Contribution to Wiley StatsRef

Stef van Buuren

2020-11-12

Contents

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

Chapter 1

MICE - Multivariate Imputation by Chained Equations

Multivariate Imputation by Chained Equations (MICE) is an algorithm to create synthetic values (imputations) for multivariate missing data. This article briefly reviews ideas similar to MICE, explains the difference between single and multiple imputation and highlights practical problems in multivariate imputation. The MICE algorithm iteratively imputes the data variable-by-variable. The text discusses the conditions needed for convergence, the issues of compatibility between the complete-data model and the imputation model, the number of iterations, the performance of the algorithm, and potential extensions.

1.1 Historic background

MICE is an acronym for *Multivariate Imputation by Chained Equations*. The term MICE refers to an algorithm to impute multivariate missing data. The user specifies the distribution of the missing data in each incomplete variable conditional on other data. For example, we could use logistic regression to impute incomplete binary variables, polytomous regression for categorical data, and linear regression for numerical data. The MICE algorithm generates multiple imputations by iteratively drawing values from these conditional distributions. The algorithm was first published as S-PLUS software (van Buuren and Groothuis-Oudshoorn, 1999). In 2006 it became widely available as an R package on CRAN (van Buuren, 2011). SAS 9.3, SPSS 17.0 and Stata 12 introduced versions of the MICE algorithm in their offerings.

Ideas similar to MICE have surfaced under other names: stochastic relaxation (Kennickell, 1991), variable-by-variable imputation (Brand, 1999), switching regressions (van Buuren et al., 1999), sequential regressions (Raghunathan et al., 2001), ordered pseudo-Gibbs sampler (Heckerman et al., 2001), partially incompatible MCMC (Rubin, 2003), iterated univariate imputation (Gelman, 2004), chained equations (van Buuren and Groothuis-Oudshoorn, 1999) and fully conditional specification (FCS) (van Buuren et al., 2006). A simple Google search reveals that "chained equations" has become the most popular name.

1.2 Multiple imputation

Multiple imputation (Rubin, 1987) is a general method to deal with incomplete data. Many analysts attempt to replace a missing entry by the "best" value according to some prediction method, a strategy known as *single imputation*. However, standard errors, confidence intervals and P-values after single imputation are correct only when all predictions are made without error, which is unrealistic in practice. Rubin realised that replacing the missing value by *one* value cannot be correct in general. His solution was brilliant and straightforward: create multiple imputations that reflect the uncertainty of the unknown value.

Rubin (1987) describes the workflow in three steps:

1. Create m completed datasets;
2. Estimate the quantities of scientific interest in each complete dataset;
3. Pool these estimates and their standard errors to a single result.

The workflow produces estimates with known statistical properties under fairly general conditions.

1.3 Practical problems in multivariate imputation

In practice, missing data can appear everywhere in the data. The MICE algorithm handles multivariate missing data problems. This section highlights some of the practical problems that we need to address.

Let Y denote the $n \times p$ matrix containing the data values on p variables for all n units in the sample. We define the *response indicator* R as a binary $n \times p$ matrix, where a "0" indicates a missing value. Symbol Y_j is the j'th column in Y. Symbol Y_{-j} indicates all columns in Y except Y_j. Symbols Y_j^{obs} and Y_j^{mis} refer to the observed and missing values in Y_j, respectively.

The basic conditional imputation model $P(Y_j^{\text{mis}}|Y_j^{\text{obs}}, Y_{-j}, R)$ specifies the distribution of the missing values Y_j^{mis} conditional on the observed data in Y_j^{obs}, on the remaining data Y_j and on the response indicator R. If we assume that the missing data are missing at random (MAR), then R drops out of the model. The rationale for conditioning on Y_{-j} is that this preserves the relations among the variables in the imputed data.

van Buuren (2018) highlighted various practical problems that occur:

- The predictors Y_{-j} themselves can contain missing values;
- "Circular" dependence can occur, where Y_j^{mis} depends on Y_h^{mis}, and Y_h^{mis} depends on Y_j^{mis} with $h \neq j$, because in general Y_j and Y_h are correlated, even given other variables;
- Variables are often of different types (e.g., binary, unordered, ordered, continuous), thereby making the application of theoretically convenient models, such as the multivariate normal, theoretically inappropriate;
- Especially with large p and small n, collinearity or empty cells can occur;
- The ordering of the rows and columns can be meaningful, e.g., as in longitudinal data;
- The relation between Y_j and predictors Y_{-j} can be complicated, e.g., nonlinear, or subject to censoring processes;
- Imputation can create impossible combinations, such as pregnant fathers.

This list is by no means exhaustive, and other complexities may appear for detailed data.

1.4 The MICE algorithm

The MICE algorithm provides an iterative solution to these problems. The procedure consists of the following steps:

1. Specify an imputation model $P(Y_j^{\text{mis}}|Y_j^{\text{obs}}, Y_{-j}, R)$ for variable Y_j with $j = 1, \ldots, p$.
2. For each j, fill in starting imputations $\dot{Y}_j^0$ by random draws from Y_j^{obs}.
3. Repeat for $t = 1, \ldots, T$.
4. Repeat for $j = 1, \ldots, p$.
5. Define $\dot{Y}_{-j}^t = (\dot{Y}_1^t, \ldots, \dot{Y}_{j-1}^t, \dot{Y}_{j+1}^{t-1}, \ldots, \dot{Y}_p^{t-1})$ as the currently complete data except Y_j.
6. Draw $\dot{\phi}_j^t \sim P(\phi_j^t|Y_j^{\text{obs}}, \dot{Y}_{-j}^t, R)$.
7. Draw imputations $\dot{Y}_j^t \sim P(Y_j^{\text{mis}}|Y_j^{\text{obs}}, \dot{Y}_{-j}^t, R, \dot{\phi}_j^t)$.
8. End repeat j.
9. End repeat t.

The algorithm starts with a random draw from the observed data and imputes the incomplete data in a variable-by-variable fashion. One iteration consists of one cycle through all Y_j. The number of iterations T can often be low, say 5 or 10. The MICE algorithm generates multiple imputations by executing the procedure in parallel m times.

1.5 Methodology

1.5.1 MCMC Conditions

The MICE algorithm is a Markov chain Monte Carlo (MCMC) method, where the state space is the collection of all imputed values. In order to converge to a stationary distribution, a Markov chain needs to satisfy three critical conditions (Roberts, 1996; Tierney, 1996):

- *irreducible*, the chain must be able to reach all interesting parts of the state space;
- *aperiodic*, the chain should not oscillate between different states;
- *recurrence*, all interesting parts can be reached infinitely often, at least from almost all starting points.

Do these properties hold for the MICE algorithm? Irreducibility is generally not a problem since the user has considerable control over the state space. This flexibility is the main attraction of the MICE algorithm.

Periodicity is a potential problem and can arise in a situation where imputation models are inconsistent. A rather artificial example of an oscillatory behavior occurs when Y_1 is imputed by $Y_2\beta+\epsilon_1$ and Y_2 is imputed by $-Y_1\beta+\epsilon_2$ for some fixed, nonzero β. The sampler will oscillate between two qualitatively different states, so the correlation between Y_1 and Y_2 after imputing Y_1 will differ from that after imputing Y_2. In general, we would like the statistical inferences to be independent of the stopping point. A way to diagnose the *ping-pong* problem, or *order effect*, is to stop the chain at different points. The stopping point should not affect statistical inferences. The addition of noise to create imputations is a safeguard against periodicity and allows the sampler to "break out" more easily.

Non-recurrence may also be a potential difficulty, manifesting itself as explosive or non-stationary behaviour. For example, if imputations are created by deterministic functions, the Markov chain may lock up. We may diagnose such from the trace lines of the sampler. As long as we estimate the parameters of imputation models from the data, non-recurrence is mild or absent.

1.5.2 Compatibility

Gibbs sampling exploits the idea that knowledge of the conditional distributions is sufficient to determine a joint distribution if it exists. The convergence of the MICE algorithm to a (multivariate) joint distribution can be guaranteed when are conditions are known to be *compatible*. For example, when conditional regressions are all linear with a normal residual, the joint corresponds to the multivariate normal distribution.

There is active literature on compatibility. We refer to van Buuren (2018) for a more extensive discussion of the topic. van Buuren et al. (2006) described a small simulation study using strongly incompatible models. The adverse effects on the estimates after multiple imputation were only minimal in the cases studied. These simulations suggested that the results may be robust against violations of compatibility. Li et al. (2012) presented three examples of problems with MICE. However, their examples differ from the usual sequential regression set up in various ways and do not undermine the validity of the approach (Zhu and Raghunathan, 2015). Liu et al. (2013) pointed out that application of incompatible conditional models cannot provide imputations from any joint model. However, they also found that Rubin's rules provide consistent point estimates for incompatible models under fairly general conditions, as long as each conditional model was correctly specified. Zhu and Raghunathan (2015) showed that incompatibility does not need to lead to divergence. While there is no joint model to converge to, the algorithm can still converge. The key to achieving convergence is that the imputation models should closely model the data. For example, include the skewness of the residuals, or ideally, generate the imputations from the underlying (but usually unknown) mechanism that generated the data.

In the majority of cases, scientific interest will focus on quantities that are more remote to the joint density, such as regression weights, factor loadings, and prevalence estimates. In such cases, the joint distribution is more like a nuisance factor that has no intrinsic value.

Apart from potential feedback problems, it appears that incompatibility seems like a relatively minor problem in practice, especially if the missing data rate is modest, and if the imputation models fit the data well. In order to evaluate these aspects, we need to inspect convergence and assess the fit of the imputations.

1.5.3 Number of iterations

When we calculate m sampling streams in parallel, we may monitor convergence by plotting one or more statistics of interest in each stream against iteration number t. Common statistics to be plotted are the mean and standard deviation of the synthetic data, as well as the correlation between different variables. The pattern should be free of a trend, and the variance within a chain should approximate the variance between chains.

In practice, a low number of iterations appears to be enough. Brand (1999) and van Buuren et al. (1999) set the number of iterations T relatively low, usually somewhere between 5 to 20 iterations. This number is much lower than in other applications of MCMC methods, which often require thousands of iterations. The imputations form the only memory in the MICE algorithm. Note that the 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 much noise into the system. Hence, the autocorrelation 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 the side-effect of speeding up convergence. Reversely, situations to watch out for occur if:

- the correlations between the Y_j are high;
- the missing data rates are high; or
- constraints on parameters across different variables exist.

The first two conditions directly affect the amount of autocorrelation in the system. The latter condition becomes relevant for customised imputation models.

A recent simulation study by Oberman et al. (2020) found that conventional convergence diagnostics like $\hat{R}$ (Gelman and Rubin, 1991) are too conservative for missing data imputation. When these diagnostics typically indicate convergence after only 30-40 iterations, the parameters estimates achieve their statistical properties between 5 and 10 iterations. It is, however, important not to rely automatically on this result as some applications can require considerably more iterations.

1.6 Performance

The MICE algorithm is extremely flexible as it allows to user to set each conditional density. Most software packages provide reasonable defaults for everyday situations, so the actual effort required from the user may be small. However, it generally pays off to go beyond the default to address particular features in the data or science, like derived variables, interaction terms, skipping pattern, multi-level data and time dependencies.

Many simulation studies provide evidence that MICE algorithm, or similar methodologies, generally yields estimates that are unbiased and that possess appropriate coverage (Brand, 1999; Raghunathan et al., 2001; Brand et al., 2003; Tang et al., 2005; van Buuren et al., 2006; Horton and Kleinman, 2007; Yu et al., 2007). Nair et al. (2013) summarise their results as

> We observe that MICE is overall the best imputation algorithm.

1.7 Future work

We may extend the MICE algorithm in various ways.

1.7.1 Skipping imputations and overimputation

By default, the MICE algorithm imputes all missing data and leaves the observed data untouched. In some cases, it may also be useful to skip imputation of specific cells. For example, we wish to skip imputation of quality of life for the deceased, or not impute customer satisfaction for people who did not buy the product. The primary difficulty with this option is that it creates missing data in the predictors, so the imputer should either remove the predictor from all imputation models or have the missing values propagated through the algorithm. Another use case involves imputing cells with observed data, a technique called *overimputation*. For example, it may be useful to evaluate whether the observed point data fit the imputation model. If all is well, we expect the observed data point in the centre of the multiple imputations. The primary difficulty with this option is to ensure that we use only the observed data (and not the imputed data) as an outcome in the imputation model. Version `3.0` of `mice` includes the `where` argument. The specification is a matrix with binary values that has the same dimensions as the data, that indicates where MICE should create imputations. We may use this matrix to specify for each cell, whether it should be imputed or not. The default is that the missing data are imputed.

1.7.2 Blocks of variables, hybrid imputation

The MICE algorithm imputes each variable separately. In some cases, it is useful to impute multiple values simultaneously, as a block. In actual data analysis sets of variables are often connected in some way. Examples are:

- A set of scale items and its total score;
- A variable with one or more transformations;
- Two variables with one or more interaction terms;
- A block of normally distributed Z-scores;
- Compositions that add up to a total;
- Set of variables that are collected together.

Instead of specifying the steps for each variable separately, it is more user-friendly to impute these as a block. Version `3.0` of `mice` includes a new `block` argument that partitions the complete set of variables into blocks. All variables within the same block are jointly imputed. The joint models need to be open to accepting external covariates. One possibility is to use predictive mean matching to impute multivariate nonresponse, where the donor values for the variables

within the block come from the same donor (Little, 1988). The main algorithm in `mice 3.0` iterates over the blocks rather than the variables. By default, each variable is a block, which gives normal behaviour.

1.7.3 Blocks of units, monotone blocks

Another way to partition the data is to define blocks of units. One weakness of the MICE algorithm is that it may become unstable when many of the predictors are imputed. Zhu (2016) developed a solution called "Block sequential regression multivariate imputation", which partitions units into blocks according to the missing data pattern. The imputation model for a given variable is modified for each block, such that only the observed data with the block can serve as a predictor. The method generalises the monotone block approach of Li et al. (2014).

1.7.4 Separate training from test data

The MICE algorithm uses all rows in the data to estimate and apply the imputation model. In practice, we sometimes wish to estimate the imputation model on one dataset and apply it to another. The `ignore` argument to the `mice()` function specifies the set of rows that MICE will ignore when creating the imputation model. The default is to include all rows. We may use the feature to split the data into a training set (on which the imputation model is built) and a test set (that does not influence the imputation model estimates). The feature is still experimental but is likely to attract interest from the data science and machine learning communities.

1.8 Conclusion

Multivariate missing data lead to analytic problems caused by mutual dependencies between incomplete variables. For general missing data patterns, the MICE algorithm is a flexible and straightforward procedure that allows for imputed values close to the data.

Bibliography

Brand, J. P. L. (1999). *Development, Implementation and Evaluation of Multiple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets*. PhD thesis, Erasmus University, Rotterdam.

Brand, J. P. L., van Buuren, S., Groothuis-Oudshoorn, C. G. M., and Gelsema, E. S. (2003). A toolkit in SAS for the evaluation of multiple imputation methods. *Statistica Neerlandica*, 57(1):36–45.

Gelman, A. (2004). Parameterization and Bayesian modeling. *Journal of the American Statistical Association*, 99(466):537–545.

Gelman, A. and Rubin, D. B. (1991). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*, 7:457–511.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. (2001). Dependency networks for inference, collaborative filtering, and data visualisation. *Journal of Machine Learning Research*, 1(1):49–75.

Horton, N. J. and Kleinman, K. P. (2007). Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models. *The American Statistician*, 61(1):79–90.

Kennickell, A. B. (1991). Imputation of the 1989 survey of consumer finances: Stochastic relaxation and multiple imputation. *ASA 1991 Proceedings of the Section on Survey Research Methods*, pages 1–10.

Li, F., Baccini, M., Mealli, F., Zell, E. R., Frangakis, C. E., and Rubin, D. B. (2014). Multiple imputation by ordered monotone blocks with application to the anthrax vaccine research program. *Journal of Computational and Graphical Statistics*, 23(3):877–892.

Li, F., Yu, Y., and Rubin, D. B. (2012). Imputing missing data by fully conditional models: Some cautionary examples and guideline. *Duke University Department of Statistical Science Discussion Paper*, 11-24.

Little, R. J. A. (1988). Missing-data adjustments in large surveys (with discussion). *Journal of Business Economics and Statistics*, 6(3):287–301.

Liu, J., Gelman, A., Hill, J., Su, Y. S., and Kropko, J. (2013). On the stationary distribution of iterative imputations. *Biometrika*, 101(1):155–173.

Nair, V., Kidambi, R., Sellamanickam, S., Keerthi, S., Gehrke, J., and Narayanan, V. (2013). A quantitative evaluation framework for missing value imputation algorithms. *arXiv preprint arXiv:1311.2276*.

Oberman, H., van Buuren, S., and Vink, G. (2020). Missing the point: Non-convergence in iterative imputation algorithms. *First Workshop on the Art of Learning with Missing Values (Artemiss) hosted by the 37th International Conference on Machine Learning (ICML)*.

Raghunathan, T. E., Lepkowski, J. M., Van Hoewyk, J., and Solenberger, P. W. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, 27(1):85–95.

Roberts, G. O. (1996). Markov chain concepts related to sampling algorithms. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pages 45–57. Chapman & Hall, London.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, New York.

Rubin, D. B. (2003). Nested multiple imputation of NMES via partially incompatible MCMC. *Statistica Neerlandica*, 57(1):3–18.

Tang, L., Unüntzer, J., Song, J., and Belin, T. R. (2005). A comparison of imputation methods in a longitudinal randomized clinical trial. *Statistics in Medicine*, 24(14):2111–2128.

Tierney, L. (1996). Introduction to general state-space Markov chain theory. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, chapter 4, pages 59–74. Chapman & Hall, London.

van Buuren, S. (2011). Multiple imputation of multilevel data. In Hox, J. and Roberts, J., editors, *The Handbook of Advanced Multilevel Analysis*, chapter 10, pages 173–196. Routledge, Milton Park, UK.

van Buuren, S. (2018). *Flexible Imputation of Missing Data. Second Edition.* Chapman & Hall/CRC, Boca Raton, FL.

van Buuren, S., Boshuizen, H. C., and Knook, D. L. (1999). Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine*, 18(6):681–694.

van Buuren, S., Brand, J. P. L., Groothuis-Oudshoorn, C. G. M., and Rubin, D. B. (2006). Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation*, 76(12):1049–1064.

van Buuren, S. and Groothuis-Oudshoorn, C. G. M. (1999). Flexible multivariate imputation by MICE. Technical Report PG/VGZ/99.054, TNO Prevention and Health, Leiden.

Yu, L.-M., Burton, A., and Rivero-Arias, O. (2007). Evaluation of software for multiple imputation of semi-continuous data. *Statistical Methods in Medical Research*, 16(3):243–258.

Zhu, J. (2016). *Assessment and Improvement of a Sequential Regression Multivariate Imputation Algorithm*. PhD thesis, University of Michigan.

Zhu, J. and Raghunathan, T. E. (2015). Convergence properties of a sequential regression multiple imputation algorithm. *Journal of the American Statistical Association*, 110(511):1112–1124.