Bayesian data analysis – Assignment 5

General information

- The recommended tool in this course is R (with the IDE R-Studio). You can download R **here** and R-Studio **here**. There are tons of tutorials, videos and introductions to R and R-Studio online. You can find some initial hints from **RStudio Education pages**.
- Instead of installing R and RStudio on you own computer, see **how to use R and RStudio remotely**.
- When working with R, we recommend writing the report using R markdown and the provided **R markdown template**. The remplate includes the formatting instructions and how to include code and figures.
- Instead of R markdown, you can use other software to make the pdf report, but the the same instructions for formatting should be used. These instructions are available also in **the PDF produced from the R markdown template**.
- Report all results in a single, **anonymous** *.pdf -file and return it to **peergrade.io**.
- The course has its own R package `aaltobda` with data and functionality to simplify coding. To install the package just run the following (upgrade="never" skips question about updating other packages):

  1. `install.packages("remotes")`
  2. `remotes::install_github("avehtari/BDA_course_Aalto", subdir = "rpackage", upgrade="never")`

- Many of the exercises can be checked automatically using the R package `markmyassignment`. Information on how to install and use the package can be found **here**. There is no need to include `markmyassignment` results in the report.
- Recommended additional self study exercises for each chapter in BDA3 are listed in the course web page.
- We collect common questions regarding installation and technical problems in a course Frequently Asked Questions (FAQ). This can be found **here**.
- Deadline for all assignments are **Sunday at 23.59**.
- If you have any suggestions or improvements to the course material, please post in the course chat feedback channel, create an issue, or submit a pull request to the public repository! Useful feedback will be thanked with bonus points.

Information on this assignment

This assignment is related to Chapters 10 and 11. The maximum amount of points from this assignment is 6.

Reading instructions: Chapter 10 and 11 in BDA3, see reading instructions **here** and **here**.

Grading instructions: The grading will be done in peergrade. All grading questions and evaluations for assignment 5 can be found **here**

Reporting accuracy: For posterior statistics of interest, only report digits for which the Monte Carlo standard error (MCSE) is zero. *Example:* If you estimate $E(\mu) = 1.234$ with $\text{MCSE}(E(\mu)) = 0.01$, you should report $E(\mu) = 1.2$. See lecture video 4.1 and the chapter notes for more information.

To use markmyassignment for this assignment, run the following code in R:

```
> library(markmyassignment)
> assignment_path <-
    paste("https://github.com/avehtari/BDA_course_Aalto/,
        blob/master/assignments/tests/assignment5.yml", sep="")
> set_assignment(assignment_path)
> # To check your code/functions, just run
> mark_my_assignment()
```

Generalized linear model: Bioassay with Metropolis (6 points)

Metropolis algorithm: Replicate the computations for the bioassay example of section 3.7 in BDA3 using the Metropolis algorithm. The Metropolis algorithm is described in BDA3 Chapter 11.2. More information on the bioassay data can be found in Section 3.7 in BDA3, and in **Chapter 3 notes**.

1. Implement the Metropolis algorithm as an R function for the bioassay data. Use the Gaussian prior as in Assignment 4, that is

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

   a) Start by implementing a function called `density_ratio` to compute the density ratio function, r in Eq. (11.1) in BDA3. Below is an example on how the function should work. You can test the function using `markmyassignment`.

```
> library(aaltobda)
> data("bioassay")
```

```
> density_ratio(alpha_propose = 1.89, alpha_previous = 0.374,
                beta_propose = 24.76, beta_previous = 20.04,
                x = bioassay$x, y = bioassay$y, n = bioassay$n)
```

```
[1] 1.187524
```

```
> density_ratio(alpha_propose = 0.374, alpha_previous = 1.89,
                beta_propose = 20.04, beta_previous = 24.76,
                x = bioassay$x, y = bioassay$y, n = bioassay$n)
```

```
[1] 0.8420882
```

   Hint! Compute with log-densities. Reasons are explained on page 261 of BDA3 and lecture video 4.1. Remember that $p_1/p_0 = \exp(\log(p_1) - \log(p_0))$. For your convenience we have provided functions that will evaluate the log-likelihood for given α and β (see `bioassaylp()` in the `aaltobda` package). Notice that you still need to add the prior yourself and remember the unnormalized log posterior is simply the sum of log-likelihood and log-prior. For evaluating the log of the Gaussian prior you can use the function `dmvnorm` from package `aaltobda`. It can be worthwhile to look up your implementation of `p_log_posterior()` in Assignment 4.

   b) Now implement a function called `Metropolis_bioassay()` which implements the Metropolis algorithm using the `density_ratio()`.

   Hint! Use a simple (normal) proposal distribution. Example proposals are $\alpha^* \sim N(\alpha_{t-1}, \sigma = 1)$ and $\beta^* \sim N(\beta_{t-1}, \sigma = 5)$. There is no need to try to find optimal proposal but test some different values for the jump scale (σ). Remember to report the one you used. Efficient proposals are dicussed in BDA3 p. 295–297 (not part of the course). In real-life a pre-run could be made with an automatic adaptive control to adapt the proposal distribution.

2. Include in the report the following:

a) Describe in your own words in one paragraph the basic idea of the Metropolis algorithm (see BDA3 Section 11.2, and lecture video 5.1).

b) The proposal distribution (related to *jumping rule*) you used. Describe briefly in words how you chose the final proposal distribution you used for the reported results.

c) The initial points of your Metropolis chains (or the explicit mechanism for generating them).

d) Report the chain length or the number of iterations for each chain. Run the simulations long enough for approximate convergence (see BDA Section 11.4, and lecture 5.2).

e) Report the warm-up length (see BDA Section 11.4, and lecture 5.2).

f) The number of Metropolis chains used. It is important that multiple Metropolis chains are run for evaluating convergence (see BDA Section 11.4, and lecture 5.2)..

g) Plot all chains for α in a single line-plot. Overlapping the chains in this way helps in visually assessing whether chains have converged or not.

h) Do the same for β.

3. In complex scenarios, visual assessment is not sufficient and $\widehat{R}$ is a more robust indicator of convergence of the Markov chains. Use $\widehat{R}$ for convergence analysis. You can either use Eq. (11.4) in BDA3 or the more recent version described **here**. You should specify which $\widehat{R}$ you used. In R the best choice is to use function `Rhat` from package `rstan`. Remember to remove the warm-up samples before computing $\widehat{R}$. Report the $\widehat{R}$ values for α and β separately. Report the values for the proposal distribution you finally used.

   a) Describe briefly in your own words the basic idea of $\widehat{R}$ and how to to interpret the obtained $\widehat{R}$ values.

   b) Tell whether you obtained good $\widehat{R}$ with first try, or whether you needed to run more iterations or how did you modify the proposal distribution.

4. Plot the draws for α and β (scatter plot) and include this plot in your report. You can compare the results to Figure 3.3b in BDA3 to verify that your code gives sensible results. Notice though that the results in Figure 3.3b are generated from posterior with a uniform prior, so even when if your algorithm works perfectly, the results will look slightly different (although fairly similar).