TREE-BASED MULTIPLE IMPUTATION METHODS

Michael Dellermann, Anatol Sluchych, and Jonah Wermter

Freie Universität Berlin

1. Motivation

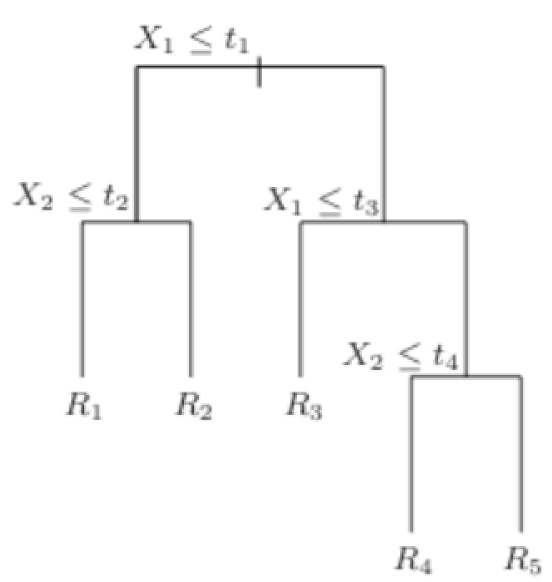
- Parametric MICE methods: conditional models to be specified for *all* variables with missing data (van Buuren & Groothuis-Oudshoorn, 2011)
- Still may fail to capture interactive and nonlinear relations among variables as well as non-standard distributions
- Tree-based methods *automatically* capture interactions, nonlinear relations, and complex distributions with no parametric assumptions or data transformations needed (Burgette & Reiter, 2010)
- Implementation in R: *mice*, *miceRanger*, and *missRanger* packages

2. Tree-based methods

Classification and regression trees (CART):

- seek to approximate conditional distribution of univariate outcome from multiple predictors
- segment predictor space into non-overlapping regions with relatively homogeneous outcomes
- segments found by recursive binary splits of predictors
- prediction for observations that fall into the same region is mean (or mode) of response values for training observations in region
- may be very non-robust and have relatively low predictive accuracy

Figure 1: Example of tree structure. Source: Hastie, Tibshirani, & Friedman (2009)



Random forest:

- *ensemble* method that addresses non-robustness and low predictive accuracy
- average predictions from B non-pruned trees constructed using B bootstrapped training sets
- *decorrelates* trees by performing each split on *randomly* chosen subset of predictors
- accurate model to impute missing values (Stekhoven & Bühlmann, 2012)

3. Imputation algorithm

4-steps algorithm:

1. Initial values for missing values filled in as follows:
   (a) Define matrix Z equal to Y_c (ordered matrix according to missingness)
   (b) Impute missing values in Y_i, $i = 1, \ldots p_1$, using tree-based method on Z and append completed version of Y_i to Z prior to incrementing i
2. Replace originally missing values of Y_i, $i = 1, \ldots p_1$, with tree-based methods on Y_{-i}
3. Repeat step 2 l times (l iterations)
4. Repeat steps 1–3 m times and obtain m imputed sets
5. Pool m datasets to one completed according to Rubin's rules

4. Comparison *mice*, *miceRanger* & *missRanger*

- Packages *mice* and *miceRanger* implement van Buuren's multivariate imputation by chained equations, *missRanger* by default single imputations (based on *missForest*)
- *mice* supports variety of imputation methods, *miceRanger* & *missRanger* only random forest
- All by default use *ranger* package for random forests (van Buuren, 2023; Mayer, 2023; Wilson, 2022), which claims to be faster and more efficient with larger data sets and complex settings than common R packages (Wilson, 2022)
  $\Rightarrow$ core functions written in C++ (faster than R, compiled vs. interpreted code) (Wright & Ziegler, 2017)
- main differences in default values and variety of analytical functions

5. Empirical simulation study

Empirical data set:

- RAND's Health Insurance Experiment: $n = 20185$, $k = 46$

Missing data mechanisms:

- p=25% and 50%
- MAR: $P(mdvis\_miss \mid xage < 25) = p$, $P(mdvis\_miss \mid mhi > 74) = p$
- MCAR: $P(income\_miss) = p$, $P(educdec\_miss) = p$

Monte Carlo simulation: $R = 100$, $M = 5$, $n = 1000$, $niter = 10$, $nrtree = 10$

- six subsets: three focus on data types, three on dataset size

6. Results

Table 1: Simulation results

Metric	Method	Bias	MSE	Coverage
mean(income)	BD	1.98	15,186	0.96
mean(income)	CC	7.39	26,003	0.96
mean(income)	mice-CART	12.44	23,660	0.94
mean(income)	mice-RF	8.48	23,922	0.94
mean(income)	miceRanger	-1.33	22,833	0.92
mean(income)	missRanger	20.02	23,362	0.86
mean(mdvis|xage>25)	BD	0.01	0.05	0.95
mean(mdvis|xage>25)	CC	0.01	0.05	0.95
mean(mdvis|xage>25)	mice-CART	0.01	0.05	1
mean(mdvis|xage>25)	mice-RF	0.01	0.05	1
mean(mdvis|xage>25)	miceRanger	0.01	0.05	0.95
mean(mdvis|xage>25)	missRanger	0.01	0.05	0.95
reg. intercept (ghindx)	BD	0.10	4.03	0.95
reg. intercept (ghindx)	CC	-1.72	28.69	0.93
reg. intercept (ghindx)	mice-CART	0.54	6.43	0.94
reg. intercept (ghindx)	mice-RF	0.78	6.03	0.95
reg. intercept (ghindx)	miceRanger	-1.30	9.55	0.89
reg. intercept (ghindx)	missRanger	-3.36	23.24	0.69

Figure 2: Imputation time per method for levels of missingness

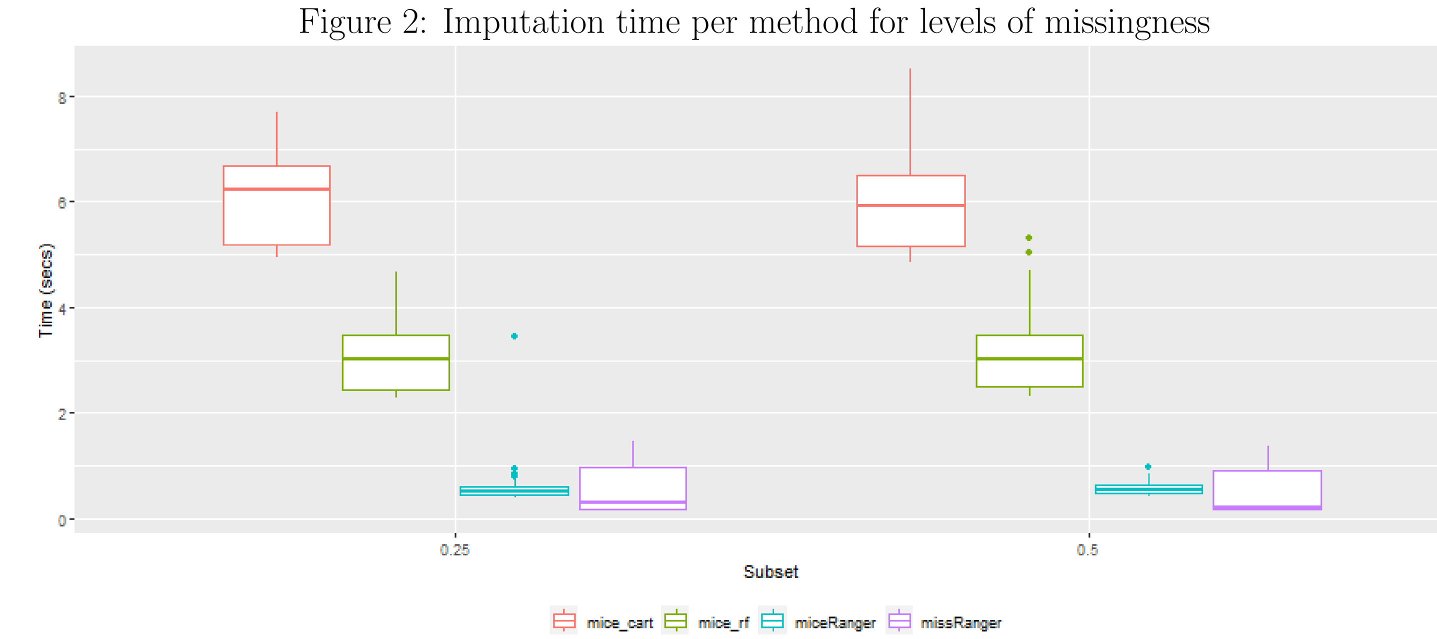


Figure 3: Imputation time per subset per method

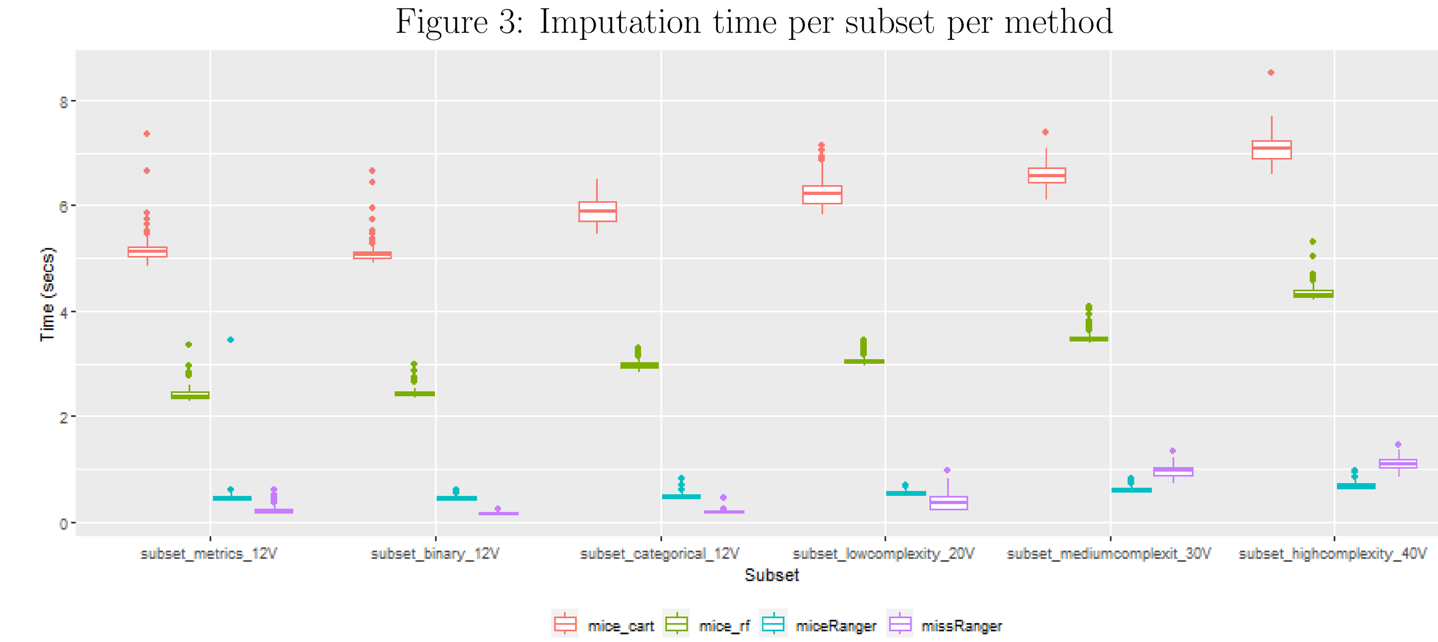
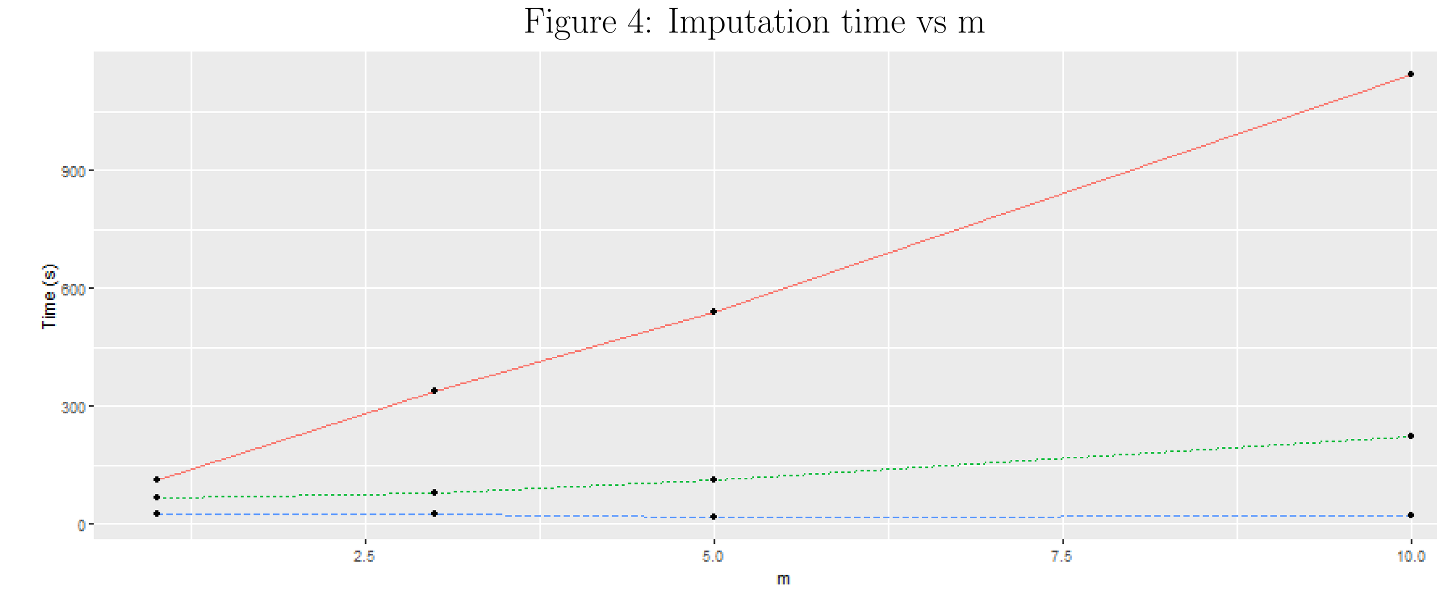


Figure 4: Imputation time vs m



7. Conclusion

- Speed & robustness: *miceRanger* & *missRanger* are ~11x faster than *mice* using standard R implementations and ~6x faster than *mice* using *ranger*.
- Scalability: *missRanger* scales best with increased number of observations, imputations and trees, *miceRanger* performing similarly
- Efficiency: Despite faster speed of *miceRanger* and *missRanger*, *mice* demonstrates superior imputation accuracy
- User-friendliness: *mice* facilitates post-imputation analysis; *miceRanger* demands manual data handling
- Practical recommendation: Choose *miceRanger* for speed, *mice* for in-depth research analysis

References

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

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*