BDA - Assignment 4

Anonymous

Contents

Bioassay model **1**
a) . . . 1
b) . . . 1

Importance sampling **2**
c) . . . 3
d) . . . 3
e) . . . 3
f) . . . 4
g) . . . 5
h) . . . 5

Import some useful libraries

```
library(aaltobda)
library(ggplot2)
library(grid)
library(gridExtra)
```

Bioassay model

Here, we study a dose-response relation model with 2 parameters (α and β) that is thoroughly explained in Section 3.7 in BDA3. We will use the same likelihood distribution and assume a bivariate normal distribution as the joint prior distribution.

a)

The marginal distributions for the priors of each parameter are $\alpha = N(0, 2^2)$ and $\beta = N(10, 10^2)$. The correlation between them is $\text{corr}(\alpha, \beta) = 0.6$. In the case of a bivariate normal distribution, we know that

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix},$$

where ρ is the correlation between α and β (i.e. $\rho = 0.6$). Therefore,

$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 10 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 2^2 & 0.6 * 2 * 10 \\ 0.6 * 2 * 10 & 10^2 \end{pmatrix} \rightarrow \boldsymbol{\mu} = \begin{pmatrix} 0 \\ 10 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 4 & 12 \\ 12 & 100 \end{pmatrix},$$

b)

We have 4000 independent draws from the posterior distribution.

```
S <- 4000  # number of draws
data("bioassay_posterior")
tail(bioassay_posterior)
```

```
##            alpha      beta
## 3995  0.06212579 10.436902
## 3996  2.16847288 10.124672
## 3997 -0.76451111  5.022993
## 3998  0.93077665  6.358671
## 3999  1.85952996  7.076051
## 4000  0.14049394  3.004310
```

The posterior means and their MCSEs are computed as

```
# Means
mu_alpha_post <- mean(bioassay_posterior$alpha)
mu_beta_post <- mean(bioassay_posterior$beta)
# MCSEs
mcse_mu_alpha_post <- sqrt(var(bioassay_posterior$alpha) / S)
mcse_mu_beta_post <- sqrt(var(bioassay_posterior$beta) / S)
```

The MCSE for $E(\alpha)$ is 0.02, thus we can report the mean of the posterior of α as $E(\alpha) = 1.0$. The MCSE for $E(\beta)$ is 0.08, thus we can report the mean of the posterior of β as $E(\beta) = 10.6$.

The 5% and the 95% quantiles are computed as

```
# Quantiles
quantile_alpha_post <-
  quantile(x = bioassay_posterior$alpha, probs = c(0.05, 0.95))
quantile_beta_post <-
  quantile(x = bioassay_posterior$beta, probs = c(0.05, 0.95))
# MCSEs of quantiles of alpha
mcse_quantile_alpha_post_lower <-
  mcse_quantile(draws = bioassay_posterior$alpha, prob = 0.05)
mcse_quantile_alpha_post_upper <-
  mcse_quantile(draws = bioassay_posterior$alpha, prob = 0.95)
# MCSEs of quantiles of beta
mcse_quantile_beta_post_lower <-
  mcse_quantile(draws = bioassay_posterior$beta, prob = 0.05)
mcse_quantile_beta_post_upper <-
  mcse_quantile(draws = bioassay_posterior$beta, prob = 0.95)
```

Given that the MCSE for the 5% and the 95% quantile of α are 0.03 and 0.04, respectively. We can report the quantiles to be $[-0.5, 2.6]$. Given that the MCSE for the 5% and the 95% quantile of β are 0.07 and 0.2, respectively. We can report the quantiles to be $[4.0, 19]$.

The Monte Carlo standard error describes the uncertainty contributed by having a finite number of simulation draws (e.g. $S = 4000$ in this case) to the total standard deviation. Furthermore, to chose the number of digits to show, we looked at the magnitude of the MCSE. We only reported as many digits as the MCSE allowed, that is, as many digits that were unaffected by MCSE.

Importance sampling

From now on, we will use importance sampling and disregard the posterior draws.

c)

Next, we implement a function that computes the log importance ratios for importance sampling given that the target distribution is the posterior and the proposal distribution is the prior described above. The logarithm of the likelihood can be computed using the `log_importance_weights` function that returns the unnormalized log-posterior for the bioassay data, assuming uniform prior.

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

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
```

The reasoning for computing the log ratios instead of ratios is purely computational. By using the logarithm of the value, we can avoid potential issues with underflow (i.e. too small of a value to be represented) or overflow (i.e. too big of a value to be stored in memory) in floating point representation.

d)

Now, we implement a function to exponentiate and normalize the log ratios that are computed by `log_importance_weights`.

```
normalized_importance_weights <- function(alpha, beta) {
  # Compute the log ratios
  log_imp_ratios <- log_importance_weights(alpha, beta)
  # Exponentiate the log ratios
  imp_ratios <- exp(log_imp_ratios)
  # Normalize the ratios (sum of all is 1)
  imp_ratios_normlz <- imp_ratios / sum(imp_ratios)
}
round(normalized_importance_weights(alpha = alpha, beta = beta), 3)
```

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

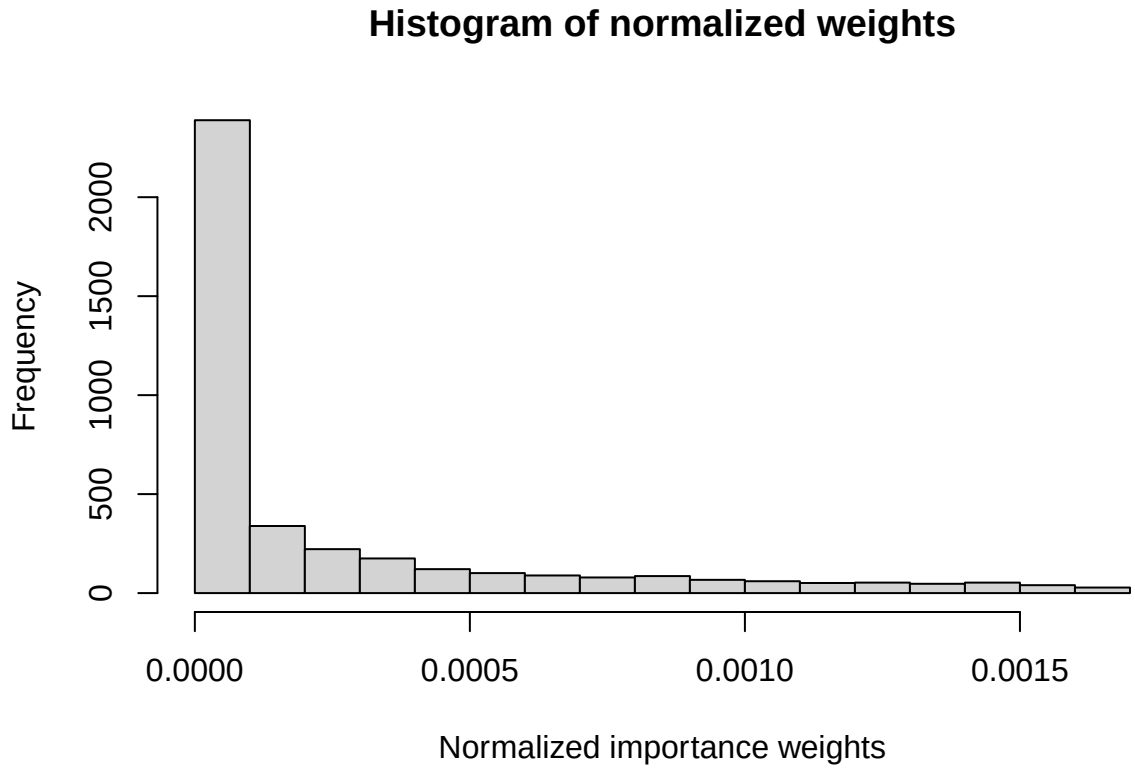
By exponentiating the log ratios, we cancel the logarithm out and we obtain the importance ratios. As we are only interested in the ratios between the weights, the scale of these weights is irrelevant to the computations that will take place afterwards, thus we can safely rescale them from 0 to 1 to avoid exceptionally large values.

e)

Here, we sample 4000 draws of α and β from the prior distribution in a).

```
# Bivariate normal distribution prior
mu_alpha_prior <- 0
sd_alpha_prior <- 2
mu_beta_prior <- 10
sd_beta_prior <- 10
rho <- 0.6

mu <- c(mu_alpha_prior, mu_beta_prior)
sigma <-
  array(c(
    c(sd_alpha_prior ^ 2, rho * sd_alpha_prior * sd_beta_prior),
```

```
    c(rho * sd_alpha_prior * sd_beta_prior, sd_beta_prior ^ 2)
  ),
  dim = c(2, 2))

# Number of draws
S <- 4000
# Random draws from bivariate normal distribution
sample <- data.frame(rmvnorm(S, mu, sigma))
colnames(sample) <- c("alpha", "beta")

# Compute the normalized importance ratios
ratios <- normalized_importance_weights(sample$alpha, sample$beta)

# Plot the histogram
hist(ratios, xlab = "Normalized importance weights", main = "Histogram of normalized weights")
```



As we see most of the normalized ratios are very small.Thus, they have very little effect on $E(h(\theta \mid y))$ (see equation 10.2 in BDA3) and they are not a reason to be concerned.

f)

Using the ratios that we just computed, we can now approximate the importance sampling effective sample size S_{eff}.

```
S_eff <- function(alpha, beta) {
  # Compute the normalized importance ratios
  ratios <- normalized_importance_weights(alpha, beta)
```

```
  return (1 / sum(ratios ^ 2))
}
round(S_eff(alpha = sample$alpha, beta = sample$beta), 3)
```

```
## [1] 1125.608
```

The effective sample size that we achieve is $S_{eff} \approx 1150$ (the number oscillates in the range $[1100, 1200]$). This is nearly 3.5 times less than the size of S.

g)

The effective sample size S_{eff} is a measure of the goodness of our choice for the proposal distribution matching the target distribution. The smaller the weights in equation 10.4 in BDA3, the larger the effective sample size is. This means that our proposed distribution was good because no large adjustments need to be done. In the extreme case of the optimum proposal distribution, the denominator would sum up to one and $S_{eff} = S$.

That is, the effective sample size represents the equivalent number of independent samples drawn from the target distribution. In conclusion, how many i.i.d. samples from the target distribution do our samples equal to.

From the histogram of the normalized importance weights, we can see that most of the weights have very low values. There are less and less ratios for larger weight values. This is not optimal, since we would like for all samples to have approximately the same weight (i.e. have a uniform-looking histogram). Since most samples are given a very small weight and few samples are given relatively larger weights, we can see that our proposal distribution could be chosen better. This is also the reason why our effective sample size is approximately 3.5 times smaller than the drawn sample S.

h)

We can compute the posterior means of α and β using importance sampling by computing the expectations using a sample from an approximation to the target distribution. Since we can not generate draws from $p(\theta \mid y)$ (θ being α and β), we can estimate $E(h(\theta \mid y))$ by drawing S samples from an approximated probability density $g(\theta)$ using

$$\frac{\frac{1}{S}\sum_{s=1}^{S} h(\theta^s)w(\theta^s)}{\frac{1}{S}\sum_{s=1}^{S} w(\theta^s)}.$$

$w(\theta^s)$ are the important ratios. Since we have previously computed the normalized weights $\tilde{w}(\theta^s)$, we know that their sum is 1 and we can reduce the denominator to 1 (given that the $\frac{1}{S}$ term is also canceled). Finally, the equation results in

$$E(h(\theta \mid y)) \approx \sum_{s=1}^{S} h(\theta^s)\tilde{w}(\theta^s)$$

Thus, we can code the function as the following.

```
posterior_mean <- function(alpha, beta) {
  ratios <- normalized_importance_weights(alpha, beta)
  posterior_mean_alpha <- sum(alpha * ratios)
  posterior_mean_beta <- sum(beta * ratios)
  return (c(posterior_mean_alpha, posterior_mean_beta))
}
round(posterior_mean(alpha = alpha, beta = beta),3)
```

```
## [1] 0.503 8.275
```

```r
# Number of draws
S <- 4000
# Random draws from bivariate normal distribution
sample <- data.frame(rmvnorm(S, mu, sigma))
colnames(sample) <- c("alpha", "beta")

# Compute the posterior mean
round(posterior_mean(alpha = sample$alpha, beta = sample$beta),3)
```

```
## [1]  0.984 10.646
```

```r
# Compute the MCSEs of the posterior mean
posterior_mean_mcse <- function(alpha, beta) {
  S_effect <- S_eff(alpha = alpha, beta = beta)
  ratios <- normalized_importance_weights(alpha = alpha, beta = beta)

  # Expectation of the square of theta -> E[theta^2]
  E_sqrd_theta <- c(sum(alpha^2 * ratios), sum(beta^2 * ratios))
  # Expectation of theta squared -> E[theta]^2
  E_theta_sqrd <- posterior_mean(alpha = alpha, beta = beta)^2
  # variance = E[theta^2] - E[theta]^2
  variance <- E_sqrd_theta - E_theta_sqrd

  mcse_mu_post <- sqrt(variance / S_effect)
  return(mcse_mu_post)
}

posterior_mean_mcse(alpha = sample$alpha, beta = sample$beta)
```

```
## [1] 0.02674953 0.13722615
```

The MCSE for the posterior mean of α is 0.03, thus we can report the posterior mean of α as $E(h(\alpha \mid y)) = 1.0$. The MCSE for the posterior mean of β is 0.14, thus we can report the posterior mean of β as $E(h(\beta \mid y)) = 11$.