Incomplete Data Analysis

Vanda Inácio

University of Edinburgh

Semester 1, 2020/2021

Multiple imputation

MI in practice – the `mice` package

- ↪ The `R` statistical software provides several useful packages for multiple imputation.
- ↪ They basically differ in the specific algorithm they implement in the imputation model (step 1). We will study in detail the `mice` package, which stands for *multiple imputation by chained equations*.
- ↪ We will start with the case where only one variable has missing values and, for now, we will also assume that such variable is continuous.
- ↪ We will later cover the case where the variable with missing values is not continuous (e.g., a binary variable) and will also learn about `mice` and multivariate missingness (i.e., several variables with missing values) as this is the most realistic case and also where `mice` shines.

Multiple imputation

MI in practice – the `mice` package

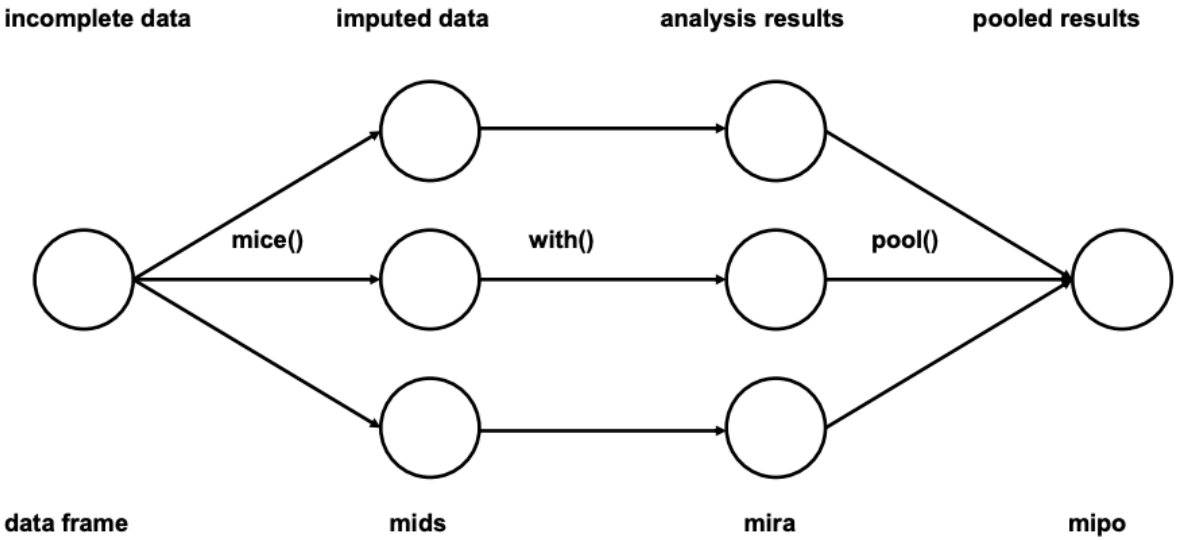

Figure 1: Main steps used in multiple imputation.

Figure from van Buuren and Groothuis-Oudshoorn (2011)

Multiple imputation

MI in practice – the `mice` package

↪ As can be seen in the figure in previous slide, there are three key functions for each stage of the multiple imputation method.

1. `mice()`: generates multiple completed datasets, where the missing values are imputed according to a chosen scheme.
2. `with()`: applies the statistical model of interest (the substantive model) to each completed dataset.
3. `pool()`: pools/combines the estimates together by Rubin's rules.

Multiple imputation

MI in practice – the `mice` package

- We start by devoting our attention to the function `mice()`, which performs the imputation step.
- The function admits several arguments and one of them is `method`, through which we specify the model to be used to impute the missing values.
- Under the normal linear model, the following methods are available:
  - `norm.nob`: implements stochastic (normal linear) regression imputation (*M* times) (improper MI).
  - `norm`: implements (normal linear) stochastic regression imputation (*M* times) with parameters drawn from their posterior distribution, so that parameters' uncertainty is taken into account (proper MI).
  - `norm.boot`: implements a frequentist counterpart of `norm`, where the parameters are estimated based on a bootstrap sample of the complete cases (proper MI).

Multiple imputation

MI in practice – the `mice` package

- `mice` also implements mean imputation (`method = mean`) and regression imputation (`method = norm.predict`) but as we already know there is very little to recommend about these methods.

- Further, because both methods are fully deterministic, the concept of multiple imputation does not really make sense, as all *M* copies of the dataset would be the same.

Multiple imputation

MI in practice – the `mice` package

- Let us elaborate a little more on the imputed values generated by `norm.nob`, `norm`, and `norm.boot`.
- Let Y stand for the variable with missing values and X for the fully observed variables. We will be assuming that Y is MAR given X.
- Further let X_{obs} indicate the subset of m rows for which y is observed and X_{mis} is the complementing subset of $n - m$ rows for which y is missing.
- The vector containing the m observed y values is denoted by y_{obs} and the vector of $n - m$ imputed values in y is indicated by $\widehat{y}_{mis}$.

Multiple imputation

MI in practice – the `mice` package

- The method `norm.nob` implements (normal linear) stochastic regression for each copy m of the data set we have made, $m = 1, \ldots, M$.

- The imputed values are of the form

$$\widehat{y}_{\text{mis}}^{(m)} = \widehat{\beta}_0 + X_{\text{mis}}\widehat{\beta}_1 + z, \quad z \sim \text{N}(0, \widehat{\sigma}^2), \quad m = 1, \ldots, M,$$

  where $\widehat{\beta}_0$ and $\widehat{\beta}_1$ and $\widehat{\sigma}^2$ are estimates obtained from fitting a normal linear regression model to $(y_{\text{obs}}, X_{\text{obs}})$.

- As we have learned last week, this method is known in the literature as improper MI as parameter uncertainty is not acknowledged since the same estimates are used for imputing all M copies of the dataset.

- This approach may lead to confidence intervals that are too narrow and this, in turn, affects the corresponding coverage of the intervals.

Multiple imputation

MI in practice – the `mice` package

- The methods `norm` and `norm.boot` implement (normal linear) stochastic regression but taking parameter uncertainty into account (proper MI).

- Specifically, in `norm`, the imputed values are of the form

$$\widehat{y}_{\text{mis}}^{(m)} = \widehat{\beta}_0^{(m)} + X_{\text{mis}}\widehat{\beta}_1^{(m)} + z, \quad z \sim \text{N}\left(0, (\widehat{\sigma}^{(m)})^2\right), \quad m = 1, \ldots, M, \tag{1}$$

  where $\widehat{\beta}_0^{(m)}$, $\widehat{\beta}_1^{(m)}$, and $\widehat{\sigma}^{(m)}$ are draws from the posterior distribution of the parameters, $p\left(\beta_0, \beta_1, \sigma^2 \mid y_{\text{obs}}, X_{\text{obs}}\right)$.

- For each imputed dataset m, we are thus using slightly different values of the parameters.

- Similarly, in `norm.boot`, the imputed values are also of the form in (1), but $\widehat{\beta}_0^{(m)}$, $\widehat{\beta}_1^{(m)}$, and $\widehat{\sigma}^{(m)}$ are the least squares/maximum likelihood estimates from a bootstrap sample of the observed data $(y_{\text{obs}}, X_{\text{obs}})$.

Multiple imputation

MI in practice – the `mice` package

- What if the normal distribution is not a good fit?
- In practice, the data could be skewed, heavy tailed, bimodal, to name only some deviations from normality.
- The effect of non-normality is general small for measures that rely on the center of the distribution, like means, but it could be substantial for estimates like a variance or a percentile (van Buuren, 2018, p. 74).
- A sensible approach is to transform the data toward normality before imputation and back-transform after imputation.
- Sometimes applying a simple function to the data, like the logarithmic is all that is needed. More generally, the transformation could be made to depend on known covariates.

Multiple imputation

MI in practice – the `mice` package

- ↪ It is also possible to directly draw imputations from non-normal distributions. For instance, Liu (1995) proposed methods for drawing imputations under the *t* distribution instead of the normal (the *t* distribution has heavier tails than the normal distribution).

- ↪ He and Raghunathan (2006) created imputations by drawing from Tukey's gh distribution, which can take many different shapes.

- ↪ We will not explore it here, but the package `ImputeRobust` (Salfran and Spiess, 2017) implements various `mice` methods for continuous distributions and, in particular, the *t*-distribution.

Multiple imputation

MI in practice – the `mice` package

- **Predictive mean matching** (`method = pmm` in `mice`) is a semiparametric approach to imputation developed for settings where the normal distribution is not a good choice. It is the default method in `mice` for continuous data.

- The main idea behind this method is to find the cases in the observed data that are similar to the cases with missing values and subsequently fill in each missing value with an observed value from one of the cases.

- In order to find similar cases, the *predicted means* (from a normal linear regression model) of complete and incomplete cases are compared.

Multiple imputation

MI in practice – the `mice` package

↪ Let $\widehat{y}_{\text{obs},i} = \widehat{\beta}_0 + X_{\text{obs},i}\widehat{\beta}_1$ be the predicted value for those with y_i observed, $i = 1, \ldots, m$, and $\widehat{y}_{\text{mis},j} = \widehat{\beta}_0 + X_{\text{mis},j}\widehat{\beta}_1$ be the predicted value for those with y_j missing, $j = m+1, \ldots, n$.

↪ There are various ways to select the donor for case j ($j = m+1, \ldots, n$):

1. The donor is the (observed) case i with the smallest absolute difference $|\widehat{y}_{\text{obs},i} - \widehat{y}_{\text{mis},j}|$.
2. A pool of donor candidates composed by the d cases with the smallest absolute difference $|\widehat{y}_{\text{obs},i} - \widehat{y}_{\text{mis},j}|$. The donor is then selected at random from the pool.
3. The pool of donor candidates is composed by all cases i for which the absolute difference is smaller than some threshold η, i.e., $|\widehat{y}_{\text{obs},i} - \widehat{y}_{\text{mis},j}| < \eta$, for some pre-specified η. The donor is then selected at random from the pool.

↪ Options 1 and 2 are implemented in `mice` (well, 1 is a particular case of 2 with $d = 1$) with the default being option 2.

Multiple imputation

MI in practice – the `mice` package

- ↪ Setting $d = 1$ is generally considered to be too low, as it may reselect the same donor over and over again and, consequently, the uncertainty about the missing values is underestimated.
- ↪ On the other hand, letting d to be very large can lead to bad matches.
- ↪ According to Morris et al. (2015), values of d between 3 and 10 provide the best results in most cases.
- ↪ The default in `mice` is $d = 5$ and this value can be changed through the argument `donors`.

Multiple imputation

MI in practice – the `mice` package

- In what regards sampling the parameters (the regression coefficients used to compute the predictive means), several approaches have been suggested in the literature:
  - *Type 0*: $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are estimated by least squares or maximum likelihood from the complete cases and used in both prediction models.
  - *Type 1*: $\widehat{\beta}_0$ and $\widehat{\beta}_1$ estimated by least squares or maximum likelihood from the complete cases are used to predict $\widehat{y}_{\text{obs},i}$ and $\tilde{\beta}_0$ and $\tilde{\beta}_1$ are draws from the posterior distribution or estimated on the basis of a bootstrap sample of the complete cases and are used to predict $\widehat{y}_{\text{mis},j}$.
  - *Type 2*: $\tilde{\beta}_0$ and $\tilde{\beta}_1$ with the same meaning as in the previous item are used to predict both $\widehat{y}_{\text{obs},i}$ and $\widehat{y}_{\text{mis},j}$.
- The three types are implemented in `mice`, with *Type 1* being the default.

Multiple imputation

MI in practice – the `mice` package

- The most obvious pitfall of predictive mean matching is the duplication of the same donor value many times and this is more likely to occur if the sample is small and the proportion of missing values is large.

- To conclude the discussion about predictive mean matching, I quote van Buuren (p. 84, 2018):

> "*The method works best with large samples, and provides imputations that possess many characteristics of the complete data. Predictive mean matching cannot be used to extrapolate beyond the range of the data (...) Also, it may not perform well with small datasets. Bearing these points in mind, predictive mean matching is a great all-around method with exceptional properties.*"

Multiple imputation

MI in practice – the `mice` package

- We have discussed thus far the use of function `mice`, i.e., how to perform step 1 in the package.
- Now having the complete datasets at hand, the conventional statistical analysis that would have been conducted had the data been complete can be performed. This is done via the function `with()` and we need to feed this function with the imputed data from the previous step and with the model we want to fit (e.g., through the use of `lm` for linear regression or `glm` for generalised linear models).
- Lastly, we use the function `pool()` to combine the results of the *M* analyses (obtained in the previous step using `with()`).

Multiple imputation

MI in practice – the `mice` package

- ↪ Depending on the nature of the variable being imputed, other types of regression models may be needed.
- ↪ The principles obtained remain, however, the same.
- ↪ When the variable to be imputed is binary, a sensible imputation model is a logistic regression model. In `mice` the procedure is implemented under the name `logreg` and it is the default in `mice` for binary variables.
- ↪ If the variable to be imputed is categorical with K unordered categories then a reasonable model is a multinomial logit model, and this is implemented in `mice` as `polyreg`.
- ↪ A categorical variable with K ordered categories is imputed by the ordered logit model or proportional odds model. Implementation is done in `polr` function.