Bayesian Statistics - Exercise 2

Objective

In this second exercise JAGS is used to obtain the results for a more complex model through Gibbs sampling. You will learn how to inspect the convergence of the samplers in R.

Help files

If you need help, the JAGS user manual or the rjags reference manual can be consulted. If you are stuck with a specific error message, a Google search may turn up possible causes and solutions.

The data

The data for this exercise were obtained in an internet survey ($N = 172$) investigating pet ownership, attitudes towards pets, and how these might be related to personality characteristics. Your research goal is to see whether people's attitude can be predicted from their level of the personality traits agreeableness and extraversion. For this purpose, you want to conduct a multiple regression analysis. The attitude towards pets and pet ownership is a continuous composite variable based on 7-point Likert scale items running from -3 to 3, representing a strongly negative or strongly positive attitude, respectively. The individual scores for the personality traits were obtained similarly, with higher scores indicating a higher level of agreeableness or extraversion.

A. Loading the SPSS dataset into R

To load an SPSS dataset into R, you first need to load the `haven` package. Now you can load the dataset into R with use of the `read_sav()` function.

B. Specifying the regression model.

Build a model file for the regression analysis, immediately using both predictors (and an intercept). You'll need to write the regression model in terms of a likelihood for each person's attitude score (using a for-loop). Tip: Remember that ~ indicates a stochastic distribution while <- indicates a deterministic relation. If you don't know where to start, take a look at the simple linear regression example in the JAGS user manual chapter 2.

As priors for the intercept and the regression coefficients, use normal distributions (= conjugate priors). You can use one multivariate normal distribution, or separate normal distributions. Note that JAGS uses precision, which is the inverse of the variance. What hyperparameters do you choose if you want low-informative priors?

Although the residual variance is of little interest, it too requires a prior distribution. A common choice is to put a gamma distribution on the precision, with small parameters like 0.001 and 0.001. (Note: if you want to get posterior results for the variance, instead of only for the precision, you should calculate it within the model file.)

C. Specifying initial values.

When you look in R at the help file for the `jags.model` function under 'Initialization' you can read that there are three methods to specify initial values with `rjags`. For now, choose option 2 to specify distinct initial values for the two MCMC chains.

D. Obtaining samples from the posterior

Now it is time to run the analysis, i.e. obtain samples from the posterior distribution, for the current model, with at least two chains.

E. Assessing convergence

There are various packages and functions for assessing convergence of the MCMC chains in R:

- Using the base R function `plot` on an MCMC.list object results in a history (trace) and density plot per monitored parameter.

- The CODA package that comes with the RJAGS package contains various functions to obtain diagnostics for MCMC output. The functions `traceplot`, `densplot`, `autocorr.plot`, and `gelman.plot` are especially useful.

- The MCMCPLOTS package produces html output with diagnostics plots, which can be useful when a Bayesian model has a large number of parameters. The function `mcmcplot` creates various diagnostics plots for each monitored parameter, including history plots and autocorrelation plots.

Create the following types of output:

- *history plots* : Do you see a random progression of values, i.e., no strong periodicities or trend? Do the estimates from the different chains overlap?

- *autocorrelation plots* : Do you see any high autocorrelations?

- *Gelman-Rubin diagnostic plot* : Did the Gelman-Rubin statistic converge to 1?

If you were not satisfied with the convergence at first, write down what did you do to improve it?

F: Substantive interpretation of the results.

If your model has converged, interpret the parameter estimates. Are the personality traits extraversion and agreeableness useful predictors of people's attitude towards pets/pet ownership?

G. Analysis with an interaction effect.

A colleague wonders if the effect of agreeableness on liking pets might be moderated by extraversion. To investigate this possibility, you should include an interaction term in the model. To obtain interpretable results, make sure that you center both predictors. Do all the necessary calculations within the model specifications, so that you don't have to change the data file at all.

What do you conclude about the potential interaction between extraversion and agreeableness?

H. Dealing with missing data.

If you're interested in missing data, try the following. In the data file, delete some values on the predictors agreeableness and extraversion. Then re-run your (original) analysis. What happens? Do you have an idea how the situation is different than before?

Include these two lines in the model likelihood (within the for-loop):

```
extraversion[i]  ~ dnorm(mu.extraversion,  tau.extraversion)
agreeableness[i] ~ dnorm(mu.agreeableness, tau.agreeableness)
```

and specify low-informative priors for these means and precisions (similar to the priors for the other parameters). In the lists with initial values, include values for these two precisions. Save the model file and re-run your analysis. Can you find out what happens now?