons' repetitons should be aggregated
  # RETURNS: selected variables, a corresponding Random Forest and OOB-error
  mtry <- ceiling(sqrt(ncol(X))) # automatically set mtry to sqrt(p)
  dat <- cbind(Y, X) # create the data
  names(dat) <- c("response", paste("V", 1:ncol(X), sep = ""))
  rankings <- matrix(NA, nrow = repetitions, ncol = ncol(X),
                               dimnames = list(1:repetitions, names(dat)[-1]))
  for (i in 1:repetitions) { # repeatedly assess ranking of variable importances
    forest <- cforest(response ~ ., data = dat,
                        controls = cforest_unbiased(mtry = mtry, ntree = ntree))
    rankings[i, ] <- varimp(forest, pre1.0_0 = T)}
  selection <- names(sort(colMeans(rankings), decreasing = T))
  errors <- matrix(NA, nrow = repetitions, ncol = ncol(X) + 1)
  for (i in 1:ncol(X)) { # do forward selection steps based on the ranking
  mtry <- min(mtry, ceiling(sqrt(i)))
  for (j in 1:repetitions) { # also repeat the computation of OOB-errors
  forest <- cforest(as.formula(paste("response", paste(selection[1:i],
                    collapse = " + "), sep = " ~ ")), data = dat,
                    controls = cforest_unbiased(mtry = mtry, ntree = ntree))
  errors[j, i + 1] <- mean((as.numeric(as.character(Y)) -
                        as.numeric(as.character(predict(forest, OOB = T))))^2)}}
  errors[, 1] <- mean((as.numeric(as.character(Y)) - # error with no predictors
  ifelse(all(Y %in% 0:1), round(mean(as.numeric(as.character(Y)))), mean(Y)))^2)
  mean.errors <- colMeans(errors); sd.errors <- apply(errors, 2, sd)
  optimum.number <- which(mean.errors <= # optimal number using the s.e. rule
              min(mean.errors) + se.rule * sd.errors[which.min(mean.errors)])[1]
  if (optimum.number == 1) { # determine the model for interpretation
   selection.int <- c(); forest.int <- c()}
  if (optimum.number != 1) {
   selection.int <- selection[1:(optimum.number - 1)]
   forest.int <- cforest(as.formula(paste("response", paste(selection.int,
                         collapse = " + "), sep = " ~ ")), data = dat,
                         controls = cforest_unbiased(mtry = mtry, ntree = ntree))}
  # determine the threshold to be exceeded for inclusion into the prediction model
  steps <- sapply(2:(ncol(X) + 1), function(x) mean.errors[x - 1] - mean.errors[x])
  threshold <- sum(abs(steps[optimum.number:length(steps)])) /
                                                ((ncol(X) + 1) - optimum.number)
  best <- which(steps <= threshold)[1] # optimal size for the prediction model
  if (best == 1) {selection.pred <- c(); forest.pred <- c()}
  if (best != 1) {selection.pred <- selection[1:(best - 1)]
   forest.pred <- cforest(as.formula(paste("response", paste(selection.pred,
                          collapse = " + "), sep = " ~ ")), data = dat,
                          controls = cforest_unbiased(mtry = mtry, ntree = ntree))}
  oob.error.int <- mean.errors[optimum.number]; oob.error.pred <- mean.errors[best]
  return(list("selection.int" = selection.int, "selection.pred" = selection.pred,
              "forest.int" = forest.int, "forest.pred" = forest.pred,
              "oob.error.int" = oob.error.int, "oob.error.pred" = oob.error.pred))
  }
```

B.5.2 Simulation Studies

Study I

R-Code used for simulation study I:

```
# load required packages
```

```
library("party"); attach(asNamespace("party")) # version 1.0-0
library(mvtnorm) # version 0.9-9992

myfunc <- function(reg = F) { # function to apply variable selection to the data
Sigma <- diag(5) # define sigma and produce the data
dat.mod <- create.dat(c(0,0,0,0,0), n = 100, sigma = Sigma, regression = reg)
# apply each method with the specified settings and save selection frequencies
N <- NAP(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                         nperm = 400, ntree = 100, alpha = 0.05)
Nap <- N$selection; Nap.b <- N$selection.bonf
G <- Gen(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                     ntree = 100, se.rule = 1, repetitions = 50)
G.i <- G$selection.int; G.p <- G$selection.pred
J <- Diaz(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                                     recompute = T, ntree = 100)
J.0 <- J$selection.0se; J.1 <- J$selection.1se
D <- Diaz(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                                     recompute = F, ntree = 100)
D.0 <- D$selection.0se; D.1 <- D$selection.1se
Alt <- ALT(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                               nperm = 400, ntree = 100, alpha = 0.05)$selection
svt <- SVT(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                             ntree = 100, folds = 5, repetitions = 20)$selection
return(list(NAP = Nap, NAP.B = Nap.b, G.i = G.i, G.p = G.p, J.0 = J.0, J.1 = J.1,
            D.0 = D.0, D.1 = D.1, ALT = Alt, SVT = svt))
}

set.seed(65) # set random seed for reproducibility of results
result.reg <-  lapply(1:5000, function(x) myfunc(reg = T)) # regression
result.clas <- lapply(1:5000, function(x) myfunc(reg = F)) # classification
```

Study II

R-Code used for simulation study II (set `regression = T` in the function `create.dat()` to produce results for the regression problem):

```
# load required packages
library("party"); attach(asNamespace("party")) # version 1.0-0
library(mvtnorm) # version 0.9-9992

myfunc <- function(x) { # function to apply variable selection to the data
Nap <- Nap.b <- G.i <- G.p <- J.0 <- J.1 <-
D.0 <- D.1 <- Alt <- svt <- vector("list", length = 11)
# create the covariance matrix
Sigma <- diag(6); Sigma[2, 4] <- Sigma[4, 2] <- .7; Sigma[3, 5] <- Sigma[5, 3] <- .7

for (s in seq(0, 1, .1)) { # vary the strength of coefficients
dat.mod <- create.dat(c(s, s, s, 1, 0, 0), n = 100, sigma = Sigma, regression = F)
# apply each method with the specified settings and save the selection frequencies
N <- NAP(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                   nperm = 400, ntree = 100, alpha = 0.05)
Nap[[which(seq(0, 1, .1) == s)]]   <- N$selection
Nap.b[[which(seq(0, 1, .1) == s)]] <- N$selection.bonf
G <- Gen(dat.mod$response, dat.mod[, which(names(dat.mod) !=  "response")],
                               ntree =  100, se.rule = 1, repetitions = 50)
G.i[[which(seq(0, 1, .1) == s)]] <- G$selection.int
G.p[[which(seq(0, 1, .1) == s)]] <- G$selection.pred
J <- Diaz(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                                recompute = T, ntree = 100)
J.0[[which(seq(0, 1, .1) == s)]] <- J$selection.0se
J.1[[which(seq(0, 1, .1) == s)]] <- J$selection.1se
D <- Diaz(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                                recompute = F, ntree = 100)
D.0[[which(seq(0, 1, .1) == s)]] <- D$selection.0se
D.1[[which(seq(0, 1, .1) == s)]] <- D$selection.1se
Alt[[which(seq(0, 1, .1) == s)]] <-
            ALT(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                nperm = 400, ntree = 100, alpha = 0.05)$selection
svt[[which(seq(0, 1, .1) == s)]] <-
            SVT(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                              ntree = 100, folds = 5, repetitions = 20)$selection
}
return(list(NAP = Nap, NAP.B = Nap.b, G.i = G.i, G.p = G.p, J.0 = J.0, J.1 = J.1,
            D.0 = D.0, D.1 = D.1, ALT = Alt, SVT = svt))
}

set.seed(1234) # set random seed for reproducibility of results
result <- lapply(1:1000, function(x) myfunc(x = x))
```

Study III

R-Code used for simulation study III (set `regression = T` in the function `create.dat()` to produce results for the regression problem):

```
# load required packages
library("party"); attach(asNamespace("party")) # version 1.0-0
library(mvtnorm) # version 0.9-9992

sigma <- diag(20); sigma[4:6, 4:6] <- .7 # create the covariance matrix
sigma[7:11, 7:11] <- .7; sigma[12:13, 12:13] <- .7; diag(sigma) <- 1

# function to assess selection frequencies and prediction errors
myfunc <- function(dat.test, sigma) {
dat.mod <- create.dat(coefs = c(3,2,1,3,2,1,3,2,1,0,0,0,0,0,0,0,0,0,0,0),
                                  n = 100, sigma = sigma, regression = F)
# apply each method and save selection frequencies and prediction errors
N <- NAP(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                         nperm = 400, ntree = 100, alpha = 0.05)
Nap <- N$selection; Nap.b <- N$selection.bonf
Nap.error <- mean((as.numeric(dat.test$response) -
                           as.numeric(predict(N$forest, newdata = dat.test)))^2)
Nap.b.error <- mean((as.numeric(dat.test$response) -
                      as.numeric(predict(N$forest.bonf, newdata = dat.test)))^2)
G <- Gen(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                     ntree = 100, se.rule = 1, repetitions = 50)
G.i <- G$selection.int; G.p <- G$selection.pred
G.i.error <- mean((as.numeric(dat.test$response) -
                       as.numeric(predict(G$forest.int, newdata = dat.test)))^2)
G.p.error <- mean((as.numeric(dat.test$response) -
                      as.numeric(predict(G$forest.pred, newdata = dat.test)))^2)
J <- Diaz(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                                     recompute = T, ntree = 100)
J.0 <- J$selection.0se; J.1 <- J$selection.1se
J.0.error <- mean((as.numeric(dat.test$response) -
                       as.numeric(predict(J$forest.0se, newdata = dat.test)))^2)
J.1.error <- mean((as.numeric(dat.test$response) -
                       as.numeric(predict(J$forest.1se, newdata = dat.test)))^2)
D <- Diaz(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                                     recompute = F, ntree = 100)
D.0 <- D$selection.0se; D.1 <- D$selection.1se
D.0.error <- mean((as.numeric(dat.test$response) -
                       as.numeric(predict(D$forest.0se, newdata = dat.test)))^2)
D.1.error <- mean((as.numeric(dat.test$response) -
                       as.numeric(predict(D$forest.1se, newdata = dat.test)))^2)
A <- ALT(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                         nperm = 400, ntree = 100, alpha = 0.05)
Alt <- A$selection
Alt.error <- mean((as.numeric(dat.test$response) -
                           as.numeric(predict(A$forest, newdata = dat.test)))^2)
S <- SVT(dat.mod$response, dat.mod[, which(names(dat.mod) != "response")],
                                       ntree = 100, folds = 5, repetitions = 20)
svt <- S$selection
svt.error <- mean((as.numeric(dat.test$response) -
                           as.numeric(predict(S$forest, newdata = dat.test)))^2)
Fo <- cforest(response ~ ., data = dat.mod, controls = cforest_unbiased(mtry =
                                 ceiling(sqrt(ncol(dat.mod) - 1)), ntree = 100))
error <- mean((as.numeric(dat.test$response) -
                                 as.numeric(predict(Fo, newdata = dat.test)))^2)

return(list(NAP = Nap, NAP.error = Nap.error, NAP.B = Nap.b, NAP.B.error = Nap.b.error,
G.i = G.i, G.p = G.p, G.i.error = G.i.error, G.p.error = G.p.error,
J.0 = J.0, J.0.error = J.0.error, J.1 = J.1, J.1.error = J.1.error,
D.0 = D.0, D.0.error = D.0.error, D.1 = D.1, D.1.error = D.1.error,
ALT = Alt, ALT.error = Alt.error, SVT = svt, SVT.error = svt.error, error.full = error))
}

set.seed(1234) # set random seed for reproducibility of results
# this is the data used for validation
mydat.test <- create.dat(coefs = c(3,2,1,3,2,1,3,2,1,0,0,0,0,0,0,0,0,0,0,0),
                                          n = 5000, sigma = sigma, regression = F)
result <- lapply(1:1000, function(x) myfunc(dat.test = mydat.test, sigma = sigma))
```

B.5.3 Empirical Evaluation

R-Code used for the empirical evaluation of the Infant Birth Weight Data (other datasets were processed the same way):

```
# load required packages
library("party"); attach(asNamespace("party")) # version 1.0-0
library(MASS) # version 7.3-17

dat <- birthwt # initialize the data
names(dat) <- c(paste("V", 1:9, sep = ""), "bwt")

# function to assess selection frequencies and prediction errors
myfunc <- function(data) {
samp <- sample(1:nrow(data), nrow(data), replace = T) # sampling step
dat.mod <- data[samp, ]; dat.test <- data[-samp, ] # test and training data
# apply each method and save the selection frequencies and prediction errors
N <- NAP(dat.mod$bwt, dat.mod[, which(names(dat.mod) != "bwt")],
                                          nperm = 400, ntree = 100, alpha = 0.05)
Nap <- N$selection; Nap.b <- N$selection.bonf
Nap.error <- mean((as.numeric(dat.test$bwt) -
                            as.numeric(predict(N$forest, newdata = dat.test)))^2)
Nap.b.error <- mean((as.numeric(dat.test$bwt) -
                       as.numeric(predict(N$forest.bonf, newdata = dat.test)))^2)
G <- Gen(dat.mod$bwt, dat.mod[, which(names(dat.mod) != "bwt")],
                                      ntree = 100, se.rule = 1, repetitions = 50)
G.i <- G$selection.int; G.p <- G$selection.pred
G.i.error <- mean((as.numeric(dat.test$bwt) -
                        as.numeric(predict(G$forest.int, newdata = dat.test)))^2)
G.p.error <- mean((as.numeric(dat.test$bwt) -
                       as.numeric(predict(G$forest.pred, newdata = dat.test)))^2)
J <- Diaz(dat.mod$bwt, dat.mod[, which(names(dat.mod) != "bwt")], recompute = T,
                                                                     ntree = 100)
J.0 <- J$selection.0se; J.1 <- J$selection.1se
J.0.error <- mean((as.numeric(dat.test$bwt) -
                        as.numeric(predict(J$forest.0se, newdata = dat.test)))^2)
J.1.error <- mean((as.numeric(dat.test$bwt) -
                        as.numeric(predict(J$forest.1se, newdata = dat.test)))^2)
D <- Diaz(dat.mod$bwt, dat.mod[, which(names(dat.mod) != "bwt")], recompute = F,
                                                                     ntree = 100)
D.0 <- D$selection.0se; D.1 <- D$selection.1se
D.0.error <- mean((as.numeric(dat.test$bwt) -
                        as.numeric(predict(D$forest.0se, newdata = dat.test)))^2)
D.1.error <- mean((as.numeric(dat.test$bwt) -
                        as.numeric(predict(D$forest.1se, newdata = dat.test)))^2)
A <- ALT(dat.mod$bwt, dat.mod[, which(names(dat.mod) != "bwt")], nperm = 400,
                                                       ntree = 100, alpha = 0.05)
Alt <- A$selection
Alt.error <- mean((as.numeric(dat.test$bwt) -
                            as.numeric(predict(A$forest, newdata = dat.test)))^2)
S <- SVT(dat.mod$bwt, dat.mod[, which(names(dat.mod) != "bwt")],
                                        ntree = 100, folds = 5, repetitions = 20)
svt <- S$selection
svt.error <- mean((as.numeric(dat.test$bwt) -
                           as.numeric(predict(S$forest, newdata = dat.test)))^2)
error.all <- mean((as.numeric(dat.test$bwt) - as.numeric(predict(
             cforest(bwt ~ ., data = dat.mod, controls = cforest_unbiased(mtry =
       ceiling(sqrt(ncol(dat.mod) - 1)), ntree = 100)), newdata = dat.test)))^2)
return(list(NAP = Nap, NAP.error = Nap.error, NAP.B = Nap.b, NAP.B.error = Nap.b.error,
G.i = G.i, G.p = G.p, G.i.error = G.i.error, G.p.error = G.p.error,
J.0 = J.0, J.0.error = J.0.error, J.1 = J.1, J.1.error = J.1.error,
D.0 = D.0, D.0.error = D.0.error, D.1 = D.1, D.1.error = D.1.error,
ALT = Alt, ALT.error = Alt.error, SVT = svt, SVT.error = svt.error, error.all = error.all))
}

set.seed(1234) # set random seed for reproducibility of results
result <- lapply(1:1000, function(x) myfunc(data = dat))
```

B.6 Chapter 6

R-Code used for the simulation studies of chapter 6:

```
# load required packages
library("party"); attach(asNamespace("party")) # version 1.0-0
library(mvtnorm) # version 0.9-9992
library(mice)    # version 2.11

sig <- matrix(c(  1, 0.3, 0.3, 0, 0, 0, # create covariance matrix
                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), ncol = 6, byrow = T)

# function used for the simulation analysis
myfunc <- function(dat.test, sigma) {
# dat.test: data frame used for the assessment of a models error
# sigma: covariance matrix used to build the training data
dat.train <- create.dat(n = 100, sigma = sigma, regression = F) # training data
dat.mis <- lapply(1, function(x) dat.train)
# lists that will contain the variable selections and corresponding errors
# of the test-based (‘TB’) and performance-based (‘PB’) approaches
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)))
  for (m in 1:4) { # induce 4 fractions of missing values
    dat.mis <- with.missings(dat, c("V2", "V5"), c("V1", "V4"), m - 1)
  for (j in 1:length(dat.mis)) { # there are 6 missing data generating processes
    train.response <- dat.mis[[j]]$response
    mean.response <- ifelse(all(train.response %in% 0:1), round(mean(
                            as.numeric(train.response))), mean(train.response))
    test.response <- as.numeric(dat.test$response)
    input <- which(names(dat.train) != "response")
    errorfunc <- function(a, b) {mean((a  -  as.numeric(predict(b, newdata = dat.test)))^2)}
    # perform variable selection
    Tb <- NAP(train.response, dat.mis[[j]][, input], nperm = 100, ntree = 100, alpha = 0.05)
    TB[[j]][Tb$selection, m, 1] <- ifelse(is.null(Tb$selection), 0, 1)
    TB.error[[j]][1, m] <- ifelse(is.null(Tb$selection), mean((test.response -
                                     mean.response)^2), errorfunc(test.response, Tb$forest))
    Pb <- Diaz(train.response, dat.mis[[j]][, input], ntree = 100)
    PB[[j]][Pb$selection, m, 1] <- ifelse(is.null(Pb$selection.1se), 0, 1)
    PB.error[[j]][1, m] <- ifelse(is.null(Pb$selection.1se), mean((test.response -
                                     mean.response)^2), errorfunc(test.response, Pb$forest))
    if (m > 1) {   # perform variable selection with a complete case analysis
    Tb <- NAP(na.omit(dat.mis[[j]])$response, na.omit(dat.mis[[j]])[, input],
                                      nperm = 100, ntree = 100, alpha = 0.05)
    TB[[j]][Tb$selection, m, 2] <- ifelse(is.null(Tb$selection), 0, 1)
    TB.error[[j]][2, m] <- ifelse(is.null(Tb$selection), mean((test.response -
                                    mean.response)^2), errorfunc(test.response, Tb$forest))
    Pb <- Diaz(na.omit(dat.mis[[j]])$response, na.omit(dat.mis[[j]])[, input], ntree = 100)
    PB[[j]][Pb$selection.1se, m, 2] <- ifelse(is.null(Pb$selection.1se), 0, 1)
    PB.error[[j]][2, m] <- ifelse(is.null(Pb$selection.1se), mean((test.response -
                                    mean.response)^2), errorfunc(test.response, Pb$forest))
    # 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) NAP(complete(imp.dat, action = x)$response,
           complete(imp.dat, action = x)[, input], nperm = 100, ntree = 100, alpha = 0.05))
    TB[[j]][, m, 3] <- rowSums(sapply(Tb, function(x)    table(x$selection)[paste("V", 1:6,
                                                                sep = "")]), na.rm = T) / 5
    TB.error[[j]][3, m] <- mean(sapply(Tb, function(x) { if (!is.null(x$selection)) {
                                errorfunc(test.response, x$forest)}
                                else mean((test.response - mean.response)^2)}))
    Pb <- lapply(1:5, function(x) Diaz(complete(imp.dat, action = x)$response,
                                complete(imp.dat, action = x)[, input], ntree = 100))
    PB[[j]][, m, 3] <- rowSums(sapply(Pb, function(x)   table(x$selection)[paste("V",
                                                     1:6, sep = "")]), na.rm = T) / 5
    PB.error[[j]][3, m] <- mean(sapply(Pb, function(x) { if (!is.null(x$selection)) {
                                errorfunc(test.response, x$forest)}
                                else mean((test.response - mean.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, Data.Driven = PB,
            Data.Driven.error = PB.error))
}

set.seed(1234) # set a random seed for reproducibility of results
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))
```

Bibliography

T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, 2010. doi: 10.1093/bioinformatics/btp630. URL http://bioinformatics.oxfordjournals.org/content/26/3/392.abstract.

T. Allison and D. V. Cicchetti. Sleep in mammals: ecological and constitutional correlates. *Science*, 194(4266):732–734, 1976. doi: 10.1126/science.982039. URL http://www.sciencemag.org/cgi/content/abstract/194/4266/732.

A. Altmann, L. Tolosi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010. ISSN 1367-4811. doi: 10.1093/bioinformatics/btq134. URL http://dx.doi.org/10.1093/bioinformatics/btq134.

K. Archer and R. Kimes. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52(4):2249–2260, 2008. ISSN 01679473. doi: 10.1016/j.csda.2007.08.015. URL http://dx.doi.org/10.1016/j.csda.2007.08.015.

P. C. Austin and J. V. Tu. Bootstrap methods for developing predictive models. *The American Statistician*, 58(2):131–137, 2004. ISSN 0003-1305. doi: 10.1198/0003130043277. URL http://www.jstor.org/stable/27643521.

R. A. Becker. *S: An Interactive Environment for Data Analysis and Graphics (His Competencies for Teaching; V. 3)*. Chapman & Hall/CRC, 1984.

Y. Benjamini and Y. Hochberg. 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, 1995. ISSN 00359246. doi: 10.2307/2346101. URL http://dx.doi.org/10.2307/2346101.

Y. Benjamini and D. Yekutieli. The Control of the False Discovery Rate in Multiple Testing under Dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001. ISSN 00905364. doi: 10.2307/2674075. URL http://dx.doi.org/10.2307/2674075.

G. Biau, L. Devroye, and G. Lugosi. Consistency of Random Forests and Other Averaging Classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008. URL http://www.jmlr.org/papers/volume9/biau08a/biau08a.pdf.

M. Bland. *An Introduction to Medical Statistics (Oxford Medical Publications)*. Oxford University Press, USA, 2000. ISBN 0192632698. URL http://www.worldcat.org/isbn/0192632698.

A.-L. Boulesteix, C. Strobl, T. Augustin, and M. Daumer. Evaluating microarray-based classifiers: An overview. *Cancer Informatics*, 6:77–97, 2008a.

A.-L. Boulesteix, C. Strobl, T. Augustin, and M. Daumer. Evaluating microarray-based classifiers: An overview. *Canc Informat*, 6:77–97, 2008b.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi: 10.1023/A:1018054314350. URL http://dx.doi.org/10.1023/A:1018054314350.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 08856125. doi: 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A:1010933404324.

L. Breiman and A. Cutler. *Random forests*. http://www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm, 2008. (accessed 03.02.2012).

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Chapman & Hall/CRC, 1 edition, 1984. ISBN 0412048418. URL http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/0412048418.

P. Bühlmann and B. Yu. Analyzing bagging. *Annals of Statistics*, 30(4):927–961, 2002.

L. F. Burgette and J. P. Reiter. Multiple imputation for missing data via sequential regression trees. *American Journal of Epidemiology*, 172(9):1070–1076, 2010. doi: 10.1093/aje/kwq260. URL http://aje.oxfordjournals.org/content/172/9/1070.abstract.

J. M. Chambers. *Graphical Methods for Data Analysis (Statistics)*. Chapman & Hall/CRC, 1983. ISBN 0412052717. URL http://www.worldcat.org/isbn/0412052717.

N. Chehata, L. Guo, and C. Mallet. Airborne lidar feature selection for urban classification using random forests. *Scanning*, XXXVIII(c):207–212, 2009. URL http://www.mendeley.com/research/airborne-lidar-feature-selection-urban-classification-using-random-forests/.

X. Chen, M. Wang, and H. Zhang. The use of classification trees for bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):55–63, 2011. ISSN 1942-4795. doi: 10.1002/widm.14. URL http://dx.doi.org/10.1002/widm.14.

D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007. doi: 10.1890/07-0539.1. URL http://www.esajournals.org/doi/abs/10.1890/07-0539.1.

R. Díaz-Uriarte and S. Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3, 2006. ISSN 1471-2105. doi: 10.1186/1471-2105-7-3. URL http://www.biomedcentral.com/1471-2105/7/3.

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

B. Efron. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983. ISSN 01621459. doi: 10.2307/2288636. URL http://dx.doi.org/10.2307/2288636.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 1 edition, 1994. ISBN 0412042312. URL http://www.worldcat.org/isbn/0412042312.

B. Efron and R. Tibshirani. Improvements on Cross-Validation: The .632+ Bootstrap Method. *Journal of the American Statistical Association*, 92(438):548–560, 1997. ISSN 01621459. doi: 10.2307/2965703. URL http://dx.doi.org/10.2307/2965703.

M. Elter, R. Schulz-Wendtland, and T. Wittenberg. The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Medical Physics*, 34(11):4164–4172, 2007. doi: 10.1118/1.2786864. URL http://link.aip.org/link/?MPH/34/4164/1.

A. Farhangfar, L. Kurgan, and J. Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recogn.*, 41(12):3692–3705, 2008. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/j.patcog.2008.05.019.

A. J. Feelders. Handling missing data in trees: Surrogate splits or statistical imputation. In *PKDD '99: Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, pages 329–334, London, UK, 1999. Springer-Verlag. ISBN 3-540-66490-4.

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL http://archive.ics.uci.edu/ml.

R. Genuer. Risk bounds for purely uniformly random forests. Rapport de recherche RR-7318, INRIA, 2010. URL http://hal.inria.fr/inria-00492231/en/.

R. Genuer, J.-M. Poggi, and C. Tuleau. Random Forests: some methodological insights. Rapport de recherche RR-6729, INRIA, 2008. URL http://hal.inria.fr/inria-00340725/en/.

R. Genuer, V. Michel, E. Eger, and B. Thirion. Random forests based feature selection for decoding fmri data. In *Proceedings Compstat 2010*, number 267, pages 1 – 8, 2010a. Paris, France.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225 – 2236, 2010b. ISSN 0167-8655. doi: DOI:10.1016/j.patrec.2010.03.014. URL http://www.sciencedirect.com/science/article/B6V15-4YNC1M2-2/2/933ac5ac7bf3d118fbaa2313fe369439.

R. Genuer, I. Morlais, and W. Toussile. Gametocytes infectiousness to mosquitoes: variable selection using random forests, and zero inflated models. Research Report RR-7497, INRIA, 2011. URL http://hal.inria.fr/inria-00550980/en/.

P. Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer, 2nd edition, 2000. ISBN 038798898X. URL http://www.worldcat.org/isbn/038798898X.

P. Good. *Introduction to Statistics through Resampling Methods and R/S-Plus*. Wiley-Interscience, New York, 2005. ISBN 0471715751.

R. H. H. Groenwold, A. R. T. Donders, K. C. B. Roes, F. E. Harrell, and K. G. M. Moons. Dealing with missing outcome data in randomized trials and observational studies. *American Journal of Epidemiology*, 175(3):210–217, 2012. doi: 10.1093/aje/kwr302. URL http://aje.oxfordjournals.org/content/175/3/210.abstract.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003. ISSN 1532-4435. URL http://portal.acm.org/citation.cfm?id=944919.944968.

S. J. Haberman. Generalized residuals for log-linear models. In *Proceedings of the 9th International Biometrics Conference*, pages 104–122, 1976.

A. Hapfelmeier and A. Horsch. Image feature evaluation in two new mammography cad prototypes. *International Journal of Computer Assisted Radiology and Surgery*, 6:721–735, 2011. ISSN 1861-6410. URL http://dx.doi.org/10.1007/s11548-011-0549-5. 10.1007/s11548-011-0549-5.

A. Hapfelmeier and K. Ulm. A new variable selection approach using random forests. *Computational Statistics & Data Analysis*, (0):–, 2012. ISSN 0167-9473. doi: 10.1016/j.csda.2012.09.020. URL http://www.sciencedirect.com/science/article/pii/S0167947312003490?v=s5.

A. Hapfelmeier, T. Hothorn, and K. Ulm. Recursive partitioning on incomplete data using surrogate decisions and multiple imputation. *Computational Statistics & Data Analysis*, (0):–, 2011. ISSN 0167-9473. doi: 10.1016/j.csda.2011.09.024. URL http://www.sciencedirect.com/science/article/pii/S0167947311003550.

A. Hapfelmeier, T. Hothorn, and K. Ulm. Random forest variable importance with missing data. 2012a. URL http://epub.ub.uni-muenchen.de/12757/.

A. Hapfelmeier, T. Hothorn, K. Ulm, and C. Strobl. A new variable importance measure for random forests with missing data. Accepted by Statistics and Computing, 2012b.

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

D. J. Harrison and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978. URL http://ideas.repec.org/a/eee/jeeman/v5y1978i1p81-102.html.

T. Hastie, R. Tibshirani, M. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. Chan, D. Botstein, and P. Brown. 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2):research0003.1–research0003.21, 2000. ISSN 1465-6906. doi: 10.1186/gb-2000-1-2-research0003. URL http://dx.doi.org/10.1186/gb-2000-1-2-research0003.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical learning*. Springer, corrected edition, 2009.

Y. He, A. M. Zaslavsky, M. B. Landrum, D. P. Harrington, and P. Catalano. Multiple imputation in a large-scale complex survey: a practical guide. *Statistical Methods in Medical Research*, 2009. doi: 10.1177/0962280208101273. URL http://smm.sagepub.com/cgi/content/abstract/0962280208101273v1.

S. G. Hilsenbeck and G. M. Clark. Practical p-value adjustment for optimally selected cutpoints. *Statistics in Medicine*, 15(1):103–112, 1996.

N. J. Horton and K. P. Kleinman. 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, 2007. ISSN 0003-1305. doi: 10.1198/000313007X172556. URL http://dx.doi.org/10.1198/000313007X172556.

T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006. doi: 10.1198/106186006X133933. URL http://pubs.amstat.org/doi/abs/10.1198/106186006X133933.

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

K. J. Janssen, Y. Vergouwe, A. R. Donders, F. E. Harrell, Q. Chen, D. E. Grobbee, and K. G. Moons. Dealing with missing predictor values when applying clinical prediction models. *Clinical chemistry*, 55(5):994–1001, 2009. ISSN 1530-8561. doi: 10.1373/clinchem.2008.115345. URL http://dx.doi.org/10.1373/clinchem.2008.115345.

K. J. Janssen, A. R. Donders, F. E. Harrell, Y. Vergouwe, Q. Chen, D. E. Grobbee, and K. G. Moons. Missing covariate data in medical research: to impute is better than to ignore. *Journal of clinical epidemiology*, 63(7):721–727, 2010. ISSN 1878-5921. doi: 10.1016/j.jclinepi.2009.12.008. URL http://dx.doi.org/10.1016/j.jclinepi.2009.12.008.

H. Jiang, Y. Deng, H.-S. Chen, L. Tao, Q. Sha, J. Chen, C.-J. Tsai, and S. Zhang. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics*, 5(1):81, 2004. ISSN 1471-2105. doi: 10.1186/1471-2105-5-81. URL http://www.biomedcentral.com/1471-2105/5/81.

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

M. A. Klebanoff and S. R. Cole. Use of multiple imputation in the epidemiologic literature. *American journal of epidemiology*, 168(4):355–357, 2008. ISSN 1476-6256. doi: 10.1093/aje/kwn071. URL http://dx.doi.org/10.1093/aje/kwn071.

B. Lausen, W. Sauerbrei, and M. Schumacher. Classification and regression trees (cart) used for the exploration of prognostic factors measured on different scales. In P. Dirschedl and R. Ostermann, editors, *Computational Statistics*, pages 483–496. Physica-Verlag, Heidelberg, 1994.

A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL http://CRAN.R-project.org/doc/Rnews/.

Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474):578–590, 2006.

M. Little, P. McSharry, S. Roberts, D. Costello, and I. Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(1):23, 2007. ISSN 1475-925X. doi: 10.1186/1475-925X-6-23. URL http://www.biomedical-engineering-online.com/content/6/1/23.

R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data, Second Edition.* Wiley-Interscience, 2 edition, 2002. ISBN 0471183865. URL http://www.worldcat.org/isbn/0471183865.

K. Lunetta, B. L. Hayward, J. Segal, and P. Van Eerdewegh. Screening large-scale association study data: exploiting interactions using random forests. *BMC Genetics*, 5(1), 2004. doi: 10.1186/1471-2156-5-32. URL http://dx.doi.org/10.1186/1471-2156-5-32.

P. Messeri, G. Lee, D. M. Abramson, A. Aidala, M. A. Chiasson, and D. J. Jessop. Antiretroviral therapy and declining aids mortality in new york city. *J Medical Care*, 4: 512–521, 2003.

F. Mosteller and J. W. Tukey. *Data analysis and regression: a second course in statistics*. Addison-Wesley Pub. Co., 1977.

K. Nicodemus. Letter to the editor: On the stability and ranking of predictors from random forest variable importance measures. *Briefings in Bioinformatics*, 2011. doi: doi:10.1093/bib/bbr016.

K. Nicodemus, J. Malley, C. Strobl, and A. Ziegler. The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics*, 11(1):110, 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-110. URL http://dx.doi.org/10.1186/1471-2105-11-110.

R. K. Pearson. The problem of disguised missing data. *SIGKDD Explor. Newsl.*, 8(1): 83–92, 2006. ISSN 1931-0145. doi: http://doi.acm.org/10.1145/1147234.1147247.

X. Qiu, Y. Xiao, A. Gordon, and A. Yakovlev. Assessing stability of gene selection in microarray data analysis. *BMC Bioinformatics*, 7(1), 2006. doi: http://dx.doi.org/10.1186/1471-2105-7-50. URL http://dx.doi.org/10.1186/1471-2105-7-50.

J. R. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1 edition, 1993. ISBN 1558602380. URL http://www.worldcat.org/isbn/1558602380.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org/. ISBN 3-900051-07-0.

A. Rieger, T. Hothorn, and C. Strobl. Random forests with missing values in the covariates, 2010. URL http://epub.ub.uni-muenchen.de/11481/.

B. Ripley. *tree: Classification and regression trees*, 2011. URL http://CRAN.R-project.org/package=tree. R package version 1.0-29.

W. Rodenburg, A. G. Heidema, J. M. A. Boer, I. M. J. Bovee-Oudenhoven, E. J. M. Feskens, E. C. M. Mariman, and J. Keijer. 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, 2008. doi: 10.1152/physiolgenomics.00167.2007. URL http://physiolgenomics.physiology.org/content/33/1/78.abstract.

D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. doi: 10.1093/biomet/63.3.581. URL http://biomet.oxfordjournals.org/cgi/content/abstract/63/3/581.

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

D. B. Rubin. Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91(434):473–489, 1996. ISSN 01621459. URL http://www.jstor.org/stable/2291635.

M. Sandri and P. Zuccolotto. Variable selection using random forests. In S. Zani, A. Cerioli, M. Riani, and M. Vichi, editors, *Data Analysis, Classification and the Forward Search*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 263–270. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-35978-4. URL http://dx.doi.org/10.1007/3-540-35978-8_30. 10.1007/3-540-35978-8_30.

W. Sauerbrei. The use of resampling methods to simplify regression models in medical statistics. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 48(3):313–329, 1999. ISSN 00359254. doi: 10.2307/2680827. URL http://dx.doi.org/10.2307/2680827.

W. Sauerbrei, P. Royston, and H. Binder. Selection of important variables and determination of functional form for continuous predictors in multivariable model building. *Statistics in medicine*, 26(30):5512–5528, 2007. ISSN 0277-6715. doi: 10.1002/sim.3148. URL http://dx.doi.org/10.1002/sim.3148.

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

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

D. Schwarz, S. Szymczak, A. Ziegler, and I. Konig. Picking single-nucleotide polymorphisms in forests. *BMC Proceedings*, 1(Suppl 1):S59, 2007. ISSN 1753-6561. URL http://www.biomedcentral.com/1753-6561/1/S1/S59.

J. Shao. Linear Model Selection by Cross-Validation. *Journal of the American Statistical Association*, 88(422):486–494, 1993. ISSN 01621459. doi: 10.2307/2290328. URL http://dx.doi.org/10.2307/2290328.

D. J. Stekhoven and P. Bühlmann. Missforest - nonparametric missing value imputation for mixed-type data. *Bioinformatics*, 2011. doi: 10.1093/bioinformatics/btr597. URL http://bioinformatics.oxfordjournals.org/content/early/2011/10/28/bioinformatics.btr597.abstract.

H. Strasser and C. Weber. On the asymptotic theory of permutation statistics. *Mathematical Methods of Statistics*, 2, 1999.

C. Strobl and A. Zeileis. Danger: High power! ? exploring the statistical properties of a test for random forest variable importance, 2008. URL http://epub.ub.uni-muenchen.de/2111/.

C. Strobl, A.-L. Boulesteix, and T. Augustin. Unbiased split selection for classification trees based on the gini index. *Computational Statistics & Data Analysis*, 52(1):483–501, 2007a. ISSN 01679473. doi: 10.1016/j.csda.2006.12.030. URL http://dx.doi.org/10.1016/j.csda.2006.12.030.

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(1):25, 2007b. ISSN 1471-2105. doi: 10.1186/1471-2105-8-25. URL http://www.biomedcentral.com/1471-2105/8/25.

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(1):25+, 2007c. ISSN 1471-2105. doi: 10.1186/1471-2105-8-25. URL http://dx.doi.org/10.1186/1471-2105-8-25.

C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):307+, 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-307. URL http://dx.doi.org/10.1186/1471-2105-9-307.

C. Strobl, J. Malley, and G. Tutz. An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, 14(4):323–348, 2009.

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

R. Tang, J. Sinnwell, J. Li, D. Rider, M. de Andrade, and J. Biernacka. Identification of genes and haplotypes that predict rheumatoid arthritis using random forests. *BMC Proceedings*, 3(Suppl 7):S68, 2009. ISSN 1753-6561. URL http://www.biomedcentral.com/1753-6561/3/S7/S68.

M. Templ, A. Kowarik, and P. Filzmoser. Iterative stepwise regression imputation using standard and robust methods. *Computational Statistics & Data Analysis*, 55(10):2793 – 2806, 2011. ISSN 0167-9473. doi: 10.1016/j.csda.2011.04.012. URL http://www.sciencedirect.com/science/article/pii/S0167947311001411.

T. M. Therneau, B. Atkinson, and B. Ripley. *rpart: Recursive Partitioning*, 2011. URL http://CRAN.R-project.org/package=rpart. R package version 3.1-49.

S. van Buuren. Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical Methods in Medical Research*, 16(3):219–242, 2007. doi: 10.1177/0962280206074463. URL http://smm.sagepub.com/cgi/content/abstract/16/3/219.

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

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

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, USA, 4th edition, 2003. ISBN 0387954570. URL http://www.worldcat.org/isbn/0387954570.

M. Wang, X. Chen, and H. Zhang. Maximal conditional chi-square importance in random forests. *Bioinformatics*, 26(6):831–837, 2010. doi: 10.1093/bioinformatics/btq038. URL http://bioinformatics.oxfordjournals.org/content/26/6/831.abstract.

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

I. R. White, P. Royston, and A. M. Wood. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine*, 30(4):377–399, 2011. ISSN 1097-0258. doi: 10.1002/sim.4067. URL http://dx.doi.org/10.1002/sim.4067.

W. B. Yahya, K. Ulm, L. Fahrmeir, and A. Hapfelmeier. k-ss: a sequential feature selection and prediction method in microarray study. *International Journal of Artificial Intelligence*, 6(S11):19–47, 2011. ISSN 0974-0635. URL http://www.ceserp.com/cp-jour/index.php?journal=ijai&page=article&op=view&path%5B%5D=932.

W. Yang and C. C. Gu. Selection of important variables by statistical learning in genome-wide association analysis. *BMC Proceedings*, 3(Suppl 7):S70, 2009. ISSN 1753-6561. URL http://www.biomedcentral.com/1753-6561/3/S7/S70.

P. Zhang. Model selection via multifold cross validation. *Annals of Statistics*, 21(1):299–313, 1993. URL http://www.jstor.org/stable/3035592.

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