Multi-state modelling software, and encouraging statistical software development.

Chris Jackson
MRC Biostatistics Unit, Cambridge, U.K.

MRC Biostatistics Unit Centenary Conference, 25 March 2014
Overview

Part 1. Software for multi-state models
▶ Two different types of multi-state model
▶ msm package for R — features, design principles, potential developments
▶ survival + mstate packages for R.

Part 2. Encouraging statistical software development in biostatistics research
▶ What’s needed, and how to do it. Start discussion…
Part I

Software for multi-state modelling — state of the art and future.
Multi-state models in continuous time

Example:

Defined by matrix $Q$ of transition intensities: instantaneous risk of moving from state $r$ to state $s \neq r$: at time $t$:

$$q_{rs}(t, \mathcal{F}(t-)) = \lim_{\delta t \to 0} P(S(t + \delta t) = s | S(t) = r, \mathcal{F}(t-))/\delta t.$$ 

e.g. Markov time-homogeneous model, $q_{rs}$ independent of \left\{ \frac{\mathcal{F}(t-)}{t} \right\}

Single period (sojourn time) in state $r \sim \text{Exp}($mean $= -1/q_{rr})$
Multi-state models in continuous time

Example:

Defined by matrix $Q$ of transition intensities: instantaneous risk of moving from state $r$ to state $s \neq r$: at time $t$:

$$q_{rs}(t, F(t-)) = \lim_{\delta t \to 0} P(S(t + \delta t) = s | S(t) = r, F(t-))/\delta t.$$ 

e.g. Markov time-homogeneous model, $q_{rs}$ independent of $\{F(t-)\}$

- Single period (sojourn time) in state $r \sim \text{Exp}(\text{mean} = -1/q_{rr})$
Data for multi-state models (1): intermittently-observed

Panel data. State only observed at a finite number of times $j$.

- Don’t know the state between these times
- e.g. chronic disease only measurable at clinic visit / screening test

- Likelihood is product of transition probabilities between states $S(t_j)$ observed at successive $t_j$ (Kalbfleisch & Lawless, JASA 1985).

\[ L(Q) = \prod_{j} P_{S(t_j),S(t_{j+1})}(t_{j+1} - t_j). \]

- Closed form for corresponding matrix $P(t) = \exp(tQ)$ only if $Q$ is constant / piecewise constant with time $t$.
- Non-Markov models difficult (see later...)

`msm` package for R — designed for this type of data
Data for multi-state models (1): intermittently-observed

Panel data. State only observed at a finite number of times $j$.

- Don’t know the state between these times
- e.g. chronic disease only measurable at clinic visit / screening test

Likelihood is product of transition probabilities between states $S(t_j)$ observed at successive $t_j$ (Kalbfleisch & Lawless, JASA 1985).

$$L(Q) = \prod_j P_{S(t_j),S(t_{j+1})}(t_{j+1} - t_j).$$

- Closed form for corresponding matrix $P(t) = \exp(tQ)$ only if $Q$ is constant / piecewise constant with time $t$.
- Non-Markov models difficult (see later...)

**msm package for R** — designed for this type of data
Data for multi-state models (2): completely-observed

Observe all changes of state

- know the complete process history.
- e.g. changes of state represent events
  - MI, stroke, periods in hospital.
- event times may be known from administrative data.
- Time-to-event data with competing event times censored.
- substantial literature on survival / competing risks
- Only Markov models supported by \texttt{msm}.
  - exponential / piecewise-exponential event times.
- Can estimate transition rates under more flexible models (e.g. Cox semi-Markov) using standard survival analysis software.

![Diagram of state transitions with 0 and 12 units of time showing transitions between State 1, State 2, State 3, and Death]
Observe all changes of state

- know the complete process history.
- e.g. changes of state represent events
  - MI, stroke, periods in hospital.
- event times may be known from administrative data.

- Time-to-event data with competing event times censored.
  - substantial literature on survival / competing risks
- Only Markov models supported by \texttt{msm}.
  - exponential / piecewise-exponential event times.
- Can estimate transition rates under more flexible models (e.g. Cox semi-Markov) using standard survival analysis software.

\textit{survival} and \textit{mstate} packages for R designed for this
msm R package for multi-state modelling

http://CRAN.R-project.org/package=msm


Used in health, finance, ecology, social science, engineering...

General and flexible. Fit continuous-time Markov models

- with any state structure / transition matrix
- covariates (proportional intensities) for any / all transitions
  - subject-specific time-constant or
  - piecewise-constant time-dependent, including time itself
- to various patterns of observation, particularly intermittently-observed...

```r
msm(state ~ time, subject=subj, data=mydata,
    covariates = list("1-2" = ~ age,
                      "2-3" = ~ age + treatment),
    qmatrix=rbind(c(0,1,1),
                  c(0,0,1),
                  c(0,0,0)), gen.inits=TRUE)
```

Chris Jackson, MRC-BSU Cambridge

Multi-state modelling, and encouraging more software
Other observation schemes

State entry time observed, but state at previous time unknown (typical for times of death)

```
# state 4 like this
msm(..., death=4, ...)
```

States only observed to be in a certain set of possible states (e.g. alive, but unknown disease severity)

```
# (put 999 in data at these times)
msm(..., censor=999, censor.states=c(1,2,3))
```

Appropriate likelihoods computed / maximised by `msm`.
(No truncated samples, or informative observation times).
Hidden Markov models

States observed with error, from a true hidden Markov chain.

\[
\begin{align*}
\text{ematrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},
\end{align*}
\]

Continuous outcome, conditional on a hidden Markov chain (many outcome distributions)

\[
\begin{align*}
\text{hmodel} &= \begin{cases} 
\text{hmmNorm}(\text{mean}=100, \text{sd}=16), \\
\text{hmmNorm}(\text{mean}=54, \text{sd}=18), \\
\text{hmmIdent}(999)
\end{cases},
\end{align*}
\]

\textit{msm} jointly estimates (with covariates on either)

- Markov transition intensities \( Q \)
- misclassification probabilities / outcome parameters
Other features

- Constrain parameters to equal other parameters or to known values — parsimony / model checking
  
  ```
  msm(...,
      constraint = list(age = c(1,1,1,2,2),
                      treatment = c(1,2,3,4,4)),
      fixedpars = c(4,9),
      ...)
  ```

- Mixture of observation types (intermittent, exact, misclassified...) in the same data
  
  ```
  msm(..., obstype = obs, obstrue = obst, ...)
  ```

- **Model assessment**: plots and formal tests of fit
  
  ```
  plot.prevalence.msm(x), pearson.msm(x), ...
  ```

---

Programming principles (1)

(I don’t always follow these! Takes time)

- Minimum input from user: sensible defaults, that advanced users can tweak.
- Complete documentation, user guides and examples.
  - Update documentation if people find something unclear / commonly misuse in same way
- Outputs that are meaningful / helpful.
  - e.g. confidence intervals instead of SEs / p-values, hazard ratios as well as log HRs.
- Useful messages. Tell the user how to fix any errors:

Data inconsistent with transition matrix

Data inconsistent with transition matrix: subject 200 moves from state 4 to state 2 at non-missing observation 1436
Programming principles (1)

(I don’t always follow these! Takes time)

- Minimum input from user: sensible defaults, that advanced users can tweak.
- Complete documentation, user guides and examples.
  - Maintain it / fix bugs. Reply helpfully to emails!
    - Update documentation if people find something unclear / commonly misuse in same way
- Outputs that are meaningful / helpful.
  - e.g. confidence intervals instead of SEs / p-values, hazard ratios as well as log HRs.
- Useful messages. Tell the user how to fix any errors:

Data inconsistent with transition matrix

Data inconsistent with transition matrix: subject 200 moves from state 4 to state 2 at non-missing observation 1436
(I don’t always follow these! Takes time)

- **Modular / maintainable code.**
- Avoid reinventing the wheel – build on others’ work
  - e.g. msm now depends on mexp (Goulet et al.), a nice library for matrix exponentiation, instead of relying on its own methods
- Automatic “unit testing”: for each package function, when updating package
  
  ```r
  stopifnot(result_from_new_version() == known_result)
  ```

- Efficiency: vectorised R code, heavy numerical work done in C
Future developments in \texttt{msm}

- Many internal cleanups, and unglamorous new features, in past year.
  - In progress: modified AIC to compare models with differently-aggregated state structures (Howard Thom)
- Future: test assumptions (Markov, time/between patient homogeneity) through fitting \textit{more complex models}...
  - ... but difficult with \textit{intermittent observations: unobserved event times / paths through states.}
  - Integrating likelihoods over possible paths / event times only feasible for simple transition structures (e.g. well/illness/death)
Semi-Markov models for panel data

- Transition intensity depends on time spent in current state.
- Difficult for panel data: don’t know time of entry into state.

Phase-type models (Titman and Sharples, Biometrics 2010; Titman, Statistics and Computing 2011)

Exponential sojourn in current state replaced by embedded Markov model with $k$ phases: sojourn time is time to state $k+1$.

May be implemented as hidden Markov models in \texttt{msm}

- many more parameters, how to interpret?
- identifiability constraints not implemented yet
Random effects models

▶ Unexplained **heterogeneity** in transition intensities between individuals / groups.

▶ Likelihood only tractable for specific cases
  ▶ e.g. **discrete** random effects distribution (Cook et al, Biometrics 2004)
  ▶ **same** random effect for all intensities (Satten, Biometrics 1999).
  ▶ **conjugate** gamma frailties, exact event times (Putter & Houwelingen, SMMR 2011), **mixture** models / mover-stayer (O'Keeffe et al. Stat. Med. 2012)…

▶ **MCMC** approaches (JAGS / BUGS / Stan software) or Monte Carlo EM (Sutradhar and Cook, JRSS C, 2008)?
  ▶ experimental facility available in **msm** to generate code to fit same model in JAGS
Time-inhomogeneous models for panel data

- Likelihood needs transition probability matrix $P$ with entry $Pr(S(t_1) = s | S(t_0) = r)$.
- Kolmogorov forward equations
  \[ \frac{dP(t_0, t_1)}{dt} = P(t_0, t_1)Q(t) \]
- $Q$ not constant or piecewise constant with time $\rightarrow$ no analytic solution.

Or numerically solve the differential equation (Titman, Biometrics 2011).

- Allows e.g. Weibull or spline functions for $Q(t)$ — smoother / more realistic than piecewise constant
- Need to solve for each distinct covariate value — hard for continuous covariates / big datasets.

Chris Jackson, MRC-BSU Cambridge
Time-inhomogeneous models for panel data

- Likelihood needs transition probability matrix $P$ with $r, s$ entry $Pr(S(t_1) = s \mid S(t_0) = r)$.
- Kolmogorov forward equations $dP(t_0, t_1)/dt = P(t_0, t_1)Q(t)$
- $Q$ not constant or piecewise constant with time $\rightarrow$ no analytic solution.

Or numerically solve the differential equation (Titman, Biometrics 2011).

- Allows e.g. Weibull or spline functions for $Q(t)$ — smoother / more realistic than piecewise constant
- Need to solve for each distinct covariate value — hard for continuous covariates / big datasets.
Future developments in msm

Ideally any new methods should work with any multi-state structure.

- or at least for common structures: e.g. progressive disease
  □-□-□-□, or progression and death □
- unmaintainable if handle too many special cases in different ways.
All transition times known: survival/mstate system

Model fitting: **survival** R package (Therneau)

- Estimate event-specific hazards \(\Rightarrow\) transition rates
  - under Cox or fully parametric models for times to each event.


- Estimate cumulative incidences from a Cox model (Breslow)
- Convert these to transition probabilities over a time period
  - Aalen-Johansen estimator for inhomogeneous Markov models
  - Individual patient simulation for semi-Markov models
- No documentation for using parametric models
  - needed e.g. for extrapolation in health economic evaluations
- Data need some awkward manipulation

Tutorials / courses to clear up **msm** vs. **mstate** confusion / make all their methods more accessible?

Chris Jackson, MRC-BSU Cambridge
Part II

Encouraging more software development in biostatistics research
Why encourage software development?

Statistical methods need accessible software.

- allows the method to be used by more people — especially non-experts
- saves time for everyone, even experts
- increases transparency / trust in research results
- good for promoting a new method
  - “the other way our ideas get out there is through software . . . software implementation is a kind of publication, indeed, one of the best kinds.” (http://andrewgelman.com/2014/03/12/publishing-journals)
  - see, e.g. popularity of DIC in BUGS
- impact, citations . . .

→ good for science (and scientists).
Types of statistical software

Ad-hoc code, often to accompany a journal paper

- Typically only usable by experts, maybe only after tweaking
- OK for reproducibility / transparency
- Bare minimum

R (or Stata) packages.

Standalone software / large libraries (e.g. BUGS, JAGS, Stan).

Chris Jackson, MRC-BSU Cambridge

Multi-state modelling, and encouraging more software
Types of statistical software

Ad-hoc code, often to accompany a journal paper

R (or Stata) packages.

➤ Documented and maintained, easy to install and use...
➤ Builds on great work done in the last 10-15 years to make R and CRAN the main platform for statistics research.
➤ Ideal that we can aim for for new methodology.

➤ see e.g. Jeff Leek (Johns Hopkins) group policy: all PhD students to develop and maintain an R package: http://simplystatistics.org/2013/10/07/the-leek-group-policy-for-developing-sustainable-r-packages/

Standalone software / large libraries (e.g. BUGS, JAGS, Stan).
Types of statistical software

- Ad-hoc code, often to accompany a journal paper
- R (or Stata) packages.
- Standalone software / large libraries (e.g. BUGS, JAGS, Stan).
  - Often to accompany a major methodological advance (e.g. MCMC).
  - Needs advanced programming / software engineering skills to develop.
  - Still needs users for feedback / bug reports / testing / support
How can we develop more accessible software?

Culture shift: software viewed as a valuable research output

- Funding bodies / grant reviewers, research assessors, PhD examiners, journal editors, supervisors and line managers...

Time and money...

People and skills...
How can we develop more accessible software?

Culture shift: software viewed as a valuable research output

Time and money...

- Priorities: consider a methodology project not finished without usable software — not an “optional extra”
- A lot of tedious work involved in writing software — but same is true for writing papers!

People and skills...
How can we develop more accessible software?

Culture shift: software viewed as a valuable research output

Time and money...

- Priorities: consider a methodology project not finished without usable software — not an “optional extra”
- A lot of tedious work involved in writing software — but same is true for writing papers!

“...[our academic culture] has traditionally put a very high premium on being clever and a relatively low premium on being willing to go through the schlep. [As applied statistics grows] the schlep becomes just as important as the clever idea. If you aren’t willing to put in the time to code your methods up and make them accessible to other investigators, then who will be?”

Jeff Leek (http://simplystatistics.org/2012/05/28/schlep-blindness-in-statistics)

People and skills...
How can we develop more accessible software?

Culture shift: software viewed as a valuable research output

Time and money...

- Priorities: consider a methodology project not finished without usable software — not an “optional extra”
- A lot of tedious work involved in writing software — but same is true for writing papers!

“creating an R package is building something. It is something you can point to and say, "I made that”. Leaving aside all the tangible benefits to your career, the profession, etc. it is maybe the most gratifying feeling you get when working on research.” (Jeff Leek, https://github.com/jtleek/rpackages)

People and skills...
How can we develop more accessible software?

Culture shift: software viewed as a valuable research output

Time and money...

People and skills...

▶ More collaborative programming, just like we do collaborative writing (tools could help e.g. GitHub)
  ▶ Informal / internal peer review by local software experts
▶ Training of students / researchers in software development
  ▶ Software user / discussion groups (e.g. for R techniques)
  ▶ Courses and online resources. A lot to be learnt from “open source” community!
▶ Collaborations with computing specialists: especially for major software projects: [http://www.timeshighereducation.co.uk/news/save-your-work-give-software-engineers-a-career-track/2006431.article](http://www.timeshighereducation.co.uk/news/save-your-work-give-software-engineers-a-career-track/2006431.article)
More than just software

“Knowledge transfer”: Link software with documentation and training
→ make methods more accessible, avoid them being misused

- Tutorial papers
  - “Vignettes”: tutorials with worked examples, packaged with the software
  - Journal of Statistical Software, R Journal, Stata Journal
    — gives author a publication!

- Short courses
Discussion

Disclaimer: My perspective is limited to MRC-BSU / biostatistics, and biased by my own enthusiasm!

- What’s your experience?
- Are we doing OK as we are?
- What else can we do?