Thesis Proposal

Methodology and Statistics for the Behavioural, Biomedical and Social Sciences

Hanne Oberman (4216318)

2019-09-20

ShinyMICE: an Evaluation Suite for Multiple Imputation

Supervised by prof. dr. Stef van Buuren, & dr. Gerko Vink

Department of Methodology and Statistics

Utrecht University

Word count:

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.

Introduction

At some point, any (social) scientist conducting statistical analyses will run into a missing data problem (Allison (2001)). Simply ignoring the missingness or using ad hoc solutions can yield wildly invalid inferences (Van Buuren (2018)). Multiple imputation (MI; Rubin (1987)) provides a framework to circumvent the *ubiquitous* problem of missing information. This technique – that is growing in popularity (Van Buuren (2018)) – entails 'guessing' the missing values in an incomplete data set several times. The variability between the resulting completed data sets represents how much uncertainty in the inferences is due to missingness (Rubin (1987)).

The R package MICE (Multiple Imputation using Chained Equations; Van Buuren and Groothuis-Oudshoorn (2011)) is the world-leading software for multiple imputation (Van Buuren (2018)). However, the package currently does not provide user-friendly means to evaluate multiply imputed data. *This is vital because of the amount of assumptions in MI algorithms. Thanks to technological developments in computing speed and data storage it has become possible to check assumptions where that was unfeasible before. The very first assumption to check is algorithmic convergence; without convergence, all 'deeper' assumptions and subsequent inferences are invalid.*

Therefore, the goal of this project is develop a MICE evaluation suite, featuring existing modules (like plots to compare the incomplete and completed data sets), and novel assessment tools (like a measure to flag algorithmic non-convergence). The practical relevance of this research project is in facilitating applied researchers in making valid inferences despite of missingness. Simultaneously it contributes to the scientific literature because there is not yet a clear-cut way of diagnosing convergence for MI algorithms. *The research question central to this project is: 'How can empirical researchers be facilitated in evaluating the validity of multiply imputed data?'.*

Literature review

Currently, "there is no clear-cut method for determining when the MICE algorithm has converged" (Van Buuren (2018), §6.5). Instead, users have to rely on visual inspection of the algorithm's iterations. Convergence is said to be reached when parameters (like means of completed variables or regression coefficients) are stable across iterations (White, Royston, and Wood (2011)). And additionally, the variation between imputations should be no larger than the variation across iterations of each individual chain (Van Buuren (2018)). *Conceptually, this visual evaluation procedure is similar to the 'Gelman-Rubin statistic' (Li et al. (2014)). Practically, however, Gelman and Rubin's convergence diagnostic $\widehat{R}$ ('potential scale reduction'; 1992) cannot be directly applied to assess convergence of MICE algorithms because the measure is famously prone to produce false positives when applied to MI data.*
That is, the over-dispersed initial state of the MICE algorithm does not conform to $\widehat{R}$'s assumption of equal starting points (source?).

Notwithstanding, it seems like Su et al. (2011) did implement $\widehat{R}$ as convergence criterion into their R package 'mi', including the conventional cut-off of 1.1. It is worth investigating the validity of the convergence statistic in the context of MICE, and the broader framework of fully conditional specification (FSC) algorithms.

The most recent literature on the topic states that convergence properties of FCS algorithms are still under debate (Takahashi (2017)), with the specific case of a MICE procedure like predictive mean matching posing an entirely open question (Murray (2018)). Van Buuren (2018) summarizes this problem as follows: "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). And that is exactly the gap in the scientific literature that this thesis will contribute to.

Potentially add something about updated (2019) version of $\widehat{R}$ and the new threshold of 1.01, see Vehtari et al. (2019). Or leave it for the Research Report.

Approach that will be used to answer the research question

During this research project I will work directly with developers of the MICE R package. I will use R Shiny (Chang et al. (2019)) to program an interactive evaluation device for the evaluation of multiply imputed data: 'ShinyMICE'. This approach does not require the use of any empirical data – personal or otherwise. Therefore, the project does not need to be reviewed/approved by the faculty's ethical committee.

The research report that is to be handed in December 13th 2019 will consist of an investigation into algorithmic convergence of MICE. Ideally, this will result in a single indicator to flag non-convergence (potentially based on the Gelman-Rubin statistic, auto-correlation, or MC error).

The second stage of the research project is of more practical nature: programming ShinyMICE. The application will at least consist of the following: 1) one or more measures to assess algorithmic convergence; 2) data visualizations (e.g., scatter-plots, densities, and cross-tabulations of the data pre- and post-imputation); and 3) statistical evaluation of relations between variables pre- versus post-imputation (i.e., chi square tests or t-tests). This part of the thesis also entails research into user interface (UI) design, local versus cloud-based data storage, and optimizing the application's efficiency.

A working beta version of ShinyMICE will be considered sufficient to proceed to the next step: writing a technical paper on the workings and usage of the software. The preferred journal for publication of this manuscript is *Journal of Statistical Software* – alternatively, publication in *The R Journal* will be attempted.

Finally, the shinyMICE package will be integrated into the existing MICE environment, and a vignette for applied researchers will be written.

Additional resources (not used)

"For models based on multiple imputed data sets, this [R hat > 1.1] is often a false positive: Chains of different submodels may not overlay each other exactly, since there were fitted to different data" (see https://cran.r-project.org/web/packages/brms/vignettes/brms_missings.html).

"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, Chapter 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).

Gelman and Rubin (1992)'s convergence criterion is defined as the ratio between the sum of the pooled within chain variance plus the between chain variance, and the pooled within chain variance.

"Why can the number of iterations in MICE be so low? First of all, realize that the imputed data Ymis form the only memory in the the MICE algorithm. Chapter 3 explained that imputed data can have a considerable amount of random noise, depending on the strength of the relations between the variables. Applications of MICE with lowly correlated data therefore inject a lot of noise into the system. Hence, the auto-correlation over t will be low, and convergence will be rapid, and in fact immediate if all variables are independent. Thus, the incorporation of noise into the imputed data has pleasant side-effect of speeding up convergence" (Van Buuren (2018), par. 4.5).

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 organizing 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".

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_puVvf32zF8h6TOZ3898uB/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)
```

```r
# . Use different priors (e.g., multivariate normal prior (i.e.,
#   dmnorm(,)) for parameters which are correlated)
# . Use other initial values
```

References

Allison, Paul D. 2001. *Missing Data*. Vol. 136. Sage publications.

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.

Gelman, Andrew, and Donald B. Rubin. 1992. “Inference from Iterative Simulation Using Multiple Sequences.” *Statistical Science* 7 (4): 457–72. https://doi.org/10.1214/ss/1177011136.

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.

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.

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.

Vehtari, Aki, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. 2019. “Rank-Normalization, Folding, and Localization: An Improved $\widehat{}R{}$ for Assessing Convergence of MCMC,” March. http://arxiv.org/abs/1903.08008.

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.