TREE-BASED MULTIPLE IMPUTATION METHODS

Michael Dellermann, Anatol Sluchych, and Jonah Wermter

Freie Universität Berlin

1. Motivation

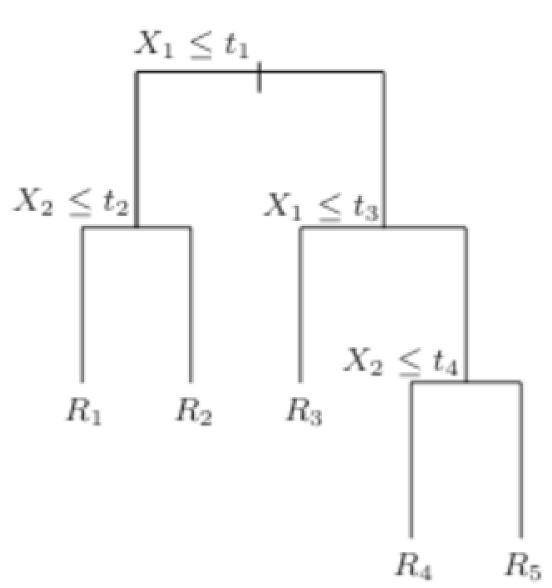
- Parametric MICE methods: conditional models to be specified for *all* variables with missing data (van Buuren & Groothuis-Oudshoorn, 2011)
- Still may fail to capture interactive and nonlinear relations among variables as well as non-standard distributions
- Tree-based methods *automatically* capture 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, 2011)

3. Imputation algorithm

4-steps algorithm:

1. Initial values for the missing values filled in as follows:
   (a) Define a matrix Z equal to Y_c (ordered matrix according to missingness)
   (b) Impute missing values in Y_i, $i = 1, ...p_1$, using tree-based method on Z and append the completed version of Y_i to Z prior to incrementing i
2. Replace the originally missing values of Y_i, $i = 1, ...p_1$, 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)
  - ⇒ 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 with $\rho = 0, \tau = 0$: $P(mdvis\_miss \mid xage < 25) = p$, $P(mdvis\_miss \mid mhi > 74) = p$
- 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	6.66	15,728	0.98
mean(income)	mice-CART	7.87	17,240	0.98
mean(income)	mice-RF	8.19	20,813	0.95
mean(income)	miceRanger	17.45	18,957	0.97
mean(income)	missRanger	3.12	17,711	0.95
mean(mdvis|xage>25)	BD	0.01	0.042	0.96
mean(mdvis|xage>25)	mice-CART	0.01	0.042	1
mean(mdvis|xage>25)	mice-RF	0.01	0.042	1
mean(mdvis|xage>25)	miceRanger	0.01	0.042	0.96
mean(mdvis|xage>25)	missRanger	0.01	0.042	0.96
reg. intercept (ghindx)	BD	0.05	4.85	0.91
reg. intercept (ghindx)	mice-CART	0.67	6.90	0.95
reg. intercept (ghindx)	mice-RF	1.34	6.17	0.96
reg. intercept (ghindx)	miceRanger	1.76	10.96	0.89
reg. intercept (ghindx)	missRanger	3.38	21.98	0.73

Figure 2: Imputation time per subset per method

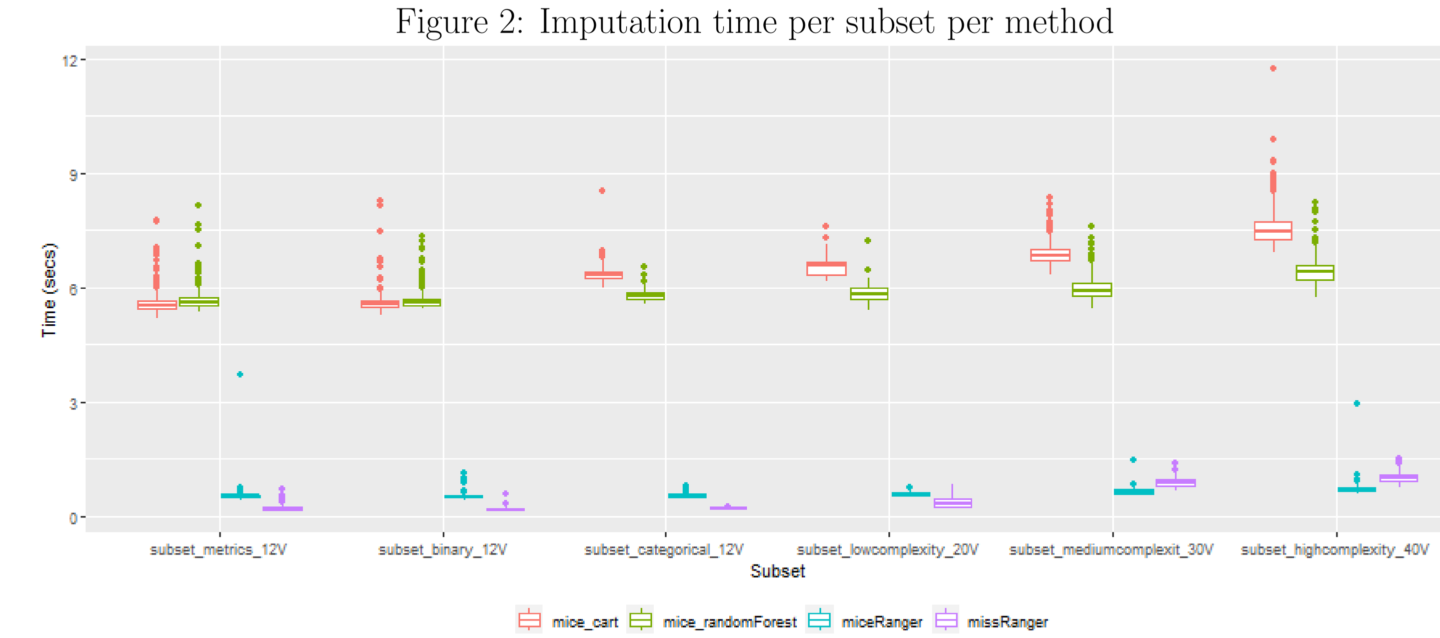
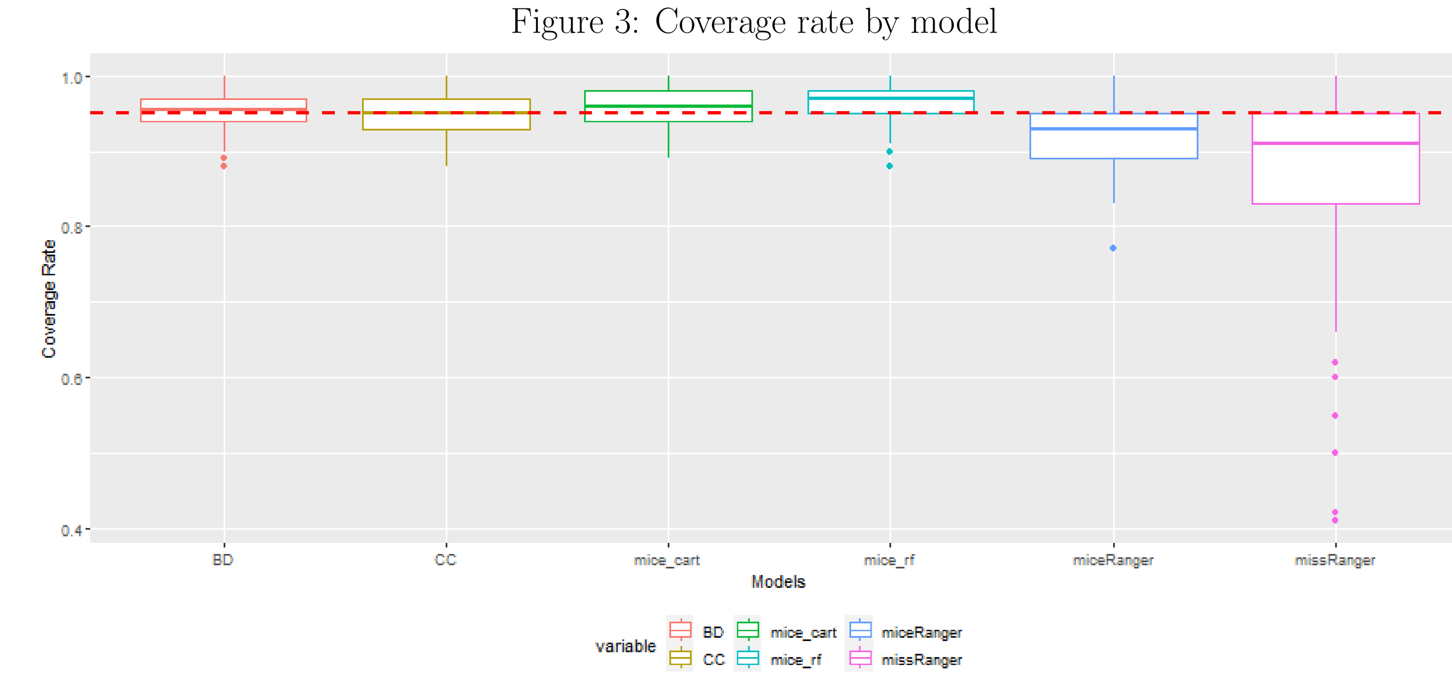


Figure 3: Coverage rate by model



7. Conclusion

- Speed: *miceRanger* and *missRanger* are about X times quicker than *mice*-CART and X times than *mice*-RF
- Accuracy & efficiency: *mice* shows lower MSE and bias for income (MCAR) and uniform accuracy for $mean(mdvis|xage > 25)$ (MAR), maintaining solid coverage.
- Robustness: *miceRanger* and *missRanger* are quicker, yet mice methods maintain consistent imputation times across different missingness levels.
- User-friendliness: *mice* simplifies post-imputation analysis with built-in pooling functions, while *miceRanger* necessitates additional manual steps for regression analysis
- Scalability with m: *missRanger* shows significant scalability, with imputation times remaining stable as the number of imputations (m) increases, demonstrating high efficiency for large-scale imputation tasks.
- Practical Recommendation: For applications prioritizing speed and computational efficiency, miceRanger is advisable for its faster imputation times across varying levels of missing data and model complexities. For research, mice is preferable for its robustness across different missingness patterns and built-in analysis features.

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.). New York: Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning with applications in R*. New York: Springer.

Mayer, M. (2023). *Package 'missranger'*. https://cran.r-project.org/web/packages/missRanger/missRanger.pdf

Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. Cambridge: MIT Press.