BDA - Assignment 4

Anonymous

Table of Contents

Exercise 1 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 $(\alpha;\beta)$, the marginal distributions are $\alpha \sim N(0;2^2)$ and $\beta \sim N(10;10^2)$, and the correlation between them is $corr(\alpha;\beta) = 0.6$. Report the mean (vector of two values) and covariance (two by two matrix) 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.

Suppose that the random variable resulting from the bivariate normal distribution is X. Because X is composed of α and β marginal distribution, X is a length-2 vector:
$X = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}$, where $\mu_\alpha = 0$, $\mu_\beta = 10$, $\sigma_\alpha = 2$, $\sigma_\beta = 10$, and $\rho = 0.6$
The mean of the bivariate normal distribution is:

$\mu_X = E[X] = \begin{bmatrix}\mu_\alpha \\ \mu_\beta\end{bmatrix} = \begin{bmatrix}0 \\ 10\end{bmatrix}$. (answer)

The covariance of the bivariate normal distribution is:

$$cov_X = E[(\alpha - \mu_\alpha)(\beta - \mu_\beta)] = \begin{bmatrix}Var(\alpha) & Cov(\alpha\beta) \\ Cov(\alpha\beta) & Var(\beta)\end{bmatrix} = \begin{bmatrix}\sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2\end{bmatrix} = \begin{bmatrix}4 & 12 \\ 12 & 100\end{bmatrix}$$

(answer)

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. Report as many digits for the mean and quantiles as the MCSEs allow. In other words, 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.). Explain in words what does Monte Carlo standard error mean and how you decided the number of digits to show.

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. In some other cases, the number of digits reported can be less than MCSE allows for practical reasons.

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

```
data("bioassay_posterior")
draws <- bioassay_posterior

S = length(draws$alpha)
mean_alpha <- mean(draws$alpha)
mean_beta <- mean(draws$beta)
cat("The mean for alpha is",mean_alpha,"\n")

## The mean for alpha is 0.9852263

cat("The mean for beta is",mean_beta,"\n")

## The mean for beta is 10.59648

cat("5% and 95% quantile for alpha is\n")

## 5% and 95% quantile for alpha is
```

```
quantile(draws$alpha, c(0.05, 0.95))

##         5%        95%
## -0.4675914  2.6102028

cat("5% and 95% quantile for beta is\n")

## 5% and 95% quantile for beta is

print(quantile(draws$beta, c(0.05, 0.95)))

##        5%       95%
##  3.991403 19.340365

##############################################################################
##################
S = length(draws$alpha)
mcse_mean_alpha <- sqrt(var(draws$alpha)/S)
mcse_mean_beta <- sqrt(var(draws$beta)/S)
cat("The MCSE for alpha mean is",mcse_mean_alpha,"\n")

## The MCSE for alpha mean is 0.01482435

cat("The MCSE for beta mean is",mcse_mean_beta,"\n")

## The MCSE for beta mean is 0.07560016

mcse_low_alpha <- mcse_quantile(draws$alpha, 0.05)[1, 1]
mcse_high_alpha <- mcse_quantile(draws$alpha, 0.95)[1, 1]
mcse_low_beta <- mcse_quantile(draws$beta, 0.05)[1, 1]
mcse_high_beta <- mcse_quantile(draws$beta, 0.95)[1, 1]
cat("5% and 95% MCSE for alpha quantile is (", mcse_low_alpha, ",", mcse_high
_alpha,")\n")

## 5% and 95% MCSE for alpha quantile is ( 0.02600412 , 0.04206342 )

cat("5% and 95% MCSE for beta quantile is (", mcse_low_beta, ",", mcse_high_b
eta,")\n")

## 5% and 95% MCSE for beta quantile is ( 0.07043125 , 0.2412129 )
```

- What is Monte Carlo standard error (MCSE)

It is an estimate of the inaccuracy of Monte Carlo samples, usually regarding the expectation of posterior samples.The MCSE approaches zero as the number of independent posterior samples approaches infinity. MCSE is essentially a standard deviation around the posterior mean of the samples, $E(\theta)$, due to uncertainty associated with using Monte Carlo methods. The acceptable size of the MCSE depends on the acceptable uncertainty associated around the marginal posterior mean, $E(\theta)$, and the goal of inference. It has been argued that MCSE is unimportant when the goal of inference is θ rather than $E(\theta)$.

- How I decided the number of digits to show

I followed the exercise instructions, in which I rounded up the mean up to n-decimals, where n is the number of zero decimals in MCSE.

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

cat("5% and 95% quantile for beta is\n")

## 5% and 95% quantile for beta is

print(quantile(draws$beta, c(0.05, 0.95)))

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

The unadjusted means are $\mu_\alpha = 0.985$ and $\mu_\beta = 10.596$

The unadjusted alpha quantiles are (-0.4675914, 2.6102028)

The unadjusted alpha quantiles are (3.991403, 19.340365)

- The MCSE values are:

MCSE for alpha mean is 0.01482435

MCSE for beta mean is 0.07560016

5% and 95% MCSE for alpha quantile is (0.02600412 , 0.04206342)

5% and 95% MCSE for beta quantile is (0.07043125 , 0.2412129)

- From the MCSE, we can derive the significant digits for the means and the quantiles

=> The mean becomes $\mu_\alpha = 1$ and $\mu_\beta = 10.6$

=> The 5% and 95% for alpha quantile becomes (-0.5 , 2.6)

=> The 5% and 95% for beta quantile becomes (4 , 19)

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

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.

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").

```
data("bioassay")
bioassay <- bioassay
# print(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)

log_importance_weights <- function(alpha, beta){
  return(bioassaylp(alpha, beta, bioassay$x, bioassay$y, bioassay$n))
}

round(log_importance_weights(alpha, beta),2)

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

# mark_my_assignment()
```

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. 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.

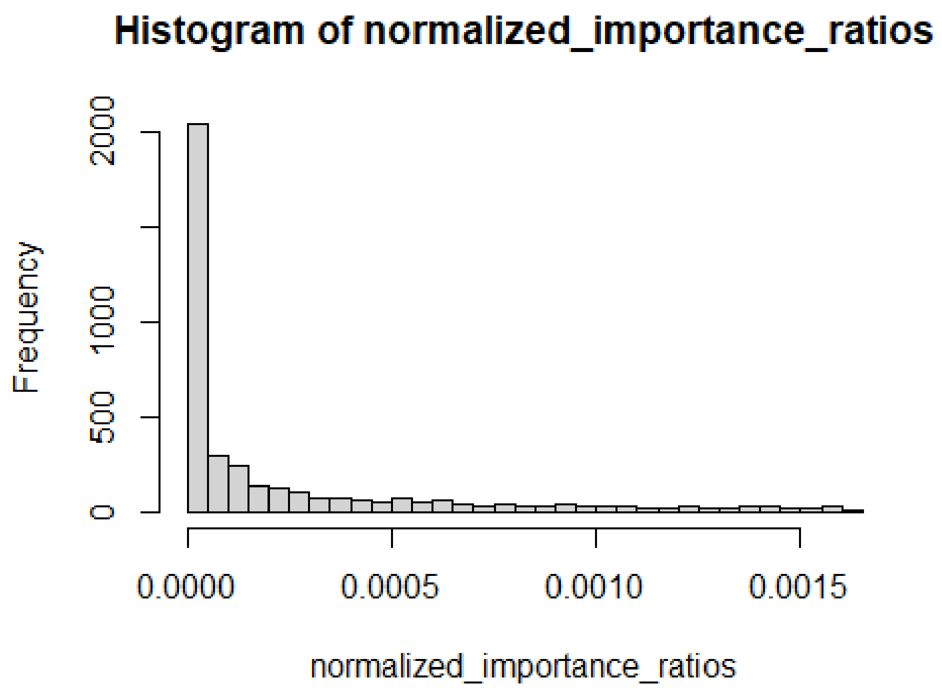
```
alpha <- c(1.896, -3.6, 0.374, 0.964, -3.123, -1.581)
beta <- c(24.76, 20.04, 6.15, 18.65, 8.16, 17.4)

normalized_importance_weights <- function(alpha, beta){
  unnormalized <- bioassaylp(alpha, beta, bioassay$x, bioassay$y, bioassay$n)
  exponentiated <- exp(unnormalized)
  ratio <- 1/sum(exponentiated)
  normalized <- exponentiated * ratio
  return(normalized)
}

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).

Hints! Use the function rmvnorm from the aaltobda package for sampling.

```
prior_mean_vector = matrix(c(0, 10), nrow=1, ncol=2, byrow = TRUE)
prior_cov_matrix = matrix(c(4, 12, 12, 100), nrow=2, ncol=2, byrow = TRUE)
nSamples <- 4000
generatedSamples <- rmvnorm(nSamples, prior_mean_vector, prior_cov_matrix)
generatedSamples <- data.matrix(generatedSamples)
alphaSamples <- generatedSamples[, 1]
betaSamples <- generatedSamples[, 2]
normalized_importance_ratios = normalized_importance_weights(alphaSamples, be
taSamples)
hist(normalized_importance_ratios, breaks = 50)
```



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.
Hint! Equation (10.4) in the course book.

```r
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)

S_eff <- function(alpha, beta){
  normalized_weights = normalized_importance_weights(alpha, beta)
  squared_normalized_weights <- normalized_weights^2
  effective_sample_size <- 1/sum(squared_normalized_weights)
  return(effective_sample_size)
}

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

## [1] 1.354

# mark_my_assignment()
round(S_eff(alpha = alphaSamples, beta = betaSamples),3)

## [1] 1149.004
```

The importance sampling effective sample size for 4000 samples is 1149.004

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).

- What the importance sampling effective sample size represents

The effective sample size (ESS) is an estimate of the sample size required to achieve the same level of precision as if that sample was a simple random sample. For example, let's call q as the approximated distribution of p. If we use $n = 1000$ samples $X_i \sim q$ and obtain an ESS of 100, then this indicates that the quality of our estimate is about the same as if we would have used 100 direct samples from $X_i \sim p$. It can be used after or during importance sampling to provide a quantitative measure of the quality of the estimated mean.

- How the effective sample size is seen in the histogram of the weights

The histograms above shows the inverse relationship between the frequency and the weights. In other words, the heavier the weights are, the more unlikely they will occur when the initial sample size (4000) is fixed. As such, ESS should occur at the mode of the histogram in E. The ESS is defined by the inverse of the sum of the importance weights. In other words, the sum of weights that corresponds to $ESS = 1149$ is $S_w = \frac{1}{1149} = 0.00087$, which happens to be at the mode of the histogram above. Therefore, ESS corresponds to most frequently occuring weights.

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. 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 ($\sqrt{Var[\theta]/S}$), but now replace S with S_{eff}. To compute $Var[\theta]$ with importance sampling, use the identity $Var[\theta] = E[\theta^2] - E[\theta]^2$.

- Explaination of the computation for importance sampling:

First, we obtain the 4000 prior samples from part (e). Then, the posterior can be calculated by important sampling with the formula: $E\big(h(\theta|y)\big) = \frac{\frac{1}{S}\sum_{S=1}^{S} h(\theta^S)w(\theta^S)}{\frac{1}{S}\sum_{S=1}^{S} w(\theta^S)}$, where the factors $w(\theta^s) = \frac{q(\theta^s|y)}{g(\theta^s)}$ is the normalized importance weights.

```
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)

posterior_mean <- function(alpha, beta){
  normalized_weights = normalized_importance_weights(alpha, beta)
  mean_weights = mean(normalized_weights)

  posterior_alpha_dist = alpha * normalized_weights
  posterior_alpha_mean = mean(posterior_alpha_dist)/mean_weights
  posterior_beta_dist = beta * normalized_weights
  posterior_beta_mean = mean(posterior_beta_dist)/mean_weights
  return(c(posterior_alpha_mean, posterior_beta_mean))
}

# round(posterior_mean(alpha = alpha, beta = beta),3)
## [1] 0.503 8.275.
round(posterior_mean(alpha = alphaSamples, beta = betaSamples),3)

## [1]  0.963 10.470

# mark_my_assignment()

posterior_mean_MCSE <- function(alpha, beta){
  normalized_weights = normalized_importance_weights(alpha, beta)
  mean_weights = mean(normalized_weights)

  posterior_alpha_dist = alpha * normalized_weights
  posterior_alpha_mean = mean(posterior_alpha_dist)/mean_weights
  posterior_beta_dist = beta * normalized_weights
  posterior_beta_mean = mean(posterior_beta_dist)/mean_weights

  posterior_alpha_squared_dist = alpha^2 * normalized_weights
  posterior_alpha_squared_mean = mean(posterior_alpha_squared_dist)/mean_weig
hts
  posterior_beta_squared_dist = beta^2 * normalized_weights
  posterior_beta_squared_mean = mean(posterior_beta_squared_dist)/mean_weight
s

  var_alpha = posterior_alpha_squared_mean - posterior_alpha_mean ^ 2
  var_beta = posterior_beta_squared_mean - posterior_beta_mean ^ 2

  sample_effective_size = S_eff(alpha, beta)
  mean_MCSE_alpha = sqrt(var_alpha/sample_effective_size)
  mean_MCSE_beta = sqrt(var_beta/sample_effective_size)
  return(c(mean_MCSE_alpha, mean_MCSE_beta))
}

round(posterior_mean_MCSE(alpha = alphaSamples, beta = betaSamples),3)

## [1] 0.027 0.139
```

The posterior means are $\mu_\alpha = 0.963$ and $\mu_\beta = 10.470$

The MCSE for the posterior alpha mean is 0.027 and for posterior beta mean is 0.139

- ⇨ The MCSE-adjusted posterior mean becomes $\mu_\alpha = 1.0$ and $\mu_\beta = 10.0$