MatchThem:: Cheat Sheet

Matching and Weighting Multiply Imputed Datasets

December 2019

0::

One of the obstacles for matching and weighting units in observational studies is that they cannot be performed in a straightforward way for units with missing values. There are 3 kinds of missing values: (1) missing-completely-at-random (missingness relies on chance); (2) missing-at-random (missingness relies on specific values in other observed variables); and (3) missing-not-at-random (missingness relies on specific values in the same unobserved variable).

2::matchthem()

Matching units in the multiply imputed datasets can be performed with calls to **matchthem()**. **matchthem()** offers a variety of options for this step:

```
> models <- matchthem(formula, datasets,
                      approach = "within",
                      method = "cem", ...)
```

Where **formula** specifies the model (e.g., **z ~ x1 + x2 + x3**) that should be used for matching, **datasets** specifies the multiply imputed datasets (produced in step **1**), **approach** specifies that matching should be performed within each imputed dataset (**within**) or an averaged propensity score across the datasets should be used for matching (**across**), and **method** specifies the method that should be used for matching. **matchthem()** supports several matching methods including **nearest** (nearest neighbor matching), **full** (full matching), **exact** (exact matching), **cem** (coarsened exact matching), etc. Several additional arguments can be passed to the function, too (e.g., **caliper**, **ratio**, etc.). The output will be saved in a **mimids** class object and can be reviewed with **summary()** and **plot()**.

3::

Average and maximum of the (absolute) standardized mean differences of the covariates in the models across the imputed datasets should be assessed to check for the extent of balance after matching and weighing. **cobalt** R package offers the required functions for this step, including **bal.tab()** and **love.plot()** (please see the vignettes of this package for details):

```
> bal.tab(models)
> bal.tab(models, abs = TRUE)
```

4::

The causal effect estimate in each matched or weighted dataset can be calculated with calls to **with()**:

```
> results <- with(models, svyglm(y ~ z,
                      family = binomial))
```

Where **models** specifies the **mimids** or **wimids** object. The output will be saved in a **mimira** class object

---

Dataset with Missing Values

Imputing the Missing Data in the Dataset

(Amelia or mice R packages)

Matching or Weighting the Multiply Imputed Datasets

(matchthem() or weightthem() functions)

Assessing Balance on the Matched or Weighted Datasets

(cobalt R package)

Matching	Weighting
Beta₁	Beta₁

Analyzing the Matched or Weighted Datasets

(with() function)

Beta$_1$ Beta$_2$ Beta$_3$ Beta$_4$ Beta$_5$

Pooling the Causal Effect Estimates

(pool() function)

Pooled Causal Effect

Matching   Weighting

1::

There are several solutions to address the problem of missing data, but a popular one is to multiply impute the missing values. Multiply imputing the missing values (i.e., replacing the missing values with a set of plausible values and random noise) is credible when missingness is ignorable (i.e., completely-at-random or at-random).

There are several R packages to multiply impute the missing data. **MatchThem**, in this version, is compatible with imputed datasets by **Amelia** and **mice** packages (please see vignettes of these packages for details):

```
> datasets <- mice(data, m = 5)
```

2::weightthem()

Weighting units in the multiply imputed datasets can be performed with calls to **weightthem()**. **weightthem()** offers a variety of options for this step:

```
> models <- weightthem(formula, datasets,
                      approach = "within",
                      method = "ps",
                      estimand = "ATT", ...)
```

Where **formula** specifies the model (e.g., **z ~ x1 + x2 + x3**) that should be used for weighting, **datasets** specifies the multiply imputed datasets (produced in step **1**), **approach** specifies that weights should be calculated within each imputed dataset (**within**) or calculated using an averaged propensity score across the datasets (**across**), **method** specifies the method that should be used for weighting, and **estimand** specifies the desired estimand that the weights should target. **weightthem()** supports several weighting methods including inverse probability weighting (**ps**) and generalized boosted model weighting (**gbm**). Several additional arguments can be passed to the function, too (e.g., **by**, **stabilize**, etc.). The output will be saved in a **wimids** class object and can be reviewed with **summary()**.

5::

The causal effect estimates from analyzing each matched or weighted dataset in the **mimira** object can be pooled using **pool()**:

```
> output <- pool(results, dfcom = NULL)
```

Where **results** specifies the **mimira** object and **dfcom** specifies the degree of freedom in the analysis (when not specified, it is extracted from the dataset with the lowest number of units or the first imputed dataset). The output will be saved in a **mimipo** class object and can be reviewed with **summary()**:

```
> summary(output, conf.int = TRUE)
```