Missing The Point: Non-Convergence in Iterative Imputation Algorithms

Hanne Oberman

17 september 2021

Abstract

Iterative imputation has become the de facto standard to accommodate for the ubiquitous problem of missing data. While it is widely accepted that this technique can yield valid inferences, these inferences all rely on algorithmic convergence. Our study provides insight into identifying non-convergence in iterative imputation algorithms. We show that these algorithms can yield correct outcomes even when a converged state has not yet formally been reached. In the cases considered, inferential validity is achieved after five to ten iterations, much earlier than indicated by diagnostic methods. We conclude that it never hurts to iterate longer, but such calculations hardly bring added value.

Contents

1 Notation in this manuscript:

- square brackets are comments for myself/stuff I need to review
- bullet points are things I need to expand
- asterisks denote a weak 'segway' between sentences

2 Intro

Iterative imputation has become the de facto standard to accommodate missing data in social scientific research [ref: e.g., rubin after 18 years paper? or murray 2018?]. The technique enables researchers to draw valid inferences when faced with [phrasing: despite of?] incomplete observations. Every missing cell in an incomplete dataset is imputed (i.e., filled in) using an imputation algorithm. And once the dataset is completed, an analysis of scientific interest can be performed. This analysis will yield unbiased and confidence-valid estimates when both the missing data problem and the scientific problem are appropriately considered. The validity of this whole process naturally depends on the convergence of the algorithm that was used to to generate the imputed values. Yet, there has not been a systematic study on how to evaluate the convergence of iterative imputation algorithms.

Determining whether an algorithm has converged is not trivial, especially in the context of iterative imputation.* It is challenging to arrive upon a single point at which convergence has been reached. Since the aim of iterative imputation is to converge to a distribution and not to a single point, the algorithm may produce some fluctuations even after it has converged. Because of this property, it may be more desirable to focus on *non*-convergence. A widely accepted practice is visual inspection of the algorithm (Raghunathan and Bondarenko 2007; Van Buuren 2018). Diagnosing non-convergence through visual inspection may, however, be undesirable: 1) it may be challenging to the untrained eye, 2) only severely pathological cases of non-convergence may be diagnosed, and 3) there is not an objective measure that quantifies convergence (Van Buuren 2018). [add argument: "Inspection of such plots is a notoriously unreliable method of assessing convergence and in addition is unwieldy when monitoring a large number of quantities of interest, such as can arise in complicated hierarchical models" (Gelman et al., 2013, p. 285).] Therefore, a quantitative diagnostic method to assess convergence would be preferred. Fortunately, there are non-convergence identifiers for other iterative algorithms, but the validity of these identifiers has not been systematically evaluated on imputation algorithms.

In this study, we explore different methods for identifying non-convergence in iterative imputation algorithms. We evaluate whether these methods are able to cover the extent of the non-convergence, and we also investigate the relation between non-convergence and the validity of the inferences. We translate the results of our simulation study into guidelines for practice, which we demonstrate by means of a case study.

3 Simulation Set-Up

We investigate non-convergence in iterative imputation through model-based simulation in R (version 4.0.3; R Core Team 2020). We provide a summary of the simulation set-up in Algorithm 1; the complete script and technical details are available from github.com/hanneoberman/MissingThePoint.

Algorithm 1: simulation set-up (pseudo-code)

```
for each simulation repetition
  1. simulate complete data
  for each missingness condition
    2. create missingness
    for each iteration
      3. impute missingness
      4. perform analysis of scientific interest
      5. apply non-convergence identifiers
      6. pool results across imputations
      7. compute performance measures
    8. combine outcomes from all iterations
  9. combine outcomes from all missingness conditions
10. aggregate outcomes from all simulation repetitions
```

3.1 Aims

With this simulation, we assess the impact of non-convergence on the validity of scientific estimates obtained using the imputation package `{mice}` (Van Buuren and Groothuis-Oudshoorn 2011). Inferential validity is reached when estimates are both unbiased and have nominal coverage across simulation repetitions ($n_{sim} = 1000$). To assess non-convergence, we terminate the algorithm after a varying number of iterations ($n_{it} = 1, 2, ..., 50$). We differentiate between nine different missingness scenarios that are defined by the data generating mechanism.

3.2 Data generating mechanism

Data are generated in each simulation repetition for a complete set of $n_{obs} = 1000$ cases (i.e., before inducing missingness). We define three multivariately normal random variables, let

$$\begin{pmatrix} Y \\ X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & & \\ 0.5 & 1 & \\ -0.5 & 0.5 & 1 \end{pmatrix} \right].$$

The complete set is amputed according to nine missingness conditions. We use a 3×3 factorial design consisting of three missingness mechanisms and three proportions of incomplete cases.

- we use the three missingness mechanisms MCAR, MAR, and MNAR with defaults settings from `mice::ampute()` (e.g., right-tailed MAR)
- we use a multivariate missing data pattern and make 25%, 50%, and 75% of cases incomplete

3.3 Estimands

We impute the missing data five times ($m = 5$) using Bayesian linear regression imputation with `mice` (Van Buuren and Groothuis-Oudshoorn 2011). On each imputed dataset, we perform multiple linear regression to predict outcome variable Y from the other two variables

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2,$$

where $\hat{Y}$ is the predicted value of the outcome. Our estimands are the regression coefficient β_1 and coefficient of determination ρ^2 that we obtain after pooling the regression results across the imputations.

3.4 Diagnostic methods

- non-convergence in iterative algorithms is diagnosed using an identifier and a parameter
- the parameter can be any statistic that we track across iterations, for example the average imputed value per imputation (i.e., chain means)
- the identifier is a calculation of some sort that quantifies non-convergence
- identifiers are historically focused on either non-mixing between chains or non-stationarity within chains
- the popular non-mixing identifier rhat has recently been updated by Vehtari et al, and should now work for non-stationarity as well
- just to be sure, we also use autocorrelation to quantify trending within chains
- we apply these identifiers to four different univariate and multivariate parameters
- the univariate parameters are those commonly used for visual inspection: chain means and chain variances

- one multivariate parameter that is of immediate interest for our estimand is the estimated value itself: regression coefficient beta_1
- we also propose a new multivariate parameter that is not dependent on the model of scientific interest: the first eigenvalue of the variance-covariance matrix of the completed data [explain!]

We use eight different methods to diagnose non-convergence: a combination of two non-convergence identifiers—autocorrelation and $\widehat{R}$—and four parameters—chain means, chain variances, β_1, and λ.

3.5 Performance measures

As recommended by Van Buuren (2018), our performance measures are bias, confidence interval width, and coverage rate of the estimands (§ 2.5.2).

4 Simulation Results

The following figures display the simulation results for the eight diagnostic methods and four performance measures, contrasted to the number of iterations in the imputation algorithm. Within the figures, we split the results according to the missingness conditions [missingness mechanisms as line types, and proportion of incomplete cases as colors]. Note that these results are averages of the $n_{sim} = 1000$ simulation repetitions.

4.1 Diagnostic Methods

Figure 1 shows the autocorrelations in the chain means (panel A), chain variances (panel B), regression estimates (panel C), and eigenvalues (panel D).

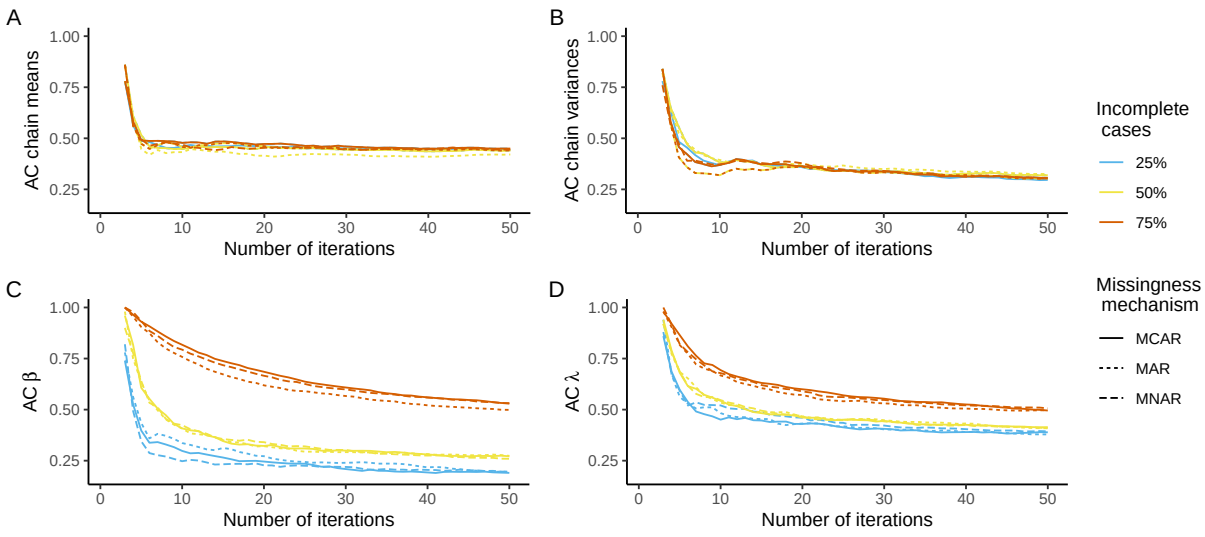

Figure 1: Autocorrelation (AC) with different parameters.

Autocorrelation in the chain means decreases rapidly in the first few iterations (see 1A). The decrease is substantive until $n_{it} \geq 6$. This means that there is some initial trending within chains, but the average imputed value quickly reaches stationarity. These results hold irrespective of the missingness condition.

Autocorrelation in the chain variances show us something similar (see 1B). The number of iterations that is required to reach non-improving autocorrelations is somewhat more ambiguous than for chain means, but generally around $n_{it} \geq 10$. We do not observe a systematic difference between missingness conditions here either.

There is more autocorrelation in the scientific estimates than in the chain means and chain variances (see 1C). We observe the highest autocorrelations in conditions where 75% of cases are incomplete. Overall, the autocorrelations reach a plateau when $n_{it} \geq 20$ to 30. There is no clear effect of the missingness mechanisms.

The autocorrelation in the eigenvalues exhibits a similar trend to the autocorrelation in the scientific estimates (see 1D). Trending in this parameter diminishes when $n_{it} \geq 20$.

Figure 2 shows the potential scale reduction factor in the chain means (panel A), chain variances (panel B), regression estimates (panel C), and eigenvalues (panel D).

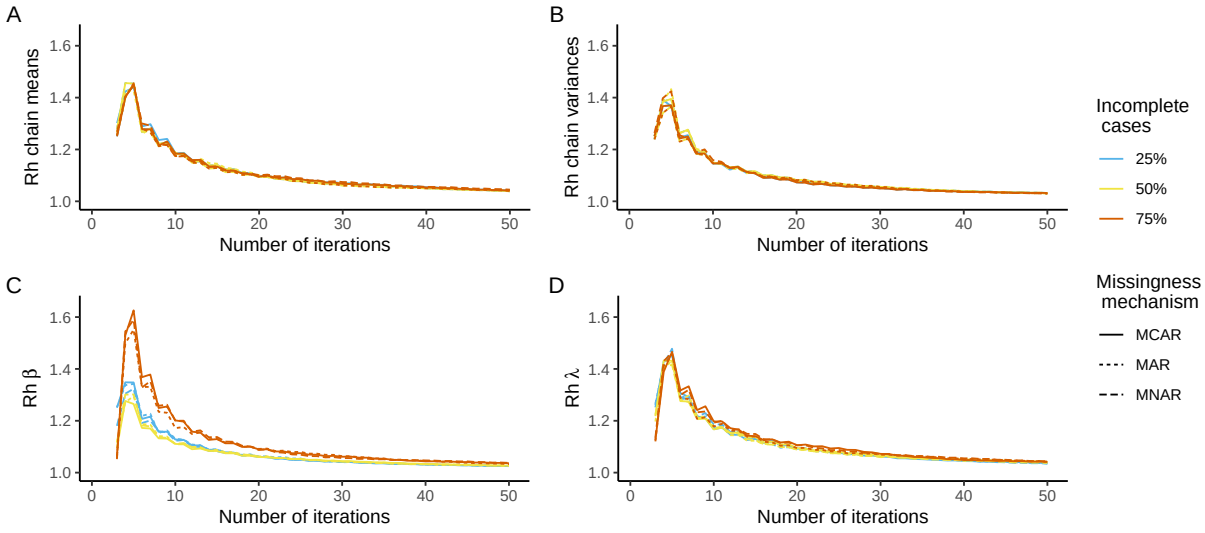

Figure 2: Potential scale reduction factor (Rh) with different parameters.

We observe that $\widehat{R}$-values of the chain means generally decreases as a function of the number of iterations (see 2A). An exception to this observation is the initial increase when $3 \leq n_{it} \leq 5$ [interpret?? due to initialization or is there really more mixing initially??]. After the first couple of iterations, the mixing between chain means generally improves until $n_{it} \geq 30$ to 40. There is no apparent differentiation between the missingness conditions.

The mixing between chain variances mimics the mixing between chain means almost perfectly (see 2B). Irrespective of the missingness condition, the $\widehat{R}$-values taper off around $n_{it} \geq 30$.

With the regression estimate as parameter we observe very similar $\widehat{R}$-values than with chain means and chain variances (see 2C). We do, however, see some differences between missingness conditions. Conditions where 75% of the cases are incomplete show more extreme non-mixing. The overall trend remains the same: about 30 iterations are required before mixing stops improving substantially.

$\widehat{R}$-values of the eigenvalues show a trend similar to the chain means and chain variances (see 2D). [add something about conditions??]

[These rhat plots all show some initialization before the fifth iteration: is rhat useful before that??]

4.2 Performance Measures

In Figure 3 we show the performance measures: bias in the regression estimate (panel A), bias in the coefficient of determination (panel B), the empirical coverage rate of the regression estimate (panel C) and the average confidence interval width of this estimate (panel D).

We see that within a few iterations the bias in the regression estimate approaches zero (see 3A). When $n_{it} \geq 6$, even the worst-performing conditions (e.g., with a proportion of incomplete cases of 75%) produce

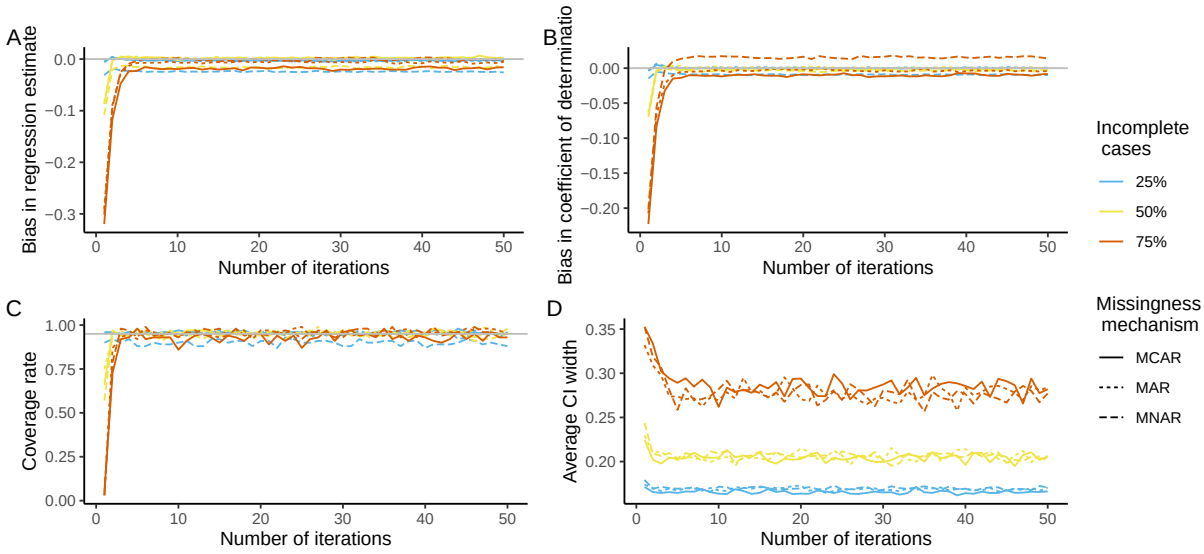

Figure 3: Performance measures.

stable, non-improving estimates [regression coefficient is underestimated because there is less info to estimate the relation??].

Equivalent to the bias in the regression estimate, the bias in the coefficient of determination tapers off within a couple of iterations (see 3B). We observe stable estimates in all conditions when $n_{it} \geq 6$ [interpret the over-estimation in MNAR+75% condition?].

Nominal coverage is quickly reached (see 3C). After just three iterations, the coverage rates are non-improving in every missingness condition [but MNAR with 5% incomplete cases does not reach nominal coverage –> due to bias in the estimate in combination with very narrow CI (see CIW!)].

The average confidence interval width decreases quickly with every added iteration until a stable plateau is reached (see 3D). Depending on the proportion of incomplete cases this takes up-to $n_{it} \geq 9$.

5 Discussion

Our study found that—in the cases considered—inferential validity was achieved after five to ten iterations, much earlier than indicated by the non-convergence identifiers. Of course, it never hurts to iterate longer, but such calculations hardly bring added value.

- Convergence diagnostics keep improving substantially until n_it = 20-30
- Performance measures do not improve after n_it = 9
- [methodological explanation is that rhat and ac have a lag (few it to inform your statistic) –> will always indicate convergence slower than inferential validity is reached]
- Univariate thetas may under-estimate non-convergence.
- Determining non-stationarity with lambda is more difficult than with qhat :(

In short, the validity of iterative imputation stands or falls with algorithmic convergence—or so it's thought. We have shown that iterative imputation algorithms can yield correct outcomes even when a converged state has not yet formally been reached. Any further iterations just burn computational resources without improving the statistical inferences.

6 Case Study

- We use real data: the `boys` dataset from the `mice` package
- We are interested in predicting age from the other variables, in particular in the regression coefficient of `hgt`
- We compare non-convergence identified using visual inspection versus rhat in the chain variances, scientific estimate and lambda.
- The figures show results of a `mice` run with 20 iterations but otherwise default settings.

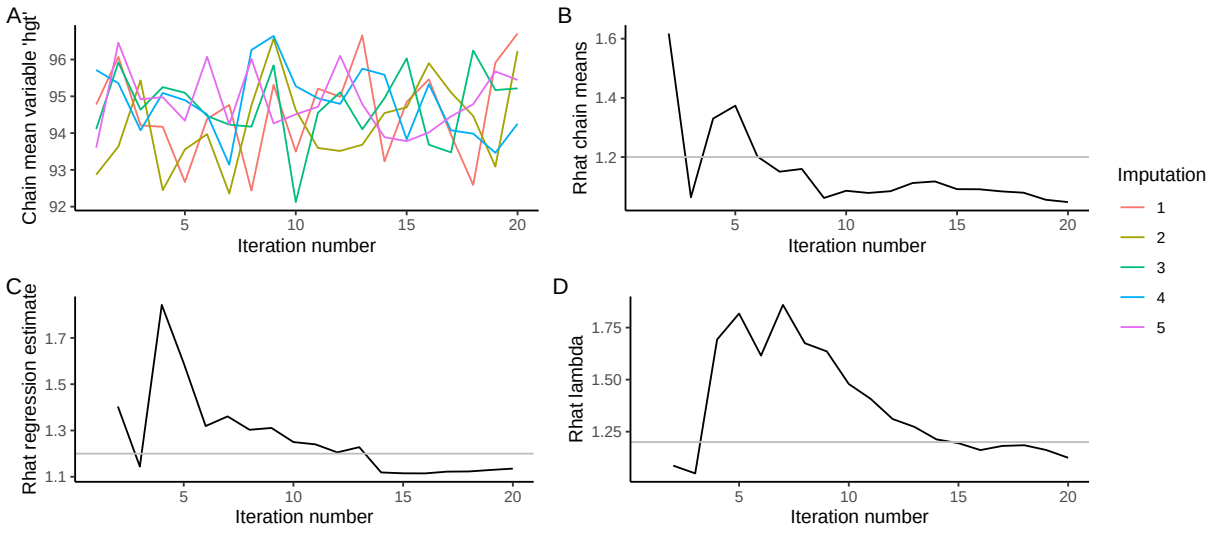

Figure 4: Case study.

From the traceplot of the chain means (see 4A) it seems that mixing improves up-to 10 iterations, while trending is only apparent in the first three iterations.

This figure (4B) shows that 7 iterations are required before the $\widehat{R}$-values of the chain means drop below the threshold for non-convergence.

The $\widehat{R}$-values for the scientific estimate reaches the threshold much sooner, when $n_{it} = 14$ (see 4C).

According to the $\widehat{R}$-values with λ as parameter, at least 15 iterations are required (see 4D).

References

Raghunathan, Trivellore, and Irina Bondarenko. 2007. "Diagnostics for Multiple Imputations." SSRN Scholarly Paper ID 1031750. Rochester, NY: Social Science Research Network. https://papers.ssrn.com/abstract=1031750.

Van Buuren, Stef. 2018. *Flexible Imputation of Missing Data*. Chapman and Hall/CRC. https://stefvanbuuren.name/fimd/.

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.