BDA - Assignment 4

Anonymous

Contents

Loading package.

```
library(aaltobda)
```

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. 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 $corr(\alpha, \beta) = 0.6$.

```
rho <- 0.6
mu_a <- 0; s_a <- 2
mu_b <- 10; s_b <- 10
```

Mean (vector of two values):

```
# Mean
mu <- c(mu_a, mu_b)
mu
```

```
## [1]  0 10
```

Mean: $mean = \begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix} = \begin{pmatrix} 0 \\ 10 \end{pmatrix}$

Covariance (two by two matrix):

```
# Covariance matrix
cov <- matrix(c(s_a^2, s_a*s_b*rho, s_a*s_b*rho, s_b^2), 2)
cov
```

```
##      [,1] [,2]
## [1,]    4   12
## [2,]   12  100
```

Covariance: $cov = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} = \begin{pmatrix} 4 & 12 \\ 12 & 100 \end{pmatrix}$

b)

You are given 4000 independent draws from the posterior distribution of the model. Load the draws with .

```
data("bioassay_posterior")
```

Report the mean separately for both α and β:

```
c(mean(bioassay_posterior[,1]), mean(bioassay_posterior[,2]))
```

```
## [1]  0.9852263 10.5964813
```

MCSEs for the means:

```
S <- dim(bioassay_posterior)[1] # number of values
MCSE_alpha <- sqrt(var(bioassay_posterior[,1]) / S)
MCSE_beta <- sqrt(var(bioassay_posterior[,2]) / S)
c(MCSE_alpha, MCSE_beta)
```

```
## [1] 0.01482435 0.07560016
```

After taking the MCSEs into account, we report the means:

Mean of α: 1.0

Mean of β: 10.6

Report the 95 % quantiles separately for both α and β.

```r
quantile(bioassay_posterior[,1], c(0.05, 0.95)) # alpha
```

```
##         5%        95%
## -0.4675914  2.6102028
```

```r
quantile(bioassay_posterior[,2], c(0.05, 0.95)) # beta
```

```
##        5%       95%
##  3.991403 19.340365
```

MCSEs for the quantiles:

```r
# alpha, 5% and 95%, function from the aalto_dba package
c(mcse_quantile(bioassay_posterior[,1], 0.05), mcse_quantile(bioassay_posterior[,1], 0.95))
```

```
## $mcse
## [1] 0.02600412
##
## $mcse
## [1] 0.04206342
```

```r
# beta, 5% and 95%, function from the aalto_dba package
c(mcse_quantile(bioassay_posterior[,2], 0.05), mcse_quantile(bioassay_posterior[,2], 0.95))
```

```
## $mcse
## [1] 0.07043125
##
## $mcse
## [1] 0.2412129
```

After taking the MCSEs into account, we report the quantiles:

5% and 95% quantiles of α: -0.5 and 2.6

5% and 95% quantiles of β: 4.0 and 19

Explain in words what Monte Carlo standard error means and how you decided the number of digits to show:

Monte Carlo is an estimate of inaccuracy in Monte Carlo methods, that refers to the uncertainty due to the finiteness of the number of simulation draws. The higher the number of draws, the more confident a received value is. When rounding according to the MCSE, we discard the non-significant digits, as they carry no useful information. I decided the number of digits to show based on the examples given in the assignment, which says that when MCSE is 0.0012345, I report two digits after the comma sign of my given measure.

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). Explain in words why it's better to compute log ratios instead of ratios.

```r
data("bioassay")

# Bioassay data
x <- bioassay[,"x"]
y <- bioassay[,"y"]
n <- bioassay[,"n"]

log_importance_weights <- function(alpha, beta) {
  # function from the aalto_dba package
  return(bioassaylp(alpha, beta, x, y, n))
}
```

d)

Implement a function for computing normalized importance ratios from the unnormalized log ratios in c). In other words, exponentiate the log ratios and scale them such that they sum to one. Explain in words what is the effect of exponentiating and scaling so that sum is one.

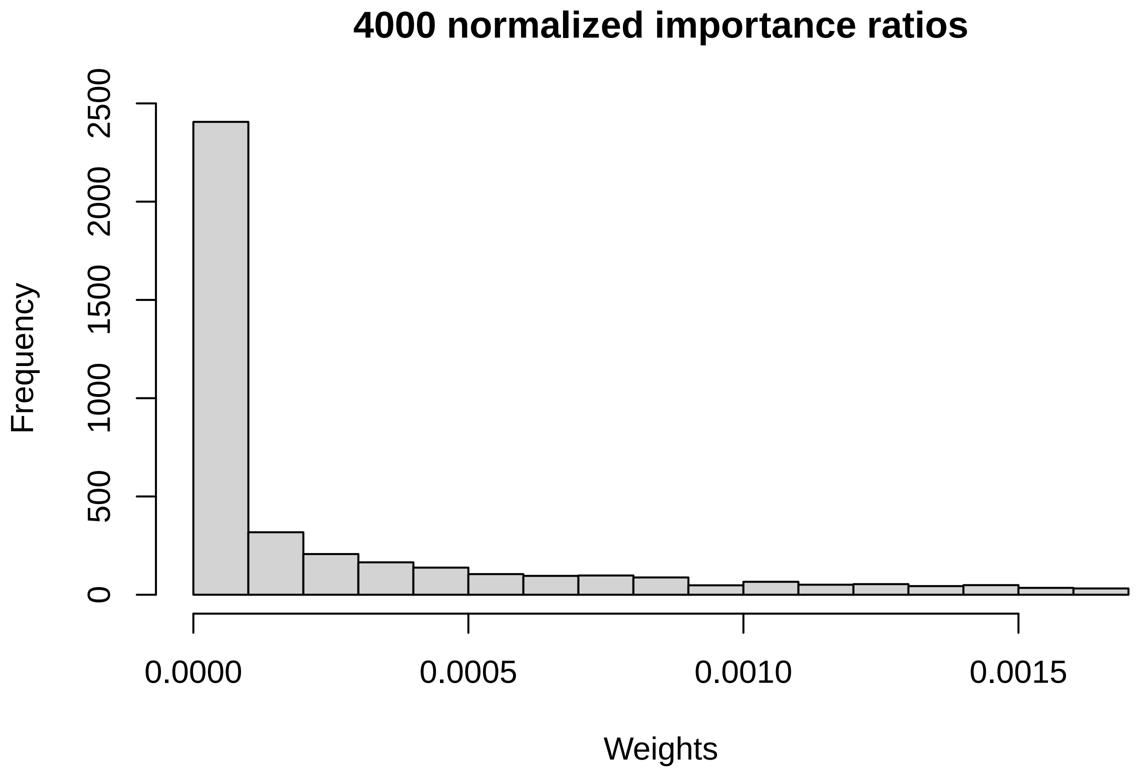
```r
normalized_importance_weights <- function(alpha, beta) {
  log_ratio <- log_importance_weights(alpha, beta)
  return (exp(log_ratio) / sum(exp(log_ratio))) # exponentiate and scale
}
```

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.

```r
# Sample 4000 draws from the prior distribution
sample <- rmvnorm(4000, mu, cov)

# Plot a histogram
hist(normalized_importance_weights(sample[,1], sample[,2]), breaks=20,
     main="4000 normalized importance ratios", xlab="Weights")
```



f)

Using the importance ratios, compute the importance sampling effective sample size $\S_eff$ and report it.

```
S_eff <- function(alpha, beta) {
  # Equation 10.4 from the course book
  return (1 / sum(normalized_importance_weights(alpha, beta)^2))
}

size_eff <- S_eff(alpha = sample[,1], beta = sample[,2])

round(size_eff, 3)
```

```
## [1] 1134.843
```

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).

The importance sampling effective sample size (S_eff) represents how accurate the importance sampling is by indicating how large part of the sampling draws affect our what we are trying to calculate. When S_eff is small, a histogram of weights looks like a lot of low values. A large S_eff, with lots of large weights in the histogram, means that the estimate is noisy.

h)

Implement a function for computing the posterior mean using importance sampling, and compute the mean using your 4000 draws. Explain in your own words the computation for importance sampling.

Report the means for α and β:

```r
calculate_posterior_mean <- function(alpha, beta){
  log_ratio = log_importance_weights(alpha, beta)
  alpha_mean = sum(alpha*exp(log_ratio))/sum(exp(log_ratio))
  beta_mean = sum(beta*exp(log_ratio))/sum(exp(log_ratio))
  return(c(alpha_mean, beta_mean))
}

calculate_posterior_mean(sample[,1], sample[,2])
```

```
## [1]  0.986493 10.676481
```

Report the Monte Carlo standard errors (MCSEs) for the means for α and β:

```r
mean_MCSEs <- function(alpha, beta, size_eff){
  log_ratio = log_importance_weights(alpha, beta)

  # Compute alpha_var with importance sampling
  ea2 = sum((alpha^2)*exp(log_ratio))/sum(exp(log_ratio))
  e2a = (sum(alpha*exp(log_ratio))/sum(exp(log_ratio)))^2
  alpha_var = ea2-e2a

  # Compute beta_var with importance sampling
  eb2 = sum((beta^2)*exp(log_ratio))/sum(exp(log_ratio))
  e2b = (sum(beta*exp(log_ratio))/sum(exp(log_ratio)))^2
  beta_var = eb2-e2b

  # MCSE with the formula sqrt(var/S_eff)
  alpha_mean_MCSE = sqrt(alpha_var/size_eff)
  beta_mean_MCSE = sqrt(beta_var/size_eff)

  return(c(alpha_mean_MCSE, beta_mean_MCSE))
}

mean_MCSEs(sample[,1], sample[,2], size_eff)
```

```
## [1] 0.02730425 0.13837668
```

After taking the MCSEs into account, we report the means:

Mean of α: 1.0

Mean of β: 11