Hybrid Imputation Models Through Blocks

Stef van Buuren

2018/8/1, JSM 2018, Vancouver

Imputing multivariate missing data

Strategies

- Joint model (JM)
- Fully conditional specification (FCS)

Questions

- Can these be complementary strategies?
- When could hybrid imputation be useful?

Joint model (JM)

- Specify joint model $P(Y, X)$
- Derive $P(Y^{\mathrm{mis}}|Y^{\mathrm{obs}}, X)$
- Draw imputations $\dot{Y}^{\mathrm{mis}}$ with Gibbs sampler

JM

Pro

- Conditionals are compatible
- Yield correct statistical inference under assumed JM
- Efficient parametrization possible
- Known theoretical properties

Con

- Lack of flexibility
- Can assume more than the complete data problem
- Leads to unrealistically large models

Fully conditional specification (FCS)

- Specify $P(Y^{\mathrm{mis}}|Y^{\mathrm{obs}}, X)$
- Draw imputations $\dot{Y}^{\mathrm{mis}}$ with Gibbs sampler

FCS example: Multivariate Imputation by Chained Equations (MICE)

- Specify imputation models $P(Y_j^{\mathrm{mis}} | Y_j^{\mathrm{obs}}, Y_{-j}, X)$
- Fill in starting imputations
- And iterate

Fully conditional specification (FCS)

Pro

- Extremely flexible, close to the data
- Subset selection of predictors
- Modular, can preserve valuable work
- Appears to work very well in practice
- Easy to explain

Con

- Theoretical properties only known in special cases
- Possible incompatibility
- No computational shortcuts

Hybrids of JM and FCS

- Partition variables into b *blocks* $h = 1, \ldots, b$
- Example:

b	partioning	model
4	$\{Y_1\}, \{Y_2\}, \{Y_3\}, \{Y_4\}, X$	MICE
2	$\{Y_1, Y_2, Y_3\}, \{Y_4\}, X$	hybrid
1	$\{Y_1, Y_2, Y_3, Y_4\}, X$	JM

JM embedded within FCS

b	h	target	predictors	type
2	1	$\{Y_1, Y_2, Y_3\}$	Y_4, X	mult
2	2	Y_4	Y_1, Y_2, Y_3, X	univ

FCS embedded within FCS

b	h	j	target	predictors	type
2	1	1	Y_1	Y_2, Y_3, Y_4, X	univ
2	1	2	Y_2	Y_1, Y_3, Y_4, X	univ
2	1	3	Y_3	Y_1, Y_2, Y_4, X	univ
2	2	1	Y_4	Y_1, Y_2, Y_3, X	univ

Consequence of blocks in `mice()`

- Main iteration loop over `blocks`
- `dim(predictorMatrix)`: $b \times p$
- `length(methods)`: b (instead of p)

Example: Multilevel data

```r
library(mice, warn.conflicts = FALSE)
```

```
## Loading required package: lattice
```

```r
library(miceadds)
```

```
## * miceadds 2.13-63 (2018-07-05 17:26:53)
```

```r
d <- brandsma[, c("sch", "lpo", "iqv", "den")]
```

FCS multilevel (c.f. Chapter 7, FIMD2)

```r
meth <- make.method(d)
meth[c("lpo", "iqv", "den")] <- c("2l.pmm", "2l.pmm",
                                  "2lonly.pmm")
pred <- make.predictorMatrix(d)
pred["lpo", ] <- c(-2, 0, 3, 1)
pred["iqv", ] <- c(-2, 3, 0, 1)
pred["den", ] <- c(-2, 1, 1, 0)
imp <- mice(d, pred = pred, meth = meth, seed = 418,
            m = 10, print = FALSE)
```

Idea 1: Embed joint model for multilevel data

```
# mitml::jomoImpute / called from mice
d$den <- as.factor(d$den)
blk <- make.blocks(d, "collect")
fm <- list(collect = list(lpo + iqv ~ 1 + (1 | sch),
                          den ~ 1))
imp <- mice(d, meth = "jomoImpute", blocks = blk,
            form = fm, print = FALSE, seed = 1,
            maxit = 2, m = 10, n.burn = 100)
```

Idea 2: Model-based imputation

- Define complete-data model $P(Y_1|Y_2, X)$
- Specify imputation model $P(Y_2|X)$ (no Y_1!)
- Rejection sampling to target $P(Y_2|Y_1, X)$
- Wu 2010, Bartlett 2015, Erler 2016
- R Software: `smcfcs`, `mdmb`, `jomo`
- Imputation compatible with complete-data model
- Can preserve deterministic relations
- Useful for strong, pre-specified complete-data models

Idea 3: Multivariate predictive mean matching

- Idea: Impute vector instead of scalar value
- Use redundancy analysis (van den Wollenberg 1977)
- Predictive mean matching using redundancy predictor
- Expected result: consistency within blocks, faster convergence

Idea 4: Imputing measurement scales

- Idea: Create module for imputing items + sum score
- Scale: items + sum score; some items are missing
- Items predicted by
  1. other items in scale
  2. sum scores other scales
- Expected result: simplification of model specification

predictorMatrix simplification: current method

```
##     age a1 a2 s_a b1 b2 b3 s_b
## age   0  0  0   1  0  0  0   1
## a1    1  0  1   0  0  0  0   1
## a2    1  1  0   0  0  0  0   1
## s_a   0  1  1   0  0  0  0   0
## b1    1  0  0   1  0  1  1   0
## b2    1  0  0   1  1  0  1   0
## b3    1  0  0   1  1  1  0   0
## s_b   0  0  0   0  1  1  1   0
```

predictorMatrix simplification: new method

```
blocks <- list(age = "age",
               A = c("a1", "a2", "s_a"),
               B = c("b1", "b2", "b3", "s_b"))
```

```
##     age a1 a2 s_a b1 b2 b3 s_b
## age   0  0  0   1  0  0  0   1
## A     1  0  0   0  0  0  0   1
## B     1  0  0   1  0  0  0   0
```

Idea 5: Combine imputation models from overlapping data

Source 1: $\{Y_1, Y_2\}$ given X - prefitted

Source 2: $\{Y_1, Y_3\}$ given X - prefitted

b	h	target	predictors	type
2	1	$\{Y_1, Y_2\}$	Y_3, X	mult
2	2	$\{Y_1, Y_3\}$	Y_2, X	mult

Conclusion

- Blocks are conceptually straightforward extension
- `blocks` implemented in `mice 3.0`
- Documentation still in the works
- Easy to specify hybrids of JM and FCS
  - Idea 1: Embed joint model for multilevel data
  - Idea 2: Model-based imputation
  - Idea 3: Multivariate predictive mean matching
  - Idea 4: Imputing measurement scales
  - Idea 5: Combine imputation models