Continuous Time Dynamic Modeling in `dynr` and `OpenMx`

Michael D. Hunter

School of Psychology
Georgia Institute of Technologyr

July 9, 2017
Dynamic Modeling Workshop
IMPS
New York City, New York

Plan

- Discrete vs Continuous Time
- Software
- General Modeling
- Examplles

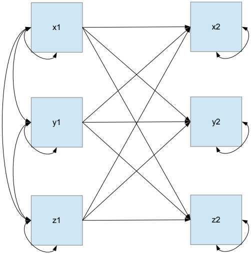
Discrete-Time Parameters Depend on the Lag

Simulation Design

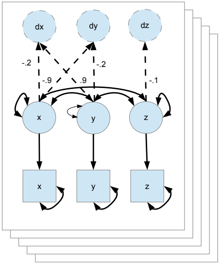
- 2-occasion data
- 3 variables
- 100 individuals
- Continuous generating process
- Sampled in discrete time
- Measurement Interval: 7 hours to 1 year
- Fit discrete- & continuous-time structural equation models to each sampling interval
- Compare parameters across conditions

Discrete-Time Parameters Depend on the Lag

Condidate Models

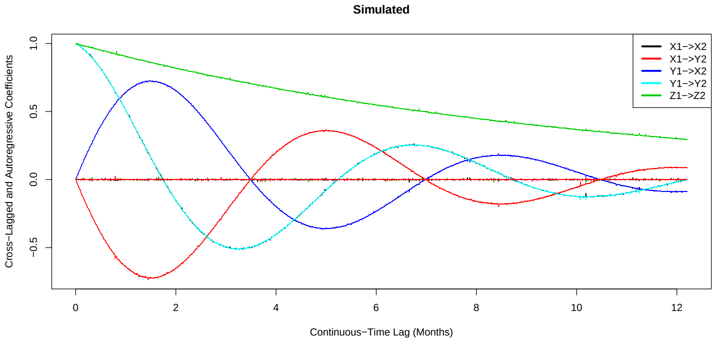


Cross-lagged Panel Model

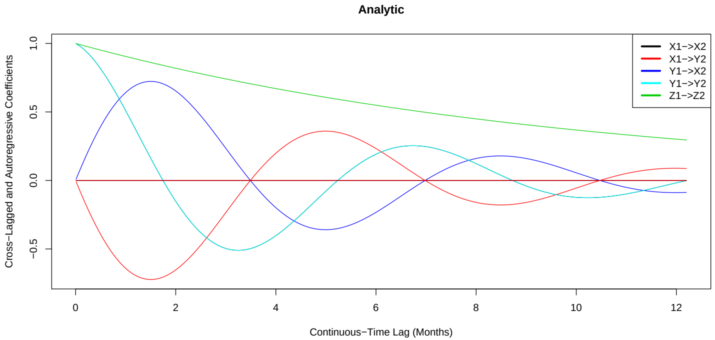


Continuous-Time Structural Equation Model

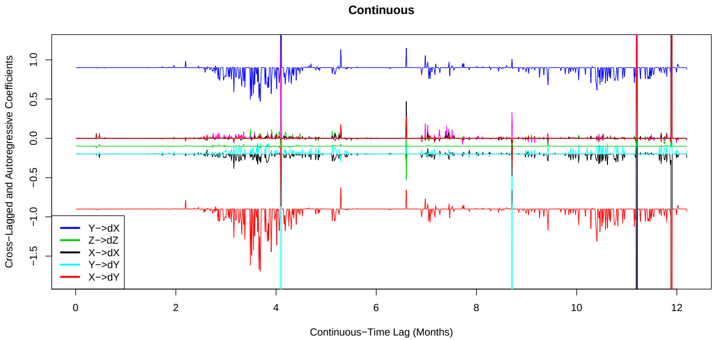
Discrete-Time Parameters Depend on the Lag



The Dependence has an Analytic Form



Continuous-Time Parameters Do Not Depend on Lag



Summary

- One-occasion mediation has no causal direction.
- Cross-lagged panel designs obtain different estimates for each measurement interval.
- Continuous-time models obtain constant estimates for every measurement interval.
- Estimated discrete-time parameters match those analytically predicted.
- Continuous-time models even work on two-occasion data!

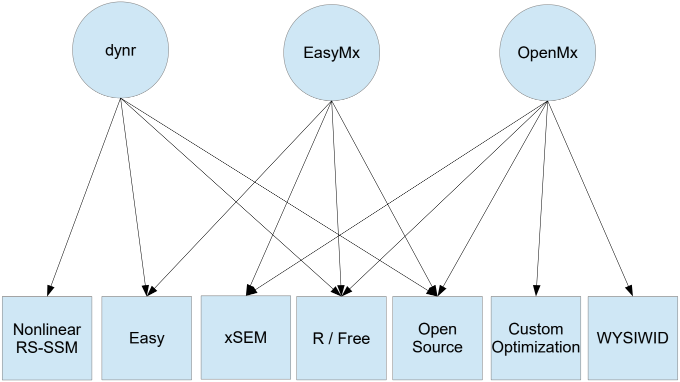
Implications

- Researchers using different lags will find different results, even when the process is the same!
- Researchers using the same lag will find different results, depending on which lag they pick!
- Also applies to pre-test post-test designs!

Limitations

- Continuous convergence problems?
- Operation with real data?
- Mis-specified/imperfect models?

Software


dynr
EasyMx
OpenMx
Nonlinear RS-SSM
Easy
xSEM
R / Free
Open Source
Custom Optimization
WYSIWID

OpenMx $\geq$ 2.0

Recent Features

- objective function $= \underbrace{\text{expectation function}}_{\text{data generation}} + \underbrace{\text{fit function}}_{\text{data comparison}}$
- Multiple optimizers: SLSQP, NPSOL, CSOLNP
- New Models: Item factor analysis, GREML, state space, LISREL
- New Helpers: `mxFactorScores`, `mxGenerateData`, `mxMI`, `mxRefModels`, `mxTryHard`

OpenMx $\geq$ 2.7

More Recent Features

- Handy S3 methods: `logLik`, `confint`, `anova`, `coef`, `simulate`
- New Fit Functions for multigroup, mixtures, hidden Markov models, weighted least squares
- Utilities: `mxSE`, `mxAutoStart`, nonparametric and parametric `mxBootstrap`, `mxCheckIdentification`, `mxGetExpected`

Why dynr?

Why make a new package for this?

- Linear and nonlinear models
- Multiple subjects
- Intuitive interface
- Fast
- Great reporting
- Combine dynamics for continuous and categorical latent variables

dynr preparation

- Gather data with `dynr.data()`
- Prepare *recipes* with
  - `prep.measurement()`
  - `prep.*Dynamics()`
  - `prep.initial()`
  - `prep.noise()`
  - `prep.regimes()` (optional)
- *Mix* recipes and data into a model with `dynr.model()`
- *Cook* model with `dynr.cook()`
- *Serve* results with
  - `summary()`
  - `plot()`
  - `dynr.ggplot()`
  - `plotFormula()`
  - `printex()`

Example 0

OpenMx

See `StateSpaceExpectationCodeListings.R` for OpenMx examples of

1. Process factor analysis in discrete and continuous time
2. Use of `mxSE` to compute standard error on transformed parameters
3. Hypothesis testing of nested models
4. Kalman scores
5. Data generation
6. Multisubject models

State Space Model

Measurement

- Structural Equation Measurement Model

$$\boldsymbol{y}_i = \Lambda \boldsymbol{\eta}_i + K\boldsymbol{x}_i + \boldsymbol{\varepsilon}_i \quad \text{with} \quad \boldsymbol{\varepsilon}_i \sim \mathcal{N}\left(\mathbf{0}, \Theta\right) \tag{1}$$

- State Space Measurement Model

$$\boldsymbol{y}_i = \Lambda \boldsymbol{\eta}_i + K\boldsymbol{x}_i + \boldsymbol{\varepsilon}_i \quad \text{with} \quad \boldsymbol{\varepsilon}_i \sim \mathcal{N}\left(\mathbf{0}, \Theta\right) \tag{2}$$

State Space Model

Transition/Structural

- Structural Equation Structural Model

$$\boldsymbol{\eta}_i = B\boldsymbol{\eta}_i + \Gamma\boldsymbol{x}_i + \boldsymbol{\zeta}_i \quad \text{with} \quad \boldsymbol{\zeta}_i \sim \mathcal{N}(\mathbf{0}, \Psi) \tag{3}$$

- State Space Structural Model in Discrete Time

$$\boldsymbol{\eta}_{i+1} = B\boldsymbol{\eta}_i + \Gamma\boldsymbol{x}_i + \boldsymbol{\zeta}_i \quad \text{with} \quad \boldsymbol{\zeta}_i \sim \mathcal{N}(\mathbf{0}, \Psi) \tag{4}$$

- State Space Structural Model in Continuous Time

$$\frac{d\boldsymbol{\eta}}{dt} = B\boldsymbol{\eta}_i + \Gamma\boldsymbol{x}_i + \boldsymbol{\zeta}_i \quad \text{with} \quad \boldsymbol{\zeta}_i \sim \mathcal{N}(\mathbf{0}, \Psi) \tag{5}$$

Notation

Structure

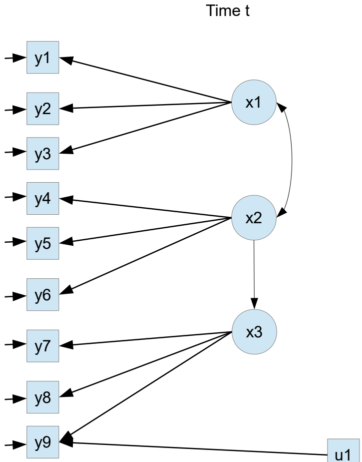
Durbin & Koopman	$\boldsymbol{\alpha}_t$	$=$	$\boldsymbol{T}_t \boldsymbol{\alpha}_{t-1} + \boldsymbol{R}_t \boldsymbol{\eta}_t$
LISCOMP	$\boldsymbol{\eta}_t$	$=$	$\boldsymbol{B}_t \boldsymbol{\eta}_{t-1} + \boldsymbol{\Gamma}_t \boldsymbol{x}_t + \boldsymbol{\zeta}_t$
West & Harrison	$\boldsymbol{\theta}_t$	$=$	$\boldsymbol{G}_t \boldsymbol{\theta}_{t-1} + \boldsymbol{w}_t$
Kalman	$\boldsymbol{x}_t$	$=$	$\boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{B}_t \boldsymbol{u}_t + \boldsymbol{w}_t$
Åstrom & Murray	$\boldsymbol{x}_t$	$=$	$\boldsymbol{A}_t \boldsymbol{x}_{t-1} + \boldsymbol{B}_t \boldsymbol{u}_t + \boldsymbol{v}_t$
OpenMx	$\boldsymbol{x}_t$	$=$	$\boldsymbol{A}_t \boldsymbol{x}_{t-1} + \boldsymbol{B}_t \boldsymbol{u}_t + \boldsymbol{q}_t$

Notation

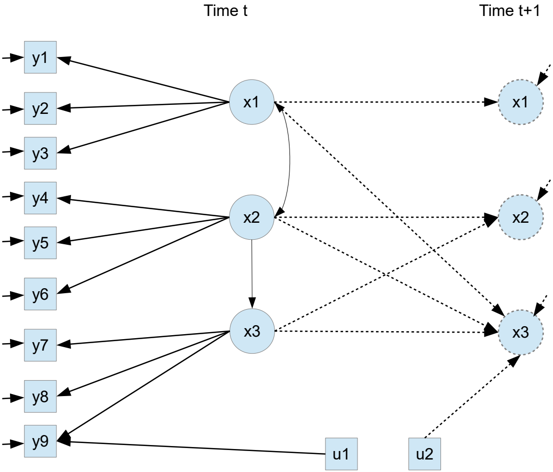
Measurement

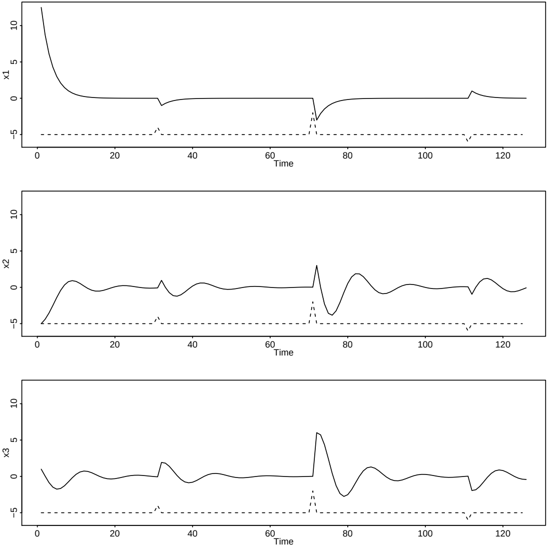
Durbin & Koopman	$\boldsymbol{y}_t$	$=$	$\boldsymbol{Z}_t\boldsymbol{\alpha}_t + \boldsymbol{\varepsilon}_t$
LISCOMP	$\boldsymbol{y}_t$	$=$	$\boldsymbol{\Lambda}_t\boldsymbol{\eta}_t + \boldsymbol{K}_t\boldsymbol{u}_t + \boldsymbol{\varepsilon}_t$
West & Harrison	$\boldsymbol{a}_t$	$=$	$\boldsymbol{F}_t\boldsymbol{\theta}_t + \boldsymbol{v}_t$
Kalman	$\boldsymbol{z}_t$	$=$	$\boldsymbol{H}_t\boldsymbol{x}_t + \boldsymbol{v}_t$
Åstrom & Murray	$\boldsymbol{y}_t$	$=$	$\boldsymbol{C}_t\boldsymbol{x}_t + \boldsymbol{D}_t\boldsymbol{u}_t + \boldsymbol{w}_t$
OpenMx	$\boldsymbol{y}_t$	$=$	$\boldsymbol{C}_t\boldsymbol{x}_t + \boldsymbol{D}_t\boldsymbol{u}_t + \boldsymbol{r}_t$

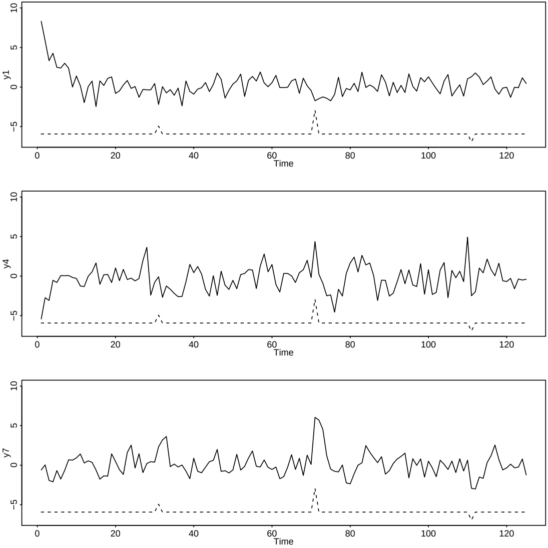
Structural Equation Models



State Space Models




x1
10
5
0
−5
0
20
40
60
80
100
120
Time
x2
10
5
0
−5
0
20
40
60
80
100
120
Time
x3
10
5
0
−5
0
20
40
60
80
100
120
Time


y1
y4
y7
Time
Time
Time
10
5
0
−5
0
20
40
60
80
100
120

Structural Equation Modeling: A Multidisciplinary Journal, 00: 1–18, 2017

ISSN: 1070-5511 print / 1532-8007 online
DOI: https://doi.org/10.1080/10705511.2017.1369354

Routledge
Taylor & Francis Group

TEACHER'S CORNER

State Space Modeling in an Open Source, Modular, Structural Equation Modeling Environment

Michael D. Hunter
University of Oklahoma Health Sciences Center

Georgia Tech

Damped and Forced Harmonic Oscillator

- As a second-order system

$$\frac{d^2x}{dt^2} = -kx - c\frac{dx}{dt} + \zeta \tag{6}$$

- As a vector of first-order systems

$$\begin{pmatrix} \frac{dx}{dt} \\ \frac{d^2x}{dt^2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -k & -c \end{pmatrix} \begin{pmatrix} x \\ \frac{dx}{dt} \end{pmatrix} + \begin{pmatrix} 0 \\ \zeta \end{pmatrix} \tag{7}$$

- The Measurement Model

$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ \frac{dx}{dt} \end{pmatrix} + \epsilon \tag{8}$$

Example 1a

dynr

See `LinearSDE.R` for dynr examples of

1. Model specification of a damped and forced harmonic oscillator
2. Bound setting
3. Model reporting with `printex`, `plotFormula`, `plot`, and `autoplot`
4. Kalman scores

Example 1b

OpenMx

See `StateSpaceContinuous.R` for OpenMx examples of

1. Undamped linear oscillator
2. Damped linear oscillator
3. Process factor analysis

Example 2

dynr

See `PFA.R` for dynr examples of

1. Process factor analysis
2. `prep.loadings()` function for easier factor loadings specification

Example 3

OpenMx

See `ModificationIndexCheck.R` for OpenMx examples of

1. Using modification indices with state space models
2. Model modification "on the fly"
3. Data generation from a model

Questions?

Thank You

`mhunter43@gatech.edu`