Working Paper

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(The Hague, Netherlands, 24-26 April 2017)

An automatic procedure for selecting weights in `kNN` imputation

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

I. INTRODUCTION

1. The k nearest neighbor method is based on donor observations. An aggregation of the k values of the nearest neighbors is used as imputed value. The kind of aggregation depends on the type of the variable.

2. 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 the weighted mean of the contributions of each variable, where the weight should represent the importance of the variable.

3. An important decision when applying k-Nearest-Neighbour imputation is the selection of proper weights for the distance variables. Based on the `kNN` function of R package `VIM` [Kowarik and Templ, 2016], several automatic ways to estimate the weights are presented and compared and the influence on the imputation results is presented.

4. Random forest and Lasso regression are evaluated to be used as methods to estimate the optimal weights or to perform variable selection for `kNN` imputation. R packages `randomForest` [Liaw and Wiener, 2002] and `glmnet` [Friedman et al., 2010] were used for these purposes. Furthermore, the decomposition of a categorical variable in its binary contrasts and weighting of these contrasts is tested.

II. Weighting approach

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 very strongly influenced by the selection of the weights. If not correlated/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.

5. 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 this choice might be hard. Therefore, automatic solutions to identify variables are tested.

A. Random forest importance

6. 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, it is a vector with a value for each predictor. So before applying `kNN`, random forest is applied and the estimated importance is used as weight vector for the `kNN` procedure.

7. Two variants of these method are used in the simulation study:

knnw: The weights are defined by the importance measure generated by a random forest procedure.

knnw2: The weights are defined by the importance measure generated by a random forest procedure but with binomial dummy variables instead of categorical variables with $k > 2$ levels.

B. Lasso shrinkage

8. The Lasso method for estimating a regression model is used to perform shrinkage or variable selection [Tibshirani, 1996], it results in zero coefficients for variables, which are presumably not important for estimating the model. In the weight vector the weights of variables with coefficient zero are set to zero and all the rest is set to one.

9. Two variants of this method are tested:

knnl.l1se: Variable selection is performed with Lasso using the largest value of lambda such that the error is within 1 standard error of the minimum mean cross-validated error (cvm).

knnl.lmin: Variable selection is performed with Lasso using the value of lambda that gives the minimum cvm.

C. Random forest importance

III. Simulation study

A. Generated data

10. For all simulations presented in this paper, we use the same setting used in [Templ et al., 2011], where we randomly generate data with $n = 500$ observations and p variables correlated with correlation ρ from a multivariate normal distribution. The population mean of (the non-outlier part of) each variable is fixed at 10. Based on the multivariate normal distribution, variables with a binary and a semi-continuous distribution are constructed by the following procedures:

Binary variable:: A binary variable y with values $y_1, \ldots, y_n$ is created on the basis of a variable x with values $x_1, \ldots, x_n$ from the generated multivariate normally distributed data by

$$y_i = \begin{cases} 0 & \text{with} \quad P(y_i = 0) = 1 - F_{N(\mu,\sigma^2)}(x_i) \\ 1 & \text{with} \quad P(y_i = 1) = 1 - P(y_i = 0) = F_{N(\mu,\sigma^2)}(x_i) \end{cases} ,$$

for $i = 1, \ldots, n$. $F_{N(\mu,\sigma^2)}$ denotes the distribution function of x, a normal distribution with mean μ and variance σ^2. Hence, if x_i is high (low) the probability that y_i becomes zero is high (low). Depending on the choice of μ, the ratio of zeros and ones differs (default 50% zeros on average)

Multinomial variable:: A categorical variable c with k levels and values $c_1, \ldots, c_n$ is created on the basis of a variable x with values $x_1, \ldots, x_n$ from the generated multivariate normally distributed data by

$$y_i = \begin{cases} 1 & \text{if} \quad x_i < q_1 \\ 2 & \text{if} \quad q_1 \leq x_i < q_2 \\ \ldots & \\ k & \text{if} \quad q_{k-1} \leq x_i < q_k \end{cases}$$

for $i = 1, \ldots, n$ and q_j denotes the quantile with probability $\frac{100(j-1)}{k}$ from the distribution function $F_{N(\mu,\sigma^2)}$ of x, a normal distribution with mean μ and variance σ^2. Therefore, the k categories are approximately evenly distributed.

Semi-continuous variable:: Without loss of generality, we set the constant part of the variable from a semi-continuous distribution to zero. We use two variables from the multivariate normally distributed data. One variable is used to generate a binary variable y with values $y_1, \ldots, y_n$. This is done in the same way as above for binary distributed variables. A second variable $\tilde{x}$ with values $\tilde{x}_1, \ldots, \tilde{x}_n$ determines the non-constant part of the semi-continuous variable z with values $z_1, \ldots, z_n$ by

$$z_i = \begin{cases} 0 & \text{if} \quad y_i = 0 \\ \tilde{x}_i & \text{if} \quad y_i = 1 \end{cases},$$

for $i = 1, \ldots, n$.

11. **Random variables** are added by drawing from an uncorrelated univariate normal distribution.

12. These procedures allow that the correlation structure generated for the multivariate normally distributed data is also reflected in the variables with mixed distribution. In order to avoid complicated notation, the resulting data values are denoted by x_{ij}^{orig}, with $i = 1, \ldots, n$ and $j = 1, \ldots, p$, and the imputed values by x_{ij}^{imp}.

B. Error measures

13. 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

14. For data sets constructed as described in Section A, containing 2 numerical variables, 2 categorical variables, 2 mixed variables, no binary variable and a certain number of random variables *nRandom*, the simulations were carried out 100 times for each combination of $nRandom \in (0, 5, 10, 15, 20)$ and $\rho \in (0.1, 0.3, 0.5, 0.7, 0.9)$.

15. Five different approaches to `kNN` imputation were compared, four of them were described in section II: **knnw**, **knnw2**, **knnl.l1se** and **knnl.lmin**. These methods are compared with **knn** without weighting (weights equal to one for all variables).

16. The results for numerical variables show a clearer pattern than those for categorical variables since the correlation has not been affected as much by discretization. In general, we can see that variable selection and weighting tends to improve the results for data with a higher number of random variables and a high correlation between the non-random variables. For low correlations, Lasso selects 0 variables which is why no results are shown for knnl.l1se and knnl.lmin in such cases. It can also be seen, that knnl.l1se tends to select less variables than knnl.lmin, results for knnl.lmin begin to show for lower correlations than those for knnl.l1se. As to the weighting approach with random forest, we can see that using binomial contrasts instead of categorical variables generally has no effect in this simulation setting.

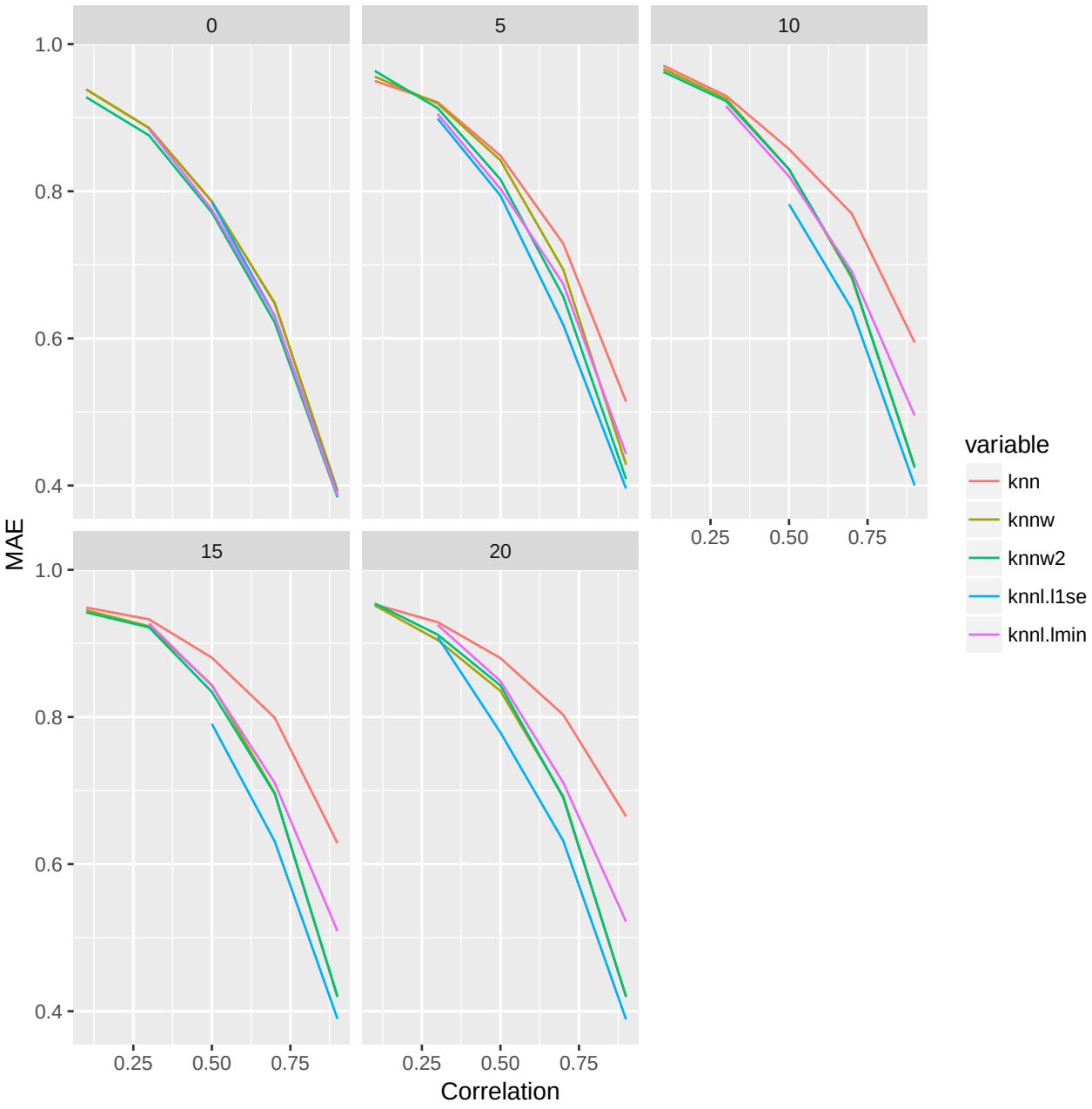

FIGURE 1. Results for continuous variables

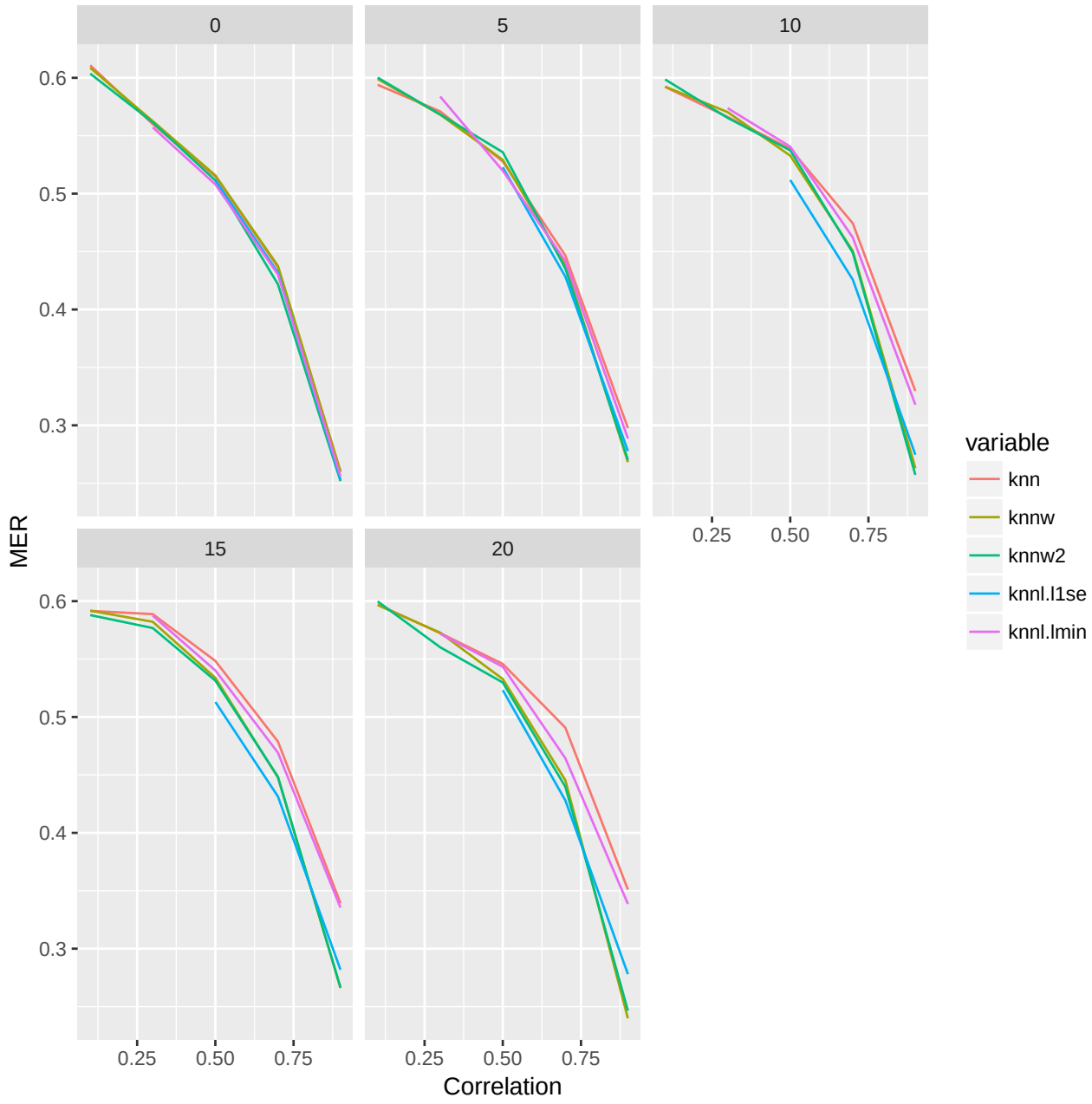

FIGURE 2. Results for categorical variables

IV. Conclusion and future work

17. The presented results are at a very preliminary stage and just the beginning for a better automatic approach of defining the weights for distance variables in the `kNN` procedure. Moreover, the data set used is constructed for the specific purpose and real life data should be used in subsequent study to test if the promising results are just due to the properties of this data set.

18. Another scenario is not to estimate the model (random forest or lasso) on the whole data set, but one a sample to increase performance of this step.

References

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

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.

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

M. Templ, A. Kowarik, and P. Filzmoser. Iterative stepwise regression imputation using standard and robust methods. *Comput Stat Data Anal*, 55(10):2793–2806, 2011.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.