Nonlinear Bayesian multilevel structural equation modeling

Holger Brandt

University of Kansas

KU Summer Stats Camp

Structure

1. Intro to Bayesian modeling
2. Simple models in stan
   - Regression model
   - Multilevel model
   - Structural equation model
   - Multilevel SEM
3. Models and path diagrams
   - Estimation methods
4. Summary

Conditional vs. unconditional probabilities

- Unconditional probability to have a heart diseases $P(H = 1)$, or to follow a high fat diet $P(D = 1)$, i.e. the marginal probabilities for some events
- Conditional probability to have a heart disease when following a high fat diet $P(H = 1|D = 1)$, i.e. the probability for some event for a specific subgroup (if you only select people with a high fat diet, how many of them have a heart disease?)

Example conditional vs. unconditional probabilities

	$H = 0$	$H = 1$	
$D = 0$	.60	.05	.65
$D = 1$	.20	.15	.35
	.80	.20	1.00

- The probability to have a heart disease is $P(H = 1) = 20\%$.
- The *joint* probability to have a person who follows a high fat diet *and* to have a heart disease is $P(H = 1, D = 1) = 15\%$
- Conditional probabilities (of having a heart attack when following a high fat diet):

$$P(H = 1|D = 1) = \frac{P(H = 1, D = 1)}{P(D = 1)} = \frac{.15}{.35} = .43 \tag{1}$$

and for having a heart attack when not following a high fat diet

$$P(H = 1|D = 0) = \frac{P(H = 1, D = 0)}{P(D = 0)} = \frac{.05}{.65} = .08 \tag{2}$$

Bayes theorem

- The Bayes rule relates to conditional probabilities to each other:

$$P(D = 1|H = 1) = \frac{P(H = 1|D = 1) \cdot P(D = 1)}{P(H = 1)} \tag{3}$$

Model estimation

- A (probability) model for a data set $\mathbf{y}$ is formulated using a *conditional density* $f(\mathbf{y}|\boldsymbol{\theta})$ for given parameter values $\boldsymbol{\theta}$.
- This density is often written as:

$$LL(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) = f(\mathbf{y}|\boldsymbol{\theta}) \tag{4}$$

- These parameter values are typically unknown and need to be estimated.
- This is often conducted by finding those values that have the highest probability to produce the data:

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\theta} LL(\boldsymbol{\theta}) \tag{5}$$

and is called maximumum likelihood estimate.

Model estimation

- For a cfa the density is given by

$$LL(\boldsymbol{\theta}) = \log|\boldsymbol{\Sigma}(\boldsymbol{\theta})| - \log|\mathbf{S}| + \mathrm{tr}(\mathbf{S}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}) - p \tag{6}$$

- Characteristics:
  - No assumption about $\boldsymbol{\theta}$ (e.g., plausible range)
  - Comparison of estimates only after estimation with $\hat{\boldsymbol{\theta}}$ and $Var(\hat{\boldsymbol{\theta}})$

Model estimation: including further information

- Assume that $\mathbf{y}$ and $\boldsymbol{\theta}$ both are random and have a probability distribution (density).
- Then the conditional density $f(\mathbf{y}|\boldsymbol{\theta})$ is

$$f(\mathbf{y}|\boldsymbol{\theta}) = \frac{f(\mathbf{y}, \boldsymbol{\theta})}{f(\boldsymbol{\theta})} \tag{7}$$

  where $f(\mathbf{y}, \boldsymbol{\theta})$ is the joint distribution of the data and the parameters, and $f(\boldsymbol{\theta})$ is a prior distribution (marginal density of $\boldsymbol{\theta}$)
- However, what we are actually interested in is

$$f(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}, \boldsymbol{\theta})}{f(\mathbf{y})} \tag{8}$$

- . . . which is the probability (distribution) for a parameter (vector) given that we observed some data.
- However, $f(\mathbf{y}, \boldsymbol{\theta})$ is typically unknown.

Model estimation: The Bayes rule

- Both equations can be combined using the Bayes rule from above:

$$f(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})}{f(\mathbf{y})} \tag{9}$$

- . . . or equivalently

$$\underbrace{f(\boldsymbol{\theta}|\mathbf{y})}_{\text{posterior}} = \underbrace{f(\mathbf{y}|\boldsymbol{\theta})}_{\text{likelihood}} \cdot \underbrace{f(\boldsymbol{\theta})}_{\text{prior}} \cdot \underbrace{f(\mathbf{y})^{-1}}_{\text{constant}} \tag{10}$$

- which is typically expressed as

$$f(\boldsymbol{\theta}|\mathbf{y}) \propto f(\mathbf{y}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta}) \tag{11}$$

where $\propto$ means "proportional to"

Bayes: Define estimator

- Estimator:

$$\hat{\boldsymbol{\theta}}_{mode} := \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}|\mathbf{y}) \tag{12}$$

- If the prior is a constant, i.e. $f(\boldsymbol{\theta}) \propto 1$, it follows that

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{mode} &= \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}|\mathbf{y}) & (13)\\ &= \arg\max_{\boldsymbol{\theta}} \underbrace{f(\mathbf{y}|\boldsymbol{\theta})}_{\text{likelihood}} \cdot \underbrace{f(\boldsymbol{\theta})}_{\propto 1} & (14)\\ &= \arg\max_{\boldsymbol{\theta}} LL(\boldsymbol{\theta}) = \hat{\boldsymbol{\theta}}_{ML} & (15)\end{aligned}$$

which is the ML estimator from above.

Bayes estimators with real (non-constant) priors

- From the equations above, we can rewrite the model to get some more information about the importance and the characteristics of the prior

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{mode} &= \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}|\mathbf{y}) && (16)\\
&= \arg\max_{\boldsymbol{\theta}} f(\mathbf{y}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta}) && (17)\\
&= \arg\max_{\boldsymbol{\theta}} \log f(\mathbf{y}|\boldsymbol{\theta}) + \log f(\boldsymbol{\theta}) && (18)\\
&= \arg\max_{\boldsymbol{\theta}} LL(\boldsymbol{\theta}) + \log f(\boldsymbol{\theta}) && (19)
\end{aligned}$$

- $\log f(\boldsymbol{\theta})$ can be viewed as a *penalty term* to the ML estimate.

Bayes estimators with real (non-constant) priors

- If the prior $\log f(\boldsymbol{\theta})$ has mass on a specific $\boldsymbol{\theta}_0$ then
  - ...the $LL(\boldsymbol{\theta})$ has only a minor influence on the estimate $\hat{\boldsymbol{\theta}}_{mode}$
  - ...and $\hat{\boldsymbol{\theta}}_{mode} \approx \boldsymbol{\theta}_0$
- With increasing sample size, the impact of the prior vanishes, i.e. in large sample sizes the ML estimator and the Bayes estimator will produce the same estimates.

Example for consequences of (wrong) priors

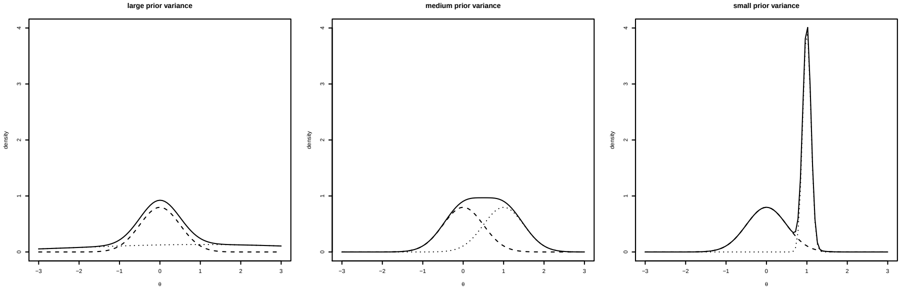


Figure: Consequences of priors (dotted lines) for posterior distribution (solid lines) for the same data set (dashed lines).

Thought experiment for prior selection

- Run experiment 1 → $\mathbf{y}_1$, ML estimate based on $f(\boldsymbol{\theta}|\mathbf{y}_1) \propto f(\mathbf{y}_2|\boldsymbol{\theta}) \cdot 1$
- Run experiment 2 → $\mathbf{y}_2$
- Now, we can use the information from the first experiment for the second experiment:

$$f(\boldsymbol{\theta}|\mathbf{y}_2) \propto f(\mathbf{y}_2|\boldsymbol{\theta}) \cdot f(\mathbf{y_1}|\boldsymbol{\theta}) \tag{20}$$

Thought experiment for prior selection

- Then we have a Bayesian estimator

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{mode} &= \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}|\mathbf{y}_2) && (21)\\
&= \arg\max_{\boldsymbol{\theta}} f(\mathbf{y}_2|\boldsymbol{\theta}) \cdot f(\mathbf{y_1}|\boldsymbol{\theta}) && (22)\\
&= \arg\max_{\boldsymbol{\theta}} LL_2(\boldsymbol{\theta}) + LL_1(\boldsymbol{\theta}) && (23)
\end{aligned}$$

- If $\mathbf{y}_1, \mathbf{y}_2$ are iid then it holds in general that $LL(\boldsymbol{\theta}) = \sum_{i=1}^{n} LL_i(\boldsymbol{\theta})$ and hence

$$\hat{\boldsymbol{\theta}}_{mode} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n_1+n_2} LL_i(\boldsymbol{\theta}) \qquad (24)$$

which is the ML estimate for merged data sets 1 and 2.

More on priors

- For each parameter in a model, a prior distribution needs to be chosen.
- This includes two parts:
  - Choose a distribution type (e.g., normal)
  - Choose specific so called *hyperparameters* for the distribution (e.g., mean and variance)
- Categorization of priors:
  - *Uninformative*: e.g., uniform distribution
  - *Weakly informative*: e.g., normal distribution with medium variance (e.g., 2)
  - *Informative*: e.g., normal distribution with small variance (e.g., .1)

Typical priors

- *Normal* distribution, e.g., for
    - Means
    - Regression coefficients
    - Covariances
- *Inverse Gamma* distribution, e.g., for variances
- *Half-Cauchy* distribution, e.g., for standard deviation
- *Inverse Wishart* distribution, e.g., for covariance matrices
- Lk_j distribution for correlation matrices
- *Beta* distribution, e.g., for correlation coefficients or probabilities (bounded at 0 and 1)

Derivation of posterior distribution

- The posterior distribution is typically a multivariate (conditional) distribution for which no simple expression exists.
- As a consequence, so called sampling procedures are used that sample from the posterior distribution
- The marginal distributions are then used for inference (e.g., mean, standard deviation, median, percentiles, mode)
- There exist different sampling procedures, which can be subsumed under *Monte Carlo Markov Chain* (MCMC) procedures, for example,
    - Gibbs
    - Metropolis Hastings
    - Hamiltonian Monte Carlo (HMC)
    - ...

Basic principles of MCMC

- In each iteration i, a draw from the distribution $f(z_{i-1})$ is conducted. Each draw only depends on the previous sample and iteration.

$$f(z_0) \to z_1 \to f(z_1) \to z_2 \to f(z_2) \to z_3 \dots \tag{25}$$

- It can be shown that $f(z_{i>t})$ approximates the true posterior distribution $f(z)$ after t iterations (e.g., $t = 1000$)

→ Stationary distribution that ignores the starting values z_0.

- The initial iterations are called *burnin* and are discarded
- After stationarity is achieved, further samples are drawn and used for the final estimates

Basic principles Gibbs

- The Gibbs sampler samples from a full conditional distribution
- Example: Data set $\mathbf{y}$ and two parameters μ, σ^2:
    1. Sample from $f(\mu|\sigma_0^2, \mathbf{y})$: μ_1
    2. Sample from $f(\sigma^2|\mu_1, \mathbf{y})$: σ_1^2
    3. Sample from $f(\mu|\sigma_1^2, \mathbf{y})$: μ_2
    4. Sample from $f(\sigma^2|\mu_2, \mathbf{y})$: σ_2^2
    5. $\ldots$

Practical issues: Convergence

- To ensure that a model converges, several *chains* are used that use different sets of starting values: If the model converges, the parameter estimates from all chains should be similar
- Pairs of parameter estimates (e.g., from draws z_1, z_2) can be highly correlated (autocorrelation). It is meaningful to skip some of the iterations in between. This is called thinning and the amount of thinning is indicated by the thinning factor (e.g., a factor of 3 uses samples $z_1, z_4, z_7 \ldots$)
- Convergence checks:
  - Rhat statistic
  - Trace plots
  - Density plots

Practical issues: Rhat statistic

- Aka Potential scale reduction (PSR) or Gelman-Rubin convergence criterion
- Assesses within vs. between chain variability:

$$Rhat = \frac{\sqrt{W + B}}{W} \tag{26}$$

- For convergence $Rhat \to 1$. As a rule of thumb $Rhat < 1.1$ indicates convergence.
- If model has not converged, use more iterations.

Examples in R and stan

1. Regression model
2. Multilevel model

Regression model: Implementation in stan

- Assume the following simple regression model with interaction effect:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{1i} + \epsilon_i \tag{27}$$

  with normal residual $\epsilon_i \sim N(0, \sigma)$.

- This model implies the following distribution of y_i:

$$y_i \sim N(\underbrace{\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{1i}}_{=\hat{y}_i}, \sigma) \tag{28}$$

$$= N(\hat{y}_i, \sigma) \tag{29}$$

  where $\hat{y}_i$ is the conditional mean (or expected value) for y_i for subject i given predictor values x_{1i}, x_{2i} and parameters $\beta_0, \ldots, \beta_3$.

Regression model: Implementation in stan

- For a model in stan, one needs to specify this mean and this variance as well the priors for the parameters
    - Formulate a mean structure for y: `muy=`$\beta_0 + \ldots$
    - Formulate a statistical model for y: `y` $\sim$ `normal(muy,sigmay)`
- Typical (weakly to non-informative) priors here are:
    - Regression coefficients: $\beta \sim N(0, 1)$
    - Residual variance (sigmay): $\sigma \sim Cauchy(0, 2.5)$ which is the Half Cauchy distribution

Regression model: Implementation in stan

We now switch to R and implement the model for the pisa data set (demo).

Random intercept model: Implementation in stan

- Assume the following extension of the regression model using a clustering (e.g. student i in school j):

$$y_{ij} = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{1ij} x_{2ij} + u_{0j} + \epsilon_{ij} \tag{30}$$

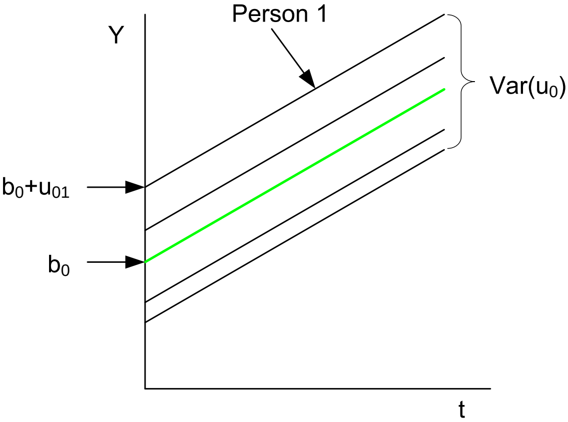
  with normal residual $\epsilon_i \sim N(0, \sigma)$ and random term u_{0j} that indicates cluster-specific deviations from the overall intercept.

- This model implies the following distribution of y_{ij} and u_{0j}:

$$y_{ij} \sim N(\beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{1ij} x_{2ij} + u_{0j}, \sigma) \tag{31}$$

$$u_{0j} = N(0, \sigma_u) \tag{32}$$

Illustration of random intercept model



Random intercept model: Implementation in stan

- For a model in stan, one needs
  - Formulate a mean structure for y: `muy=`$\beta_0 + \ldots + u_0$
  - Formulate a statistical model for y: `y` $\sim$ `normal(muy,sigmay)`
  - Formulate a statistical model for u: `u0` $\sim$ `normal(0,sigmau)`
- Typical (weakly to non-informative) priors here are:
  - Regression coefficients: $\beta \sim N(0,1)$
  - (Residual) variance (sigmay,sigmau): $\sigma \sim Cauchy(0,2.5)$ which is the Half Cauchy distribution

Random intercept model: Implementation in stan

We now switch to `R` and implement the model for the pisa data set (exercise).

Random intercept and slope model: Implementation in stan

- Assume the following extension of the random intercept model using random slopes for the two linear effects:

$$\begin{aligned} y_{ij} = {} & \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{1ij} x_{2ij} \\ & + u_{0j} + u_{1j} x_{1ij} + u_{2j} x_{2ij} + \epsilon_{ij} \end{aligned} \tag{33}$$

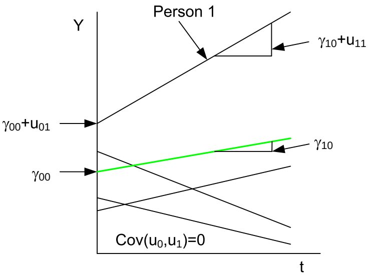
  with normal residual $\epsilon_i \sim N(0, \sigma)$ and random terms u_{0j}, u_{1j}, u_{2j} that indicate cluster-specific deviations from the overall trajectories.

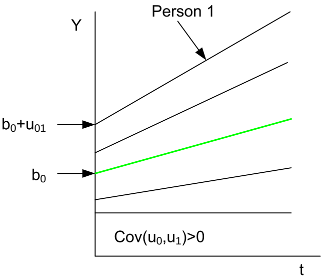
- This model implies the following distribution of y_{ij} and u_{0j}:

$$\begin{aligned} y_{ij} \sim {} & N(\beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{1ij} x_{2ij} \\ & + u_{0j} + u_{1j} x_{1ij} + u_{2j} x_{2ij}, \sigma) \end{aligned} \tag{34}$$

$$u_{kj} = N(0, \sigma_{uk}), k = 0, 1, 2 \tag{35}$$

Illustration of random intercept and slope model





Random intercept and slope model: Implementation in stan

- For a model in stan, one needs
  - Formulate a mean structure for y: `muy=`$\beta_0 + \ldots + u_0 + \ldots$
  - Formulate a statistical model for y: `y` $\sim$ `normal(muy,sigmay)`
  - Formulate a univariate model for each u: e.g., `u0` $\sim$ `normal(0,sigmau0)`
  - Or, a multivariate formulation, which in general is a meaningful alternative: `u` $\sim$ `multi_normal(0,Sigmau)`.
- We choose the univariate model for now because
  - The variances in this example are close to zero
  - Multivariate distributions should be reparameterized using a Cholesky decomposition in stan (see below)

Random intercept and slope model: Implementation in stan

We now switch to R and implement the model for the pisa data set (exercise).

SEM: Implementation in stan

- For a SEM in stan, one needs to include a model for each observed and latent variable.
- Latent variables are treated as parameters that are unknown. Using the sampler, one directly generates scores (idea: factor scores) for each person and hence can directly use them for the model formulation.
- For each variable measurement or structural equations need to be specified as well as a statistical model. This model can be univariate (e.g., for indicator variables) or multivariate (e.g., for latent predictors that correlate).

SEM: Measurement model

- For each observed variable x, y, one formulates a measurement equation, for example

$$x_i = \tau_x + \lambda_x \xi_i + \delta_i \tag{36}$$

$$y_i = \tau_y + \lambda_y \eta_i + \epsilon_i \tag{37}$$

- which results in a statistical model of

$$x_i \sim N(\tau_x + \lambda_x \xi_i, \sigma_\delta) \tag{38}$$

$$y_i \sim N(\tau_y + \lambda_y \eta_i, \sigma_\epsilon) \tag{39}$$

  and can be implemented in the same way as the regression model above.

- Constraints and fixed parameters can directly be included e.g., by having

$$x_{1i} \sim N(\xi_i, \sigma_\delta) \tag{40}$$

  for a scaling variable x_{1i} and factor loading of 1 and intercept 0.

SEM: Structural model

- For each dependent variable η, one formulates a structural equation, for example

$$\eta_i = \beta_0 + \beta_1\xi_{1i} + \beta_2\xi_{2i} + \beta_3\xi_{1i}x_{2i} + \zeta_i \tag{41}$$

- which results in a statistical model of

$$\eta_i \sim N(\beta_0 + \beta_1\xi_{1i} + \beta_2\xi_{2i} + \beta_3\xi_{1i}x_{2i}, \sigma_\zeta) \tag{42}$$

and can be implemented in the same way as the regression model above.

SEM: Structural model

- For predictor variables ξ_1, ξ_2, one typically formulates a multivariate model using a vector of means and covariance matrix.
- In stan there are several possibilities to do that
  1. Generate a covariance matrix and use a multivariate (normal) distribution for ξ
  2. Generate a correlation matrix and standard deviations, and use a multivariate (normal) distribution for ξ
  3. Use a Cholesky transformation and generate univariate normal variables that are transformed to ξ

  where the advantages for estimation increase from 1 to 3 (i.e. 1 is least preferable). In our example this leads to a sampling time efficiency of the factor 3.

SEM: Structural model

1 Covariance matrix: use
`matrix[dim(xi),dim(xi)] phi;` as parameter and
`wishart(nu,Sigma);` as prior
with hyperpriors nu (degrees of freedom), Sigma (covariance matrix).

SEM: Structural model

2 Correlation matrix + SD's: use

  - `corr_matrix[#(xi)] rho;`
  - `vector<lower=0>[#(xi)] sigmaxi;`

  as parameters for correlation matrix and vector of SD's (`dim(xi)` is the number of ξ's) and

  - `rho ∼ lkj_corr(2);`
  - `sigmaxi ∼ cauchy(0,2.5);`

  as priors

- Formulate the covariance matrix in the transformed parameters as
  - `phi = quad_form_diag(rho,sigmaxi);`
- and xi is then
  - `xi ∼ multi_normal(muxi,phi);`

(3) Cholesky transformation

- Each matrix $\boldsymbol{\xi}$ of correlated variables can be formulated as

$$\boldsymbol{\xi} = \boldsymbol{\mu} + \mathbf{A}\mathbf{z} \tag{43}$$

  where $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ are uncorrelated, standard normal variables and $\mathbf{A}$ is a lower triangle matrix (Cholesky matrix).
- This procedure has the advantage that univariate z's can be generated and then be transformed to correlated variables with arbitrary mean vector and covariance matrix.
- In stan this involves several steps but is more stable and way faster.
- In general, it holds that if you can transform a multivariate problem into a univariate one, then you should do that!

(3) Cholesky transformation in stan for 2 predictor variables

- Parameters:
  - `vector[2] muxi;` // mean xi
  - `vector<lower=0>[2] sigmaxi;` // SD xi
  - `cholesky_factor_corr[2] L1;`
  - `matrix[N,2] zi;` // these are uncorrelated standard normal variables
- Transformed parameters:
  - `matrix[N,2] xi;`
  - `xi = muxi + zi*diag_pre_multiply(sigmaxi,L1)';` // this generates xi

(3) Cholesky transformation in stan for 2 predictor variables

- Model:
  - `to_vector(zi) ~ normal(0,1);`
  - `muxi ~ normal(0,1);` // means
  - `sigmaxi ~ cauchy(0,2.5);` // SDs
  - `L1 ~ lkj_corr_cholesky(2);` // Cholesky matrix
- Generated quantities:
  - `matrix[2,2] phi;` // covariance matrix for output
  - `phi = diag_pre_multiply(sigmaxi,L1)* diag_pre_multiply(sigmaxi,L1)';`

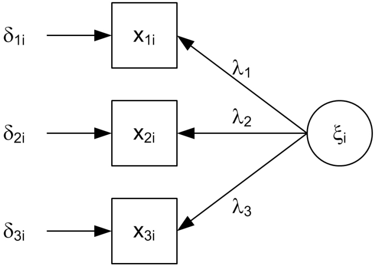
Final remark on covariance/correlation matrices

- One can also use the correlation matrix directly. In this case, the variances of the latent variables are used as a scaling.
- Then, only option 2 and 3 can be used. In both cases, one does not need the SD's of the latent variables but estimates factor loadings for all indicators.

SEM: Implementation in stan

We now switch to R and implement the SEM for the Kenny-Judd data set (exercise and demo).

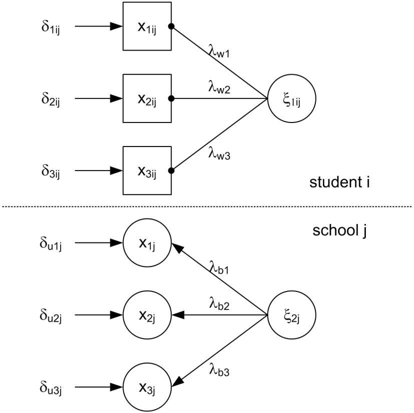
Single level factor model



Each indicator has the measurement model:

$$x_i = \lambda \xi_i + \delta_i \tag{44}$$

Two level factor model (using Muthén type notation)



Two level factor model: Equations and interpretation

- Each indicator has a measurement model that includes within and between components:

$$x_{ij} = \lambda_w \xi_{1ij} + \lambda_b \xi_{2j} + \delta_{ij} + \delta_{uj} \tag{45}$$

    - ξ_{1ij} is a within level factor representing the same individual characteristics across items (e.g., individual math skills)
    - ξ_{2j} is a between level factor representing the same school-specific characteristics across items (e.g., average math skill level in school)
    - δ_{ij} is a within level residual representing individual characteristics independent of the factors (e.g., individual attention deficits)
    - δ_{uj} is a between level residual (random effect) representing school-specific characteristics independent of the factors (e.g., this specific math item was exercised a lot in this school but not in others)

Two level factor model with constraints

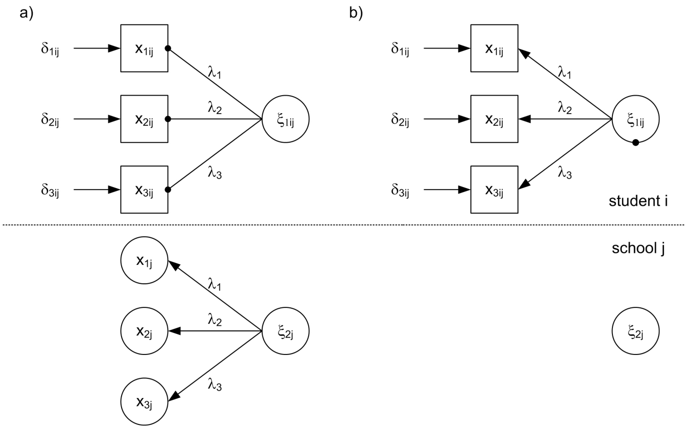


Figure: All school-specific deviations are due to differences in the latent factor. a) and b) represent equivalent models.

Two level factor model: Equations and interpretation

- Each indicator has a measurement model that includes within and between components:

$$x_{ij} = \lambda\xi_{1ij} + \lambda\xi_{2j} + \delta_{ij} + \underbrace{\delta_{uj}}_{=0} \tag{46}$$

$$= \lambda(\xi_{1ij} + \xi_{2j}) + \delta_{ij} \tag{47}$$

  - ξ_{1ij} and ξ_{2j} are a within and between level factors representing the same individual and school-specific characteristics across items (e.g., individual math skills).
  - In this formulation, both factors have the same interpretations (because factor loadings are constrained across levels).
  - The model decomposes within and between characteristics on the latent level only. (This can be tested by estimating the variances of δ_{uj} first).

Multilevel SEM 1 with random intercept

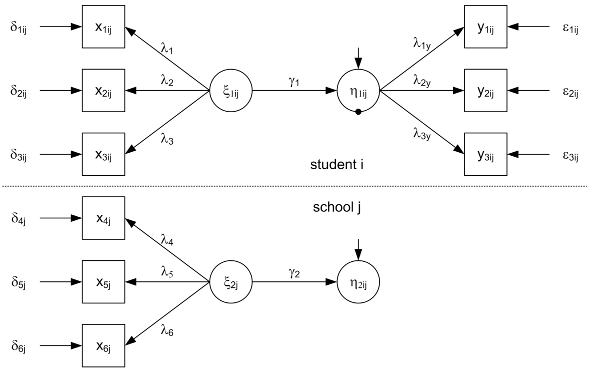


Figure: Here ξ_2 is a conceptually different between level factor with school-level indicators (e.g., SES factor of the school area)

Multilevel SEM 1: Equations and interpretation

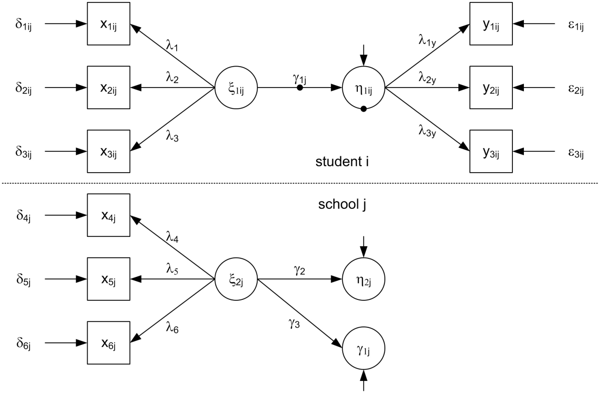
- Again, all school specific aspects of the within level variables are due to differences at the latent level. ξ_1 does not have a school-level variance component (it could be reading attitude, that does only differ across students but not across schools)
- The dependent variable has the following structural model:

$$\eta_{1ij} = \eta_{2j} + \gamma_1 \xi_{1ij} + \zeta_{1ij} \tag{48}$$

$$\eta_{2j} = \alpha + \gamma_2 \xi_{2j} + \zeta_{2j} \tag{49}$$

  - γ_1, γ_2 are fixed effects of the within and between level factors ξ_1, ξ_2.
  - α is the intercept
  - ζ_{1ij} is a residual on the within level
  - ζ_{2j} is a residual on the between level. In manifest mlm, this is often denoted as u_0

Multilevel SEM 2 with random slope



Multilevel SEM 2: Equations and interpretation

- The difference to ML-SEM 1 before is that γ_{1j} is a random slope. In principle any parameter in the model on the within level (also factor loadings) can be formulated as random term that is incorporated in the between level (but one should have a hypothesis on it).
- The structural model is now given by:

$$\eta_{1ij} = \eta_{2j} + \gamma_{1j}\xi_{1ij} + \zeta_{1ij} \tag{50}$$

$$\eta_{2j} = \alpha + \gamma_2\xi_{2j} + \zeta_{2j} \tag{51}$$

$$\gamma_{1j} = \gamma_1 + \gamma_3\xi_{2j} + \zeta_{3j} \tag{52}$$

  - γ_1, γ_2 are fixed effects. γ_3 is a cross-level interaction.
  - α is the intercept
  - ζ_{1ij} is a residual on the within level
  - ζ_{2j}, ζ_{3j} are residuals on the between level. In manifest mlm, thee are often denoted as u_0, u_1

Estimation methods for ML SEM

1. Pseudobalanced ML (Muthén's ML/MUML) [outdated]
2. Two phase direct estimation [outdated]
3. Weighted least squares (WLS) [fairly outdated]
4. Full maximum likelihood [e.g., in Mplus]
5. Bayesian estimator [e.g., in stan, jags, or Mplus]

In combination with nonlinear SEM, only the last two are applicable. The Mplus implementation (with LMS) is very limited with the number of nonlinear terms that can be modeled (and it takes very long for computation)

Likelihood function for linear latent two level model (see Rabe-Hesketh & Skrondal, 2006)

- Level 2: $n^{(3)}$ schools j sampled with probabilities π_j, $w_j = 1/\pi_j$
- Level 1: $n_j^{(2)}$ students i sampled from each school with probabilities $\pi_{i|j}$, $w_{i|j} = 1/\pi_{i|j}$
- $n_{ij}^{(1)}$ items k used for each students
- Log likelihood function:

$$LL = \sum_{j=1}^{n^{(3)}} \log \int \exp \left(\sum_{i=1}^{n_j^{(2)}} \log \int \exp \left[\sum_{k=1}^{n_{ij}^{(1)}} LL \left(y_{kij} | \zeta_{ij}^{(2)}, \zeta_j^{(3)} \right) \right] g(\zeta_{ij}^{(2)}) d\zeta_{ij}^{(2)} \right) g(\zeta_j^{(3)}) d\zeta_j^{(3)} \quad (53)$$

where $\zeta_{ij}^{(2)}, \zeta_j^{(3)}$ indicate the latent (random) variable that are integrated out.

FIML for nonlinear models

- Even for linear models, the likelihood function needs to be approximated using numerical methods.
- The main challenge for nonlinear models is that the inner part of the function $(LL\left(y_{kij}|\zeta_{ij}^{(2)}, \zeta_{j}^{(3)}\right))$ is replaced with the log likelihood function used in LMS or NSEMM. This needs further (nested) numerical approximation.
- In principle, these ML-SEM can be extended to mixture models. Again, this might lead to a strong computational burden.
- Two example codes for a random intercept and a random slope model is provided for Mplus. The random slope model needs further care because the computational burden is too high (for demo version use first 2 indicators for each construct).

Bayesian implementation

- The Bayesian implementation in stan is straightforward.
- It is a combination of the mlm and the sem files that we used this morning and can directly be integrated into a single file.
- Here, again a Cholesky decomposition for the latent factors is meaningful.

Summary

- The Bayesian framework is a very flexible framework in which almost any model can be specified.
- For complex nonlinear ML-SEM an identification needs to be ensured (even more when semiparametric models are tried to be implemented).
- A careful investigation of priors and convergence is of utter importance. Packages that automatically set priors are not recommended even though they seem to be optimal from an applied perspective.
- Overall: Nonlinearity, nonnormality and clustered data structures often occur in social sciences. They need to be addressed thoroughly and adequately.
- Future research is needed, for example to develop model fit measures.

Thank you for your attention.