Nonlinear modeling – Bayesian Modeling and Multilevel SEM

Holger Brandt

June 07, 2017

1 A simple regression model

The first data set is based on the PISA data set used before and is provided in `pisa_manifest.dat`. As you might have guessed, the students were clustered in schools (which were clustered in cities, counties, departments, states, countries etc.). Here, we first use manifest data to start some simple coding in stan. The data set contains a school id, the reading skills, the online activities and the reading attitude. All variables (except the school id) were standardized (for illustrative purposes).

1. Run a simple regression model that predicts the reading skills using the online activities and the reading attitude. Include an interaction between the two predictors. Ignore the clustering for now.
2. Interpret the results.
3. Now formulate the model syntax for a Bayesian regression in `stan`. The syntax for the first example is provided in `regression.stan` for you (we will go over the syntax step by step). Run the model.
   (a) Check the convergence of the model.
   (b) Interpret the results. Compare it to the OLS regression.

2 Some multilevel models

For this exercise, we will use the same data as before, but we will incorporate the clustering in schools in the analysis.

1. Run a simple random intercept model that predicts the reading skills using the online activities and the reading attitude accounting for the clustering in schools. Include an interaction between the two predictors. Use the lmer package that provides an ML estimator for multilevel models.

2. Do the results change compared to the regression model above? Is there an effect of the clustering? [We will not go into detail about model comparison here because it is different in Bayesian modeling.]

3. Now extend the model syntax for a Bayesian multilevel model using the syntax for the regression model. You will need to account for the following steps (see also `mlm1_start.stan`) in the model syntax:

   - *Data* is extended including now: number of schools (**Ns**), cluster indicator (**school**).
   - *Parameters* include a vector for the random intercept **u0** (length **Ns**) and a standard deviation for the random intercept (**sigmau0**).
   - *Transformed parameters* still need **muy** but the random intercept **u0** must be added in the model formulation.
   - The *model* now needs specifications for the distribution of **u0** and a prior for **sigmau0**.

4. Check the convergence, interpret the results and compare it to the lmer-results.

5. Extend the model to also include a random slope for the two predictor variables (both are within level predictors). First, run the model in lmer. Use uncorrelated random terms.

6. Then, formulate a model in stan that assumes uncorrelated random terms. Test the model. Do you need the random slope?

   For the extension of the random intercept model you need to add the following aspects (all other things are as above):

   - *Data:* same as before.
   - In the model (name the file `mlm2_start.stan`): *Parameters:* include an additional vectors for the random slopes **u1** and **u2** (again with length **Ns**) and their SD's (**sigmau1**,**sigmau2**).
   - Adequately extend the model formulation for **muy**.
   - *Model:* Specifications for the distributions of **u1,u2** and priors for **sigmau1,sigmau2**.

3 A simple structural equation model

The next exercise again uses the Kenny-Judd data set from Day 3 (where we started with the constrained approach; `data_kj.dat`). We will now implement an SEM with interaction effect in stan. As you will notice, this is quite straightforward because the products of the latent variables can directly be included.
This first example is a demonstration and we will go step by step through the syntax of `sem1.stan`. One of the main practical aspects is the naming of parameters etc. (to keep an overview). Here, we use the following convention (later you can use whatever you want).

- Regression coefficients are in a vector called **b1**. The intercept is separate (**b0**).
- Factor loadings are vectors with the letter **l** (small L) followed by **x** or **y**.
- (Residual) SD's are vector names **sigma** followed by the respective variable (e.g., **x**, i.e. **sigmax**).
- Correlation matrices are names **rho**, covariance matrices **phi**. If more than one matrix exist, add again the variables' name (e.g., **xi**)
- (Conditional) means are vectors names **mu** followed by the respective variable (e.g. **muxi** or **mueta**)

1. Test the interaction model for the Kenny-Judd model. Check the convergence and the results.
2. Add quadratic effects to the model (name the file `sem1a.stan`). You need to adjust the length of **b1** (was 3) and change the equation for **mueta**. What do you conclude about the nonlinear structure in the data set?
3. The model syntax also provides a "Cholesky version" of the model (`sem1b.stan`). In fact, this is the recommended version in stan because multivariate distributions are complicated to sample from. We will go in detail if we have time for it (otherwise you can test it at home using `sem1b.stan`).

4 A multilevel structural equation model

The final data set returns to our PISA example in `pisa_mlsem.dat`. We will consider only the two-level version with clustering in schools.

1. The first model only includes reading skills and online activities (demo, code is provided in `mlsem1.stan`):

   - Formulate a model that includes a random intercept for the reading skills.
   - Each latent variable (reading skills and online activities) loads on 3 indicators. Use the first indicator as a scaling variable.
   - Include a quadratic and a cubic effect for online activities.

2. Final model: We now also include the reading attitude. Add the following elements (exercise):

   - Add a measurement model for reading attitude.
   - Use a multivariate formulation for the latent predictors (either by using `multi_normal` and a covariance matrix as in `sem1.stan` or by using the Cholesky version as in `sem1b.stan`)
   - Include linear and quadratic effects for online activities and reading attitude as well as their interaction. Remember to extend the vector **b1** for that.