Tutorial on Fitting Regime-Switching ODE using *dynr*

Sy-Miin Chow

October 11, 2015

Model Descriptions

This example illustrates how to use functions in the dynr package to fit a regime-switching linear ODE model of the form:

$$\frac{d\eta_1(t)}{dt} = -r_1\eta_1(t) + a_{12}[\eta_2(t) - \eta_1(t)]$$

$$\frac{d\eta_2(t)}{dt} = -r_2[\eta_2(t) - \eta_{20}] + a_{21}[\eta_2(t) - \eta_1(t)]$$

- if $S_t = 1$: r_1 and r_2 are freely estimated; $a_{12} = a_{21} = 0$;
- if $S_t = 2$: $r_1 = r_2 = 0$; a_{12} and $a_{21} =$ are freely estimated.
- The transition between regimes is governed by a transition matrix with elements: $\begin{bmatrix} p_{11} & 1 - p_{11} \\ p_{21} & 1 - p_{21} \end{bmatrix}$, where p_{11} and p_{21} are functions of the covariates $x_{1,it}$ and $x_{2,it}$ as:

$$p11 = \frac{\exp(a_0 + a_1 + a_3 * x_{2,it} + a_5 * x_{1,it})}{exp(0) + exp(a_0 + a_1 + a_3 * x_{2,it} + a_5 * x_{1,it})}$$

$$p21 = \frac{\exp(a_0 + a_2 * x_{2,it} + a_4 * x_{1,it})}{exp(0) + exp(a_0 + a_2 * x_{2,it} + a_4 * x_{1,it})}.$$

Load library and format data

```
require(dynr)
```

```
## Loading required package: dynr
## Loading required package: numDeriv
## Loading required package: ggplot2
## Loading required package: reshape2
## Loading required package: plyr
```

```
thedata = read.table('/Users/symiin/Dropbox/BREKFIS/dynr/da
thedata$V6 = as.numeric(thedata$V6)
data <- dynr.data(thedata, id="V1", time="V2",observed=past
                  covariates=paste0('V', 5:6))
```

Model specification

- The parameters to be optimized are:
  $[\log(r_1), \log(r_2), \log(a_{12}), \log(a_{21}), log(\sigma_{e1}), log(\sigma_{e2}), a_0, a_1, a_2, a_3, \eta_{20}$
- We now specify the model structure and control parameters
- xstart, ub and lb contain, respectively, the initial values for the parameters, upper, and lower bounds

```
model <- dynr.model(
                num_regime=2,
                dim_latent_var=2,
                xstart=c(rep(log(.1), 4), log(10.0), log(10.0
                ub=c(rep(10, 6), rep(20, 4), 1000, 20, 20),
                lb=c(rep(-10, 6), rep(-20, 4), 0, -20, -20),
                options=list(maxtime=1*60, maxeval=20)
)
```

Running the model

```
res <- dynr.run(model, data)
summary(res)
```

Some default dynr plots (work in-progress)

```
plot(res, data=data, graphingPar=list(cex.main=1, cex.axis=

dynr.ggplot(res, data.dynr=data, states=c(1,2), names.state
```