Working Paper

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Workshop on Statistical Data Editing
(Neuchâtel, Switzerland, 18-20 September 2018)

Using Random Forest to Improve Single Imputation Precision - A Case Study

Prepared by Johannes Gussenbauer, Alexander Kowarik and Angelika Meraner, Statistics Austria, Vienna, Austria

I. INTRODUCTION

1. The k-Nearest-Neighbor method (kNN) is based on donor observations. A missing value is imputed based on the aggregation of the k values nearest to the missing values, the so called k nearest neighbors. Depending on the type of variable different aggregation estimates can be used.
The distance computation of the function `kNN` in the R package `VIM` for defining the nearest neighbors is based on an extension of the Gower distance [Gower, 1971], which can handle distance variables of the type binary, categorical, ordered, continuous and semi-continuous. The distance between two observations is then defined as the weighted mean of the contributions of each variable, where the weight should represent the importance of the variable.

2. When applying k-Nearest-Neighbour imputation the value for k and weights for the distance variables can be choosen. The value for k is in many cases set to 5, which yields in general acceptable results. However the selection of proper weights for the distance variables is heavily depended on the problem at hand and can be quite challenging. Using an automized procedure for estimating weights can, for instance, be achieved by using the random forest algorithm and its embedded procedure for estimating variable importance. Especially for a large number of distance variables a proper choice for the weights can counteract the curse of dimensionality, which can limit the discriminative power of distance metrices in high dimensional vector spaces.

3. Another approach for improving the performance of the k-Nearest-Neighbour method is the use of additional distance variables, which are extracted from the data by applying a statistical model beforehand. For our case study we chose the random forest algorithm [Breiman, 2001] to estimate the variable, containing missing values, on the subset with fully observed observations.

4. Based on the `kNN` function of R package `VIM` [Kowarik and Templ, 2016] and the R package `ranger` [Wright and Ziegler, 2017b] the performance for data imputation with and without weight estimation are applied to real life data based on the austrian register for households. The variables of interests are a semi continous variable, person income, and an ordered variable, education. In addition the performance increase for including the predicted variable, using the random forest algorithm, in the set of distance variables is tested with and without weight adjustment.

II. Weighting approach

1. In the used variant of the Gower Distance [Kowarik and Templ, 2016], as in the original one [Gower, 1971], each variable k has a contribution $\delta_{i,j,k}$ in the range of 0 and 1 weighted with a weight w_k. The distance

$$d_{i,j} = \frac{\sum_{k=1}^{p} w_k \delta_{i,j,k}}{\sum_{k=1}^{p} w_k} \ , \tag{1}$$

is therefore strongly influenced by the selection of weights. If uncorrelated/random variables are added to the distance computation in the `kNN` procedure and all are given equal weights, noise is added to the imputation and the estimation suffers. This issue becomes more severe for large number of distance variables.

2. Obviously, an informed choice on the importance of a specific variable by a data expert might be a good start to define weights. However, when imputing a large number of variables or when having a large number of variables available as possible distance variables can make the situation more complex.

A. Random forest importance

1. Of course, random forest itself provides the capabilities to be used in imputation, but it can also be used to generate a weight vector, namely the importance estimated when computing the random forest model. In the R package `ranger` the variable importance for a regression tree is measured with the response variance and in the case of a classification tree with the Gini index, see [Wright and Ziegler, 2017a]. Before applying `kNN`, random forest is applied and the estimated importance is used as weight vector for the `kNN` procedure.

2. Two variants of these method are used in the case study:

kNN: The weights are all set to 1, without using the importance from the random forest algorithm
kNNw: The weights are defined by the importance measure generated by a random forest procedure.

B. Feature as distance variable

1. The `kNN` method could also be improved by using additional features as distance variables, if these features are correlated with the variable containing missing values. Predicting the variable of interest and using the prediction as an additional distance variable results in so called predictive mean matching (see [Little and Rubin, 2014]). In our case study we used the random forest algorithm to predict the variable of interest.

2. Three variants of this method are tested:

kNNmean: Using only the predicted variable as distance variable for the `kNN` method.
kNNmeandist: Using the predicted variable in addition to all other available variables for the `kNN` method.
kNNmeandistw: Using the predicted variable and all other available variables as distance variables and applying weights defined by the importance measure generated by a random forest procedure.

III. Case study

A. Register data

1. The data used in this case study is based on the austrian register for households and containes personal variables like income, education, gender, and region of birth as well as household variables like county and degree of urbanization. To showcase the performance of the different methods for variable sampling size a subsample is generated from the whole population by simple random sampling. The sampling size n was chosen to be in $\{500, 1000, 2000, 3500, 5000\}$.

2. The variables of interest were chosen to be income, a semi-continous variable, and education, an ordered variable. For each variable a percentage p of the observations were set to missing according to different missing value structures (see [Rubin, 1976]), namely

MCAR: Missing completely at random occurs when missingness of a variable is independent of observable as well as unobservable variables.

MAR: Missing at random occurs when missingness of a variable can be explained or modeled by variables with complete information.

MNAR: Missing not at random occurs when missingness of a variable is depended on the variable itself.

3. For MCAR the observations which are set to missing were selected by simple random sampling. In the case of MAR the variable of interest is predicted using the random forest algorithm and observations are set to missing according to a probability vector v, defined by

$$v_i = \exp(-|x_i - \hat{x}_i|) \quad i = 1, \ldots, n$$

with x_i and $\hat{x}_i$ as the true and predicted values for the variable of observation i.
MNAR is also simulated: For the variable income the lowest and highest 10% of income greater 0 are sampled with a probability 10 times higher than the rest of the observations. In the case of education one value of education is randomly selected for which every observation is sampled with a probability 10 times higher than the other observations.

B. Error measures

1. For numerical variables, the prediction error is measured using the mean absolute error

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| x_{ij}^{imp} - x_{ij}^{orig} \right|,$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, p$.
For categorical variables we use the misclassification error rate

$$\text{MER} = \frac{1}{n} \sum_{i=1}^{n} I(x_{ij}^{imp} \neq x_{ij}^{orig}),$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, p$, where $I(x_{ij}^{imp} \neq x_{ij}^{orig}) = 1$ if $x_{ij}^{imp} \neq x_{ij}^{orig}$ and 0 else.

C. Results

1. Using the data described in section A simulations were carried out 100 times for each combination of missing type $m.type \in \{MCAR, MAR, MNAR\}$, probability of missing $p \in \{0.01, 0.025, 0.05, 0.1, 0.2\}$ and the sample size $n \in \{500, 1000, 2000, 3500, 5000\}$. The number of trees for the random forest

algorithm was set to 500. Prior to running the estimations the income variables was redefined as $\log(income + 1)$ to reduce skewness in the data.

2. Six different approaches for missing value imputation were compared, five of them were described in section II: **kNN**, **kNNw**, **kNNmean**, **kNNmeandist**, **kNNmeandistw**. These methods are compared with applying the random forest algorithm directly to impute missing values, **ranger**. For the variable income this was done in 2 steps, first modelling if an indiviual has positive income and afterwards modelling the value of income on the subset of individuals with positive income.

3. The results for variable income show, that using the random forest algorithm performed almost always worse than one of the kNN methods. This does seem strange but is however the result of predicting 0 income, for which the kNN methods were in general more successful. Looking at the differences regarding the structure of missing values, we can only conclude that for MAR the performance generally increases, which is to be expected. Using the predicted value as distance variable, **kNNmean**, does result in improved precision, compared **kNNw**. However when using it in addition to the other variables, **kNNmeandist**, the precision gain always outperformes **kNNw** as well as **kNNw**. In addition weighting them the predicted as well as all other distance variables using the variable importance does, altough slightly, increase precision yet again.
Looking at the results for variable education does show quite different picture. Method **ranger** does outperform methods **kNN** and **kNNw** and increases in precision for higher number of missing values. This can not be observed for the methods using kNN. Using the predicted value for education, **kNNmean**, does seem to yield better results than using it in addition to all other distance variables. Setting weights for all distance variables, **kNNmeandistw**, achieves roughly the same precision as using only the predicted value as distance variable.

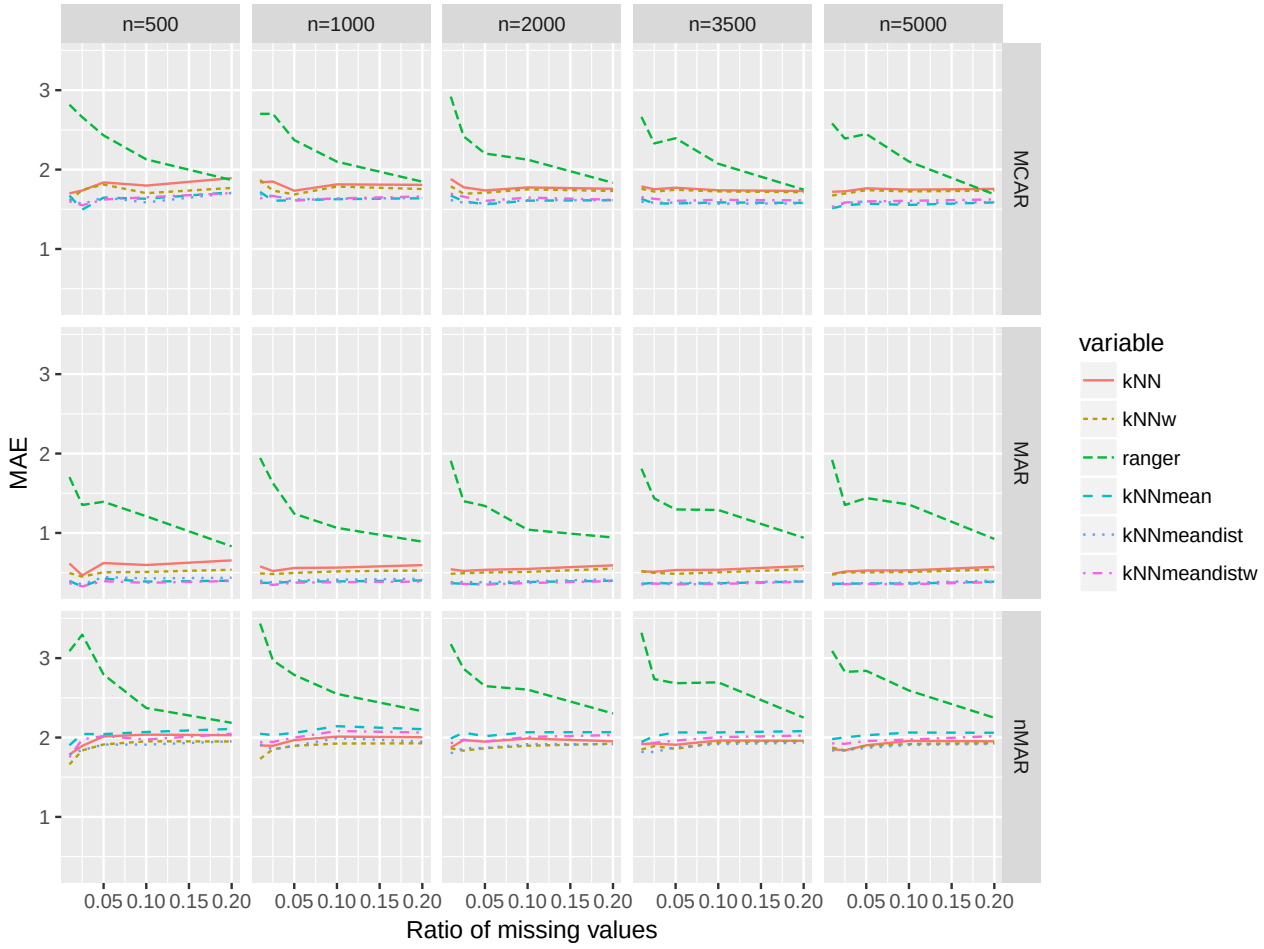

FIGURE 1. Results for variable income

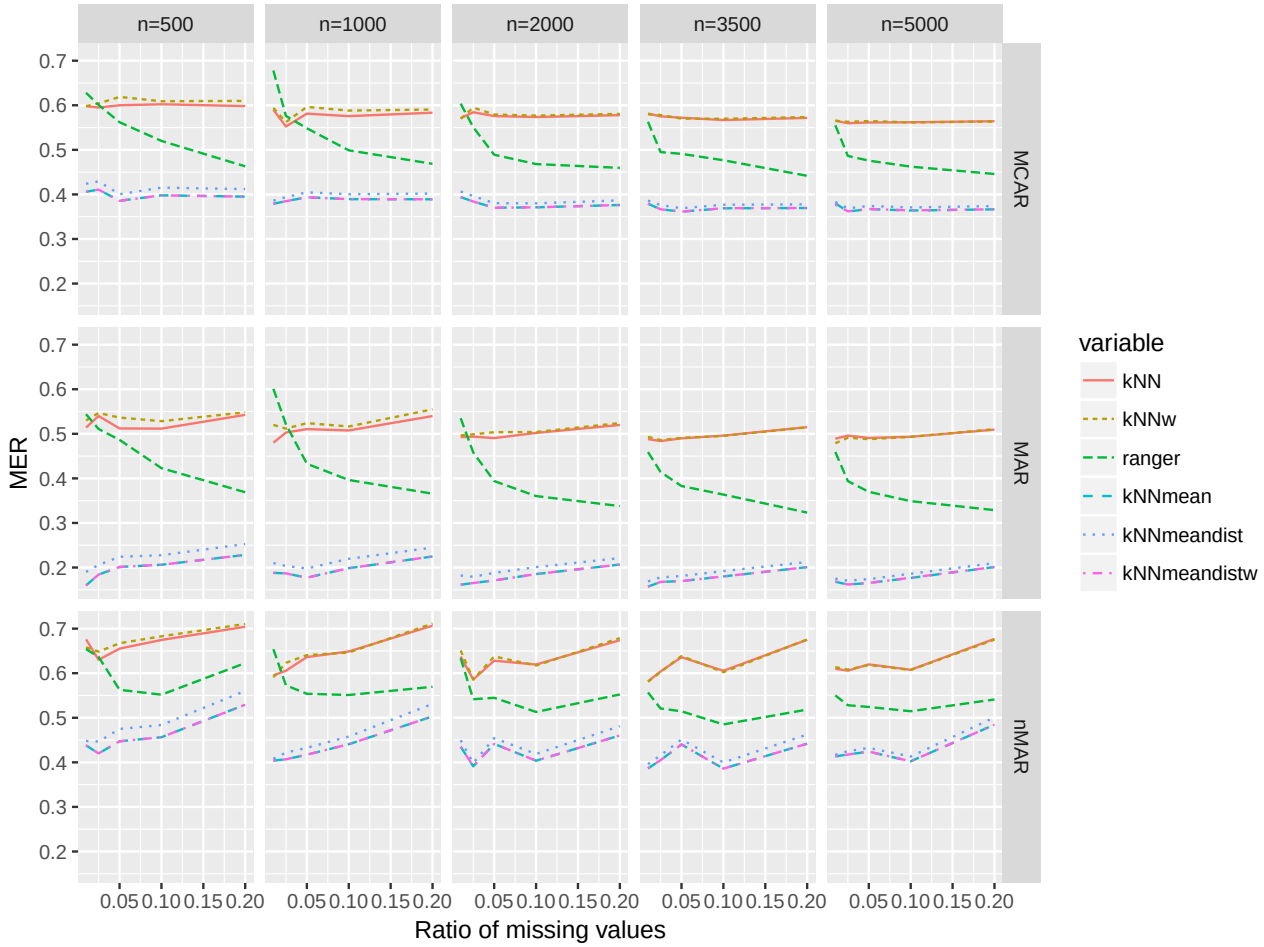

FIGURE 2. Results for variable education

IV. Conclusion and future work

1. The results show that using the random forest algorithm instead of kNN does not always yield better results and that this might be depended on the type of variable. For the ordered variable the precision for random forest was higher than when using simple kNN or weighted kNN and vice versa for the semicontinous variable. However to get a clearer picture on this more tests on a multitude of variables are needed.

2. Regarding precision improvement the results showed that using the predicted value as additional distance variable improved in general the precision of kNN. Additionally setting weights, using the variable importance, lead to the highest precision in almost every case. This showcases the potential precision gain when integrating engineered features for missing value imputation.
Using multiple engineered features, in combination with different weights, might improve precision even further for kNN, however this needs further research.

3. The different mechanism for creating missing values did not give a lot additional insight apart from the fact that MAR resulted in lower error measures for all the methods. It should however be noted that these mechanisms should still be improved.

4. The methods **kNNw**, **kNNmean**, **kNNmeandist** and **kNNmeandistw** have been implemented into the function `kNN()` from R-package `VIM`.

addRF: Setting the parameter `addRF` to `TRUE` will create predicted values of the variables of interest. When imputing missing values for a variable the corresponding predicted values are used as additional distance variables.
onlyRF: If `onlyRF` is set to `TRUE` only the predicted values are used as distance variables.
weights: The parameter `weights` can be set to `"auto"` which leads to automatic weight selection using the variable importance from the random forest algorithm. Additional distance variables, created with `addRF=TRUE`, will also be considered in this step.

References

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

J.C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

Alexander Kowarik and Matthias Templ. Imputation with the R package VIM. *Journal of Statistical Software*, 74(7):1–16, 2016. doi: 10.18637/jss.v074.i07.

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2014.

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

Marvin Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software, Articles*, 77(1):1–17, 2017a. ISSN 1548-7660. doi: 10.18637/jss.v077.i01. URL https://www.jstatsoft.org/v077/i01.

Marvin N. Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17, 2017b. doi: 10.18637/

jss.v077.i01.