BDA - Assignment 6

Anonymous

Contents

Import some useful libraries

```
library(aaltobda)
library(rstan)
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
## Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file
```

```
library(bayesplot)
```

```
## This is bayesplot version 1.7.2
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

```
##    * Does _not_ affect other ggplot2 plots
```

```
##    * See ?bayesplot_theme_set for details on theme setting
```

```
# rstan options
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)

# Import the data
data("bioassay")
```

Generalized linear model: Bioassay with Stan

Exercise 1

In this exercise, we model a dose-response relation for the bioassay data using the Stan probabilistic programming language. We assume a Gaussian prior distribution characterized by

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \sim \mathrm{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0), \text{ where } \boldsymbol{\mu}_0 = \begin{bmatrix}0\\ 10\end{bmatrix} \text{ and } \boldsymbol{\Sigma}_0 = \begin{bmatrix}2^2 & 10\\ 10 & 10^2\end{bmatrix}.$$

The model is built in Stan as follows

```
code_bioassay <- "bioassay_model.stan"
writeLines(readLines(code_bioassay))
```

```
## // The input data is a vector 'x' of length 'N'.
## data {
##   int<lower=0> N;     // total number of rows
##   vector[N] x;        // dose level
##   int<lower=0> n[N];  // number of animals
##   int y[N];  // number of deaths (positive outcomes)
##
##   // parameters of prior distribution
##   vector[2] mu;
##   matrix[2, 2] Sigma;
##
## }
##
## // The parameters accepted by the model. Our model
## // accepts the 2D parameter 'theta'.
## parameters {
##   vector[2] theta;
## }
##
## transformed parameters {
##   vector[N] rel;
##   rel = theta[1] + theta[2]*x;  // dose-response relation
## }
##
## // The model to be estimated.
## model {
##   theta ~ multi_normal(mu, Sigma);  // prior
##   y ~ binomial_logit(n, rel);       // likelihood
## }
```

Now, we can prepare the data in R with

```
bioassay_data <- list(N = length(bioassay$x),
                      x = bioassay$x,
                      n = bioassay$n,
                      y = bioassay$y,
                      mu = c(0, 10),
                      Sigma = matrix(data = c(4, 10, 10, 100), nrow = 2, ncol = 2))
```

Then, we can get a fit with the following R command and sample from the posterior distribution.

```
max_iters <- 2000
n_chains <- 5
fit <- stan(file = 'bioassay_model.stan', data = bioassay_data, chains = n_chains,
            iter = max_iters, warmup = floor(max_iters/2))
```

Exercise 2

In order to check the convergence of the chains, we can use `stan::monitor` to compute $\hat{R}$ and other convergence analytics. $\hat{R}$ is an indicator of the between- and within-chain variance of the estimates. If the between- or within-chain estimates are generally not agreeing, we will get a larger value. A larger value thus tells that we should continue with further simulations to improve our inference of the target distribution. If the value of $\hat{R}$ is less than 1.05, it is considered safe to use the sample.

```
monitor(fit)
```

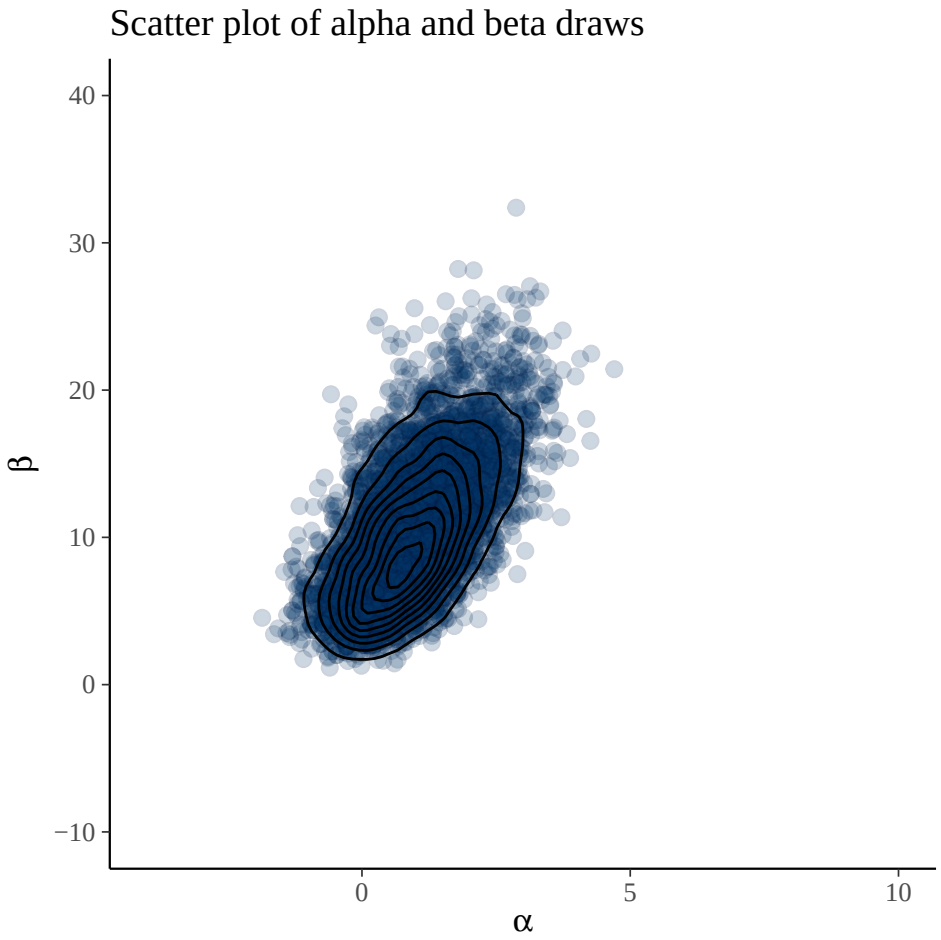
```
## Inference for the input samples (5 chains: each with iter = 2000; warmup = 0):
##
##             Q5  Q50  Q95 Mean  SD  Rhat Bulk_ESS Tail_ESS
## theta[1]  -0.4  0.9  2.6  1.0 0.9     1     1644     1830
## theta[2]   4.0  9.9 19.0 10.5 4.6     1     1629     1870
## rel[1]   -14.4 -7.6 -3.2 -8.0 3.5     1     1911     2080
## rel[2]    -4.2 -2.0 -0.6 -2.2 1.1     1     3047     2587
## rel[3]    -0.8  0.4  1.8  0.5 0.8     1     1941     2000
## rel[4]     2.9  8.2 15.9  8.6 4.0     1     1478     1930
## lp__      -9.1 -6.8 -6.1 -7.1 1.0     1     1806     2282
##
## For each parameter, Bulk_ESS and Tail_ESS are crude measures of
## effective sample size for bulk and tail quantities respectively (an ESS > 100
## per chain is considered good), and Rhat is the potential scale reduction
## factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

As we can see, α (i.e. `theta[1]`) and β (i.e. `theta[2]`) both have an $\hat{R}_\alpha = \hat{R}_\beta = 1$. Both results are below 1.05, which means that we can accept the sample as the chains have converged.

Exercise 3

Here, we plot the draws from α and β in a scatter plot to compare the results with Figure 3.3b in BDA3. The x- and y-axis limits are adjusted to capture the same grid as in the book figure.

```
p <- mcmc_scatter(fit, pars = c('theta[1]', 'theta[2]'), alpha = 0.2)
(p + labs(title = "Scatter plot of alpha and beta draws",
          x = expression(alpha),
          y = expression(beta)) + stat_density_2d(color = "black", size = .5)
  + xlim(-4, 10) + ylim(-10, 40))
```



All in all, the results look quite similar with some slight difference due to the different choice of prior (uniform prior compared to Gaussian).

Exercise 4

This exercise has been written and compiled in Windows using R and the package `Rstan`. I had some installation issues in the beginning since I did not follow the instructions of installing the `Rstan` from source as I had already used it in the previous assignment. The problems disappeared when I followed the instructions. I thought about using Aalto's jupyter when I run into problems, but I resolved those problems locally in the end.

The hardest part about the assignment was getting the hang of Stan syntax and how the model is defined. I believe this will get easier with practice.