

Dynamic Structural Equation Models

Asparouhov, Muthén and Hamaker

May 19, 2016

- Motivation
- General DSEM framework and estimation
- DIC
- Empirical example
- Simulation Studies: ARMA, DAFS, WNFS, Two-level AR, Three-level AR, TVEM
- Unevenly spaced and individual-specific times of observations

Motivation

- Merge "time series" and "structural equation" modeling concepts in a generalized modeling framework in Mplus V8
- Until recently, most dynamic structural equation models were focused on the case $N=1$, due to connection with econometrics, ARMA models and Kalman filter estimation. Most social science and biostatistics/epidemiological applications have $N > 1$. Thus time-series SEM model must be a two-level model where the cluster variable is the individual and we have a long sequence of observations for that individual.
- Two distinct sources of correlation: within individual correlations can be due to subject-specific effect (two-level modeling) or it can be due to correlation due to proximity (autocorrelation) of the observations (time series modeling). The two types of correlations are easy to parse out from the data in sufficiently long longitudinal data.

Motivation continued

- The need for this modeling arises if we want to see continuum of latent variable constructs and from the fact that we don't know how often to measure the latent construct. If we measure too often we have large autocorrelation and false independence, if we measure too infrequently we can miss important changes and portray incomplete picture
- Ignoring the autocorrelation will produce incorrect SE and hide a substantial amount of the underlying relationship in the data.
- When the autocorrelation is substantial and we don't account for it, we essentially ignore the best predictors - the values from the previous time period
- Current modeling approach to long longitudinal data using ML estimation: A. Two-level modeling (ignoring proximity correlation and assuming observations are independent conditional on the individual growth trajectory) B. Multivariate model (computationally intractable beyond time length of 100 observations)

- ML is not a feasible estimation method for two-level time-series model, and is fairly complex even for single level.
- Bayesian estimation is feasible. Adding "one additional correlation" results in changing one step in the MCMC

DSEM framework continued

- The within level model includes previous periods latent variables - modeling the correlations in consecutive periods.
- Only as predictors, not as dependent variables (if we include it as predictor we will have a repeating equation)
- Prior latent variables can affect current latent variables DAFS (direct autoregressive factor score) or current observed variables directly WNFS (white noise factor score)
- One generalized framework to include many models
- L is the lag variable: how many time period back are included in the model. $L=1$ mean that just the latent variables in the previous period need to be included.

$$Y_{1,it} = v_1 + \sum_{l=0}^L \Lambda_{1,l} \eta_{1,i,t-l} + \varepsilon_{1,it}$$

$$\eta_{1,i,t} = \alpha_1 + \sum_{l=0}^L B_{1,l} \eta_{1,i,t-l} + \Gamma_1 x_{1,it} + \xi_{1,it}$$

- Let Y_{it} be an observed vector of measurements for individual i at time t .
- Let η_{it} be a latent variable vector for individual i at time t .
- Let X_{it} be a observed vector of covariates individual i at time t .
- Similarly Y_i , X_i and η_i are individual-specific variables
- Similarly Y_t , X_t and η_t are time specific variables
- Main decomposition equation

$$Y_{it} = Y_{1,it} + Y_{2,i} + Y_{3,t}$$

- $Y_{2,i}$, $Y_{3,t}$ are the "individual" and "time" specific contribution. These are latent variables. $Y_{1,it}$ is the residual.

DSEM framework continued

- The usual structural equations at level 2 and 3.

$$Y_{2,i} = v_2 + \Lambda_2 \eta_{2,i} + \varepsilon_{2,i}$$

$$\eta_{2,i} = \alpha_2 + B_2 \eta_{2,i} + \Gamma_2 x_{2,i} + \xi_{2,i}$$

$$Y_{3,t} = v_3 + \Lambda_3 \eta_{3,t} + \varepsilon_{3,t}$$

$$\eta_{3,t} = \alpha_3 + B_3 \eta_{3,t} + \Gamma_3 x_{3,t} + \xi_{3,t}$$

- Random intercepts, loadings, slopes, variances: every within level parameter s can be random. s_1 non-random within level parameter, $s_{2,i}$ is individual random effect part of $\eta_{2,i}$ and $s_{3,t}$ is time specific random effect part of $\eta_{3,t}$

$$s = s_1 + s_{2,i} + s_{3,t}$$

- There are 3 models in the above specification
- TYPE=CROSSCLASSIFIED; CLUSTER=ID TIME. Modeling time specific effects. Requires the time scale to be comparable across individuals: time varying effect model. Currently time specific effects excludes loadings and variances.
- TYPE=TWOLEVEL; CLUSTER=ID. No time specific random effects (non-random time specific effects are ok). This will be the most common and introductory model to be used.
- TYPE=GENERAL. Single level. N=1 case. No random effects. All observations are correlated. Multivariate econometrics. Kalman filter. ARMA. It can also be used for small N , as a wide multivariate spread.

- The above model assumes normality
- Ordered polytomous and Binary dependent variables using the underlying Y^* approach
- Missing data: MAR likelihood based treatment via MCMC imputation
- The above model extends the Version 7 model time-intensive cross-classified SEM model with the time-series capabilities
- Missing data imputation using DSEM - much more realistic when there is autocorrelation in the data as the missing data will be imputed from the neighbouring observations rather than from the average for the person
- MCMC based distributions for all model parameters, latent variables, and individual-specific or time specific random effects

DSEM language: latent variable lag

- ETA&1 refers to ETA with lag 1, ETA&2 refers to ETA with lag 2, etc ...
- Lag variables ETA&i are used only as predictors - not as dependent variables. Using ETA as dependent variable suffices to specify the model.
- ETA by Y1-Y5 (& L); specifies the lag L and that ETA&1, ETA&2, ..., ETA&L will be in the model
- DAFS model: ETA by Y1-Y5 (& 1); ETA on ETA&1;
- WNFS model: ETA by Y1-Y5 (& 1); Y1-Y5 on ETA&1;

DSEM extension: observed variable lag

- For type=twolevel and type=general (single level) models observed variable lag variables can also be included in the model. Not available yet for type=cross-classified
- Language: lagvar=Y(L); specifies that $Y\&1, Y\&2, \dots, Y\&L$ can be used in the model
- It can be used with observed dependent variables Y and independent variables X
- For single level and without missing data this is just a convenience. You can as well enter the data yourself. With missing data however this is not true. Rely on Mplus full information approach for missing data even in single level DSEM.
- For twolevel $Y\&j$ is used only on the within level and it refers to $Y_{w,i,t-j}$ where $Y_{i,t} = Y_{i,b} + Y_{i,t,w}$
- $Y_{w,i,t-j}$ is a "latent covariate", it does not have an observed quantity even if there is no missing data.

- If there is no missing data one can in principle use instead of $Y_{w,i,t-j}$, the variable $Y_{i,t-j} - \overline{Y}_{i,t}$, i.e., replace the latent $Y_{i,b}$ with its estimating quantity $\overline{Y}_{i,t}$ however that leads to some bias.
- The bias is more complicated than the bias in standard multilevel models, where the bias is known precisely
- The bias is most likely not of practical significance if $T > 200$, little difference between $Y_{i,b}$ and $\overline{Y}_{i,t}$, although SE are likely to still be biased as the observed analogue does not accommodate the uncertainty in the estimate which is of the same magnitude as the SE for the model parameters.
- In the presence of missing data there is no "observed covariate approach". Just like in single level missing values on covariates implies that observations are deleted which creates bigger mess than usual for time series models.

- Deleting a row is highly consequential in time series models. For example in AR(1) if the autocorrelation between Y_t and Y_{t-1} is ρ the correlation between Y_t and Y_{t-2} is ρ^2 . Deleting a row (if Y_{t-1} is missing) changes the model for Y_t .
- Search statmodel.com for "latent covariate" for more info on that topic.
- Summary: The best approach is the full information approach developed with the lagvar=Y(L); command for both single and twolevel models.
- Currently available only for continuous variables, not categorical, and not with cross-classified models

The development of that "observed lag" algorithm and command was done AFTER the seemingly more advanced "latent variable lag" algorithm. Four reasons:

- 1. Creating a latent variable behind the observed variable requires fixing a variance to zero in Mplus and with Bayes that will lead to slow convergence in many cases. Using directly observed lag variables yields fast convergence.
- 2. In this modeling framework DIC is conditional on all within level latent variables which leads to unusable DIC of minus infinity with fake latent variables.
- 3. Unnecessarily complicated input language using fake latent variables.
- 4. Multiple papers in psychological methods that can help with early adoption.

```

title: ARMA(2,1)

montecarlo:
  names = y;
  nobs = 500;
  nreps = 100;
  lagvar=y(2);

analysis: estimator=bayes;

model montecarlo:
  f by y@1 (&1); f*0.6;
  y on f&1*0.4 y&1*0.5 y&2*0.2;
  y@0.00000001;

model:
  f by y@1 (&1); f*0.6;
  y on f&1*0.4 y&1*0.5 y&2*0.2;
  y@0.01;
    
```

$$Y_t = \mu + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \varepsilon_t + \beta_3 \varepsilon_{t-1}$$

DSEM example: single level ARMA(2,1) results

MODEL RESULTS									
		Population	ESTIMATES		S. E.	M. S. E.	95% Cover	% Sig	
			Average	Std. Dev.	Average			Coeff	
F	BY	1.000	1.0000	0.0000	0.0000	0.0000	1.000	0.000	
Y	ON								
F&1		0.400	0.3875	0.2143	0.2429	0.0456	0.960	0.293	
Y	ON								
Y&1		0.500	0.5178	0.2079	0.2373	0.0431	0.949	0.859	
Y&2		0.200	0.1765	0.1754	0.1933	0.0310	0.960	0.111	
Intercepts									
Y		0.000	-0.0016	0.0540	0.0466	0.0029	0.869	0.131	
Variances									
F		0.600	0.5809	0.0373	0.0392	0.0017	0.939	1.000	
Residual Variances									
Y		0.010	0.0100	0.0000	0.0000	0.0000	1.000	0.000	

DSEM Stationarity

- The models as described assumes that the distribution of $Y_t|X_t$ stabilizes i.e. is stationary. If the observed Y_t data does not exhibit a stationary distribution and the distribution changes with time then covariates X_t that change over time need to be included (for example including T as covariate will model a distribution that has a mean changing linearly over time)
- Alternative method for dealing with non-stationarity is to use ARIMA methodology. Instead of modeling Y_t , model $Y_t - Y_{t-1}$ or more generally $(1 - L)^d Y_t$ as a stationary process
- How to check that estimated model is stationary? MA (moving average) has no effect - always stationary. The WNFS is always stationary. AR (autoregressive) portion should have all root of the lag-equation > 1 , i.e., for AR(1) $Y_t = \beta Y_{t-1} + \varepsilon$, $|\beta| < 1$ for stationarity and it is the autocorrelation. In multivariate form the eigenvalues of $|\beta|$ must be less than 1 by absolute value and $\beta = V^{-1}C$ is not the autocorrelation coefficients and could be bigger than 1.

DSEM Stationarity continued

- How to really check if a model is stationary: generate data in Mplus and see if it explodes
- Consider the simple DAFS model

$$Y_{it} = \mu_i + \Lambda \eta_{it} + \varepsilon_{it}$$

$$\eta_{it} = \alpha + \beta \eta_{i,t-1} + \xi_{it}$$

- This model is stationary if and only if $|\beta| < 1$. If there are more than one factors and η_{it} is a vector then β is a matrix and the process is stationary if and only if the eigenvalues of the matrix are smaller than 1 by absolute value.
- Note also that in the univariate case

$$Var(\eta_{it}) = Var(\xi_{it}) / (1 - \beta^2), E(\eta_{it}) = \alpha / (1 - \beta)$$

- If $\beta \rightarrow 1$ then the mean and the variance will $\rightarrow \infty$

DSEM Estimation

- Estimating issues as an addition to the Mplus Version 7 algorithm.
- 1. [Between Parts | observed data] can't really be computed as the likelihood is messy and the observations are not independent given the between level random effects $\Rightarrow \eta$ has to become part of MCMC and the between level random effects are now [Between Parts | observed data, η]. Once this is done we can use two-level FIML formulas. An efficiency gain in mixing quality that is being lost.
- 2. $[\eta_{it} | *]$ used in the MCMC estimation now uses $2 * L + 1$ measurement equations and one regression equation and depends on

$$Y_{t,i}, Y_{t+1,i}, \dots, Y_{t+L,i},$$

$$X_{t,i}, X_{t+1,i}, \dots, X_{t+L,i},$$

$$\eta_{t-L,i}, \dots, \eta_{t+L,i}$$

- 3. Initial Conditions.
 - Now we have $\eta_{i,t=0}, \eta_{i,t=-1}, \dots$ in the model. In the case of the observed lag variables we have $Y_{i,t=0}, Y_{i,t=-1}$. We treat these as auxiliary parameters with their own prior.
 - If sequences are long such as $T > 50$ it doesn't matter. For smaller sequences it does.
 - Mplus offers 2 options
 - A. Use our default automatic priors, in the first 100 MCMC iterations we update the priors from the sample statistics of η or $Y_{i,t}$, then we discard those 100 MCMC iterations.
 - B. Specify a normal prior for these auxiliary parameters in model prior. Difficult to use in practice even for experts especially when variables are not standardized. Nevertheless you have access to it.

```
model:
  f by y@1 (&1); f*0.6;
  y on f&1*0.4 y&1*0.5 y&2*0.2;
  y@0.01;

model prior: f~N(0,0.6); y~N(0,1);
```

- Note also that η and Y with negative indices become important in simulation studies as well. By default we start at 0 then draw and discard the first 10 draws. After that the simulation distribution stabilizes and is independent of these initial values.
- In Version 8 restriction that is likely to stay: random loadings can vary across individual but not across time. Random slopes however can vary across individual and time with current implementation using type=crossclassified. Observed lag variables not available with cross-classified models.

- 4. The posterior distribution of η_{it} changes across t . It is the same in the middle of the sequence but changes at the end or the beginning as there are fewer observation (the sequence is truncated).
- 5. Missing data and observed variables lag: The posterior distribution of [missing data of observed lag variables | *] now involves multiple rows because the missing observation participates in several different times t . If the variable has lag L the size of the missing data imputation model becomes $L + 1$ times bigger.
- 6. For observed lag variables $Y_{it} = Y_{b,i} + Y_{w,it}$ the posterior distribution for $Y_{b,i}$ is now more complicated. In a simple example such as AR(1) it works like this

$$Y_{w,it} = \mu + B \cdot Y_{w,it-1} + \varepsilon_{it}$$

$$Y_{it} = \mu + (I - B) \cdot Y_{b,i} + B \cdot Y_{it-1} + \varepsilon_{it}$$

Thus the "between part" is multiplied by $(I - B)$ in this case.

- Do not be surprised or alarmed if Mplus converges in 20 seconds and WinBugs takes 24 hours.
- Here are the four main reasons:
 1. Mplus uses Fortran (fastest computational environment)
 2. Mplus uses parallel computing so each chain is computed separately (the number of chains will not increase the computing time as long as you have enough CPUs on your computer (most come with 8 - Quad Core i7 processors). Use the PROC=number_chains command
 3. Mplus uses the largest updating blocks possible - complicated to program but gives the best mixing quality
 4. Mplus uses sufficient statistics

- DIC is currently the only information criterion implemented in Mplus for comparing DSEM models.

$$D(\theta) = -2\log(p(Y|\theta))$$

$$p_D = \bar{D} - D(\bar{\theta})$$

$$DIC = D(\bar{\theta}) + 2p_D$$

- Despite the clear definition with the above formulas, there is substantial variation in what DIC actually is. The source of the variation is the definition of θ . For example in a factor analysis model, one way to compute DIC is using the observed Y implied variance covariance matrix, and never really involve the factor. You can also treat the factor as parameters. In that case likelihood = $[Y|\eta]$. Also different treatment of missing values: parameters or not a part of the likelihood.

- Different definitions of DIC are not comparable. You can compare only if they are using the same likelihood definition.
- Software implementations not clear about what definition they use, even published peer reviewed articles limited to the above definition, but no full details on what likelihood is being used
- Different definitions of DIC have different power to detect significance.
- An easy way to figure out what DIC is being computed: look at p_D - estimated number of parameters
- Among the different definitions of DIC: the smaller p_D the better. Fewer parameters to be integrated, faster convergence, more accurate estimate, more power, if p_D is large then it is not unusual to have a situation when model estimation has converged but DIC estimation has not

- Looking at p_D is surprisingly useful, you can detect estimation problems with it. If p_D is not close enough to what you expect it to be most likely you need to run many more MCMC iterations. If p_D is negative you most certainly should not consider the model as converged.
- p_D is the "estimated number of parameters" it does not match the actual number of parameters. Consider random effect with zero variance, this is equivalent to a non-random parameter. p_D has a way of telling this and will not consider these random effects to be different parameters but one parameter. Similarly highly correlated random effects will count as one not two random effects. The more correlated random effects are the more discounted they are. For example two random effects with correlation 0.75 could count as 1.5 parameters. If the correlation is 0 they will count as 2 parameters, if the correlation is 0.95 they will count as 1.2 parameters. All these numbers are approximate.

- All latent variables are conditioned on, between and within level latent variables. Likelihood is now straightforward to compute as the within cluster variables are independent. Large number of parameters.
- In some models this definition becomes useless. Consider a model like ARMA

$$Y_t = \beta_1 Y_{t-1} + \varepsilon_t + \beta_2 \varepsilon_{t-1}$$

The likelihood used by DIC is $[Y_t|Y_{t-1}, \varepsilon_t, \varepsilon_{t-1}]$ which has zero conditional variance and - infinity as likelihood. If your model requires the specification Y@0; on the within level DIC will likely be useless and you should be able to see that in the large negative DIC value. p_D still ok.

- You can use DIC to compare complex hypothesis such as a whole set of parameters is zero, or a set of random effects is zero. You can use DIC to compare the need for random loadings (fixed v.s. random loadings). You can use DIC to compare non-nested models. If testing just one parameter instead use the confidence/credibility interval.
- DIC is somewhat too sensitive in this framework. Consider this example. DSEM with 1 factor can not be compared with DSEM with 2 factors due to the fact that the factors are conditioned on and treated as parameters. Thus the DIC values for the two models are not comparable.
- General trend: DIC will loose its appeal as the answer to all questions. We use Bayes estimation to be able to estimate models that we can't compute the likelihood, i.e., where DIC is not computable in traditional sense. [Model test] will probably be available soon for Bayes.

- The example without AR is from Asparouhov and Muthén (2015), "General Random Effect Latent Variable Modeling: Random Subjects, Items, Contexts, and Parameters", in Advances in Multilevel Modeling for Educational Research. The data comes from Jahng, Wood, Trull (2008)
- 21 items measuring mood for 84 subjects, two groups according to DSM-IV classification.
- 76 to 186 observations per person all within 4 weeks: violation of equally spaced times.
- Primary interest is to determine if the group variable increases the factor variance: $\beta = \text{SIGMA ON GROUP}$
- All Version 7 features apply to DSEM: random loadings, random indicator mean, random factor mean, random factor variance

Input file for empirical example

```
model:
%within%
s1-s21 | f by jittery-scornful (& 1);
f@1; rho | f on f&1;

%between%
rho; rho on group;
f; f on group;
s1-s21 jittery-scornful;
[s1-s21*1] (m1-m21);
sigma by s1-s21*1 (m1-m21);
sigma; sigma on group;
```

Results from empirical example: borderline personality disorder data

Model	DIC	p_D	p	β
Full DSEM model	436628	13473	113	.46[.11,.92]
DSEM with non-random AR	436749	13553	111	.24[-.04,.58]
DSEM without AR	437276	14087	110	.29 [-.01,.67]
Twollevel model	460351	3366	110	.29[.04,.65]

Conclusions from empirical example

- Full DSEM model gets the best DIC.
- Dynamic modeling affects point estimates, standard errors, significance
- The Full DSEM model separates mood variation as momentary and global. Fast changing mood (small ar) v.s. slow changing mood (large ar).
- Sine curve amplitude effect: Sigma on Group
- Sine curve frequency effect: Rho on Group
- DSEM-DIC not on the same scale as Twolevel-DIC

Alternative individual-specific variances for a variable.

- Usually we use this for observed variables. For latent variables the random loading common factor model is more natural.
- Method 1
 - $\text{Sigma} \mid E1 \text{ by } E; E1@1; E@0$
 - $E = \text{Sigma} * E1$, where $E1$ is standard normal
 - Sigma is between level random effect (regress on predictors)
 - $\text{Var}(E \mid ID=i) = \text{Sigma}_i^2$
- Method 2
 - Replace $E@0$ with $E@0.5$
 - $\text{Var}(E \mid ID=i) = 0.5 + \text{Sigma}_i^2$
 - 0.5 is the minimum variance, should be viewed as "prior min". If you get it wrong you may truncate some variances.
 - Improves convergence especially with missing data
 - Recommendation: the constant should be 50% of the non-random variance estimate

Simulation Study - Twolevel AR(1) with random intercept and autocorrelation

```

MONTECARLO: NAMES ARE y;
NOBS = 10000;
NREP = 100;
NCSIZES = 1;
CSIZES = 100(100);
lagvar= y(1);

ANALYSIS: TYPE IS TWOLEVEL RANDOM;
estimator=bayes; biter=(300); proc=2;

MODEL:

%WITHIN%
y*1;
s1 | y on y&1;

%BETWEEN%
[y*2 s1*0.2];
y*0.7 s1*0.02;
y with s1*0;
    
```

Results - Twolevel AR(1) with random intercept and autocorrelation

MODEL RESULTS							
	Population	ESTIMATES Average	Std. Dev.	S. E. Average	M. S. E.	95% Cover	% Sig Coeff
Within Level							
Residual Variances							
Y	1.000	1.0006	0.0144	0.0142	0.0002	0.940	1.000
Between Level							
Y							
S1	0.000	0.0000	0.0131	0.0160	0.0002	1.000	0.000
Means							
Y	2.000	2.0190	0.0925	0.0863	0.0088	0.940	1.000
S1	0.200	0.1987	0.0166	0.0176	0.0003	0.970	1.000
Variances							
Y	0.700	0.7225	0.0938	0.1110	0.0092	0.960	1.000
S1	0.020	0.0215	0.0047	0.0046	0.0000	0.940	1.000
CORRELATIONS AND MEAN SQUARE ERROR OF THE TRUE FACTOR VALUES AND THE FACTOR SCORES							
		CORRELATIONS		MEAN SQUARE ERROR			
		Average	Std. Dev.	Average	Std. Dev.		
S1		0.8118	0.0407	0.0831	0.006		
Y		0.9874	0.0028	0.1303	0.010		

Simulation Study - Twolevel AR(1) with random intercept, autocorrelation and variance

```

MONTECARLO: NAMES ARE y;
            NOBS = 10000;
            NREP = 100;
            NCSIZES = 1;
            CSIZES = 100(100);
            lagvar= y(1);

ANALYSIS: TYPE IS TWOLEVEL RANDOM;
estimator=bayes; biter=(300); proc=2;

MODEL:

%WITHIN%
y@1; e@1;
s1 | y on y@1;
s2 | e by y;

%BETWEEN%
[y*2 s1*0.2 s2*1];
y*0.7 s1*0.02 s2*0.01;
y with s1-s2*0; s1 with s2*0;

```

Results - Twolevel AR(1) with random intercept, autocorrelation and variance

		Population	ESTIMATES Average	Std. Dev.	S. E. Average	M. S. E.	95% Cover	% Sig Coeff
Within Level								
Variances								
E		1.000	1.0000	0.0000	0.0000	0.0000	1.000	0.000
Residual Variances								
Y		0.100	0.1000	0.0000	0.0000	0.0000	1.000	0.000
Between Level								
Y	WITH							
S1		0.000	0.0007	0.0150	0.0154	0.0002	0.960	0.040
S2		0.000	0.0011	0.0112	0.0122	0.0001	0.930	0.070
S1	WITH							
S2		0.000	-0.0005	0.0021	0.0023	0.0000	0.930	0.070
Means								
Y		2.000	2.0035	0.0762	0.0874	0.0058	0.970	1.000
S1		0.200	0.2031	0.0179	0.0172	0.0003	0.930	1.000
S2		1.000	1.0000	0.0121	0.0131	0.0001	0.960	1.000
Variances								
Y		0.700	0.7516	0.1130	0.1170	0.0153	0.920	1.000
S1		0.020	0.0184	0.0039	0.0039	0.0000	0.920	1.000
S2		0.010	0.0111	0.0022	0.0026	0.0000	0.950	1.000

Simulation Study - Twolevel DAFS Lag 3 model

```

montecarlo:
names = y1-y5;
NOBS = 10000;
NREP = 100;
NCSIZES = 1;
CSIZES = 100(100);

ANALYSIS: TYPE IS TWOLEVEL;
estimator=bayes;
biter=(500); proc=2;

model montecarlo:
%within%
f by y1-y5*1 (& 3);
y1-y5*1;
f@1;
f on f&1*0.4 f&2*0.2 f&3*0.1;

%between%
fb by y1-y5*0.5; fb@1; y1-y5*0.2;

```

Results - Twolevel DAFS Lag 3 model

F	BY							
Y1		1.000	0.9992	0.0134	0.0120	0.0002	0.900	1.000
Y2		1.000	1.0010	0.0127	0.0120	0.0002	0.950	1.000
Y3		1.000	1.0001	0.0138	0.0120	0.0002	0.910	1.000
Y4		1.000	0.9997	0.0128	0.0120	0.0002	0.930	1.000
Y5		1.000	0.9988	0.0134	0.0119	0.0002	0.910	1.000
F	ON							
F&1		0.400	0.4024	0.0121	0.0123	0.0002	0.980	1.000
F&2		0.200	0.1983	0.0143	0.0139	0.0002	0.950	1.000
F&3		0.100	0.0997	0.0138	0.0126	0.0002	0.910	1.000

Simulation Study - Twolevel WNFS Lag 5 model

```
montecarlo:
  names = y1-y5;
  NOBS = 10000;
  NREP = 100;
  NCSIZES = 1;
  CSIZES = 100(100);

ANALYSIS:  TYPE IS TWOLEVEL;
           estimator=bayes;
           biter=(500); proc=2;

model montecarlo:
  %within%
  f by y1-y5*1 (& 5);
  y1-y5*1;
  f@1;
  y1-y5 on f&1*0.4 f&2*0.2 f&3*0.3 f&4*0.2 f&5*0.1;

  %between%
  fb by y1-y5*0.5; fb@1; y1-y5*0.2;
```

Results - Twolevel WNFS Lag 5 model

Y1	ON								
F&1		0.400	0.4012	0.0155	0.0160	0.0002	0.960	1.000	
F&2		0.200	0.1989	0.0168	0.0161	0.0003	0.940	1.000	
F&3		0.300	0.3021	0.0145	0.0165	0.0002	0.980	1.000	
F&4		0.200	0.1998	0.0179	0.0167	0.0003	0.950	1.000	
F&5		0.100	0.0985	0.0155	0.0164	0.0002	0.960	1.000	

Simulation Study - Twolevel DAFS-WNFS Combo Lag 1 model - ARMA(1,1) factor

```
montecarlo:
  names = y1-y5;
  NOBS = 10000;
  NREP = 100;
  NCSIZES = 1;
  CSIZES = 100(100);

ANALYSIS:  TYPE IS TWOLEVEL;
           estimator=bayes;
           biter=(500); proc=2;

model montecarlo:
  %within%
  f by y1-y5*1 (& 1);
  y1-y5*1;
  f@1;
  f on f&1*0.4;
  y1-y5 on f&1*0.6;

  %between%
  fb by y1-y5*0.5; fb@1; y1-y5*0.2;
```

Results - Twolevel DAFS-WNFS Combo Lag 1 model - ARMA(1,1) factor

F	ON								
F&1		0.400	0.3992	0.0154	0.0139	0.0002	0.930	1.000	
Y1	ON								
F&1		0.600	0.5980	0.0216	0.0218	0.0005	0.970	1.000	
Y2	ON								
F&1		0.600	0.6020	0.0207	0.0220	0.0004	0.990	1.000	
Y3	ON								
F&1		0.600	0.5999	0.0208	0.0219	0.0004	0.980	1.000	
Y4	ON								
F&1		0.600	0.6006	0.0215	0.0219	0.0005	0.940	1.000	
Y5	ON								
F&1		0.600	0.5985	0.0231	0.0219	0.0005	0.920	1.000	

Note that this is counterintuitive from SEM perspective, but not from time series perspective. The model is essentially a factor analysis model with ARMA(1,1) factor

Summary of literature: my favorite (read in this order)

- Time series, ARMA literature, econometrics textbooks. W.H. Green, "Econometric Analysis"
- Comparisons of Four Methods for Estimating a Dynamic Factor Model (2008), by Zhang Hamaker Nesselroade
- Bayesian Estimation of Categorical Dynamic Factor Models (2007), by Zhang and Nesselroade
- On the Nature of SEM Estimates of ARMA Parameters (2002) by Hamaker, Dolan, Molenaar
- A Multilevel AR(1) Model: Allowing for Inter-Individual Differences in Trait-Scores, Inertia, and Innovation Variance (2015) by Jongerling, Laurenceau, Hamaker
- Conditions for the Equivalence of the Autoregressive Latent Trajectory Model and a Latent Growth Curve Model With Autoregressive Disturbances (2005) by Hamaker
- A comparison of Inverse-Wishart prior specifications for covariance matrices in multilevel autoregressive models (2016) by Schuurman, Grasman, Hamaker

Subject-specific times of observations

- The basic model assumes that observations are taken at equally spaced time.
- If times are subject-specific we slice the time grid in sufficiently refined grid and enter missing data for the times where observation is not taken.
- For example if several observations are taken during the day, and at different times for each individual, we slice the day in 24 hour periods and place the corresponding observations in the hour slots.
- Data from the next simulation looks like this for day 1 for individual 1 and 2.

Subject-specific times of observations: subject 1 day 1

y1	y2	y3	y4	y5	T	ID
999	999	999	999	999	1	1
999	999	999	999	999	2	1
999	999	999	999	999	3	1
999	999	999	999	999	4	1
999	999	999	999	999	5	1
999	999	999	999	999	6	1
999	999	999	999	999	7	1
999	999	999	999	999	8	1
999	999	999	999	999	9	1
999	999	999	999	999	10	1
999	999	999	999	999	11	1
999	999	999	999	999	12	1
5.026193	0.327383	1.017519	0.701296	-0.55917	13	1
999	999	999	999	999	14	1
1.628885	1.652829	2.324074	1.800932	4.013447	15	1
999	999	999	999	999	16	1
4.376545	1.652831	2.098822	6.188234	2.913506	17	1
1.534865	0.631455	-0.29779	2.798775	1.37025	18	1
0.359654	1.476764	-0.43374	0.348777	1.382437	19	1
999	999	999	999	999	20	1
999	999	999	999	999	21	1
999	999	999	999	999	22	1
999	999	999	999	999	23	1
999	999	999	999	999	24	1

Subject-specific times of observations: subject 2 day 1

y1	y2	y3	y4	y5	T	ID
999	999	999	999	999	1	2
-4.10343	-2.63665	-0.71492	-1.18325	-2.38467	2	2
999	999	999	999	999	3	2
-1.49744	0.268575	-1.94601	-0.70893	-2.5558	4	2
999	999	999	999	999	5	2
999	999	999	999	999	6	2
999	999	999	999	999	7	2
999	999	999	999	999	8	2
0.158731	-0.49435	2.052861	2.387179	0.932945	9	2
999	999	999	999	999	10	2
999	999	999	999	999	11	2
999	999	999	999	999	12	2
999	999	999	999	999	13	2
999	999	999	999	999	14	2
999	999	999	999	999	15	2
999	999	999	999	999	16	2
999	999	999	999	999	17	2
999	999	999	999	999	18	2
-0.2193	-2.14029	0.050116	-0.42015	-1.00061	19	2
999	999	999	999	999	20	2
999	999	999	999	999	21	2
999	999	999	999	999	22	2
999	999	999	999	999	23	2
999	999	999	999	999	24	2

Subject-specific times of observations - simulation study

```
montecarlo:
  names = y1-y5 u;
  NOBS = 30000;
  NREP = 100;
  NCSIZES = 1;
  CSIZES = 100(300);
  categorical=u;
  generate=u(1);
  within=u;
  missing=y1-y5;

model missing: [y1-y5@-15]; y1-y5 on u@30;

ANALYSIS: TYPE IS TWOLEVEL; estimator=bayes;
          biter=10000(500); proc=2;

model montecarlo:
  %within%
  [u$1*-0.83];
  f by y1-y5*1 (& 1);
  y1-y5*1;
  f@1;
  f on f&1*0.4;

  %between%
  fb by y1-y5*0.5; fb@1; y1-y5*0.2;
```

Subject-specific times of observations - simulation study results

- 80% missing values, 20% present, in hourly scale that means 4 or 5 observations a day
- 99% convergence rate, 96 minutes computation for 100 replications, 1 min per replication

		Population	ESTIMATES Average	Std. Dev.	S. E. Average	M. S. E.	95% Cover	% Sig Coeff
Within Level								
F	BY							
	Y1	1.000	1.0015	0.0166	0.0180	0.0003	0.990	1.000
	Y2	1.000	1.0017	0.0172	0.0178	0.0003	0.949	1.000
	Y3	1.000	1.0001	0.0183	0.0179	0.0003	0.960	1.000
	Y4	1.000	1.0034	0.0157	0.0180	0.0003	0.970	1.000
	Y5	1.000	1.0010	0.0192	0.0180	0.0004	0.939	1.000
F	ON							
	F&1	0.400	0.3981	0.0234	0.0211	0.0005	0.889	1.000

Subject-specific times of observations - summary of results

- Similar results for 85% missing and 90% missing, however, convergence is visibly slower for 90% missing values than for 80% missing values and non-convergence is more likely
- It appears that optimal time discretization would be about 80% to 85% missing values inserted to represent the distance between the observations
- Very likely any information contained in the unequal distances in the observations would be extracted well using the 80% to 85% missing values.
- It appears that this problem is solved with this simple setup and there is no need to develop Brownian motion continuous modeling.
- Mplus now implements a utility that will setup the missing data for you, given the precise times of observations

Subject-specific times of observations algorithm

- $tvarinterval = t(0.1)$;
- Split the time axis in bins of size 0.1. Then place each observation in the correct bin
- Repeat these steps until each bin contains no more than 1 observation
 - find a bin with more than 1 observations
 - locate the nearest empty bin (look up or down)
 - move one of the extra observation to fill in the the empty bin but keep order of the observations so the extra observation bumps the rest of the observations towards the empty bin
- Fill in the remaining bins with missing values and set the time as $T=1,2, \dots$ and T is the bin number.
- Other algorithms are possible. The algorithm discretizes the time scale and approximates the times of observations with times from the discretized grid.

Empirical Example: Jahng, Wood, Trull (2008)

```
variable:
  names=user t y1-y21 group;
  usevar=y1-y21 group;
  between=group;
  cluster=user;
  tvarinterval = t(0.1);

data: file is Seuming_PM2008_addtimestamp2.dat;

ANALYSIS: TYPE = twolevel random; estimator=bayes; proc=2;

model:
  %within%
  f1 by y1 (&1)
    y2-y21 (a1-a20);
  f1 on f1&1;

  %between%
  f2 by y1
    y2-y21*1 (a1-a20);
  f2 on group;
```

Empirical Example Results: autocorrelation coefficient for the factor. Continuous time modeling

Grid size	ρ	implied	convergence in seconds
0.05	0.788	-	1139
0.1	0.673	0.621	97
0.15	0.640	0.489	97
ignore time	0.598	-	78

- Implied values are from the most refined model 0.788^2 and 0.788^3
- Some gain in precision going from 0.1 to 0.05, big gain from going from 0.15 to 0.05
- All other parameters in the model remain unchanged
- The smaller the grid size the better the precision but also the slower the convergence

Simulated Example Results. Continuous time modeling

- Using one generated data set from the 80% missing simulation. Consider the results for the autocorrelation parameter under various analysis. True value=0.4. Note that AR meaning depends on the time interval between observations. Note that the true grid value that we know is sufficient is 1. Note that the model is almost the same as the empirical example.

Grid size	ρ	implied	$ \rho - \text{implied} $
0.5	0.633	-	-
1	0.398	0.401	0.003
1.5	0.303	0.254	0.049
2	0.229	0.161	0.068
ignore time	0.102	-	-

- You can see in this controlled experiment that "implied - estimated" quantity clearly discovers the correct grid size

Why do we need lag modeling for the latent variables?

- What happens if we just model the lag of the observed sum score? Generate large data set 2000 individuals with 50 time points and 3 indicators: Two-level DAFS

```
montecarlo:
  names = y1-y3;
  NOBS = 100000;
  NREP = 1;
  NCSIZES = 1;
  CSIZES = 2000(50);
  save=OL.dat;

ANALYSIS: TYPE IS TWOLEVEL;
  estimator=bayes;
  biter=(2000); proc=2;

model montecarlo:
  %within%
  f by y1-y3*1 (&1);
  y1-y3*1;
  f@1;
  f on f&1*0.5;

  %between%
  fb by y1-y3*0.5; fb@1; y1-y3*0.2;
```


Two-level AR(1): Observed and Latent

- Two-level Latent AR(1)

$$Y_{it} = \mu + Y_i + F_{it}$$

$$F_{it} = \rho F_{i,t-1} + \varepsilon_{it}$$

- Two-level Observed AR(1)

$$Y_{it} = \mu + Y_i + F_{it}$$

$$Z_{it} = Y_{it} - \bar{Y}_{it}$$

$$F_{it} = \rho Z_{i,t-1} + \varepsilon_{it}$$

Two-level Latent AR(1) for the sum score

```
variable:
  names = y1-y3 C;
  usevar=y;
  lagvar=y(1);
  cluster=c;

define: y=y1+y2+y3;

data: file=OL.dat;

ANALYSIS: TYPE IS TWOLEVEL;
          estimator=bayes;
          biter=(2000); proc=2;

model:
  %within%
  y on y&1*0.5;
  %between%
  y;

savedata: file is OL1.dat;
```

Two-level Observed AR(1) for the sum score

```
variable:
  names = y z C;
  usevar=y z;
  cluster=c;
  within=z;

define: center z (groupmean);

data: file=OL1.dat;

ANALYSIS: TYPE IS TWOLEVEL;
          estimator=bayes;
          biter=(2000); proc=2;

model:
  %within%
  y on z*0.5;
  %between%
  y;
```

Why do we need lag modeling for the latent variables? Results

- Autocorrelation ρ true value = 0.5

Model	ρ	bias
Two-level DAFS	0.492	0.008
Two-level Latent AR(1)	0.391	0.109
Two-level Observed AR(1)	0.360	0.140

- Two-level Latent AR(1) for the sum score bias due to not accounting for measurement error of the factor
- Two-level Observed AR(1) for the sum score yields further bias due to not accounting for measurement error in the lag variable

Three-level AR(1) model

- Y_{idt} is the observed value for individual i on day d at time t

$$Y_{idt} = \mu + Y_i + E_{it} + F_{id} + G_{idt}$$

$$G_{idt} = \rho_1 G_{id,t-1} + \varepsilon_{1,idt}$$

$$F_{id} = \rho_2 F_{i,d-1} + \varepsilon_{2,id}$$

- Two type of autocorrelation parameter, ρ_1 is the autocorrelation within the day, ρ_2 is the autocorrelation between the days
- Maybe take out E_{it} ?
- Model has 7 parameters: 4 variances, 2 autocorrelations, 1 intercept
- Data consists of 100 individuals, observed for 100 days, with 10 observations per day

Three-level AR(1) model - simulation study

```
montecarlo:
  names = y1-y10;
  NOBS = 10000;
  NREP = 100;
  NCSIZES = 1;
  CSIZES = 100(100);

ANALYSIS: TYPE IS TWOLEVEL;
estimator=bayes; biter=(500); proc=2;

model montecarlo:
  %within%
  f by y1-y10@1 (& 1);
  y1-y10@0.01;
  f1 by y1@1; f2 by y2@1; f3 by y3@1;
  f4 by y4@1; f5 by y5@1; f6 by y6@1;
  f7 by y7@1; f8 by y8@1; f9 by y9@1;
  f10 by y10@1;
  f1-f10*1 (1);
  f*0.5;
  f on f&1*0.3;
  f2-f10 pon f1-f9*0.5 (2);
  f with f10@0;

  %between%
  fb by y1-y10@1; fb*0.4; y1-y10*0.1;
  [v1-v10*0] (3);
```

Three-level AR(1) model - simulation study results

		Population	ESTIMATES		S. E.	M. S. E.	95% Cover	% Sig
Within Level			Average	Std. Dev.	Average			Coeff
F	ON							
F&1		0.300	0.2955	0.0133	0.0150	0.0002	0.970	1.000
F2	ON							
F1		0.500	0.4993	0.0042	0.0040	0.0000	0.940	1.000
Variances								
F1		1.000	1.0007	0.0049	0.0049	0.0000	0.930	1.000
Residual Variances								
F		0.500	0.4994	0.0150	0.0126	0.0002	0.880	1.000
Between Level								
Intercepts								
Y1		0.000	0.0037	0.0855	0.0572	0.0072	0.810	0.190
Variances								
FB		0.400	0.4310	0.0570	0.0658	0.0042	0.970	1.000
Residual Variances								
Y1		0.100	0.1007	0.0058	0.0051	0.0000	0.940	1.000

Three-level AR(1) model with subject-specific times of observations

- Using 50% missing data. Approximately 5 randomly spaced times of observations per day
- 5 observations a bit too low to obtain good autocorrelation parameter. Sequence is too short? Mixing estimation?
- Add the commands:
missing=y1-y10;
model missing: [y1-y10*0];

Three-level AR(1) model with subject-specific times of observations - simulation results

		Population	ESTIMATES Average	Std. Dev.	S. E. Average	M. S. E.	95% Cover	% Sig Coeff
Within Level								
F	ON							
F&1		0.300	0.2864	0.0210	0.0166	0.0006	0.810	1.000
F2	ON							
F1		0.500	0.4428	0.0540	0.0188	0.0062	0.360	1.000
Variances								
F1		1.000	1.0444	0.0429	0.0159	0.0038	0.450	1.000
Residual Variances								
F		0.500	0.4694	0.0307	0.0189	0.0019	0.560	1.000
Between Level								
Intercepts								
Y1		0.000	0.0097	0.0665	0.0589	0.0045	0.890	0.110
Variances								
FB		0.400	0.3821	0.0640	0.0608	0.0044	0.940	1.000
Residual Variances								
Y1		0.100	0.0925	0.0098	0.0057	0.0002	0.660	1.000

Asparouhov, Muthén and Hamaker

Muthén & Muthén 65/70

Two-level AR(1) TVEM - simulated example

- General TVEM framework: multivariate, multilevel, time-series, continuous and categorical dependent variables.
- Consider the following example N=500, T=50

$$Y_{it} = \mu_t + Y_i + F_{it}$$

$$F_{it} = \rho F_{i,t-1} + \beta_t X_{it} + \varepsilon_{it}$$

$$\mu_t = f_1(t) = \log(t)$$

$$\beta_t = f_2(t) = a + bt + ct^2 = 0.001 \cdot t \cdot (50 - t)$$

- $f_1(t)$ and $f_2(t)$ are arbitrary functions of t - no specific functional form is assumed in the model

Asparouhov, Muthén and Hamaker

Muthén & Muthén 66/70

Two-level AR(1) TVEM - input file

```
variable:
  names are y x t c;
  within=x;
  cluster=c t;

data:file=new1.dat;

ANALYSIS: TYPE = CROSS random;
estimator=bayes; biter=(5000); process=2;

model:
  %within%
  y@0.05; eta by y@1 (&1);
  eta on eta&1*0.5; eta*1.2;
  s | eta on x;

  %between t%
  y: s: [s];

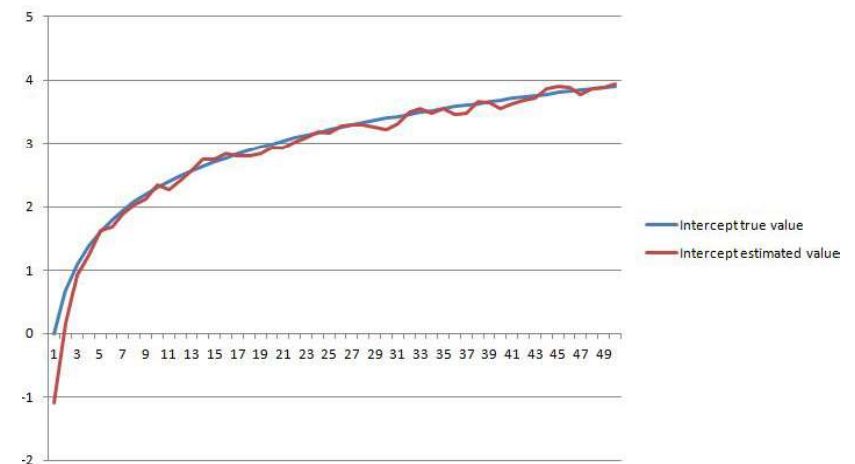
  %between c%
  y*0.5; [y*0]; s@0; [s@0];

savedata: file is TVEM.dat; save=FS(100);
```

Asparouhov, Muthén and Hamaker

Muthén & Muthén 67/70

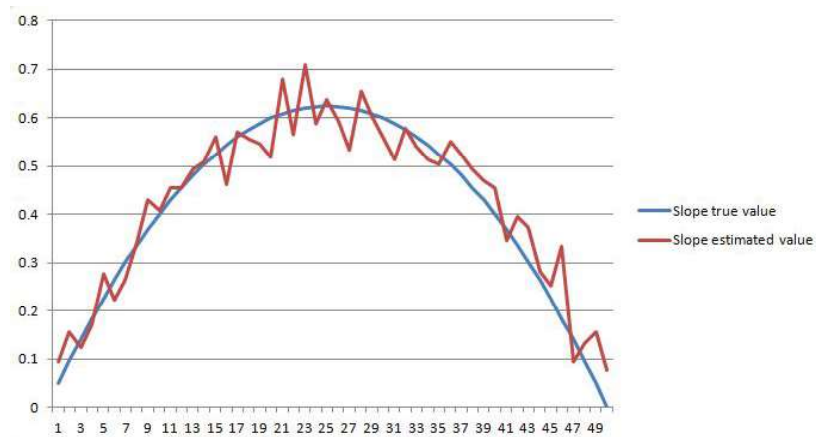
Two-level AR(1) TVEM - intercept results



Asparouhov, Muthén and Hamaker

Muthén & Muthén 68/70

Two-level AR(1) TVEM - slope results



Dynamic Latent Class Analysis

In my talk tomorrow we will discuss the general framework implemented for release in Mplus Version 8.0 and Version 8.1 which simultaneously incorporates time series modeling, hidden Markov modeling, multilevel modeling and mixture modeling. The framework can be useful to study continuous and categorical latent variable development across time. Specific examples and simulation studies are discussed such as:

- Multilevel Mixture Models
- Multilevel Latent Transition Models with cluster specific transition probabilities
- Multilevel Mixtures of Dynamic Structural Equation Models
- Dynamic Latent Class Analysis
- Multilevel Hidden Markov Model
- Multilevel Markov Switching Autoregressive Models
- Multilevel Markov Switching Kalman Filter Models