Uppsala University

Bayesian Statistics and 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 many tutorials, videos and introductions to R and R-Studio online. You can find some initial hints from **RStudio Education pages**.
- When working with R, we recommend writing the report using R markdown and the provided **R markdown template**. The template 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 you should use the same instructions for formatting. These instructions are also available in **the PDF produced from the R markdown template**.
- We supply a Google Colab notebook that you can also use for the assignments. We have included the installation of all necessary R packages; hence, this can be an alternative to using your own local computer. You can find the notebook **here**. You can also open the notebook in Colab **here**.
- Report all results in a single and *anonymous* pdf. Note that no other formats are allowed.
- The course has its own R package `bsda` 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("MansMeg/BSDA", subdir = "rpackage", upgrade="never")`

- Many of the exercises can be checked automatically using the R package `markmyassignment`. you can find information on how to install and use the package **here**. There is no need to include `markmyassignment` results in the report.
- You can find common questions and answers regarding the installation and technical problems in Frequently Asked Questions (FAQ).
- You can find deadlines and information on how to turn in the assignments in Studium.
- You are allowed to discuss assignments with your friends, but it is not permitted to copy solutions directly from other students or the internet. Try to solve the actual assignment problems with your code and explanations. Do not share your answers publicly. We compare the answers with the "urkund" system. We will report all suspected plagiarism.
- 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 **here**.

- It is *mandatory* to include the following parts in all assignments (these are included already in the template):

  1. Time used for reading: How long time took the reading assignment (in hours)
  2. Time used for the assignment: How long time took the basic assignment (in hours)
  3. Good with assignment: Write one-two sentences of what you liked with the assignment/what we should keep for next year.
  4. Things to improve in the assignment: Write one-two sentences of what you think can be improved in the assignment. Can something be clarified further? Did you get stuck on stuff unrelated to the content of the assignment etc.

- You can find information on how each assignment will be graded and how points are assigned **here**. **Note!** This grading information can change during the course, for example, if we find errors or inconsistencies or do additions to the assignments. Please feel free to comment on these grading instructions, ideally before turning in your assignment, if you think something is missing or is incorrect.

- To pass (G) the assignment, you need 70% of the total points. To pass with distinction (VG), you need 90% of the total points. See the grading information on the point allocations for each assignment.

- On cheating: You are not allowed to show your assignments (text or code) to anyone. Only discuss the assignments with your fellow students. The student that show their assignment to anyone else could also be considered to cheat. Similarly, on zoom labs, only screen share when you are in a separate zoom room with teaching assistants. You are not allowed to use large language models, such as ChatGPT, to write assignments.

- All mathematics need to be done in digital form to simplify grading and commenting. Hence, it is not allowed to write math on paper and add an image in the assignment. If you have difficulties to write math in latex, see https://editor.codecogs.com/.

Information on this assignment

This assignment is related to Chapters 10 and 11.

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

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 $E(\mu) = 1.2$.

When computing the $\hat{R}$ diagnostics, you only need to include two decimals.

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

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

Don't include `markmyassignment` results in the report.

Generalized linear model: Bioassay with Metropolis

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.

1. Implement the Metropolis algorithm as an R function for the bioassay data. Use the Gaussian prior

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \sim \mathrm{N}\left(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right), \qquad \text{where} \quad \boldsymbol{\mu}_0 = \begin{bmatrix}0\\ 10\end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_0 = \begin{bmatrix}2^2 & 12\\ 12 & 10^2\end{bmatrix}.$$

   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(bsda)
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] 1.305179
```

```
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] 0.7661784
```

   Hint! Compute with log-densities. Reasons are explained on page 261 of BDA3. 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 `bsda` 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 `bsda`.

   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).

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).

e) Report the warm-up length (see BDA Section 11.4).

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. If you have a large sample, make sure that the plot is readable by decreasing opacity or only plotting a random subset of the samples. 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).