Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

A Note on Convergence Diagnostics for Multiple Imputation Algorithms

Research Report (Preparation for Research Master Thesis, Research Seminar)
Methodology and Statistics for the Behavioural, Biomedical and Social Sciences

Hanne Oberman
Utrecht University

Abstract

This note investigates how algorithmic convergence of multiple imputation procedures—methodology to circumvent the ubiquitous problem of missing data—may be diagnosed. We conclude that potential scale reduction factor $\widehat{R}$ and autocorrelation may be appropriate convergence diagnostics, but their performance differs from conventional applications to iterative algorithmic procedures. These results will be implemented into an interactive framework for the evaluation of multiple imputation methods that is under development as research Master thesis project.

Keywords: multiple imputation, convergence, **mice**, R.

1. Introduction

At some point, any scientist conducting statistical analyses will run into a missing data problem (Allison 2001). Missingness is problematic because statistical inference cannot be performed on incomplete data without employing *ad hoc* solutions (e.g., list-wise deletion), which may yield wildly invalid results (Van Buuren 2018). A popular answer to the ubiquitous problem of missing information is to use the framework of multiple imputation (MI), proposed by Rubin (1987). MI is an iterative algorithmic procedure in which each missing data point is 'imputed' (i.e. filled in) several times. The variability between imputations is used to reflect how much uncertainty in the inference is introduced by the missingness. Therefore, MI can provide valid inferences despite missing information.

To obtain valid inferences with MI, the variability between imputations should be properly represented (Rubin 1987; Van Buuren 2018). If this variability is under-estimated, confidence intervals around estimates will be too narrow, which can yield spurious results. However, over-estimation of the variance between imputations results in unnecessarily wide confidence intervals, which can be costly because it lowers the statistical power. Since both of these situations are undesirable, imputations and their variability should be evaluated. Evaluation measures, however, are currently missing or under-developed in MI software, like the world-leading **mice** package (Van Buuren and Groothuis-Oudshoorn 2011) in R (Core Team 2019).

The goal of this research project is to develop novel methodology and guidelines for evaluating MI methods. These tools will subsequently be implemented in an interactive evaluation framework for multiple imputation to aid applied researchers in drawing valid inference from incomplete datasets. This note provides the theoretical foundation towards the diagnostic evaluation of the convergence of MI algorithms. For reasons of brevity, this note only focuses on the MI algorithm implemented in **mice** Van Buuren and Groothuis-Oudshoorn (2011). To this end, we will evaluate how convergence of the imputation algorithm can be diagnosed.

The convergence properties of the MI algorithm in **mice** are investigated through model-based simulation. The results of this simulation study are guidelines for assessing convergence of MI algorithms. These guidelines will be implemented in an interactive evaluation tool for **mice**, 'ShinyMICE', which is currently under development. All programming code used in this note is available from github.com/gerkovink/ShinyMICE/simulation.

1.1. Terminology

The intended audience of this note consists of anyone who uses multiple imputation to solve missing data problems. Basic familiarity with MI methodology is assumed.[1] This note follows notation and conventions of Van Buuren and Groothuis-Oudshoorn (2011).

Let Y denote an $n \times p$ matrix containing the data values on p variables for all n units in a sample. The collection of observed data values in Y is denoted by Y_{obs}, and will be referred to as 'incomplete' or 'observed' data interchangeably. The missing part of Y is denoted by Y_{mis}. Which parts of Y are missing is determined by the 'missingness mechanism'. This note only considers a 'missing completely at random' (MCAR) mechanism, where the probability of being missing is equal for all $n \times p$ cells in Y.

Figure 1.1 provides an overview of the steps involved with MI. Missing data in Y is 'imputed' (i.e., filled in) m times. The imputed data is combined with observed data Y_{obs} to create m completed data sets. On each completed data set, the analysis of scientific interest (or 'complete data model') is performed. The quantity of scientific interest (e.g., a regression coefficient) is denoted with Q. Since Q is estimated on each completed data set, m separate $\hat{Q}$ values are obtained. These m values are combined into a single pooled estimate $\bar{Q}$.

This note focuses on the algorithmic properties of the imputation step. The algorithm employed within this step has an iterative nature. That is, before drawing m imputed values for each missing data point in Y_{mis}, a 'chain' of potential values is sampled. Each of the m chains starts with an initial value, drawn randomly from Y_{obs}. The chains are terminated after a predefined number of iterations. Only the ultimate sample that a chain lands on is imputed, and subsequently used in the analysis and pooling steps. The collection of samples across

[1] For the theoretical foundation of MI, see Rubin (1987). For an accessible and comprehensive introduction to MI from an applied perspective, see Van Buuren (2018).

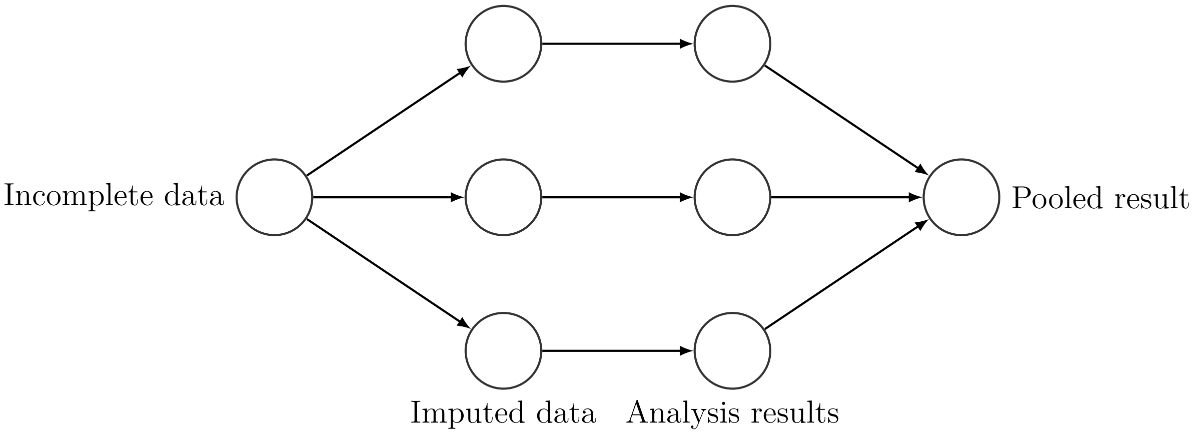

Figure 1: Scheme of the main steps in multiple imputation ($m = 3$). Adapted from (Van Buuren 2018, § 1.4.1).

iterations (between the initial value and the imputed value) for each of the m imputations will be referred to as an 'imputation chain'. The total number of iterations per chain will be denoted with T, where t varies over the integers $1, 2, \dots, T$.

1.2. Theoretical Background

There is no scientific consensus on the convergence properties of multiple imputation algorithms (Takahashi 2017). Some default techniques in **mice** might not yield converged states at all (Murray 2018). Algorithmic convergence should, therefore, be monitored carefully. In that respect is the current evaluation practice insufficient. That is, the recommendation is to visually inspect imputation chains for signs of non-convergence, which may be challenging to the untrained eye (Van Buuren 2018, § 6.5.2). Quantitative evaluation of the convergence of MI algorithms would be preferred.

The multiple imputation algorithm in **mice** is a special case of Markov chain Monte Carlo (MCMC) methods—a framework of iterative algorithmic procedures. Within this framework, several convergence diagnostics have been proposed, but the application of these measures on MI algorithms has not been systematically studied (Van Buuren 2018). This section provides a brief review of: **[INMIDDELS INCORRECT]**1) how convergence is defined in the general context of iterative algorithmic procedures; 2) what diagnostic tools are available; and 3) which of these may apply to MI algorithms.

Loosely speaking, MCMC algorithms aim to sample values from an unknown target distribution (e.g., the posterior distribution of a parameter in Gibbs samplers, or the distribution of missing values in MI algorithms). Convergence is reached once the algorithm samples exclusively from the target distribution (Cowles and Carlin 1996). Since this target distribution is unknown by definition, convergence can only be monitored by evaluating signs of non-convergence (Hoff 2009). In practice, the definition of convergence consists of two components: 'mixing' of chains together, and 'stationarity' of individual chains (Gelman, Carlin, Stern, Dunson, Vehtari, and Rubin 2013, p. 284). The mixing component is satisfied when chains intermingle such that the only difference between the chains is caused by the randomness induced by the algorithm. Stationarity refers to the absence of trending across iterations within chains.

[COULD BE SHORTER!] Most convergence diagnostics target either of the two components. The stationarity component of convergence may be evaluated with autocorrelation (AC; Schafer 1997; Gelman *et al.* 2013), numeric standard error (or 'MC error'; Geweke 1992), and Raftery and Lewis's (1991) procedure to determine the effect of trending within chains. The mixing component can be assessed with the potential scale reduction factor $\widehat{R}$ (a.k.a. 'Gelman-Rubin statistic'; Gelman and Rubin 1992). With an adapted version of $\widehat{R}$, proposed by Vehtari, Gelman, Simpson, Carpenter, and Bürkner (2019), we might also evaluate the stationarity component of convergence. This would make $\widehat{R}$ a general convergence diagnostic. The application of $\widehat{R}$ to assess stationarity has not been thoroughly investigated. Therefore, this study employs both $\widehat{R}$ and autocorrelation to investigate convergence, as recommended by (Cowles and Carlin 1996, p. 898). Other convergence diagnostics are outside the scope of this study.[2]

The convergence diagnostics under consideration are $\widehat{R}$ and autocorrelation. Perfect convergence is reached when the variance between imputation chains is equivalent to the variance within chains ($\widehat{R} = 1$), and there is no dependency between subsequent iterations of imputation chains ($AC = 0$).

To define $\widehat{R}$, we follow notation by (Vehtari *et al.* 2019, p. 5). Let M be the total number of chains, T the number of iterations per chain, and θ the scalar summary of interest (e.g., chain mean or chain variance). For each chain ($m = 1, 2, \dots, M$), we estimate the variance of θ, and average these to obtain within-chain variance W.

$$W = \frac{1}{M} \sum_{m=1}^{M} s_j^2, \text{ where } s_m^2 = \frac{1}{T-1} \sum_{t=1}^{T} \left(\theta^{(tm)} - \bar{\theta}^{(\cdot m)} \right)^2 .$$

We then estimate between-chain variance B as the variance of the collection of average θ per chain.

$$B = \frac{T}{M-1} \sum_{m=1}^{M} \left(\bar{\theta}^{(\cdot m)} - \bar{\theta}^{(\cdot \cdot)} \right)^2 , \text{ where } \bar{\theta}^{(\cdot m)} = \frac{1}{T} \sum_{t=1}^{T} \theta^{(tm)}, \quad \bar{\theta}^{(\cdot \cdot)} = \frac{1}{M} \sum_{m=1}^{M} \bar{\theta}^{(\cdot m)}$$

From the between- and within-chain variances we compute a weighted average, $\widehat{\text{var}}^{+}$, which over-estimates the total variance of θ. $\widehat{R}$ is then obtained as a ratio between the over-estimated total variance and the within-chain variance:

$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^{+}(\theta|y)}{W}}, \text{ where } \widehat{\text{var}}^{+}(\theta|y) = \frac{N-1}{N} W + \frac{1}{N} B.$$

[2] All of these methods evaluate the convergence of univariate scalar summaries (e.g., chain means or variances). These convergence diagnostics cannot diagnose convergence of multivariable statistics (i.e., relations between scalar summaries). Van Buuren (2018) proposed to implement multivariable evaluation through eigenvalue decomposition MacKay and Mac Kay (2003). This method is outside of the scope of the current study.

We can interpret $\widehat{R}$ as potential scale reduction factor since it indicates by how much the variance of θ could be shrunken down if an infinite number of iterations per chain would be run (Gelman and Rubin 1992). This interpretation assumes that chains are 'over-dispersed' at $t = 1$, and reach convergence as $T \to \infty$. Over-dispersion implies that the initial values of the chains are 'far away' from the target distribution and each other. The iteration at which chains are independent of their initial values is the point where mixing occurs. This is diagnosed with $\widehat{R}$-values close to one. High $\widehat{R}$-values thus indicate slow mixing of chains, which is a sign of non-convergence. The conventionally acceptable threshold for this convergence diagnostic was $\widehat{R} < 1.2$ (Gelman and Rubin 1992). Recently, Vehtari *et al.* (2019) proposed a more stringent threshold: $\widehat{R} < 1.01$.

Following the same notation, we define autocorrelation as the correlation between two subsequent θ-values within the same chain[3](Lynch 2007, p. 147).

$$AC = \left(\frac{T}{T-1}\right) \frac{\sum_{t=1}^{T-1}(\theta_t - \bar{\theta}^{(\cdot m)})(\theta_{t+1} - \bar{\theta}^{(\cdot m)})}{\sum_{t=1}^{T}(\theta_t - \bar{\theta}^{(\cdot m)})^2}.$$

We can interpret AC-values as a measure of dependence within chains. Positive AC-values indicate recurrence (i.e., similarity between θ-values of subsequent iterations), which may lead to trending. Negative AC-values occur when the θ-values of subsequent iterations diverge from one-another. Dependence in this sense does not threaten the stationarity component of convergence. On the contrary even—it may increase the variance of θ. **speed up convergence???** As convergence diagnostic, the interest is in positive AC-values.[4]

1.3. Simulation Hypothesis

This study evaluates whether $\widehat{R}$ and AC could diagnose convergence of multiple imputation algorithms. There is, however, no baseline measure available to evaluate their performance against. [5] We will assess the appropriateness of the two convergence diagnostics in comparison to evaluation criteria recommended by (Van Buuren 2018, § 2.5.2). The performance diagnostics comprise of average bias, average confidence interval width, and empirical coverage rate across simulations.[6]

[**HIER NOG ZOOI**] We hypothesize that $\widehat{R}$ may not be an appropriate diagnostic, because it assumes over-dispersed initial values. In **mice**, initial values of the algorithm are chosen randomly from the observed data. Therefore, we cannot be certain that the initial values are over-dispersed. Without over-dispersed initial states, $\widehat{R}$ may falsely diagnose convergence (Brooks and Gelman 1998). This suggests that $\widehat{R}$ would not be sensitive enough to flag non-convergence of MI algorithms. However, an empirical finding by Lacerda, Ardington, and Leibbrandt (2007) shows that $\widehat{R}$ can indicate non-convergence at unconventionally long chain lengths. While the default number of iterations in **mice** is five, $\widehat{R} > 1.1$ was observed after 50 iterations, and $\widehat{R} > 1.01$ up-to 150 iterations. This suggests that $\widehat{R}$ would over-estimate non-convergence of MI algorithms.

[3] In this study we only consider AC at lag 1, i.e., the correlation between the t^{th} and $t + 1^{th}$ iteration of the same chain.

[4] The magnitude of AC-values can also be evaluated statistically, but that is outside of this note's scope.

[5] The current practice of visually inspecting imputation chains for signs of non-mixing or non-stationarity can only diagnose severely pathological cases of non-convergence.

[6] We could also look at distributional characteristics, and plausibility of imputed values, see Vink (n.d.). For now, this is outside of the scope of this study.

We hypothesize that $\widehat{R}$ will over-estimate non-convergence. No hypothesis was formulated about AC, results are exploratory.

2. Methods

We use model-based simulation in R to perform multiple imputation under 100 simulation conditions. In each condition, the MI algorithm comprises a different imputation chain length ($T = 1, 2, \ldots, 100$). The number of simulation runs per condition is 1000. For each simulation condition (i.e., number of iterations) and each repetition (i.e., simulation run) we compute convergence diagnostics ($\widehat{R}$ and autocorrelation) and simulation diagnostics (bias, confidence interval width and coverage). These diagnostics are aggregated across repetitions to obtain averages per simulation condition. The simulation set-up consists of several steps, summarized in the pseudo-code below.[7]

```
# pseudo-code of simulation
simulate data
for (number of simulation runs from 1 to 1000)
  for (number of iterations from 1 to 100)
    create missingness
    impute the missingness
    compute convergence diagnostics
    perform analysis
    pool results
    compute simulation diagnostics
aggregate convergence and simulation diagnostics
```

2.1. Data simulation

The point of origin for all simulation conditions is a finite population of $N = 1000$. The data are simulated to solve a multiple linear regression problem, where dependent variable Y is regressed on independent variable X, and covariates Z_1 and Z_2: $Y \sim \beta_1 X + \beta_2 Z_1 + \beta_3 Z_2$. The quantity of scientific interest is regression coefficient β_1. The data generating model is a multivariate normal distribution with means structure μ and variance-covariance matrix Σ.

$$\begin{pmatrix} X \\ Z_1 \\ Z_2 \\ \epsilon \end{pmatrix} \sim N \left[\begin{pmatrix} 12 \\ 3 \\ 0.5 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 & 4 & 1.8 & 0 \\ 4 & 16 & 4.8 & 0 \\ 1.8 & 4.8 & 9 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix} \right]$$

Outcome variable Y is subsequently calculated as $Y = 2X + .5Z_1 - Z_2 + \epsilon$.

[7] The complete R script of the simulation study is available on Github.

2.2. Amputation

The complete dataset is 'amputed' once in each simulation repetition. That is, the **mice** function `ampute()` is used to impose a missingness mechanism upon the data. The missingness is univariate, and the probability to be missing is the same for all variables, namely 20% (`prop = 0.8, mech = "MCAR"`). This leaves 20% of the rows completely observed.

2.3. Imputation

Missing data points are imputed with the **mice** function `mice()`. All simulations are performed with Bayesian linear regression imputation (`method = "norm"`), and five imputation chains (`m = 5`). The number of iterations varies between simulation conditions (`maxit = 1, 2, ..., 100`).

2.4. Convergence Diagnostics

We investigate convergence by extracting not only the imputed values of each MI procedure (the T^{th} iteration), but also information (θ-values) from the intermediate iterations ($t = 1, 2, \ldots, T$). The θs of interest are chain means and chain variances per variable. We compute convergence diagnostics separately for each of the four variables in the dataset, and report the maximum (absolute) value as the 'worst linear predictor' of convergence [**Rephrase or footnote??**]. $\widehat{R}$ is computed by implementing the definition given in **EQUATION NR**. AC is computed with **stats** function `acf()`.

2.5. Analysis

To estimate the quantity of scientific interest, Q, we perform multiple linear regression on each completed dataset with the **stats** function `lm()`. We obtain an estimated regression coefficient per imputation, which are pooled into a single estimate, $\bar{Q}$. We use the **mice** function `pool()` to get variance estimates according to Rubin's 1987 rules, and utilize these to implement finite population pooling as per (Vink and van Buuren 2014).

2.6. Simulation diagnostics

We compute bias as the difference between $\bar{Q}$ and Q. Confidence interval width (CIW) is defined as the difference between the lower and upper bound of the 95% confidence interval around $\bar{Q}$ (CI95%). We compute the CI95% bounds as $\bar{Q} \pm 2.66 \times SE_{\bar{Q}}$, where 2.66 is the quantile of a t-distribution with $m - 1$ degrees of freedom, and $SE_{\bar{Q}}$ is the square root of the pooled variance estimate. From bias and CIW, we calculate empirical coverage rates. Coverage rate is the proportion of simulations in which Q is between the bounds of the CI95% around $\bar{Q}$.

3. Results

Figures 2 and 3 display results per simulation condition ($T = 1, 2, \ldots, 100$). A subset of simulation conditions is presented in Table 1.

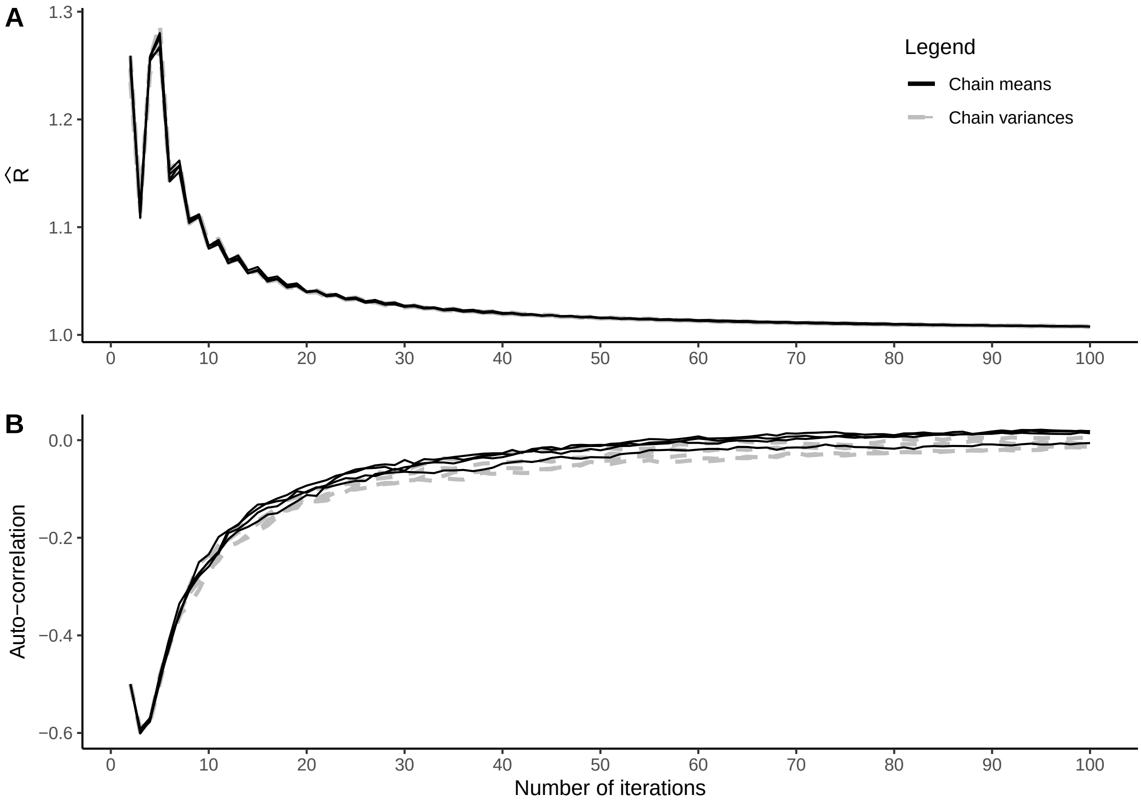

Figure 2: Convergence diagnostics over 1000 MCMC simulations.

3.1. Convergence diagnostics

It is apparent that there is a relation between convergence diagnostics $\widehat{R}$ and AC, and the number of iterations per simulation condition(T). Generally speaking, we see a trend towards convergence as T increases with each simulation condition ($\widehat{R}$-values approach one, and AC-values approach zero). The convergence diagnostics show more or less equivalent trends for chain means versus chain variances. We, therefore, only discuss the $\widehat{R}$ and AC-values of chain means.

Figure 2A shows that $\widehat{R}$-values generally decrease with increasing imputation chain lengths. The decline stabilizes somewhere between the simulation conditions $T = 30$ and $T = 50$. The downward trend is most pronounced for $T = 3$, and between $T = 5$ and $T = 10$. In the intervening conditions ($3 \leq T \leq 5$), however, we observe a steep increase in $\widehat{R}$-values. This increase implies that convergence cannot be diagnosed at $T = 3$, as would be the case based on the conventional threshold $\widehat{R} < 1.2$. [**This may be spurious??**] According to the widely used threshold $\widehat{R} < 1.1$, convergence can be diagnosed for conditions where $T > 9$. If we use the recently recommended threshold $\widehat{R} < 1.01$, we would conclude that we cannot diagnose convergence in any simulation condition. [**Add something about $\widehat{R} < 1$?**]

The AC-values displayed in Figure 2B are almost uniformly increasing as a function of T. The only AC-value that deviates from this observation is for $T = 2$. The lowest AC-value is obtained for $T = 3$. At $T = 5$, the AC-value reaches the level observed at $T = 2$. Between $T = 10$ and $T = 30$ the increase plateaus off, and AC-values are stable around $T = 70$. AC

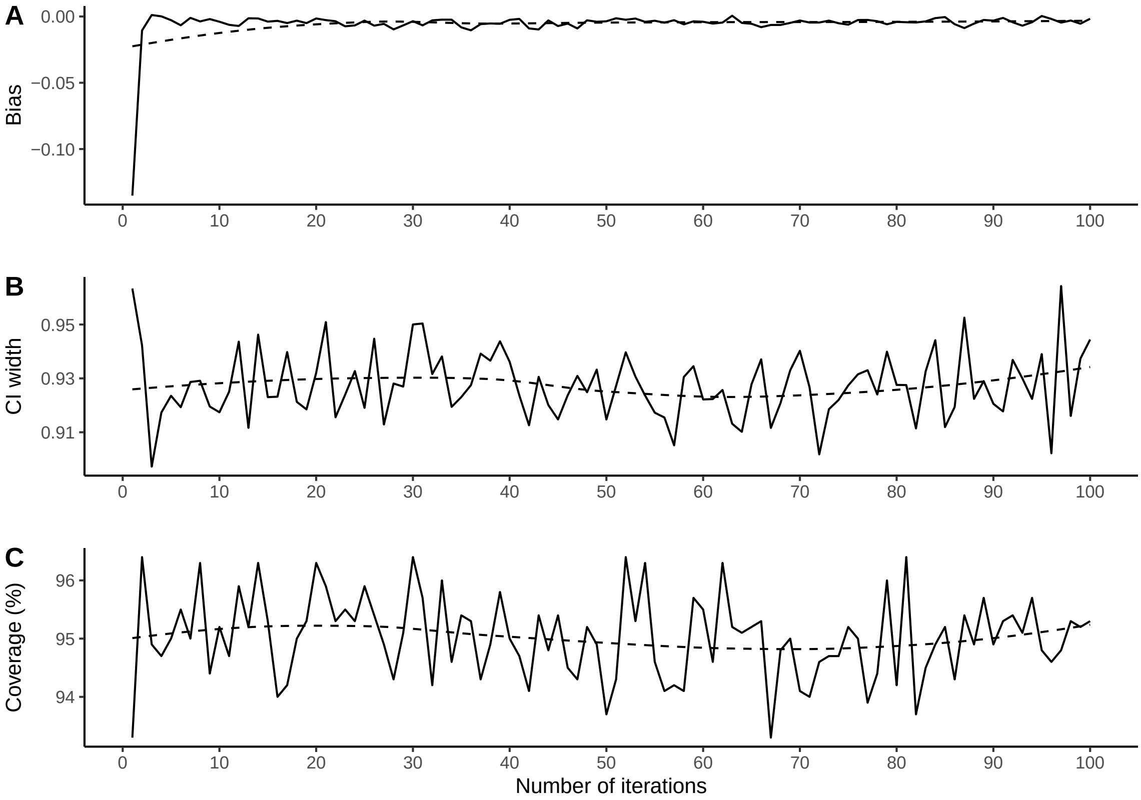

Figure 3: Simulation diagnostics over 1000 MCMC simulations.

is then indifferentiable from zero, indicating stationarity. According to this diagnostic, none of the simulation conditions show signs of non-convergence, since we only observe negative or zero AC-values.

Taken together, we see that convergence may be diagnosed in simulation conditions where $T > 10$.

3.2. Simulation diagnostics

We use average bias, average confidence interval width, and coverage rate as performance measures to evaluate $\widehat{R}$ and AC. We make the general observation that the simulation diagnostics behave as theorized for most simulation conditions (bias around zero, coverage rate around 95%).

Figure 3A shows that bias is fairly stable across simulation conditions. The condition $T = 1$ clearly deviates from this trend with a negative bias. The bias for $T = 2$ is below average, but within the range of fluctuations. Conditions where $T > 3$ **are OK, Loess line is flat for** $T > 20$.

CIW is only clearly divergent for the simulation condition $T = 1$. However, we want to be on the safe side, and not under-estimate the variance. Otherwise, we might end up with spurious inferences. Conditions where $T > 3$ have similar CIWs, which seems to be stable across iterations. **Loess line is never completely flat.**

Table 1: Simulation and convergence diagnostics over 1000 MCMC simulations.

T	Bias	CI width	Cov. rate	$\widehat{R}_{mean}$	$\widehat{R}_{var}$	AC_{mean}	AC_{var}
1	-0.137	0.954	0.932	NA	NA	NA	NA
2	-0.006	0.932	0.953	1.650	1.632	-0.500	-0.500
3	0.002	0.929	0.944	1.314	1.306	-0.660	-0.659
4	0.003	0.933	0.957	1.461	1.457	-0.733	-0.735
5	0.004	0.935	0.954	1.475	1.472	-0.705	-0.706
6	0.001	0.934	0.956	1.258	1.256	-0.656	-0.646
7	0.002	0.930	0.948	1.265	1.269	-0.591	-0.585
8	0.004	0.930	0.954	1.181	1.178	-0.495	-0.516
9	0.003	0.931	0.956	1.187	1.187	-0.442	-0.459
10	0.003	0.952	0.943	1.141	1.140	-0.403	-0.423
15	0.002	0.942	0.965	1.100	1.100	-0.276	-0.289
25	0.005	0.934	0.955	1.057	1.057	-0.143	-0.159
50	-0.002	0.929	0.959	1.027	1.026	-0.053	-0.075
100	0.000	0.920	0.946	1.013	1.013	0.022	-0.017

Empirical coverage rate seems more or less stable for conditions where $T > 1$. We see some over-coverage at $T = 2$. We might over-estimate the variance, but better than under-estimating it. On average, the coverage rate is somewhat higher than the expected nominal coverage of 95% Neyman (1934), namely 95.3%. Loess line is never flat.

From the simulation diagnostics, we see that simulation conditions where $T > 3$ are generally OK. The performance measures indicate that for the current data, MI procedure, and quantity of interest, the algorithm should have maxit argument of four or more. This is, however, a simple simulation problem, with MCAR missingness mechanism, univariate missingness, and moderate fmi's [REMOVE?? COMPUTE??]. harder problems require more iterations.

There is a discrepancy between what the convergence diagnostics and the performance measures indicate. While $T = 4$ is OK with respect to simulation quantities, $T = 3$ is the worst possible value according to the convergence diagnostics, and $T = 4$ provides marginally better convergence diagnostics.

4. Summary and discussion

This note shows that the convergence of multiple imputation algorithms may be evaluated with convergence diagnostics $\widehat{R}$ and AC. These measures, however, indicate non-convergence at imputation chain lengths that produce acceptable simulation diagnostics. This may be due to the quantity of scientific interest chosen. More 'complicated' Qs (e.g., higher order effects or variance components) might show bias, under- or over-coverage at higher T.

At least $\widehat{R}$ does not under-estimate non-convergence. It's better to be too strict than too lenient. According to this simulation study, the recently proposed threshold $\widehat{R}$ is too stringent for MI algorithms, and the conventionally acceptable $\widehat{R} < 1.2$ is too lenient.

Summary. This note illustrates that conventional convergence diagnostics behave differently on MI data than other MCMC methods like Gibbs samplers in Bayesian analyses. The most recent recommended threshold for $\widehat{R}$ ($\widehat{R} < 1.01$) may be too stringent for MI data. **However, using the 1.2 threshold may be too lenient because this was observed at three iterations and then again after six. If we would use 1.2, we might miss this increase after three iterations. The 1.1 threshold seems OK. Also, ten iterations are computationally not terrible. With increased performance and storage capacities, doubling the default is fine I guess.**

From the simulation diagnostics, it appears that as little as four iterations might be sufficient to obtain unbiased, confidence valid estimates. Convergence diagnostics $\widehat{R}$ and autocorrelation, however, indicate that convergence may only be reached after twenty or even forty iterations.

R hat below 1. $\widehat{R}$ could theoretically not be smaller than one, yet it occurred several times in this study (see online appendix XYZ). This can happen when the number of simulations is smaller than in 'regular' MCMC processes [explain that fewer iterations is an advantage of MI, not a disadvantage compared to MCMC]. Increasing Rhat values can happen if the initial values were not appropriately over-dispersed (Brooks and Gelman 1998, p 438).

Negative ACs. autocorrelation is dangerous when positive. These autocorrelations are negative. Still, we want the iterations to be stable and independent. Default maxit is five iterations now, should this be different? Are the 'waves' most pronounced at iteration 5? But why the dip??

Assessing the stationarity component of convergence with AC might be redundant, since high AC-values are implausible in MI procedures. That is, the randomness induced by the MI algorithm effectively mitigates the risk of dependency within chains. AC would thus not be informative of the convergence of MI algorithms.

The observed dip in AC implies that default maxit value of five iterations is the worst possible number of iterations.

Future research. This study considered only an MCAR missingness mechanism. Necessary but not sufficient to demonstrate the performance of convergence diagnostics. This is just a proof of concept. Further research is needed to investigate the performance under violation of convergence, e.g. dependency between predictors (very high correlations).

Computational details

The results in this paper were obtained using R 3.6.1 Core Team (2019) with the **mice** 3.6.0.9000 package Van Buuren and Groothuis-Oudshoorn (2011). R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`.

Acknowledgments

This note is part of a Research Master's thesis project. It was written by the sole author (Hanne Oberman, BSc.), with guidance from Master thesis supervisors prof. dr. Stef van Buuren and dr. Gerko Vink, and research seminar mentor dr. Daniela Cianci.

References

Allison PD (2001). *Missing data*. Sage publications.

Brooks SP, Gelman A (1998). "General Methods for Monitoring Convergence of Iterative Simulations." *Journal of Computational and Graphical Statistics*, **7**(4), 434–455. doi:10.1080/10618600.1998.10474787.

Core Team (2019). *R: A language and environment for statistical computing*. Vienna, Austria. Tex.organization: R Foundation for Statistical Computing, URL https://www.R-project.org/.

Cowles MK, Carlin BP (1996). "Markov chain Monte Carlo convergence diagnostics: a comparative review." *Journal of the American Statistical Association*, **91**(434), 883–904. ISSN 0162-1459.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2013). *Bayesian Data Analysis*. CRC Press LLC, Philadelphia, PA, United States.

Gelman A, Rubin DB (1992). "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science*, **7**(4), 457–472. doi:10.1214/ss/1177011136.

Geweke J (1992). "Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments." *Bayesian statistics*, **4**, 641–649.

Hoff PD (2009). *A First Course in Bayesian Statistical Methods*. Springer Texts in Statistics. Springer New York, New York, NY. doi:10.1007/978-0-387-92407-6.

Lacerda M, Ardington C, Leibbrandt M (2007). "Sequential regression multiple imputation for incomplete multivariate data using Markov chain Monte Carlo." *Technical report*, University of Cape Town, South Africa.

Lynch SM (2007). *Introduction to applied Bayesian statistics and estimation for social scientists*. Springer Science & Business Media.

MacKay DJ, Mac Kay DJ (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Murray JS (2018). "Multiple Imputation: A Review of Practical and Theoretical Findings." *Statistical Science*, **33**(2), 142–159. doi:10.1214/18-STS644.

Neyman J (1934). "On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection." *Journal of the Royal Statistical Society*, **97**(4), 558–625. doi:10.2307/2342192.

Raftery AE, Lewis S (1991). "How many iterations in the Gibbs sampler?" *Technical report*, Washington University Seatle, Department of Statistics, United States.

Rubin DB (1987). *Multiple Imputation for nonresponse in surveys*. Wiley series in probability and mathematical statistics Applied probability and statistics. Wiley, New York, NY.

Schafer JL (1997). *Analysis of incomplete multivariate data*. Chapman and Hall/CRC.

Takahashi M (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**, 37. doi:10.5334/dsj-2017-037.

Van Buuren S (2018). *Flexible imputation of missing data*. Chapman and Hall/CRC.

Van Buuren S, Groothuis-Oudshoorn K (2011). "mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software*, **45**(1), 1–67. ISSN 1548-7660. doi:10.18637/jss.v045.i03. URL https://www.jstatsoft.org/index.php/jss/article/view/v045i03.

Vehtari A, Gelman A, Simpson D, Carpenter B, Bürkner PC (2019). "Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC." ArXiv: 1903.08008, URL http://arxiv.org/abs/1903.08008.

Vink G (????). "Towards a standardized evaluation of multiple imputation routines."

Vink G, van Buuren S (2014). "Pooling multiple imputations when the sample happens to be the population." *arXiv:1409.8542 [math, stat]*. ArXiv: 1409.8542 version: 1, URL http://arxiv.org/abs/1409.8542.

Affiliation:

Hanne Ida Oberman, BSc.
Methodology and Statistics for the Behavioural, Biomedical and Social Sciences
Department of Methodology and Statistics
Faculty of Social and Behavioral Sciences
Utrecht University
Heidelberglaan 15, 3500 Utrecht, The Netherlands
E-mail: h.i.oberman@uu.nl URL: https://hanneoberman.github.io/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd