Thesis Proposal

Methodology and Statistics for the Behavioural, Biomedical and Social Sciences

Hanne Oberman

2019-09-10

ShinyMICE: an Evaluation Suite for Multiple Imputation

Supervised by Prof. Dr. Stef van Buuren, & Dr. Gerko Vink

Department of Methodology and Statistics

Utrecht University

Requirements: max 750 words (excl. references)

In general the proposal should focus on the relevancy of the project, the gap in the scientific literature, and the feasibility to finish the project within 8 months. Possibly specify a 'step 2' to pursue after the main objective is reached.

Proposal by supervisors

The MICE package in R is a world-leading software package for multiple imputation. When using the mice function to solve missing data, vast amounts of information are calculated and stored. However, the MICE package currently lacks a user-friendly means of assessing this information. This project focuses on developing and programming novel means of evaluating, presenting and organising this information in a web-browser based local app that allows for 1) comparing the imputed data to the observations, 2) inspecting the algorithmic convergence, 3) inspecting multivariate distributions, 4) the plausibility of the imputed data, and so on. During this project you will work closely with the developers of MICE in R and the coding components in this project will focus on R and Shiny.

The objective of the research is to 1) create a Shiny app to investigate and evaluate multiply imputed data sets, 2) implement it in MICE in R, 3) write an instructional vignette, 4) write a technical paper on the workings and usage of the software aimed at e.g. the R Journal or Journal of Statistical Software. The expected output is a state-of-the-art shiny app that can find its way into MICE.

Introduction

Research goals...

develop a valid method to investigate the plausibility of multiply imputed data based on:

- models (imputation and non-response)
- data features (cross-tabs, point estimates, aggregate statistics, etc.)
- assumptions
- algorithmic convergence

Literature review

"There is no clear-cut method for determining when the MICE algorithm has converged" (Van Buuren (2018), §6.5).

"Convergence is diagnosed when the variance between different sequences is no larger than the variance within each individual sequence" (Van Buuren (2018), §6.5).

"Several expository reviews are available that assess convergence diagnostics for MCMC methods (Cowles and Carlin 1996; Brooks and Gelman 1998; El Adlouni, Favre, and Bobée 2006). Cowles and Carlin (1996) conclude that "automated convergence monitoring (as by a machine) is unsafe and should be avoided." No method works best in all circumstances. The consensus is to assess convergence with a combination of tools. The added value of using a combination of convergence diagnostics for missing data imputation has not yet been systematically studied" (Van Buuren (2018), §6.5).

Abayomi, Gelman, and Levy (2008)

Bartlett et al. (2015)

Li et al. (1991)

Rubin (1987)

Rubin (1996)

Vink (n.d.)

Van Buuren and Groothuis-Oudshoorn (2011)

Chang et al. (2019)

Schafer and Graham (2002)

Cowles and Carlin (1996)

Zhu and Raghunathan (2015)

"As MICE is an iterative procedure, it is important that convergence is achieved. This may be checked by computing, at each cycle, the means of imputed values and/or the values of regression coefficients, and seeing if they are stable" (White, Royston, and Wood (2011), p. 394).

"A common diagnostic tool is to plot one or more parameters against the iteration number and assess convergence by how different the variance between different sequences is relative to the variance within each individual sequence, similar to the Gelman-Rubin statistic (Gelman and Rubin, 1992) used in Markov chain Monte Carlo (MCMC) diagnostics" (Li et al. (2014), §4.3).

"R.hat: The value of the R^ statistic used as a convergence criterion. The default is 1.1 (Gelman and Rubin 1992; Gelman, Carlin, Stern, and Rubin 2004)" (Su et al. (2011) p. 4).

"Our mi offers two ways to check the convergence of the multiple imputation procedure. By default, mi() monitors the mixing of each variable by the variance of its mean and standard deviation within and between different chains of the imputation. If the R^ statistic is smaller than 1.1, (i.e., the difference of the within and between variance is trivial), the imputation is considered converged (Gelman, Carlin, Stern, and Rubin 2004). Additionally, by specifying mi(data, check.coef.convergence = TRUE, ...), users can check the convergence of the parameters of the conditional models" (Su et al. (2011), p. 13).

"the convergence properties of FCS are currently under debate due to possible incompatibility (Li, Yu, and Rubin 2012; Zhu and Raghunathan 2015)" (Takahashi (2017), p.)

"The convergence properties of FCS in general settings is still mostly an open question. The behavior of FCS algorithms under non- or quasi-Bayesian imputation procedures like PMM is entirely an open question" (Murray (2018), p.19)

"Imputers who do choose to use FCS should use flexible univariate models wherever possible and take care to assess apparent convergence of the algorithm, for example by computing traces of pooled estimates or other statistics and using standard MCMC diagnostics (Gelman et al., 2013, Chpater 11). It may also be helpful to examine the results of many independent runs of the algorithm with different initializations and to use random scans over the p variables to try to identify any convergence issues and mitigate possible order dependence" (Murray (2018), p. 19).

Approach

R Shiny...

ShinyMICE

- single measure to assess whether algorithm converged (based on Gelman-Rubin statistic?)
- data visualizations pre and post imputation (scatterplots, densities, cross-tabs)
- statistical evaluation of relations between variables pre and post imputation (chi square or t-tests)

References

Additional resources

Bayes course on convergence:

```
# E. Assessing convergence
# ------------------------------------------------------------------------
```

```
# History plot, autocorrelation plot
# History plots show the sampled parameters over the iterations (excluding
# the burn-in).
# The development/pattern in these plots gives an indication of convergence.
# When the history plot is stable (a fat catterpillar), convergence is reached.
# Autocorrelation can be measured at many lags.
# High autocorrelation indicates slow mixing of the random path.

# mcmcplot(mcmcout = samples)
# #sigma seems fine, but b doesn't:
# #even at lag 5 there is still quite some autocorrelation: therefore we need to center the predictors!
# #otherwise we could use a million iterations to deminish this effect

# Gelman-Rubin diagnostic
# The Gelman and Rubin statistic requires you run
# the sampler/algorithm at least twice: These runs are
# referred to as multiple chains.
# It compares the variance between chains to the variance
# within chains (G-R statistic = T/W = (pooled within chain
# var + between chain var)/pooled within chain var.
# It's the red line in the plot, and should be near 1.
# See Gibbs sampler presentation (week 2)
# https://drive.google.com/file/d/1ABHm8ala3c_puVuf32zF8h6TOZ3898uB/view
# pdf p. 41, slide nr 29.
# gelman.plot(samples)
# #this appears to be okay with a really small deviation from 1

# MC error (OPTIONAL?)
# MC error = SD/sqrt(number of iterations)
# SD represents the variation across iterations
# MC error thus represents how much the means differ w.r.t. the iterations
# MC error decreases as number of iterations increases.
# It should not be larger than 5% of the sample standard deviation

# History and density plot (OPTIONAL)
# plot(samples)

# Autocorrelation plots (OPTIONAL)
# autocorr.plot(samples)

# If parameters did not converge, you may:
# . Use (many) more iterations
# . Use a different parametrization (e.g., center predictors)
# . Use different priors (e.g., multivariate normal prior (i.e.,
#   dmnorm(,)) for parameters which are correlated)
# . Use other initial values
```

Abayomi, Kobi, Andrew Gelman, and Marc Levy. 2008. "Diagnostics for Multivariate Imputations." *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 57 (3): 273–91. https://doi.org/10.1111/j.1467-9876.2007.00613.x.

Bartlett, Jonathan W, Shaun R Seaman, Ian R White, and James R Carpenter. 2015. "Multiple Imputation of Covariates by Fully Conditional Specification: Accommodating the Substantive Model." *Statistical Methods in Medical Research* 24 (4): 462–87. https://doi.org/10.1177/0962280214521348.

Chang, Winston, Joe Cheng, J. J. Allaire, Yihui Xie, Jonathan McPherson, RStudio, jQuery Foundation (jQuery library and jQuery UI library), et al. 2019. *Shiny: Web Application Framework for R* (version 1.3.2). https://CRAN.R-project.org/package=shiny.

Cowles, Mary Kathryn, and Bradley P Carlin. 1996. "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review." *Journal of the American Statistical Association* 91 (434): 883–904.

Li, Fan, Michela Baccini, Fabrizia Mealli, Elizabeth R. Zell, Constantine E. Frangakis, and Donald B. Rubin. 2014. "Multiple Imputation by Ordered Monotone Blocks with Application to the Anthrax Vaccine Research Program." *Journal of Computational and Graphical Statistics* 23 (3): 877–92. https://doi.org/10.1080/10618600.2013.826583.

Li, Kim-Hung, Xiao-Li Meng, Trivellore E Raghunathan, and Donald B Rubin. 1991. "Significance Levels from Repeated P-Values with Multiply-Imputed Data." *Statistica Sinica*, 65–92.

Murray, Jared S. 2018. "Multiple Imputation: A Review of Practical and Theoretical Findings." *Statistical Science* 33 (2): 142–59. https://doi.org/10.1214/18-STS644.

Rubin, Donald B. 1987. *Multiple Imputation for Nonresponse in Surveys*. Wiley Series in Probability and Mathematical Statistics Applied Probability and Statistics. New York, NY: Wiley.

———. 1996. "Multiple Imputation After 18+ Years." *Journal of the American Statistical Association* 91 (434): 473–89. https://doi.org/10.2307/2291635.

Schafer, Joseph L, and John W Graham. 2002. "Missing Data: Our View of the State of the Art." *Psychological Methods* 7 (2): 147.

Su, Yu-Sung, Andrew E. Gelman, Jennifer Hill, and Masanao Yajima. 2011. "Multiple Imputation with Diagnostics (Mi) in R: Opening Windows into the Black Box" 45 (2): 1–31. https://doi.org/10.7916/D8VQ3CD3.

Takahashi, Masayoshi. 2017. "Statistical Inference in Missing Data by MCMC and Non-MCMC Multiple Imputation Algorithms: Assessing the Effects of Between-Imputation Iterations." *Data Science Journal* 16 (0): 37. https://doi.org/10.5334/dsj-2017-037.

Van Buuren, Stef. 2018. *Flexible Imputation of Missing Data*. Chapman and Hall/CRC.

Van Buuren, Stef, and Karin Groothuis-Oudshoorn. 2011. "Mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software* 45 (1): 1–67. https://doi.org/10.18637/jss.v045.i03.

Vink, Gerko. n.d. "Towards a Standardized Evaluation of Multiple Imputation Routines."

White, Ian R., Patrick Royston, and Angela M. Wood. 2011. "Multiple Imputation Using Chained Equations: Issues and Guidance for Practice." *Statistics in Medicine* 30 (4): 377–99. https://doi.org/10.1002/sim.4067.

Zhu, Jian, and Trivellore E. Raghunathan. 2015. "Convergence Properties of a Sequential Regression Multiple Imputation Algorithm." *Journal of the American Statistical Association* 110 (511): 1112–24. https://doi.org/10.1080/01621459.2014.948117.