Multiple imputation for lifecourse data

Brendan Halpin, Dept of Sociology, University of Limerick

Royal Statistical Society/SLLS June 9 2015
MICT: Gap-filling MI for lifecourse data

- Multiple imputation for categorical time series
- Particularly appropriate for life course history data
  - Spells in states, occasional transitions
  - Where missingness also tends to be consecutive
- More longitudinally coherent than MICE
- More appropriate to categorical time-series than approaches like Amelia
- Easy to use Stata add-on, computationally efficient
- Integrated with Stata’s MI framework
  - uses its imputation engine
  - uses Stata’s post-imputation estimation framework
Update

- This work updates work previously presented in 2012/3 (Halpin, 2012, 2013)
- Today I present new tests of performance using simulated and real data
- Now covers initial and terminal gaps as well as internal gaps
- Uses Stata’s imputation engine rather than home-brewed version
- Integrates with Stata’s MI infrastructure for post-imputation estimation
- Packaged as an easy-to-use Stata add-on
Missingness is endemic in longitudinal data

- Ever increasing availability of longitudinal data such as labour market, fertility, family formation, or residential histories
- But very subject to missingness, more than cross-sectional data
  - Repeated collection: attrition, contradiction
  - Demands of retrospection, etc
Some methods more affected than others

- Some methods can deal with missingness well
  - e.g. duration models can "censor" data from the first occurrence of missing onwards,
- Others require full data
- And throwing away data is wasteful, even where it does not introduce bias
Missingness in longitudinal data

Missingness is not random in lifecourse data

- Volatile life courses will be more prone to missingness
  - more likely to miss data collection point
  - less redundancy in the data (gap less likely to be papered over)
- Very often the information loss due to missingness is trivial
  - in practice lots of redundancy
  - a shame to throw the data away
- Hence we impute!
Rubin established the notion (1987)
- Draw several imputations from the predictive distribution of the imputation model
- Analyse each separately
- Combine the results according to "Rubin’s Rules"
Multiple imputation with missingness in multiple variables

- Straightforward with single variable to impute
- A bit more complicated if there are multiple incomplete variables
- If missingness is monotone, a sequence of single imputations is possible
Monotone missing
Monotone missing

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Monotone missing

![Diagram showing monotone missing data pattern](image-url)
Monotone missing
Gaps in longitudinal data
Imputation by gap-filling
Simulations and results
References

Multiple Imputation

Monotone missing
## Multiple Imputation

### Monotone missing

![Monotone missing pattern](image-url)
### Non-Monotone missing

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Non-monotonic, unordered

---

### Gaps in longitudinal data

- Imputation by gap-filling
- Simulations and results

### Multiple Imputation

---

### References

---
Non-Monotone missing

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Non-monotonic, partly ordered
Non-Monotone missing
Two approaches: Joint Modelling and MICE

- If non-monotonic, two approaches
  - "Joint Modelling" (i.e., model the joint distribution $P(Y,X,R)$)
  - MI by chained equations (van Buuren, 2007; van Buuren and Groothuis-Oudshoorn, 2011; Royston, 2009; White et al., 2011)

- The former has better theoretical foundations, but has substantial difficulties if some variables are categorical

- The latter is less well theorised but is flexible and experience says it works well, including for categorical variables.
Multiple Imputation

MICE

- A separate equation for each imputed variable
- Thus allows logit, ordinal logit, multinomial logit as appropriate
- Deals with the joint nature of imputation by an iterative chain:
  - First, cheaply impute all missing observations (e.g., hot-deck)
  - Then re-impute using earlier imputations and observed data
  - Repeat until convergence occurs (often quite soon)
- While the theoretical base is not fully established, it works well
- It improves on joint modelling particularly for categorical variables (van Buuren, 2007; Allison, 2005)
MICE not for time-series

- Existing implementations of MICE are not adapted for time-series
- As yet, no mechanisms for treating lags and leads as "passively imputed"
- Currently difficult to express models that take longitudinality properly into account
- Brute force approaches fail: high numbers of highly collinear variables
- As I show below, it tends to impute data with too little longitudinal stability (transition rates too high)
Multiple Imputation

Other MI software

- MI for time-series does exist
- In particular, Amelia (for R and Stata) (Honaker and King, 2010)
- However, this depends on joint imputation based on a MVN joint distribution
- As mentioned above, this is poor for categorical data (van Buuren, 2007; Allison, 2005)
Hence MICT: filling gaps with nearest info

- Treat explicitly as a time-series, use lags and leads to predict
- Focus on gaps rather than variables to orient sequence of imputation
- Effectively monotone missingness in this framework
- One model per unit of length of longest gap, not one per incomplete variable
The algorithm

Chained gap-healing

- Begin with longest gap, predict first (or last) element
- Then predict last (or first) of next shortest gap length (including longer gaps already reduced)
- Until no gaps remain
- Important to begin fill from edges
  - Least distance from observed data
  - But each gap has two edges: to begin pick one at random and impute
  - Then the other edge (of the newly shortened gap) has better data than the former, so alternate
The algorithm

Sketching gap closure

- Five unit gap
  - XXX.....XXX
- Three unit gap
  - XXX...XXXXX
The algorithm

Sketching gap closure

- Five unit gap
  - XXX.....XXX
  - XXX....iXXX

- Three unit gap
  - XXX...XXXXX
  - XXXi...IXXX
  - XXXI..iIXXX
  - XXXIi.IIXXX
  - XXXIIiIIXXX

The algorithm aims to fill in gaps in longitudinal data. The sketching gap closure approach is outlined with examples for five and three unit gaps, showing how data can be imputed. Further details on simulations and results are provided in the associated section.
The algorithm

Sketching gap closure

- Five unit gap
- Three unit gap

- XXX.....XXX
- XXX....iXXX
- XXXi...IXXX
- XXXX....XXX
- XXXX...iXXX
- XXXX...IXXX

- XXXX...XXXXX
The algorithm

Sketching gap closure

- Five unit gap
  - XXX.....XXX
  - XXX....iXXX
  - XXXi...IXXX
  - XXXI..iIXXX

- Three unit gap
  - XXX...XXXXX
  - XXX...XXXXX
  - XXX...XXXXX
  - XXX...XXXXX
Sketching gap closure

- Five unit gap
  - XXX....XXX
  - XXX....iXXX
  - XXXi....IXXX
  - XXXi..iIXXX
  - XXXiI..IIXXX

- Three unit gap
  - XXX...XXXX
  - XXX...XXXX
  - XXX...XXXX
  - XXX...XXXX
  - XXXXi...XXXXX
The algorithm

Sketching gap closure

- **Five unit gap**
  - XXX....XXX
  - XXX....iXXX
  - XXXi....IXXX
  - XXXI..iIXXX
  - XXXIi.IIXXX
- **Three unit gap**
  - XXX...XXXX
  - XXX...XXXX
  - XXX...XXXX
  - XXX..iXXXXX
  - XXXIi.IXXXXX
MICT for Stata

- Implemented as a Stata add-on: Multiple Imputation for Categorical Time-series (MICT: soon available in SSC)
- Key added value is handling the updating of lag and lead vars, defining the sequence of operations
- Predictive model: at least prior and subsequent states, but can be more sophisticated
  - summaries of prior and subsequent histories
  - time-varying effects
  - fixed individual-level variables
  - other time-dependent variables (fully observed or simply imputed)
- Analogous models for initial and terminal gaps
Examples

- Some demonstrations
  1. Real data with simulated missing: compare imputed with observed
  2. Simulated data with simulated missing: compare MICT and MICE using very simple data
  3. Real data with real gaps, using a fairly complex model
  4. Real data with real gaps, using more complex model that takes data collection context into account
Real data with simulated missing

- Data (McVicar and Anyadike-Danes, 2002):
  - 6 years of monthly data
  - Labour market histories of Northern Irish youth
- Insertion of missingness at random
  - Each month has a 1.25% chance of being missing, but
  - But 67% chance if the previous month is missing
- ⇒ consecutive runs of missingness, MCAR wrt observed data
Default imputation model

```
mi impute mlogit `_mct_state` i.`_mct_next` i.`_mct_last` . . .
```

- where `_mct_state` is the internal copy of the state variable
- `_mct_last` and `_mct_next` are respectively the most recent and nearest future observation
- Initial and terminal gaps are imputed using only respectively subsequent and prior information.

```
use mvadmar
mict_prep state, id(id)
mict_impute
```
Defining better imputation models

- Default imputation model is very simple:
  \[ Y = f(X_{t-lag}, X_{t+lead}) \]
- Implicitly assumes a zero-order Markov process with time-constant transition rates
- We can over-ride the built-in models by redefining the programs
  - mict_model_gap
  - mict_model_initial, and
  - mict_model_terminal
Over-ride internal gap model

capture program drop mict_model_gap
program define mict_model_gap
mi impute mlogit _mct_state
    i._mct_next##c._mct_t i._mct_last##c._mct_t
    _mct_before* _mct_after*,
    add(1) force augment
end
Built-in variables

- Variables _mct_before1 to _mct_beforeC and _mct_after1 to _mct_afterC are built-in.
- The proportion of time before and after the gap spent in each of the C categories of the state variable.
- ⇒ incorporate history beyond zero-order in a simple way.
- Interactions i._mct_next##c._mct_t, i._mct_last##c._mct_t allow for time-varying transition rates.
- Other variables can also be entered:
  - fixed individual variables
  - variables indicating time-dependent state in other domains.
Three cases, with 10 imputations
Examining the imputations

- Four cases of gaps embedded in a single state:
  - nearly always filled with that state
  - one example of it being filled with another plausible state

- Two examples of gaps between two different states
  - Imputations mostly randomise the point of transition
  - A few imputations interpolate spells in other states

- One gap spans a complete spell: this is very unlikely to be imputed
A good if not perfect performance: lots of redundancy in lifecourse histories

One particular worry: spells that are entirely missing are not recovered
  - It may be that the process generating missingness is related to spell structure

Below I consider a way of partially addressing this
Comparing MICT with MICE

- Very difficult to fit a model like this with the conventional MICE framework
- Models with "everything in" will fail computationally
  - too many variables
  - much too collinear
- Models with more refined prediction equations are very hard to express
- Neither mi impute nor ice are adapted for lags and leads, etc.
Strategy: Compare performance on very simple data

- Generate simple simulated data where a very simple model is correct
- To wit, 36-element long, 4-categories, with fixed transition rates and a zero-order Markov process
- MCAR runs of missingness
- Zero-order $\Rightarrow$ only adjacent last and next observations carry information with which to impute
- MICT uses only _mct_last and _mct_next as predictors
- MICE uses only immediately adjacent states, $X_{t-1}$ and $X_{t+1}$
Simulated data with simulated missing

Royston’s ICE

```stata
ice m.m1 m.m2 m.m3 m.m4 m.m5 m.m6 m.m7 m.m8 m.m9 m.m10 ///
m.m11 m.m12 m.m13 m.m14 m.m15 m.m16 m.m17 m.m18 ///
m.m19 m.m20 m.m21 m.m22 m.m23 m.m24 m.m25 m.m26 ///
m.m27 m.m28 m.m29 m.m30 m.m31 m.m32 m.m33 m.m34 ///
m.m35 m.m36, ///
saving(ice, replace) persist m(10) cycles(10) ///
eq(m1: i.m2 , ///
m36: i.m35 , ///
m2: i.m1 i.m3, ///
m3: i.m2 i.m4, ///
[ ... ]
m35: i.m34 i.m36)
```
Simulated data with simulated missing data

Stata’s `mi impute chained`

```stata
mi set flong
mi register imputed m*
mi impute chained ///
    (mlogit, omit(i.m3 i.m4 [...] i.m34 i.m35 i.m36 )) m1 ///
    (mlogit, omit(i.m4 [...] i.m34 i.m35 i.m36 )) m2 ///
    (mlogit, omit(i.m1 [...] i.m34 i.m35 i.m36 )) m3 ///
    (mlogit, omit(i.m1 i.m2 [...] i.m34 i.m35 i.m36 )) m4 ///
    (mlogit, omit(i.m1 i.m2 i.m3 [...] i.m34 i.m35 i.m36 )) m5 ///
    [...] ///
    (mlogit, omit(i.m1 i.m2 i.m3 i.m4 [...] )) m35 ///
    (mlogit, omit(i.m1 i.m2 i.m3 i.m4 [...] i.m34 )) m36 ///
, add(10) force augment
```
Simulated data with simulated missing

Some imputations, MICT and MICE
Simulated data with simulated missing

Too many transitions

- Inspection suggests that mi impute and ice are more prone to interpolating spells in other states
- Is this a systematic feature?
- Calculate the difference between the observed and impute number of spells for each case
- Use mi estimate to carry out a t-test using Rubin’s rules

\[ H_0 : N_{obs} = N_{imp} \]

<table>
<thead>
<tr>
<th>Method</th>
<th>Difference</th>
<th>Std. Err.</th>
<th>t</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICT</td>
<td>-.0058</td>
<td>.0326</td>
<td>-0.18</td>
<td>0.859</td>
</tr>
<tr>
<td>mi impute</td>
<td>.2733</td>
<td>.0305</td>
<td>8.95</td>
<td>0.000</td>
</tr>
<tr>
<td>ice</td>
<td>.3962</td>
<td>.0434</td>
<td>9.14</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Simulated data with simulated missing

MICT has greater longitudinal consistency

- 10 imputations, 2000 sequences, 3 methods
- With a very simple data set, MICT outperforms MICE in terms of longitudinal consistency
From simulation to a real example

- The first example used real data with simulated missing
- The second example used simulated data with simulated missing
- Now an example with real missing data from BHPS
  - 6 years of monthly data, women who have a birth at end year 2
    - Employed full-time
    - Employed part-time
    - Unemployed
    - Not in the labour market
  - 706 fully observed sequences, 194 with gaps under 12 months, c400 with bigger gaps but with data that can be used for prediction
State distribution: Mothers’ labour market history

- Full time employed
- Part time employed
- Unemployed
- Non-employed

Time (0-80)
Real data with real missing

Indexplot: Mothers’ labour market history
Predictive model

```
mi impute mlogit _mct_state ///
   i._mct_next##c._mct_t##c._mct_t ///
   i._mct_last##c._mct_t##c._mct_t ///
   _mct_before*    ///
   _mct_after*
```

- Implies transition pattern that varies in a non-linear fashion
- Uses history and future distribution of states
Gaps in longitudinal data

Imputation by gap-filling

Simulations and results

Real data with real missing

Gappy sequences are differently distributed: cluster analysis

<table>
<thead>
<tr>
<th>1, 0</th>
<th>1, 1</th>
<th>2, 0</th>
<th>2, 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3, 0</th>
<th>3, 1</th>
<th>4, 0</th>
<th>4, 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5, 0</th>
<th>5, 1</th>
<th>6, 0</th>
<th>6, 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7, 0</th>
<th>7, 1</th>
<th>8, 0</th>
<th>8, 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
</tr>
</tbody>
</table>

Graphs by g5_8 and gap
Information from data collection structure

- In the initial simulation, I noted that if a gap spans a complete spell it will be lost
  - no redundancy in this case
- If missingness is related to spells this can be a systematic feature $\Rightarrow$ bias
- In the BHPS, missingness (and transition patterns) is correlated with data collection structure (Halpin, 1998):
  - Month of interview disproportionately likely to be followed by gap, or transition
  - 1st month of a reported spell likely to follow a gap/transition
Is missingness related to data collection?

Time-dependent observations

- Can bring this to bear on the imputation, improving the imputation of transition points
- A monthly observation:
  - Nothing special
  - Reported start of spell current at interview
  - Reported start of a spell in the inter-wave job history
  - Date of interview
- Improves the fit of the model, improves the timing of predicted transitions
Is missingness related to data collection?

Selected imputations without (L) and without data collection info (R)
Conclusions

- MICT creates realistic imputations of gap-prone lifecourse data
- It respects longitudinal continuity better than MICE
- It is easy to define good prediction models with MICT
- It is reasonably stable in computational terms
- Longitudinal data is often missing, and not at random: needs imputation
- Important to pay attention to the processes generating gaps too


