Dynamically modeling collaborative learning processes

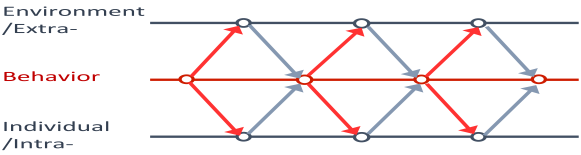
Lu Ou

ACTNext

Feb 24, 2018

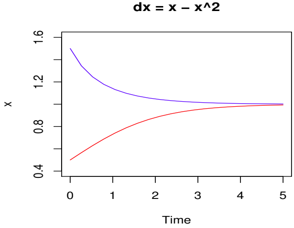
Overview

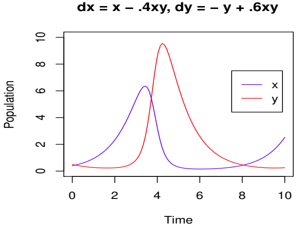
Integrated Model of Development (Gariépy 1996)



- Adaptation: goal-directed functional alignment btw extra- and intra- organimistic conditions.
- Optimization: any self-organizing system that is at equilibrium with its environment must minimize its free energy (free-energy principle; Friston 2010)
- Behavior as the leading edge of adaptation.
- Changes in adapted states.

Mathematical Representation





- Adapted state (e.g., skill level): equilibrium
- Behavior: interaction/multiplication (e.g., with a problem, a student/teacher)

Judy and Nick

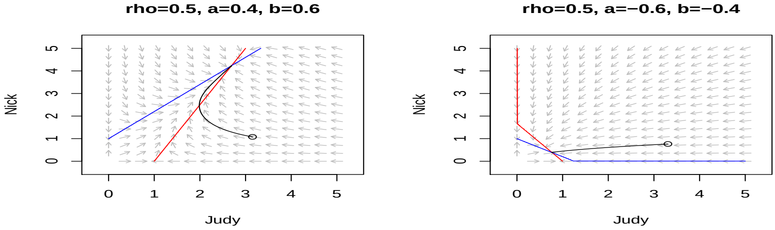
Collaboration and Competition

$$\begin{bmatrix} \frac{d\,Judy(t)}{d\,t} \\ \frac{d\,Nick(t)}{d\,t} \end{bmatrix} = \begin{bmatrix} Judy(t) - Judy(t)^2 + a Judy(t) Nick(t) \\ \rho(Nick(t) - Nick(t)^2) + b Judy(t) Nick(t) \end{bmatrix}$$

$\rho > 0$: Frequency paramters

$a = b = 0$: No interaction; Self-regulation

$a, b > 0$: Collaboration; $a, b < 0$: Competition



Extensions to multi-dimensional models

- N-subject Lotka-Volterra equations ($i, j = 1, 2, \cdots, N$)

$$\frac{dx_i}{dt} = F_i(x_1, x_2, \cdots, x_N) \triangleq \rho_i x_i (1 - \frac{x_i + \sum_{i \neq j} a_{ij} x_j}{K_i}),$$

  - ρ_i: growth parameters that determine the steepness of the logistic growth functions
  - K_i: the carrying capacity parameters that represent the limited resources
  - a_{ij} are relation parameters that specify the interaction effects between each pair of x_i
  - The system is *competitive* if $\frac{\partial F_i}{\partial x_j} \leq 0$ for $i \neq j$; and is *cooperative* if $\frac{\partial F_i}{\partial x_j} \geq 0$ for $i \neq j$.
- 3-person groups are necessary and sufficient to perform better than the best individuals on highly intellective problems (Laughlin et al. 2006).

Markov-switching
Dynamic models

Dynamic Systems approach

- ***Definition***
  - systems of elements that change over time (Thelen and Smith 1994)
  - a class of mathematical equations that describe time-based systems with particular properties
- ***Discrete-time Model***
  - time: integers $1, 2, 3, \cdots$
  - change as *differences*
- ***Continuous-time Models***
  - time: real numbers
  - change as *rate of change*
  - limit of discrete-time models

Modeling Framework

- Dynamic Model
  - Discrete-time State-Space models
    $\boldsymbol{x}_i(t_{i,j}) = F_{\boldsymbol{\theta}_{f,i}}(\boldsymbol{x}_i(t_{i,j-1})) + \zeta_i(t_{i,j}), \zeta_i(t_{i,j}) \sim N(\mathbf{0}, \Sigma_\zeta)$
  - Continuous-time Ordinary and Stochastic Differential Equation (ODE & SDE)
    $d\boldsymbol{x}_i(t) = F_{\boldsymbol{\theta}_{f,i}}(\boldsymbol{x}_i(t), t)dt + G_{\boldsymbol{\theta}_{f,i}}(t)d\boldsymbol{w}(t)$
- Measurement Model
  $\boldsymbol{y}_i(t_{i,j}) = \boldsymbol{\mu} + \boldsymbol{\Lambda}\boldsymbol{x}_i(t_{i,j}) + \epsilon_i(t_{i,j}), \epsilon_i(t_{i,j}) \sim N(\mathbf{0}, \Sigma_y)$
- Initial condition $\boldsymbol{x}_1(t_{1,j}) \sim N(\boldsymbol{x}_0, \boldsymbol{P}_0)$

What are regime-switching dynamic models?

- A regime–switching longitudinal model consists of several latent (unobserved) classes–or "regimes." Within each class, a submodel is used to described the distinct change patterns associated with the class.
- Each "regime" can be thought of as one of the stages or phases of a dynamic process.
- Individuals can switch between classes or regimes over time.
- The changes that unfold within a regime are continuous in nature.

Initial regime probabilities
Transition probabilities

Multinomial logistic regression models are used to represent the initial regime probabilities and describe each individual *i*'s transition in class membership from time *t*-1 to time *t* as

$$\Pr(S_{i1} = k | \boldsymbol{x}_{i1}, \theta) \triangleq \pi_{k,i1} = \frac{\exp(a_{k1} + \boldsymbol{b}'_{k1}\boldsymbol{x}_{i1})}{\sum_{s_1=1}^{K} \exp(a_{s1} + \boldsymbol{b}'_{s1}\boldsymbol{x}_{i1})},$$

$$\Pr(S_{it} = k | S_{i,t-1} = j, \boldsymbol{x}_{it}, \theta) \triangleq \pi_{jk,it} = \frac{\exp(a_{kt} + \boldsymbol{b}'_{kt}\boldsymbol{x}_{it})}{\sum_{st=1}^{K} \exp(a_{st} + \boldsymbol{b}'_{st}\boldsymbol{x}_{it})}$$

S_{it} = individual *i*'s class membership at time *t*

K = the number of regimes

a_{kt} = the logit intercept for the *k*th regime at time t

$\boldsymbol{x}_{it}$ = a vector of covariates for person *i* at time t

$\boldsymbol{b}_{kt}$ = a vector of logit slopes for the *k*th regime at time t

Estimation:

The dynr R package

(Ou, Hunter, and Chow 2017)

What can dynr do? Dynr Facts 1-5

1. Dynr fits discrete- and continuous-time dynamic models to multivariate longitudinal/time-series data.
2. Dynr handles linear and nonlinear dynamic models with an easy-to-use interface.
   - Dynr allows model specification in matrix and formula forms.
   - Dynr allows automatic differentiation.
3. Dynr deals with dynamic models with regime-switching properties.
   - Caveat: Only linear measurement
4. Dynr computes in C and runs fast.
5. Dynr provides ready-to-present results through LaTex equations and plots.

Models, Algorithms, and Software for the Framework in **dynr**

		Discrete-Time	Continuous-Time
Single-Regime	linear	Linear State-Space model KF **dynr**, **OpenMx**, **pomp**, **KFAS**, **dlm**, **dse**, **MKFM6**, **SsfPack**, MATLAB	Linear SDE/ODE CDEKF **dynr**, **pomp**, **OpenMx**, **ctsem**, MATLAB
	nonlinear	Nonlinear State-Space model EKF **dynr**, **pomp**, **SsfPack**, MATLAB	Nonlinear SDE/ODE CDEKF **dynr**, **pomp**, MATLAB
Multiple-Regime	linear	RS State-Space model Kim filter **dynr**, GAUSS code, MATLAB	RS SDE/ODE CD Kim filter **dynr** only
	nonlinear	RS Nonlinear State-Space model Extended Kim filter **dynr** only	RS Nonlinear SDE/ODE CD extended Kim filter **dynr** only

SDE = Stochastic Differential Equation, ODE = Ordinary Differential Equation, CD = Continuous-Discrete, RS = Regime-Switching, KF = Kalman filter (Kalman 1960), EKF = Extended Kalman filter (Anderson and Moore 1979; Bar-Shalom, Li, and Kirubarajan 2001), Kim filter = KF + Hamilton filter + Collapse procedure (Kim and Nelson 1999). Extended Kim filter was proposed by (Chow and Zhang 2013); the CD extended Kim filter is proposed by (Chow, Ou, et al. 2018).

Extended Kalman Filter

1. Time update (Prediction)
2. Measurement Update (Correction)

$$
\begin{aligned}
\boldsymbol{v}_k &= \boldsymbol{y}_k - \Lambda \hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{R}_{e,k} = \Sigma_\epsilon + \Lambda \hat{\boldsymbol{P}}_{k|k-1} \Lambda^T \\
\boldsymbol{K}_k &= \hat{\boldsymbol{P}}_{k|k-1} \Lambda^T \boldsymbol{R}_{e,k}^{-1} \\
\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{v}_k, \hat{\boldsymbol{P}}_{k|k} = \hat{\boldsymbol{P}}_{k|k-1} - \boldsymbol{K}_k \Lambda \hat{\boldsymbol{P}}_{k|k-1}
\end{aligned}
$$

3. Optimization
   $\log L(\theta) = \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{T} (-p_{i,k} \log(2\pi) - \log |\boldsymbol{R}_{e,k}| - \boldsymbol{v}'_{i,k} \boldsymbol{R}_{e,k}^{-1} \boldsymbol{v}_{i,k})$

Extended Kalman Filter

- Prediction (Discrete-time)

$$\begin{aligned} \hat{\boldsymbol{x}}_{t|t-1} &= F(t, \hat{\boldsymbol{x}}_{t-1|t-1}) \\ P_{t|t-1} &= \frac{\partial F(t, \hat{\boldsymbol{x}}(t))}{\partial \hat{\boldsymbol{x}}} P_{t-1|t-1} \frac{\partial F(t, \hat{\boldsymbol{x}}(t))}{\partial \hat{\boldsymbol{x}}}^{\mathsf{T}} + Q \end{aligned}$$

- Prediction (Continuous-time)

$$\frac{d\hat{\boldsymbol{x}}}{dt} = F(t, \hat{\boldsymbol{x}}_{t-1|t-1})$$

$$D\boldsymbol{P}(t) = \frac{\partial F(t, \hat{\boldsymbol{x}}(t))}{\partial \hat{\boldsymbol{x}}} \boldsymbol{P}(t) + \boldsymbol{P}(t)(\frac{\partial F(t, \hat{\boldsymbol{x}}(t))}{\partial \hat{\boldsymbol{x}}})^T + \boldsymbol{Q}(t)$$

- Solve these differential equations using Runge-Kutta ODE solver.

Example scripts

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()`

Collaboration and Competition

Example script

```
require(dynr)
data <- dynr.data(data.simulate,
  id="id", time="time",
  observed=c("x","y"), covariates = "cond")
meas <- prep.measurement(
  values.load=diag(1, 2),
  obs.names = c('x', 'y'),
  state.names=c('Judy', 'Nick'))
```

Collaboration and Competition

Example script

```
formula=list(
  list(Judy~ Judy-Judy^2-a*Judy*Nick,
       Nick~ rho*(Nick-Nick^2)-b*Judy*Nick),
  list(Judy~ Judy-Judy^2+c*Judy*Nick,
       Nick~ rho*(Nick-Nick^2)+d*Judy*Nick))
dynm<-prep.formulaDynamics(
  formula=formula,
  startval=c(rho = 0.5,
    a=0.5, b=0.5, c=0.5, d=0.5),
  isContinuousTime=TRUE)
```

Collaboration and Competition

Example script

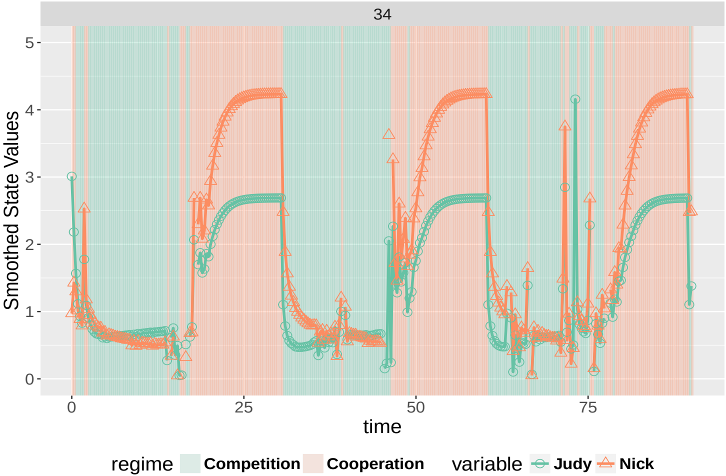
```
model <- dynr.model(
  dynamics=dynm, measurement=meas,
  noise=mdcov, initial=initial ,
  regimes=regimes , transform=trans ,
  data=data ,
  outfile="CompeteCollaborate.c")
```

Collaboration and Competition

Example script

```
# Estimate free parameters
res <- dynr.cook(model)
# Examine results
summary(res)
plotFormula(model,
  ParameterAs=res@transformed.parameters)
plot(res, dynrModel = model, style=1)
```

Results from dynr



Results from dynr

Dynamic Model

Regime 1:

$$d(\mathrm{Judy}(t)) = (\mathrm{Judy}(t) - \mathrm{Judy}(t)^2 - 0.56 \times \mathrm{Judy}(t) \times \mathrm{Nick}(t))dt$$

$$d(\mathrm{Nick}(t)) = (0.5 \times (\mathrm{Nick}(t) - \mathrm{Nick}(t)^2) - 0.37 \times \mathrm{Judy}(t) \times \mathrm{Nick}(t))dt$$

Regime 2:

$$d(\mathrm{Judy}(t)) = (\mathrm{Judy}(t) - \mathrm{Judy}(t)^2 + 0.4 \times \mathrm{Judy}(t) \times \mathrm{Nick}(t))dt$$

$$d(\mathrm{Nick}(t)) = (0.5 \times (\mathrm{Nick}(t) - \mathrm{Nick}(t)^2) + 0.6 \times \mathrm{Judy}(t) \times \mathrm{Nick}(t))dt$$

Measurement Model

$$x = \mathrm{Judy} + \varepsilon_1$$

$$y = \mathrm{Nick} + \varepsilon_2$$

Discussion

1. Modeling the start and end of the collaborative learning process
2. Reconsider the current modeling framework: relations between measurement and latent states
3. Implications for adaptive learning:
   3.1 Test to learn:
   test as intervention and source of motivation
   3.2 Continuous tutoring and monitoring
4. Application to real data.

Thank you for your attention!

Acknowledgements

- Gunter Maris, Vanessa Simmering, Benjamin Deonovic, Alina von Davier
- Sy-Miin Chow, Michael Hunter, Jiangang Hao, Zhaohua Lu, Timothy Brick, Meng Chen, Linying Ji, Sukruth Reddy, and etc.
- The ACTNext group
- The QuantDev Group at Penn State

References

- B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, 1979.
- Y. Bar-Shalom, X. R. Li, and T Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. New York, NY: John Wiley & Sons, 2001.
- Sy-Miin Chow, Lu Ou, et al. "Representing Sudden Shifts in Intensive Dyadic Interaction Data Using Differential Equation Models with Regime Switching". In: *Psychometrika* (2018).
- Sy-Miin Chow and Guangjian Zhang. "Nonlinear Regime-Switching State-Space (RSSS) Models". In: *Psychometrika* 78.4 (2013), pp. 740–768.
- Karl Friston. "The free-energy principle: A unified brain theory?". In: *Nature Reviews* 11 (2010), pp. 127–138.
- Jean-Louis Gariépy. "The Question of Continuity and Change in Development". In: *Developmental Science*. Ed. by Robert B Cairns, Glen H Elder, and Elizabeth Jane Costello. New York: Cambridge University Press, 1996. Chap. 5, pp. 78–96.

References (cont.)

- Rudolph Emil Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Journal of Basic Engineering* 82.1 (1960), pp. 35–45.
- C-J Kim and Charles R. Nelson. *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*. Cambridge, MA: MIT Press, 1999.
- Patrick R. Laughlin et al. “Groups Perform Better Than the Best Individuals on Letters-to-Numbers Problems: Effects of Group Size”. In: *Journal of Personality and Social Psychology* 90.4 (2006), pp. 644–651.
- Lu Ou, Michael D. Hunter, and Sy-Miin Chow. ***dynr**: Dynamic Modeling in R*. R package version 0.1.11-5. 2017.
- Esther Thelen and Linda B Smith. *A Dynamic Systems Approach to the Development of Cognition and Action*. Cambridge, MA: MIT Press, 1994.