Bayesian Data Analysis - Assignment 4

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 **here**.

- You can write the report with your preferred software, but the outline of the report should follow the instruction in the R markdown template that can be found **here**.

- Report all results in a single, **anonymous** *.pdf -file and return it to **peer-grade.io**.

- The course has its own R package with data and functionality to simplify coding. To install the package just run the following:

  1. `install.packages("remotes")`
  2. `remotes::install_github("avehtari/BDA_course_Aalto",`
     `subdir = "rpackage")`

- 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**.

- Additional self study exercises and solutions for each chapter in BDA3 can be found **here**.

- We collect common questions regarding installation and technical problems in a course Frequently Asked Questions (FAQ). This can be found **here**.

- If you have any suggestions or improvements to the course material, please feel free to create an issue or submit a pull request to the public repository!!

Information on this assignment

This exercise is related to Chapters 3 and 10. The maximum amount of points from this assignment is 6.

Reading instructions: Chapters 3 and 10 in BDA3, see reading instructions **here** and **here**

Grading instructions: The grading will be done in peergrade. All grading questions and evaluations for exercise 4 can be found **here**

Reporting accuracy: As many significant digits as justified by the Monte Carlo error and posterior accuracy.

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

```
> library(markmyassignment)
> exercise_path <-
+    "https://github.com/avehtari/BDA_course_Aalto/blob/master/exercises/tests/ex4.yml"
> set_assignment(exercise_path)
> # To check your code/functions, just run
> mark_my_assignment()
```

Bioassay model. In this exercise, you will use a dose-response relation model that is used in Section 3.7 of the course book and in the chapter reading instructions **here**. The used likelihood is the same, but instead of uniform priors, we will use a bivariate normal distribution as the joint prior distribution of the parameters α and β.

a) In the prior distribution for (α, β), the marginal distributions are $\alpha \sim N(0, 2^2)$ and $\beta \sim N(10, 10^2)$, and the correlation between them is $\text{corr}(\alpha, \beta) = 0.5$. Report the full mean and covariance of the bivariate normal distribution.

   – **Hint!** The mean and covariance of the bivariate normal distribution are a length–2 vector and a 2×2 matrix. The elements of the covariance matrix can be computed using the relation of correlation and covariance.

b) You are given 4000 independent draws from the posterior distribution of the model. Load the draws with `data("bioassay_posterior")`. Report the mean as well as 5 % and 95 % quantiles separately for both α and β. Report also the Monte Carlo standard errors (MCSEs) for the mean and quantile estimates. Choose the amount of digits you report for the mean and quantiles based on the MCSEs such that you leave out digits where MCSE is nonzero (Example: if posterior mean is 2.345678 and MCSE is 0.0012345, report two digits after the decimal sign, taking into account the usual rounding rule, so you would report 2.35. Further digits do not contain useful information due to the Monte Carlo uncertainty.).

   – **Note!** The answer is graded as correct only if the number of digits reported is correct! The number of significant digits can be different for the mean and quantile estimates.

   – **Hint!** Quantiles can be computed with the `quantile` function. With S draws, the MCSE for $\text{E}[\theta]$ is $\sqrt{\text{Var}[\theta]/S}$. MCSE for the quantile estimates can be computed with the `mcse_quantile` function from the `aaltobda` package.

Importance sampling. Now we discard our posterior draws and switch to importance sampling.

c) Implement a function for computing the log importance ratios (log importance weights) when the importance sampling **target distribution** is the posterior distribution, and the **proposal distribution** is the prior distribution from a). Below is a test example, the functions can also be tested with `markmyassignment`.

   – **Note!** The values below are *only* a test case. In this c) part, you only need to report the source code of your function, as it will be needed in later parts.

   – **Hints!** Use the function `rmvnorm` from the `aaltobda` package for sampling. Non-log importance ratios are given by equation (10.3) in the course book. The fact that our proposal distribution is the same as the prior distribution makes this task easier. The **logarithm** of the likelihood can be computed with the `bioassaylp` function from the `aaltobda` package. The data required for the likelihood can be loaded with `data("bioassay")`.

```
> alpha <- c(1.896, -3.6,  0.374, 0.964, -3.123, -1.581)
> beta <- c(24.76, 20.04, 6.15, 18.65, 8.16, 17.4)
> round(log_importance_weights(alpha, beta),2)
```

```
[1]   -8.95 -23.47  -6.02  -8.13 -16.61 -14.57
```

d) Implement a function for computing normalized importance ratios from the un-normalized log ratios in c). In other words, exponentiate the log ratios and scale them such that they sum to one. Below is a test example, the functions can also be tested with `markmyassignment`.

- **Note!** The values below are *only* a test case. In this d) part, you only need to report the source code of your function, as it will be needed in later parts.

```
> alpha

[1]   1.896 -3.600   0.374   0.964 -3.123 -1.581

> beta

[1] 24.76 20.04   6.15 18.65   8.16 17.40

> round(normalized_importance_weights(alpha = alpha, beta = beta),3)

[1] 0.045 0.000 0.852 0.103 0.000 0.000
```

e) Sample 4000 draws of α and β from the prior distribution from a). Compute and plot a histogram of the 4000 normalized importance ratios. Use the functions you implemented in c) and d).

f) Using the importance ratios, compute the importance sampling effective sample size S_{eff} and report it.

- **Note!** The values below are *only* a test case, you need to use 4000 draws for `alpha` and `beta` in the final report.

```
> alpha

[1]   1.896 -3.600   0.374   0.964 -3.123 -1.581

> beta

[1] 24.76 20.04   6.15 18.65   8.16 17.40

> round(S_eff(alpha = alpha, beta = beta),3)

[1] 1.354
```

- **Hint!** Equation (10.4) in the course book.
- **Note!** *BDA3 1st (2013) and 2nd (2014) printing have an error for $\tilde{w}(\theta^s)$ used in the effective sample size equation (10.4). The normalized weights equation should not have the multiplier S (the normalized weights should sum to one). Errata for the book* http://www.stat.columbia.edu/~gelman/book/errata_bda3.txt *. The later printings and slides have the correct equation.*

g) Explain in your own words what the importance sampling effective sample size represents. Also explain how the effective sample size is seen in the histogram of the weights that you plotted in e).

h) Implement a function for computing the posterior mean using importance sampling, and compute the mean using your 4000 draws. Below is an example how the function would work with the example values for **alpha** and **beta** above. Report the means for alpha and beta, and also the Monte Carlo standard errors (MCSEs) for the mean estimates. Report the number of digits for the means based on the MCSEs.

   - **Note!** The values below are *only* a test case, you need to use 4000 draws for **alpha** and **beta** in the final report.
   - **Hint!** Use the same equation for the MCSE of $E[\theta]$ as earlier ($\sqrt{\text{Var}[\theta]/S}$), but now replace S with S_{eff}. To compute $\text{Var}[\theta]$ with importance sampling, use the identity $\text{Var}[\theta] = E[\theta^2] - E[\theta]^2$.

```
> alpha

[1]  1.896 -3.600  0.374  0.964 -3.123 -1.581

> beta

[1] 24.76 20.04  6.15 18.65  8.16 17.40

> round(posterior_mean(alpha = alpha, beta = beta),3)

[1] 0.503 8.275
```