

Research Report

Methodology and Statistics for the Behavioural, Biomedical and Social Sciences

Hanne Oberman (4216318)

2019-10-15

ShinyMICE: an Evaluation Suite for Multiple Imputation

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

Department of Methodology and Statistics

Utrecht University

Goals:

- find an answer to a research question in the field of Methods and Statistics
- get work experience in the field of Methods and Statistics
- a first step in writing a scientific paper: perform research and report the findings in a short research report

Grading:

- The amount of work done
- Work independency
- Clarity and logic of the research report (clear and correct formulations, a logical structure of sections and of paragraphs within sections, no typos or errors in grammar, and flow of sections, like key questions from literature or conclusions from results).
- Nice looking and consistent lay-out of the report (conciseness in presentation of tables, figures, and formulas (if any)).
- General comments

Format (either/or):

- It is written as a (mini) thesis, with an introduction, methods section, some results (i.e., preliminary analyses, or pilot simulations), and a discussion of results. The length of the research report should be maximally 2500 words of text (without references list and/or tables and figures). Please do not include appendices, and no more than 6 tables/figures. Table and Figure captions do not count towards the word limit. An abstract may be included, but is not necessary.
- It is the first half of the thesis, i.e., there are no results included yet, but the report contains a full introduction including a literature review and a methods section that contains details about the data, instruments and/or statistical procedures. There is no (absolute) word limit but the length of the report should comply with recommendations/expectations of the proposed journal. Note that most journals have word limits or rather strong opinions on (too) lengthy manuscripts.

Introduction

- MI (and ad hoc solutions?)

Theoretical Background

- Terminology (MCAR, MAR, MNAR)
- Rubin's rules
- FCS vs. JM

Methods

Replicate Lacerda, Ardington, and Leibbrandt (2007): chains that converged within ten iterations are still flagged as non-converged at iteration fifty (Lacerda, Ardington, and Leibbrandt 2007).

“The monitoring statistic was computed for mean monthly earnings at each iteration in chains of length $k = 200$. Since calculation of the statistic requires M parallel sequences, $m = 5$ such chains were constructed. This value of m was informed by the preferred choice given in the literature on multiple imputation. The monitoring statistic computed at each iteration is presented in Figures 15 to 17 for each of the three missingness mechanisms. The red vertical line denotes ten iterations.” (Lacerda, Ardington, and Leibbrandt 2007, 49)

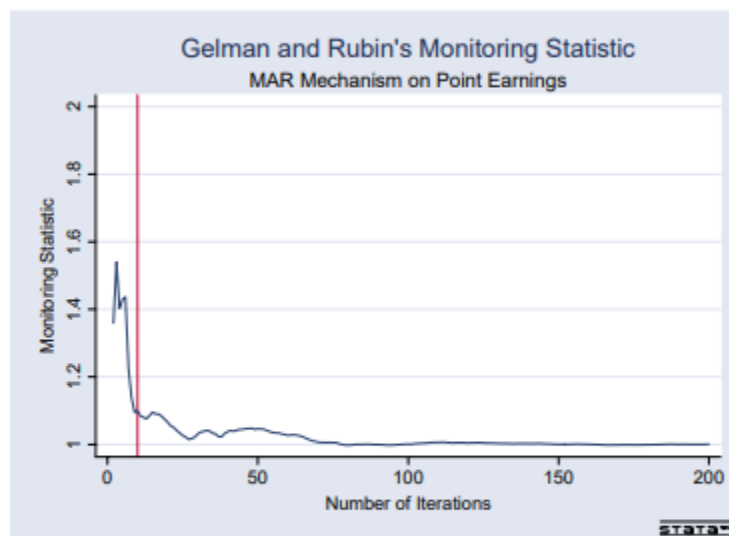


Figure 16: Gelman and Rubin's monitoring statistic at each iteration under an MAR mechanism. Red vertical line corresponds to 10 iterations.

- Potential convergence measures:

why do people say the GR stat will give false positives? perhaps the window (nr of it) is too small. if you start from dispersed, you'll end up with convergence ($GR = 1$) -> the bad thing is to stop too early. this is a false negative, not positive! possible approach: replicate and build a decision rule to solve the problem.

in mice algorithms, iterating is very fast compared to regular MCMC. therefore, we could use the average over the last 5 iterations you'll take the 'majority vote' to indicate conv. so compute GR at each iteration: band and threshold. another approach is to compute autocorrelation (which is decreasing over iterations because we add in noise. schaffer (1997, p. 129) wrote on worst linear stat and calculate the autocorr of that stat we know that the alg did converge. mice has noise so little autocorr. but we have very rich data so we're not estimating everything from scratch. not: we're talking about missing data only, not the combined data (that autocorr. is very high). autocorr is high if we can predict the miss value very well. the higher r^2 the higher the autocorr. so the harder to converge. watch out for the problem with `txp` that will converge at once because you can deductively impute the value per class. possibly use slope of mean too to see whether there is trending. maybe also compute first eigenvalue of the matrix, not just mean and sd. also in pca we have several measures for nr of components, why not do something similar for mice? however, now we use mean and sd of missing values only, and that's not possible for correlations because we'll have different ns for different correls. but if we take the compl. data that should not be a problem because that's constant. maybe look into applying pca on the imputed data and if that stays the same we know that the means and variances are stable too. patrick ... and ian white work on stata proc mi also work on convergence diagnostics -> autocorr function plot in sas of worst linear function, and likelihood ratio that you could save.

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

Additional resources (not used)

quick win: `xyplot` and `hist` have all lattice functions (all sorts of scatterplots!)

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

“For models based on multiple imputed data sets, this [$\hat{R} > 1.1$] 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 Y_{mis} 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).

References

- 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>.
- Lacerda, Miguel, Cally Ardington, and Murray Leibbrandt. 2007. “Sequential Regression Multiple Imputation for Incomplete Multivariate Data Using Markov Chain Monte Carlo.”
- 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>.
- Van Buuren, Stef. 2018. *Flexible Imputation of Missing Data*. Chapman and Hall/CRC.
- Vehtari, Aki, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. 2019. “Rank-Normalization, Folding, and Localization: An Improved \hat{R} for Assessing Convergence

of MCMC,” March. <http://arxiv.org/abs/1903.08008>.