Anna Rieger, Torsten Hothorn and Carolin Strobl

Random Forests with Missing Values in the Covariates

Technical Report Number 79, 2010
Department of Statistics
University of Munich

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

Random Forests with Missing Values in the Covariates

Anna Rieger · Torsten Hothorn · Carolin Strobl

Abstract In Random Forests [2] several trees are constructed from bootstrap- or subsamples of the original data. Random Forests have become very popular, e.g., in the fields of genetics and bioinformatics, because they can deal with high-dimensional problems including complex interaction effects. Conditional Inference Forests [8] provide an implementation of Random Forests with unbiased variable selection. Like the original Random Forests, they employ surrogate variables to handle missing values in the predictor variables.

In this paper we report the results of an extensive simulation study covering both classification and regression problems under a variety of scenarios, including different missing value generating processes as well as different correlation structures between the variables. Moreover, a high dimensional setting with a high number of noise variables was considered in each case. The results compare the performance of Conditional Inference Forests with surrogate variables to that of *knn* imputation prior to fitting.

The results show that while in some settings one or the other approach is slightly superior, there is no overall difference in the performance of Conditional Inference Forests with surrogate variables and with prior *knn*-imputation.

Keywords Surrogate Variables · *knn*-imputation · missing at random.

Carolin Strobl is supported by grant STR1142/1-1 from the German Research Foundation (Deutsche Forschungsgemeinschaft).

Anna Rieger
Deparment of Statistics
Ludwig-Maximilians-Universität
Munich, Germany E-mail: Anna.Rieger@campus.lmu.de

Torsten Hothorn
Deparment of Statistics
Ludwig-Maximilians-Universität
Munich, Germany E-mail: torsten.hothorn@stat.uni-muenchen.de

Carolin Strobl
Deparment of Statistics
Ludwig-Maximilians-Universität
Munich, Germany E-mail: carolin.strobl@stat.uni-muenchen.de

1 Introduction

In Random Forests [2] a forest of classification or regression trees is constructed from bootstrap- or subsamples of the original data, and the prediction rule is based on the majority vote or average over all trees. Random Forests are particularly popular in the fields of genetics and bioinformatics [11,1,4,5], because they can deal with high-dimensional problems involving complex interaction effects.

As an advancement of the original method, Conditional Inference Forests were developed [8]. Conditional Inference Forests follow the construction principle of Random Forests, but employ different splitting criteria in the tree building process to avoid the problem of variable selection bias [6,13,14], that can affect the original Random Forests when predictor variables of different types are compared [15].

The aim of this work is to illustrate how Conditional Inference Forests deal with missing values in the predictor variables: Conditional Inference Forests adopt the principle of surrogate variables suggested by [3], so that observations with missing values can be directly processed by the forest. As an alternative approach, k-nearest-neighbors- (*knn*-)imputation [17] of missing values before the construction of the forest is considered.

While both approaches are popular and easy to use, it has not yet been investigated whether they are comparable with respect to prediction accuracy, or whether one approach is clearly superior to the other. Therefore the two approaches are compared in this paper.

The principle of surrogate splits in tree-based methods is the following: Once a predictor variable is selected for the next split in a tree, observations that have a missing value in this variable are processed further down the tree by means of a surrogate variable, that is not missing. The surrogate variable is selected such that it is the best predictor for the split in the originally chosen variable. Advantages of this approach are that observations with missing values need not be excluded from the data set or treated prior to model fitting.

As opposed to that, imputation approaches like *knn*-imputation are used in a preliminary step to fill in missing values in the data set before the model fitting stage. In *knn*-imputation, each observation with a missing value receives the weighted mean value that the k-nearest-neighbors have in this variable. The proximity of the neighbors is computed from the remaining variables by Euclidian distance (see also section 3).

In the following, an extensive simulation study covering classification and regression problems as well as a variety of missing data mechanisms is presented in order to compare the two approaches. In the case of missing values, the advantage of simulations studies is that the missing values can be induced "on purpose", so that the "true" values are still known. This allows for quantifying the prediction error of each method by means of generating learning and test samples. All simulations and calculations were conducted with the R system for statistical computing [12].

2 Simulation design

The data sets are generated in two ways, depending on the type of the response variable. All details of the data generating processes and simulation designs are given in the following.

2.1 Data generating processes

The first data generating process (dgp) is for categorial response, i.e. classification; the second is for continuous response, i.e. regression.

2.1.1 Process for classification

The response is a binary factor with two values: $y = 1$ and $y = 2$. It is calculated by modelling the logits:

$$P(y = 2|X) = \pi(X) = \frac{\exp(X^\top \beta)}{1 + \exp(X^\top \beta)} \tag{1}$$

In doing so, X contains $x_j, j = 1, \ldots, 5$, where x_j are multivariate nomally distributed with mean 0 and with variance Σ_j, that can be varied.

The influence of the single covariates x_j is set to $\beta = (1,2,3,4,5)^\top$. With the resulting probability vector $\pi(X)$, the response is assigned class 1 or 2 at random.

In order to simulate high-dimensional settings, 45 noise variables, that are multivariate normally distributed with mean 0 and the identity matrix as covariance, are added as possible predictors.

2.1.2 Process for regression

For generating a continous response, the model "Friedman1" [7, formula (61)] was applied. In the original reference, ten independent uniformly distributed variables are used, but only five of them have influence on the response:

$$y = 10 \cdot \sin(\pi x_1 x_2) + 20 \cdot (x_3 - 0.5)^2 + 10 \cdot x_4 + 5 \cdot x_5 \tag{2}$$

Here, we again add 45 noise variables to simulate a high-dimensional setting. These noise variables are identically and independently distributed. However, in our dgp the variables that have an influence are correlated.

In order to simulate uniformly distributed correlated variables, the variables are first sampled from a multivariate normal distribution with variance 1 and the desired correlation (that is equal to the covariance if the variance is 1). In a second step, the distribution function of the standard normal distribution is applied to each random variable. The resulting marginal distributions are uniform on $[0, 1]$.

$$\begin{aligned} X &\sim F(x) \\ \Rightarrow F(X) &\sim U(0,1) \end{aligned}$$

2.2 Correlation matrices

For each of the two dgps, three different correlation (or respectively covariance) matrices were used.

The first correlation matrix (termed s1 in the results section) contains a high correlation of 0.9 for all pairs or variables:

$$\Sigma_1 = \begin{pmatrix} 1 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 1 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 1 \end{pmatrix}$$

The second correlation matrix (s2) also contains high correlations of 0.9. However, the variables are correlated only in two separate blocks, so that x_1 trough x_3 form one block and x_4 and x_5 form another block:

$$\Sigma_2 = \begin{pmatrix} 1 & 0.9 & 0.9 & 0 & 0 \\ 0.9 & 1 & 0.9 & 0 & 0 \\ 0.9 & 0.9 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0.9 \\ 0 & 0 & 0 & 0.9 & 1 \end{pmatrix}$$

The third correlation matrix (s3) again contains one constant correlation for all pairs or variables. However, now the value of the correlation is 0.1, i. e. we simulate low correlations:

$$\Sigma_3 = \begin{pmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{pmatrix}$$

2.3 Methods for creating missing values

Different methods were implemented for creating missing values in the predictor variables. The missing values were generated to be either missing completely at random (MCAR) or missing at random (MAR) [10].

2.3.1 Missing completely at random (MCAR)

The simplest method for inducing missing values completely at random is to choose a given number of locations with a random number generator and eliminate the values in this locations from the vector representing the variable. By this approach, as many locations as desired can be sampled out of the n observations and replaced by NA.

2.3.2 Missing at random

Methods for generating values missing at random (no longer *completely* at random) are more complicated: The choice of the locations that are replaced by missing values in the “missing” variable now depends on the value of a second variable, that we will term “determining” variable. Therefore, the values of the “determining” variable now have an influence on whether a value in the “missing” variable is missing or not.

Table 1 Coefficients for missings by logit modelling

response	$\gamma_{i,0}$	$\gamma_{i,k}$
categorial	-2.25	1
continous	-0.5	1

*Creation of ranks (*MAR1*)* The first procedure for MAR constructs a vector with the ranks of the “determining” variable. The probability for a missing value in a certain location in the “missing” variable is computed by dividing the rank of the location in the “determining” variable by the sum of all ranks. Because the minimum observation gets the lowest rank 1, the minimum observation produces the lowest probability to get a `NA`. The locations for `NA` in the “missing” variable are then sampled with the resulting probability vector.

*Creation of two groups (*MAR2*)* The second approach for creating values MAR divides the data set in two groups. These two groups are defined by the “determining” variable. An observation belongs to the first group if the value in the “determining” variable is greater than or equal to the median of the “determining” variable, otherwise it belongs to the second group. The probability vector has the mass 1, which is distributed between the two groups in the proportion 9:1. If an observation belongs to the first group, its probability for `NA` is 9 times higher than the probability for `NA` in the second group below the median. A single observation in the respective group has missing value probability of 0.9 or respectively 0.1 divided by the number of members in this group. The locations for `NA` are sampled with the resulting probability vector again.

*Dexter truncation (*MAR3*)* With this method, it is simply the observations with the biggest values in the “determining” variable that are replaced by `NA` in the “missing” variable, until the desired fraction of `NA` (see section 2.3.3) has been achieved.

*Symmetric truncation (*MAR4*)* This method is similar to the previous method. However, instead of deleting all values of the “missing” variable that correspond to the biggest values in the “determining” variable, the fraction of missings is cut in half and the method also cancels the observations with the smallest values in the “determining” variable.

*Logit modelling (*LOG*)* In this method for missing values, the probability for `NA` no longer depends on a single “determining” variable. It is modelled by logits:

$$P(x_i \text{ missing}) = \text{logit}\left(\gamma_{i,0} + \sum_{\substack{k=1\\k\neq i}}^{5} \gamma_{i,k} \cdot x_k\right)$$

i. e. it depends on the values of five other variables and on the coefficients $\gamma_{i,k}$. For each variable with missings desired, one needs an intercept and five coefficients. The utilised coefficients, achieved by trial and error such that the fraction of missing values in each variable approximately reached the desired value, are given in table 1. When the sum is calculated, one gets the probability vector by computing the logits. The locations for `NA` are sampled with the resulting probability vector.

Table 2 “Determining” variables and fractions of missingness.

missings in	by creation of ranks/of two groups	by dexter/symmetric truncation	fraction
x_1	x_2	x_2	20%
x_3	x_4	x_5	10%
x_4	x_5	x_5	20%
summary			50%

*Missings depending on y (*DEPy*)* For missing values depending on the value of the response y, the probability is 0.1, if the response is categorial and has value $y = 1$. Otherwise it is 0.3.

For continuous response, i. e. in the second dgp, the probability is 0.1 for values $y \geq 13$, otherwise it is 0.4. The values have been chosen by trial and error, until the desired fractions of `NA` (see section 2.3.3) had been achieved.

2.3.3 Fraction of missingness

The fraction of missing values is the same in each simulation: In the first variable x_1 20% of data are missing in the learning sample respectively in the test sample. In the third variable x_3 the amount is 10% of missings, in the forth variable there are again 20% missing. We used a maximum of 20% missing values, because [17] also worked with this upper bound. One gets altogether at the maximum 50% observations with missing data with this distribution of fractions.

Additionally, for the MAR-functions “determining” variables are needed. For missings in x_1 the “determining” variable was x_2, because they belong together to one correlation block in the high correlated block design matrix Σ_2. The variable x_5 is used as the “determining” variable for x_4 because they are correlated in each design, too. So in each block of the correlation matrix Σ_2 there is one variable with missing values (x_1 and x_4 respectively).

For variable x_3, however, x_4 was chosen as the “determining” variable just in the first two methods, because in this way one can test the assignment quality of the `cforest`-function with uncorrelated variables (in design Σ_2).

For the third and forth method (dexter truncation and symmetric truncation) of MAR one needs complete cases in the “determining” variable because of the design of the function `quantile()` used in R. So, for these methods the “determining” variable for x_3 is x_5, because x_4 has missings and is thus not applicable as “determining” variable. Since x_3 and x_5 not correlated either in the original setting, one can use x_5 as “determining” variable instead of x_4. Table 2gives a summary of this.

3 Imputation method

The missing values were imputed by means of *knn*-imputation. The procedure searches the k nearest neighbors with respect to Euklidian distance and replaces the missing value by a weighted mean over the neighbors’ values. In this case, $k = 10$ is chosen. For more details on *knn*-imputation see [17].

4 Computation of Random Forests

For computation of the Conditional Inference Forests, the function `cforest` from R package `party` was used. The following arguments were used:

As the splitting criterion, the maximum value of the linear statistic (default to be used if all predictors are of the same type as here) was used. The minimum value of the criterion necessary for splitting `mincriterion` was also set to the default value 0.1. Within each forest `ntree` = 50 trees were produced. (This low number of trees is sufficient because is the dgps used here the number of relevant predictor variable is low.) The maximum number of surrogate variables was 3. As an extra stopping criterion, the number of observations left in a node for splitting, `minsplit`, was set to 30.

5 Simulation

In total, all possible combinations of

- the two data generating processes,
- the three correlation/covariance matrices,
- the seven methods for inserting `NA` and
  - imputation with subsequent fitting of the `cforest` or
  - directly fitting the `cforest` with surrogate splits

were investigated in the simulation study. Additionally, the high-dimensional analogons for each setting (including 45 noise variable) were investigated to mimic screening studies, e.g., for gene expression data.

In order to evaluate the performance in each szenario, there are two possibilities for missing values:

- Missing values only in the learning sample and
- in both the learning and test sample.

Two test samples were generated – one for each data generating function. Each test sample contained 5000 observations to be able to reliably estimate the prediction error. The learning samples were newly generated for each simulation and contained 200 observations each. For each simulation, 100 different learning samples were generated.

The criterion for assessing the prediction accuracy for categorial response was the out-of-sample mean binomial log likelihood

$$LL = \frac{1}{n}\sum_{i=1}^{n} y_i \cdot \log(\widehat{p}_i) + (1-y_i)\cdot\log(1-\widehat{p}_i) \tag{3}$$

where $\widehat{p}_i$ is the probability for $y_i = 2$ in the test sample, which was estimated on the learning sample. The binomial log likelihood has only non-positive values.

For data sets with continous response, the out-of-sample mean squared error was used as criterion for assessing the prediction accuracy

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 \tag{4}$$

where $\widehat{y}_i$ is the predicted value for the responses in the test sample, which was estimated on the learning sample.

6 Results

The results of the simulation studies will be displayed in the following way: The distribution of the respective criterion for assessing the prediction accuracy (binomial log-Likelihood or MSE termed “risk”) is displayed by means of boxplots. For better comparison, the negative binomial log-Likelihood is plotted, so that the orientation is the same as for the MSE. Thus, a high value on the y-axis always indicates a poor prediction accuracy.

To increase the readability, not all combination of experimantal factors are presented at the same time, but the results are displayed separately for classification vs. regression problems, for missing values in the learning data only vs. missing values in both learning and test data, and for low vs. high dimensional data sets.

The experimental factors that vary within each display are the missing value generating processes and the correlation structures.

6.1 Classification problems

6.1.1 Low dimensional problems

Missing values in the learning data only In the low dimensional classification problems with missing values in the learning data only, Conditional Inference Forests with prior *knn*-imputation (termed knn in the display) in average show a slightly higher risk (i.e. a slightly worse performance) than Conditional Inference Forests with surrogate variables (termed sur in the display).

However, this effect is mediated by the higher-order interactions between the imputation approach, the missing value generating processes and the correlation structures: In the case of high correlations (s1) and missing values inserted by truncation depending on the determining variable (MAR3), in the case of high correlations (s1) and missing values inserted by a logistic model depending on five determining variables (LOG) and in the case of block correlations (s2) and missing values inserted by truncation depending on the determining variable (MAR3) the effect is reversed, such that *knn*-imputation shows a slightly lower risk (i.e. a slightly better performance) than surrogate variables, as illustrated in Figure 1.

However, in a linear model including all main effects and interaction terms, besides the significant main effect of *knn*-imputation vs. surrogate variables, only the first of these three third-order interactions reversing the effect in favor of *knn*-imputation was significant.

Missing values in both learning and test data In the low dimensional classification problems with missing values in the learning and test data, the pattern is essentially the same as before, as illustrated in Figure 2, but the effects are less pronounced.

Accordingly, there are again some significant third-order interactions (including the ones described above), but the main effect of *knn*-imputation vs. surrogate variables no longer significant.

6.1.2 High dimensional problems

In the high dimensional classification problems with missing values in the learning or learning and test data, the effects are even less pronounced (Figure not shown). Overall, there

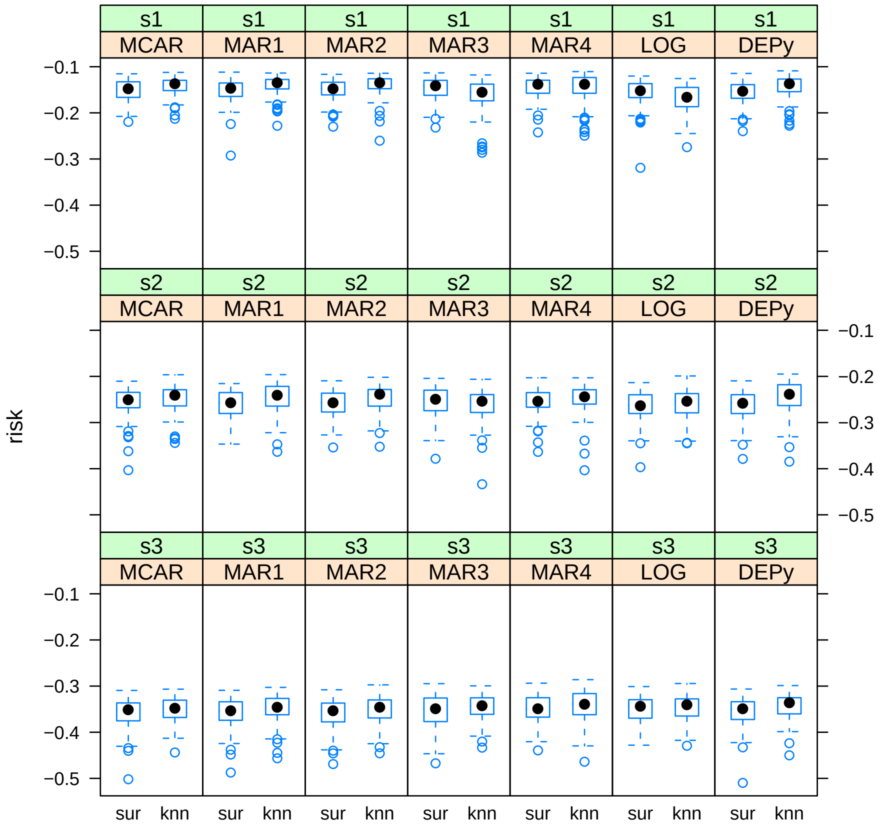

Fig. 1 Distributions of negative binomial log-Likelihood for low dimensional classification problems with missing values in the learning data only.

is no difference between the performance of Conditional Inference Forests with prior *knn*-imputation and with surrogate variables.

The only significant interaction effect, that may be worth mentioning, is again that in the case of high correlations (s1) and missing values inserted by a logistic model depending on five determining variables (LOG) *knn*-imputation shows a slightly lower risk (i.e. a slightly better performance). Apart from that, there is no significant difference for any factor combination, but again the tendency that in most cases *knn*-imputation shows a slightly higher risk (i.e. a slightly worse performance) than surrogate variables.

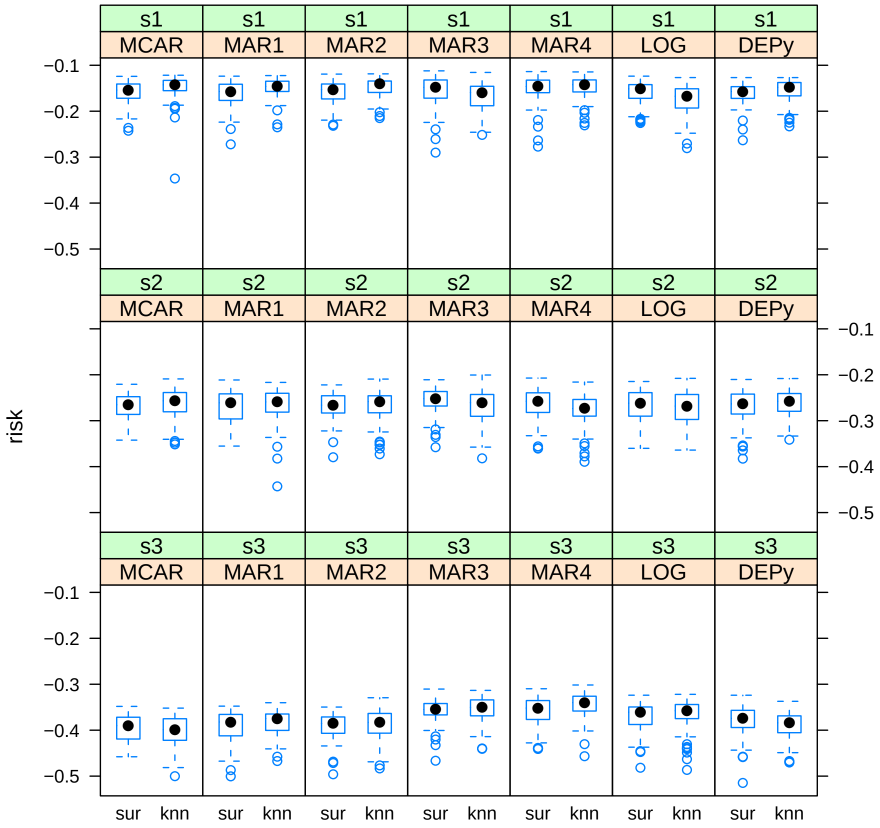

Fig. 2 Distributions of negative binomial log-Likelihood for low dimensional classification problems with missing values in the learning and test data.

6.2 Regression problems

6.2.1 Low dimensional problems

Missing values in the learning data only As opposed to the classification problems, in the low dimensional regression problems with missing values in the learning data only, *knn*-imputation shows a slightly lower risk (i.e. a slightly better performance) than surrogate variables over all factor combinations, as illustrated in Figure 3.

The main effect as well as several higher-order interactions (including the ones already found in the classification problems above) are significant. However, now the interaction effects are in the direction of the main effect, so that they increase the effect that *knn*-imputation shows a slightly lower risk (i.e. a slightly better performance) than surrogate variables in cases with high or blockwise correlations and missing values inserted depending determining variables.

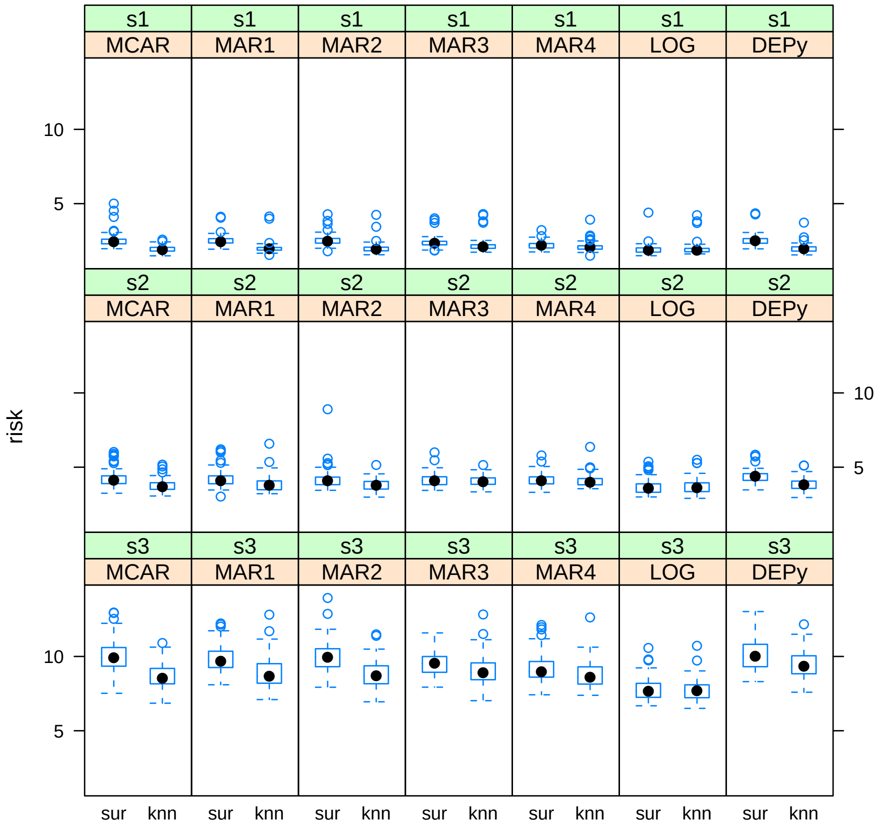

Fig. 3 Distributions of MSE for low dimensional regression problems with missing values in the learning data only.

Missing values in both learning and test data Essentially the same pattern results for low dimensional regression problems with missing values in the learning and test data (Figure not shown).

6.2.2 High dimensional problems

In the high dimensional regression problems, the effects are again less pronounced (Figure not shown). There is again no significant difference between the performance of Conditional Inference Forests with prior *knn*-imputation and with surrogate variables.

6.3 Real data set: The Los Angeles ozone pollution data

Besides the simulation results, we would like to illustrate the performance of the two methods by means of a real data set on ozone pollution.

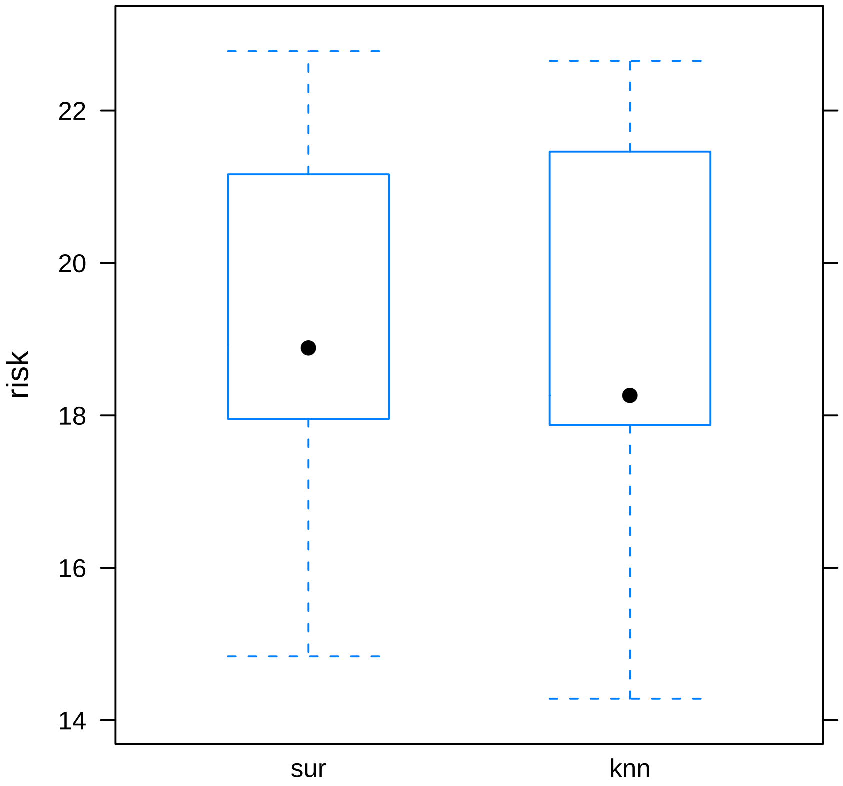

Fig. 4 Distributions of MSE over 100 bootstrap samples for Conditional Inference Forests with with surrogate splits and with prior *knn*-imputation.

The data set has 366 observations and 13 variables, including the response (V4). It is available in the R-package “mlbench” [9]. Originally there are five missing values in the response, but these observations were removed from the data set. The response is numerical, so regression trees were employed.

To again compare the two analysis approaches – Conditional Inference Forests with surrogate splits or with prior *knn*-imputation – and illustrate the variability of the results due to random sampling, we drew 100 bootstrap samples from the original data for each approach. The MSE was calculated based on the respective out-of-bag samples.

The results illustrated in Figure 4 show that in this particular example imputation leads to slightly lower prediction error in the median. However, the variability is very high in both cases and the difference is not significant.

7 Discussion and Conclusion

In summary, our results show that there is no clear advantage for either *knn*-imputation or surrogate variables for dealing with missing values in the predictor variables.

While in our simulation setup surrogate variables show a slightly better perfomance in most settings for classification problems and *knn*-imputation shows a slightly better performance in most settings for regression problems, these results may not be generalizable due to our particular choice of the parameters used for generating the data. Moreover, no significant differences could be found in high dimensional problems.

Our results do support the general understanding that the potential of imputation approaches depends on the correlation structure of the data in combination with the missing value generating process: As shown in the results section, *knn*-imputation has an advantage for those combinations with high correlations and missing values inserted depending determining variables (i.e., missing at random (MAR) as opposed to missing completely at random (MCAR)).

However, since in practice neither the missing value generating process nor the true correlation structure (of the original variables without missing values) is known, this result cannot aid the decision whether to use an imputation approach or rely on surrogate variables.

References

1. Arun K and Langmead CJ (2006) Structure based chemical shift prediction using random forests non-linear regression. In: Jiang T, Yang UC, Chen YPP, and Wong L, (Eds.) (2006) Proceedings of the Fourth Asia-Pacific Bioinformatics Conference, Taipei, Taiwan, pp. 317-326
2. Breiman L (2001) Random Forests. Machine Learning, 45, pp. 5-32
3. Breiman L, Friedman J, Olshen R and Stone C, (1984) Classification and Regression Trees. Chapman & Hall, New York, NY, USA
4. Bureau A, Dupuis J, Falls K, Lunetta KL, Hayward B, Keith TP and Eerdewegh PV (2005) Identifying SNPs predictive of phenotype using random forests. Genetic Epidemiology, 28 (2), pp. 171-182
5. Diaz-Uriarte R and de Andrés SA (2006) Gene selection and classification of microarray data using random forest. BMC Bioinformatics, 7 (3)
6. Dobra A and Gehrke J (2001) Bias correction in classification tree construction. In: Brodley CE and Danyluk AP (Eds.) (2001) Proceedings of the Seventeenth International Conference on Machine Learning (ICML), Williams College, Williamstown, MA, USA. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 90-97
7. Friedman JH (1991) Multivariate adaptive regression splines. The Annals of Statistics, 19 (1), pp. 1-67
8. Hothorn T, Hornik K and Zeileis A (2006) Unbiased Recursive Partitioning: A Conditional Inference Framework. Journal of Computational and Graphical Statistics, 15 (3), pp. 651-674
9. Leisch F and Dimitriadou E (2009) mlbench: Machine Learning Benchmark Problems. R package, version 1.1-6
10. Little R and Rubin D (2002) Statistical analysis with missing data. 2nd edn. John Wiley & Sons, Inc., Hoboken, NJ, USA
11. Lunetta KL, Hayward LB, Segal J and Eerdewegh PV (2004) Screening large-scale association study data: Exploiting interactions using random forests. BMC Genetics, 5 (32)
12. R Development Core Team (2008) R: A Language and Environment for Statistical Computing. ISBN 3-900051-07-0, http://www.R-project.org, R Foundation for Statistical Computing, Wien, Österreich
13. Shih Y (2004) A note on split selection bias in classification trees. Computational Statistics and Data Analysis 45 (3), pp. 457-466
14. Strobl C, Boulesteix AL and Augustin T (2007a) Unbiased split selection for classification trees based on the Gini Index. Computational Statistics & Data Analysis, 52 (1), pp. 483-501
15. Strobl C, Boulesteix AL, Zeileis A and Hothorn T (2007b) Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics, 8 (25)
16. Svejdar V (2007) Variablenselektion in Klassifikationsbäumen unter spezieller Berücksichtigung von fehlenden Werten. Master Thesis, Ludwig-Maximilians-Universität München
17. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D and Altman RB (2001) Missing value estimation methods for DNA microarrays. Bioinformatics, 17 (6), pp. 520-525