ST4234: Bayesian Statistics

Tutorial 7 Solution, AY 19/20

Solutions

1.

(a) The counts $(Y_1, \ldots, Y_4)$ follow a multinomial distribution with probabilities given by the vector

$$\left(\frac{1}{2}+\frac{\theta}{4},\ \frac{1}{4}(1-\theta),\ \frac{1}{4}(1-\theta),\ \frac{\theta}{4}\right).$$

Using a uniform prior for θ, the posterior density of θ is

$$\begin{aligned}
p(\theta|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\theta)p(\theta)\\
&\propto \frac{197!}{125!18!20!34!}\left(\frac{1}{2}+\frac{\theta}{4}\right)^{125}\left(\frac{1}{4}(1-\theta)\right)^{18}\left(\frac{1}{4}(1-\theta)\right)^{20}\left(\frac{\theta}{4}\right)^{34}\cdot 1\\
&\propto \left(\frac{1}{2}+\frac{\theta}{4}\right)^{125}\left(\frac{1}{4}(1-\theta)\right)^{38}\left(\frac{\theta}{4}\right)^{34}.
\end{aligned}$$

If θ is transformed to $\eta = \log\frac{\theta}{1-\theta}$, then $\theta = \frac{\mathrm{e}^\eta}{1+\mathrm{e}^\eta}$. As $\frac{\mathrm{d}\theta}{\mathrm{d}\eta} = \frac{\mathrm{e}^\eta}{(1+\mathrm{e}^\eta)^2}$, using the change-of-variable technique, the posterior density is

$$\begin{aligned}
p(\eta|\boldsymbol{y}) &= p(\theta|\boldsymbol{y})\cdot|J|\\
&\propto \left(\frac{1}{2}+\frac{\mathrm{e}^\eta}{4(1+\mathrm{e}^\eta)}\right)^{125}\left(\frac{1}{4(1+\mathrm{e}^\eta)}\right)^{38}\left(\frac{\mathrm{e}^\eta}{4(1+\mathrm{e}^\eta)}\right)^{34}\cdot\frac{\mathrm{e}^\eta}{(1+\mathrm{e}^\eta)^2}\\
&\propto \left(2+\frac{\mathrm{e}^\eta}{1+\mathrm{e}^\eta}\right)^{125}\frac{1}{(1+\mathrm{e}^\eta)^{39}}\left(\frac{\mathrm{e}^\eta}{1+\mathrm{e}^\eta}\right)^{35}.
\end{aligned}$$

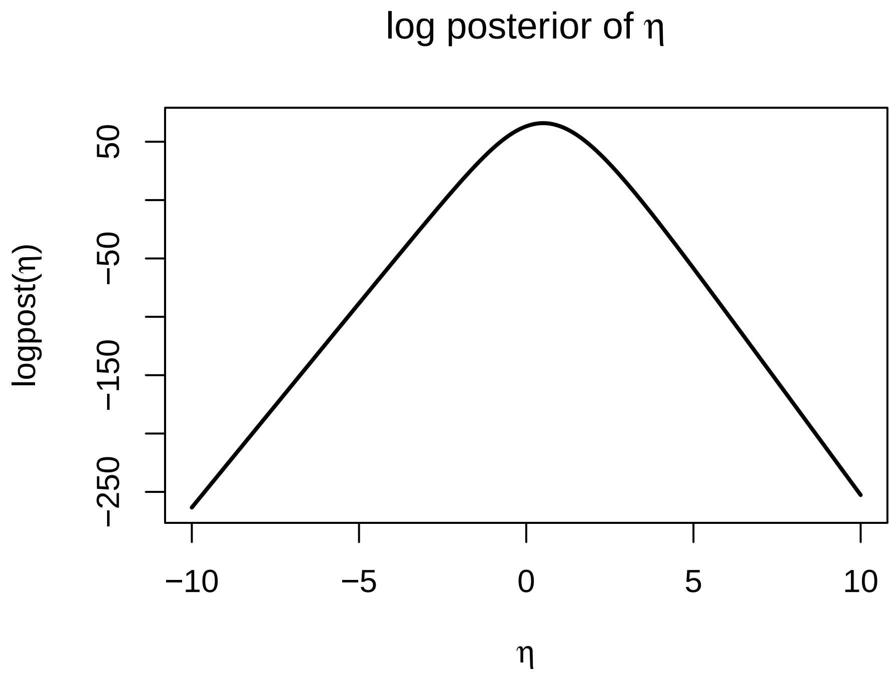
(b) The log posterior function (in terms of η) is

$$\ell(\eta) = \log p(\eta|\boldsymbol{y}) = 125\log\left(2+\frac{\mathrm{e}^\eta}{1+\mathrm{e}^\eta}\right) - 74\log(1+\mathrm{e}^\eta) + 35\eta + C,$$

where C is an additive constant not dependent on η. A normal approximation to $p(\eta|y)$ is given by $\mathrm{N}(\hat\eta, -[\ell''(\hat\eta)]^{-1})$ where $\hat\eta$ is the posterior mode. We first plot the log posterior function and identify that the mode lies in the interval [-10,10]. Then we use `optim()` with the option `method="Brent"`.

Using `optim()`, we find that the posterior mode $\hat\eta = 0.5066$ and $-[\ell''(\hat\eta)]^{-1} = 0.0473$. From this normal approximation a 95% CI for η is (0.0803, 0.9329). Transforming this interval, a 95% CI for θ is (0.5201, 0.7177).

```
# log posterior function
logpost <- function(eta) {
        125*log(2+(exp(eta)/(1+exp(eta))))-74*log(1+exp(eta))+35*eta
}

# plot the log posterior
eta.grid <- seq(from=-10, to=10, by=0.1)
plot(eta.grid, logpost(eta.grid),type="l",lwd=2,
     ylab=expression(paste("logpost(",eta,")")),xlab=expression(eta),
         main=expression(paste("log posterior of ",eta)))
```



```
# find the posterior mode using optim() with method="Brent"
(out <- optim(par=0, fn=logpost, hessian=TRUE, control=list(fnscale=-1),
              method="Brent", lower=-10, upper=10))

## $par
## [1] 0.5066004
##
## $value
## [1] 65.93283
##
## $counts
## function gradient
##       NA       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##           [,1]
## [1,] -21.13337
```

```r
(post.mode <- out$par)
```

```
## [1] 0.5066004
```

```r
(post.var <- -1/out$hessian)
```

```
##            [,1]
## [1,] 0.04731853
```

```r
# normal 95% CI for eta
(CI <- qnorm(c(0.025,0.975),mean=post.mode,sd=sqrt(post.var)))
```

```
## [1] 0.08025296 0.93294792
```

```r
# transformed 95% CI for theta
exp(CI)/(1+exp(CI))
```

```
## [1] 0.5200525 0.7176730
```

(c) Let

$$f(\eta) = \left(2 + \frac{e^\eta}{1+e^\eta}\right)^{125} \frac{1}{(1+e^\eta)^{39}} \left(\frac{e^\eta}{1+e^\eta}\right)^{35}$$

be the unnormalized posterior density of η.

To design the rejection algorithm, we need to first find a constant M such that $f(\eta) \le Mg(\eta)$ for all $\eta \in \mathbb{R}$, where $g(\eta)$ is taken to be the t_4 density with location given by the posterior mode `post.mode` and scale given by the variance in the normal approximation `post.var`. That is we need to find $M' = \log M$ such that

$$D(\theta) = \log f(\eta) - \log g(\eta) \le \log M = M' \quad \text{for all } \eta \in \mathbb{R}.$$

We first write a function named `diff` to compute the difference $D(\theta)$.

```r
# difference function = log(f) - log(g)
require(mvtnorm)

## Loading required package:  mvtnorm

df <- 4
diff <- function(eta, post.mode, post.var, df){
    logpost(eta) - dmvt(matrix(eta,ncol=1), delta=post.mode,
        sigma=matrix(post.var,1,1), df=df)
}
```

Then we maximize the difference function $D(\theta)$ by `optim()`.

```r
# maximize the difference function
(out <- optim(par=0, fn=diff, control=list(fnscale=-1),
              method="Brent", lower=-10, upper=10,
              post.mode=post.mode, post.var=post.var, df=df))

## $par
## [1] 0.7287661
##
## $value
## [1] 65.44855
##
## $counts
## function gradient
##       NA       NA
##
## $convergence
## [1] 0
##
## $message
## NULL

logM <- out$value       # M' value
```

We can see that the maximum of $D(\theta)$ is 65.44855, attained at $\eta = 0.7288$. Hence we can set M' to this maximum value of $D(\theta)$, and $M = \exp(M')$.

The rejection algorithm can then proceed as follows.

i. Generate $\eta^{(s)} \sim t_4(\hat{\eta}, -[\ell''(\hat{\eta})]^{-1})$ $(\ell(\eta) = \log f(\eta))$.
ii. Generate a random variable U from Uniform[0,1].
iii. If $U < \dfrac{f(\eta^{(s)})}{Mg(\eta^{(s)})}$, accept $\eta^{(s)}$; otherwise reject $\eta^{(s)}$.

iv. Repeat steps 1–3 until the desired sample $\{\eta^{(s)}|s = 1, \ldots, S\}$ is obtained. The members of this sample will then be random variables from $p(\eta|\boldsymbol{y})$.

The R function for rejection sampling is as follows.

```
# rejection sampling function
rej_sampling <- function(S, post.mode, post.var, df, logM){
  eta.samples <- rep(0,S)
  s <- 0
  T <- 0
  while (s < S){
    T <- T+1
    eta <- rmvt(1, delta=post.mode, sigma=matrix(post.var,1,1), df=df)
    U <- runif(1)
    if (log(U) < diff(eta, post.mode, post.var, df) - logM){
      s <- s+1
      eta.samples[s] <- eta
    }
  }
  accept.rate <- S/T
  list(accept.rate=accept.rate, eta.samples=eta.samples)
}

set.seed(4234)
S <- 10^4
output <- rej_sampling(S, post.mode, post.var, df, logM)
output$accept.rate

## [1] 0.8873114

# CI for eta
quantile(output$eta.samples, c(0.025,0.975))

##       2.5%      97.5%
## 0.08130297 0.94648013

# CI for theta
theta.samples <- 1/(1+exp(-output$eta.samples))
quantile(theta.samples, c(0.025,0.975))

##      2.5%     97.5%
## 0.5203146 0.7204068
```

We draw 10^4 samples of η using this function. The acceptance rate is 88.73%. A

95% CI for η is (0.081, 0.946) and a 95% CI for θ is (0.520, 0.720).

2.(a) We first need to write down the log posterior function. For each of the two $\boldsymbol{y}$ values, we optimize this log posterior function, find the mode of the posterior, as well as the normal approximation variance. From page 16 in Chapter 5 notes, if $\widehat{\theta}$ is the posterior mode of θ, then the variance of normal approximation is given by

$$\widehat{\sigma}^2 = 1/\left[\frac{y_1}{(2+\hat{\theta})^2} + \frac{y_2+y_3}{(1-\hat{\theta})^2} + \frac{y_4}{\hat{\theta}^2}\right].$$

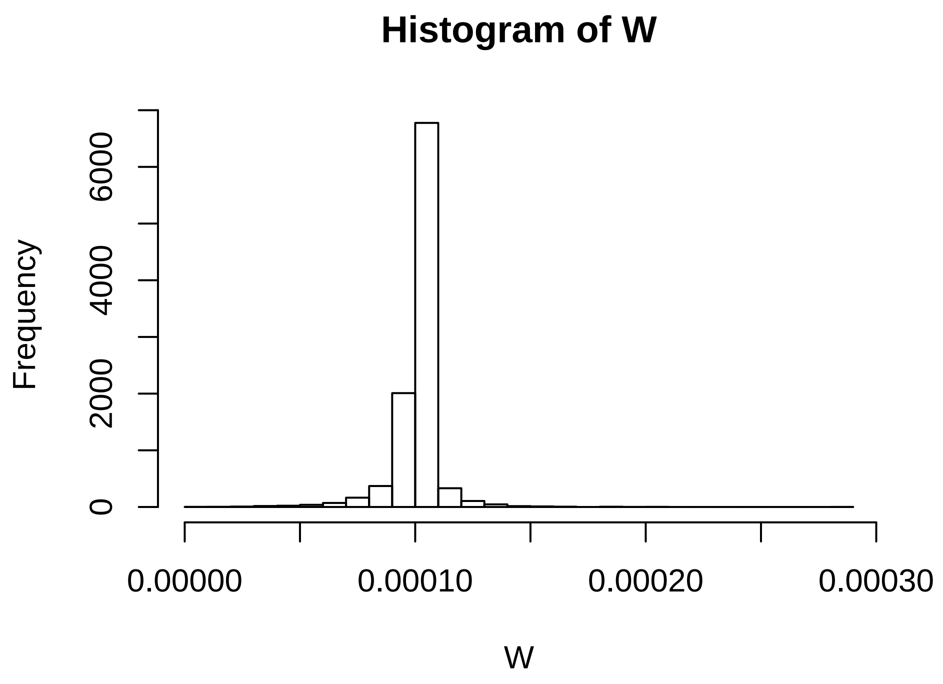
We draw 10^4 samples $\{\theta^{(s)}:\ s=1,\ldots,S\}$ from $N(\widehat{\theta},\widehat{\sigma}^2)$. The importance weights are given by $w(\theta^{(s)}) = \frac{\exp(\ell(\theta))}{g(\theta)}$ for each sample, where $\ell(\theta)$ is the log posterior function and $g(\theta)$ is the density of $N(\widehat{\theta},\widehat{\sigma}^2)$. We then compute the normalized weights $W_s = \frac{w(\theta^{(s)})}{\sum_{s=1}^S (w(\theta^{(s)})}$ for $s=1,\ldots,S$, and plot a histogram of all W_s. The importance sampling estimate of $\mathrm{E}(\theta|\boldsymbol{y})$ is $\widehat{\theta}_{IS} = \widehat{\sum}_{s=1}^S W_s\theta^{(s)}$, and the standard error is

$$\sqrt{\sum_{s=1}^S \left\{W_s\left[\theta^{(s)} - \widehat{\theta}_{IS}\right]\right\}^2}.$$

We first run this procedure for $\boldsymbol{y} = (125, 18, 20, 34)$. We obtain that the importance sampling estimate of $\mathrm{E}(\theta|\boldsymbol{y})$ is 0.6230 with standard error 5.19×10^{-4}. The true posterior mean is 0.6228. Compared to the Laplace approximation method in Chapter 5, the importance sampling estimate is more accurate than approximation method 1 (which gives 0.6268), and is about as accurate as approximation method 2 (which gives 0.6227).

```r
# first, define the log posterior function
logpost <- function(theta, Y) {
        Y[1]*log(2+theta) + (Y[2]+Y[3])*log(1-theta) + Y[4]*log(theta)
}
# Y=(125,18,20,34)
Y <- c(125,18,20,34)
# optimize logpost()
# find the posterior mode and variance of normal approximation
# formulas can be found in Chapter 5 notes
(out <- optimize(logpost, interval=c(0,1), Y=Y, maximum=TRUE))

## $maximum
## [1] 0.6268298
##
## $objective
## [1] 67.3841
```

```r
post.mode <- out$maximum
post.var <- 1/(Y[1]/(2+post.mode)^2+(Y[2]+Y[3])/
                 (1-post.mode)^2+Y[4]/post.mode^2)
# importance sampling: draw samples from normal
set.seed(4234)
S <- 10^4
theta.samples <- rnorm(S, mean=post.mode, sd=sqrt(post.var))
# raw log weights
logw <- logpost(theta.samples,Y=Y) -
        dnorm(theta.samples, mean=post.mode, sd=sqrt(post.var), log=TRUE)
w <- exp(logw - max(logw))      # raw weights
W <- w/sum(w)            # normalized weights
hist(W, breaks=30)              # histogram of normalized weights
```



```r
(est <- sum(W*theta.samples))          # IS estimate of E(theta|y)

## [1] 0.6230117

(se <- sqrt(sum((W*(theta.samples -est))^2)))   # standard error

## [1] 0.0005189213
```

Next, we run this procedure for $\boldsymbol{y} = (14, 0, 1, 5)$. In this case, you need to be careful. **Because the posterior mode of θ is close to 1, when we use**

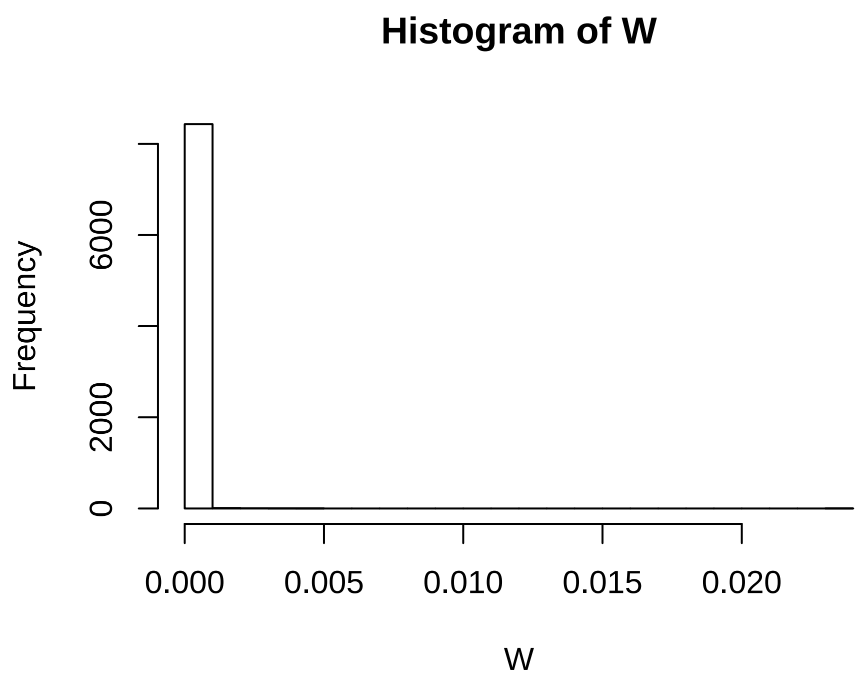
importance sampling with the normal proposal, the normal approximation distribution will generate a lot of samples outside the interval $[0, 1]$. These samples should receive zero importance weights since the posterior density outside $[0, 1]$ is zero. If $f(\theta)$ is the unnormalized posterior density, then obviously $f(\theta^{(s)}) = 0$ if $\theta^{(s)} \notin [0, 1]$, which means that $w(\theta^{(s)}) = f(\theta^{(s)})/g(\theta^{(s)}) = 0/g(\theta^{(s)}) = 0$ if $\theta^{(s)} \notin [0, 1]$, and the corresponding $W_s = 0$ as well.

In the R codes below, we can achieve this by removing NaNs from the logarithm of raw weights `logw`. In this case, out of 10^4 samples from the normal distribution, over 1500 samples of $\theta^{(s)}$ fall outside the $[0, 1]$ interval, and we identify and remove them by using the function `is.nan()`. You may use other method to achieve the same goal.

The importance sampling estimate of $\mathrm{E}(\theta|\boldsymbol{y})$ is 0.8316 with standard error 7.96×10^{-2}. The true posterior mean is 0.8311. Compared to the Laplace approximation method in Chapter 5, the importance sampling estimate is more accurate than both approximation method 1 (which gives 0.9034) and approximation method 2 (which gives 0.8275). However, the importance weights are highly imbalanced as seen from the histogram.

```
# Y=(14,0,1,5)
Y <- c(14,0,1,5)
# optimize logpost()
# find the posterior mode and variance of normal approximation
# formulas can be found in Chapter 5 notes
(out <- optimize(logpost, interval=c(0,1), Y=Y, maximum=TRUE))

## $maximum
## [1] 0.9034396
##
## $objective
## [1] 12.07723

post.mode <- out$maximum
post.var <- 1/(Y[1]/(2+post.mode)^2+(Y[2]+Y[3])/
                  (1-post.mode)^2+Y[4]/post.mode^2)
# importance sampling: draw samples from normal
set.seed(4234)
S <- 10^4
theta.samples <- rnorm(S, mean=post.mode, sd=sqrt(post.var))
# raw log weights
logw <- logpost(theta.samples,Y=Y) -
```

```
        dnorm(theta.samples, mean=post.mode, sd=sqrt(post.var), log=TRUE)

## Warning in log(1 - theta):  NaNs produced

# logw contains NaNs for those theta's falling outside [0,1]
# remove all NaNs from logw
# find the indices where logw is not NaN, i.e. theta is inside [0,1]
index <- which(is.nan(logw)==FALSE)
(length(index))          # number of theta samples inside [0,1]

## [1] 8454

w <- exp(logw[index] - max(logw[index]))        # raw weights
W <- w/sum(w)            # normalized weights
hist(W, breaks=30)               # histogram of normalized weights
```



```
(est <- sum(W*theta.samples[index]))            # IS estimate of E(theta|y)

## [1] 0.831613

(se <- sqrt(sum((W*(theta.samples[index] -est))^2)))     # standard error


## [1] 0.007964838
```

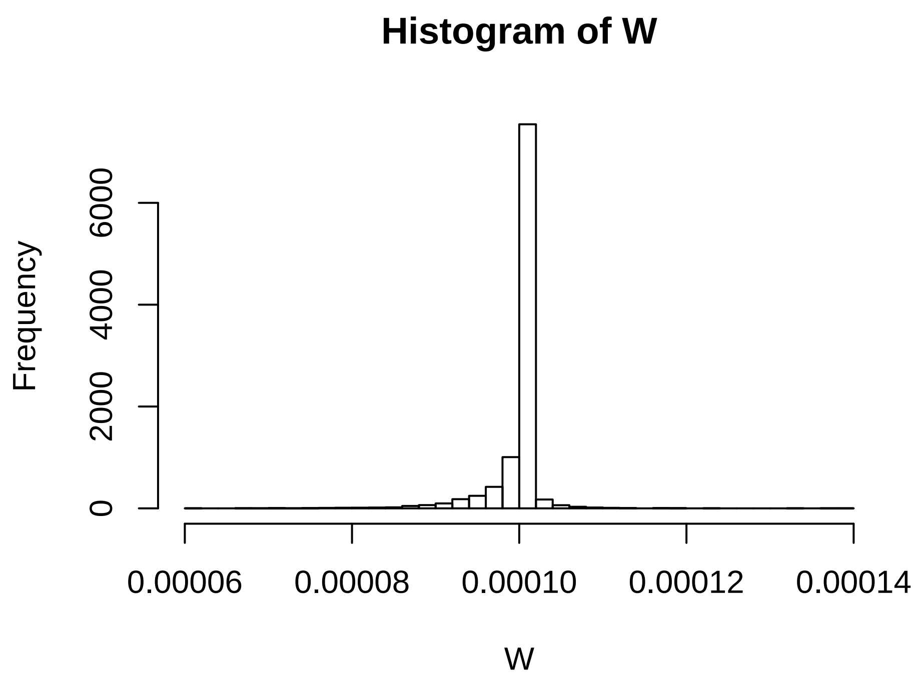
(b) We first decide the values of a and b in the proposal beta distribution. The mode of Beta(a, b) is $\frac{a-1}{a+b-2}$, and the variance is $\frac{ab}{(a+b)^2(a+b+1)}$. We can first write a function to search for the right values of a and b, by matching these two values with the posterior mode and the variance from normal approximation. In the R codes below, this is done by optimizing the function `beta.para()`, which computes the squared error from the fitted to the observed values. Then we use the solution of (a, b) to perform importance sampling, similar to Part (a).

We first run the procedure for $\boldsymbol{y} = (125, 18, 20, 34)$. The beta proposal is approximately Beta$(54.77, 33.01)$. We obtain that the importance sampling estimate of $\mathrm{E}(\theta|\boldsymbol{y})$ is 0.6223 with standard error 5.10×10^{-4}. The true posterior mean is 0.6228. This estimate is slightly less accurate than the importance sampling estimate from normal approximation in Part (a) (0.6230) and approximation method 2 in Chapter 5 (0.6227), but is still more accurate than approximation method 1 (0.6268).

```r
# Y=(125,18,20,34)
Y <- c(125,18,20,34)
(out <- optimize(logpost, interval=c(0,1), Y=Y, maximum=TRUE))

## $maximum
## [1] 0.6268298
##
## $objective
## [1] 67.3841

post.mode <- out$maximum
post.var <- 1/(Y[1]/(2+post.mode)^2+(Y[2]+Y[3])/
                 (1-post.mode)^2+Y[4]/post.mode^2)
# find beta parameters: define a 2d loss function
beta.para <- function(para, post.mode, post.var) {
        a <- para[1]; b <- para[2]
        ab.mode = (a-1)/(a+b-2)
        ab.var = a*b/(a+b)^2/(a+b+1)
        return((ab.mode-post.mode)^2+(ab.var-post.var)^2)
}
(out.ab <- optim(par=c(5,5), fn=beta.para,
                post.mode=post.mode, post.var=post.var))

## $par
## [1] 54.77294 33.01322
##
## $value
## [1] 5.701211e-11
```

```
##
## $counts
## function gradient
##      121       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
a <- out.ab$par[1]
b <- out.ab$par[2]
# importance sampling: draw samples from Beta(a,b)
set.seed(4234)
S <- 10^4
theta.samples <- rbeta(S, a, b)
# raw log weights
logw <- logpost(theta.samples,Y=Y) -
        dbeta(theta.samples, a, b, log=TRUE)
w <- exp(logw - max(logw))      # raw weights
W <- w/sum(w)           # normalized weights
hist(W, breaks=30)              # histogram of normalized weights
```



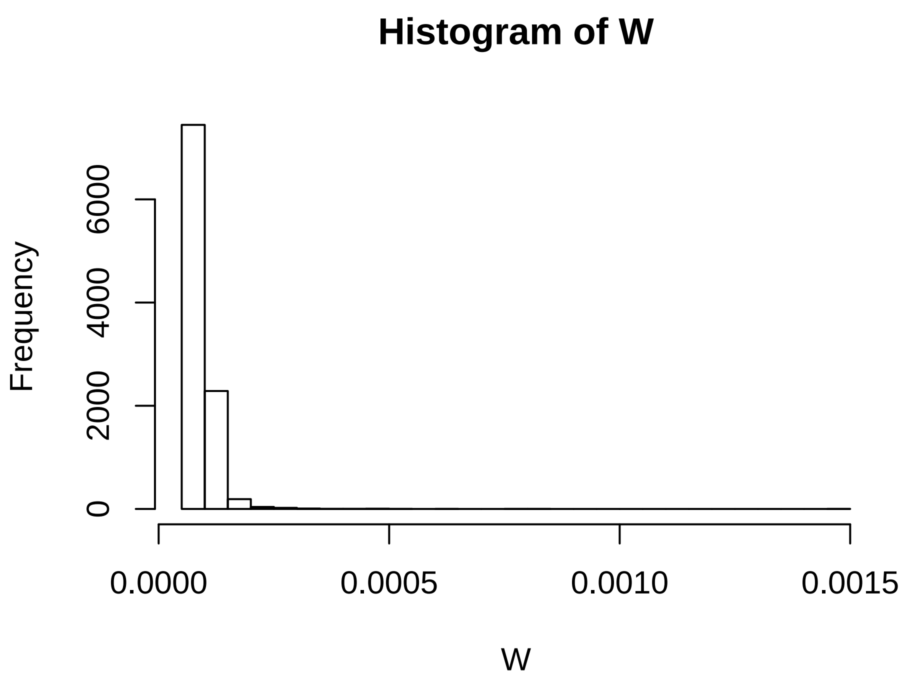
```
(est <- sum(W*theta.samples))          # IS estimate of E(theta|y)

## [1] 0.622338

(se <- sqrt(sum((W*(theta.samples -est))^2)))   # standard error

## [1] 0.0005102165
```

We then run the procedure for $\boldsymbol{y} = (14, 0, 1, 5)$. The beta proposal is approximately Beta$(11.86, 2.16)$. Note that we will no longer encounter the problem as in Part (a), because all samples from the beta distribution is inside the interval $[0, 1]$. We obtain that the importance sampling estimate of $\mathrm{E}(\theta|\boldsymbol{y})$ is 0.8288 with standard error 1.68×10^{-2}. The true posterior mean is 0.8311. This estimate is slightly less accurate than the importance sampling estimate from normal approximation in Part (a) (0.8316), but is more accurate than approximation method 1 (0.9034) and approximation method 2 (0.8275) in Chapter 5. The histogram of normalized weights seems slightly better than that in Part (a).

```
# Y=(14,0,1,5)
Y <- c(14,0,1,5)
(out <- optimize(logpost, interval=c(0,1), Y=Y, maximum=TRUE))

## $maximum
## [1] 0.9034396
##
## $objective
## [1] 12.07723

post.mode <- out$maximum
post.var <- 1/(Y[1]/(2+post.mode)^2+(Y[2]+Y[3])/
                 (1-post.mode)^2+Y[4]/post.mode^2)
# find beta parameters: define a 2d loss function
beta.para <- function(para, post.mode, post.var) {
        a <- para[1]; b <- para[2]
        ab.mode = (a-1)/(a+b-2)
        ab.var = a*b/(a+b)^2/(a+b+1)
        return((ab.mode-post.mode)^2+(ab.var-post.var)^2)
}
(out.ab <- optim(par=c(5,5), fn=beta.para,
                 post.mode=post.mode, post.var=post.var))

## $par
## [1] 11.863307  2.161017
```

```
##
## $value
## [1] 3.143529e-10
##
## $counts
## function gradient
##       89       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
a <- out.ab$par[1]
b <- out.ab$par[2]
# importance sampling: draw samples from Beta(a,b)
set.seed(4234)
S <- 10^4
theta.samples <- rbeta(S, a, b)
# raw log weights
logw <- logpost(theta.samples,Y=Y) -
        dbeta(theta.samples, a, b, log=TRUE)
w <- exp(logw - max(logw))      # raw weights
W <- w/sum(w)           # normalized weights
hist(W, breaks=30)              # histogram of normalized weights
```



```
(est <- sum(W*theta.samples))            # IS estimate of E(theta|y)

## [1] 0.8288104

(se <- sqrt(sum((W*(theta.samples-est))^2)))     # standard error

## [1] 0.001681054
```

3. From Tutorial 6 Question 2, we have

$$\log p(\theta|y) = \underbrace{\theta y - n\log(1+e^{\theta}) - \frac{(\theta-\mu)^2}{2\sigma^2}}_{\ell(\theta)} + C,$$

To design the rejection algorithm, we need to first find a constant M such that $\exp(\ell(\theta)) \le Mp(\theta)$ for all $\theta \in \mathbb{R}$, where $p(\theta)$ is the density of $\mathrm{N}(0, .25^2)$. That is, we need to find $M' = \log M$ such that

$$\ell(\theta) - \log p(\theta) \le \log M = M' \quad \text{for all } \theta \in \mathbb{R}.$$

Let $D(\theta) = \ell(\theta) - \log p(\theta)$. We write a function `diff()` to compute $D(\theta)$ and use `optim()` to find its maximum value. We can see that $M' = -0.4673558$ and we take $M = \exp(M')$.

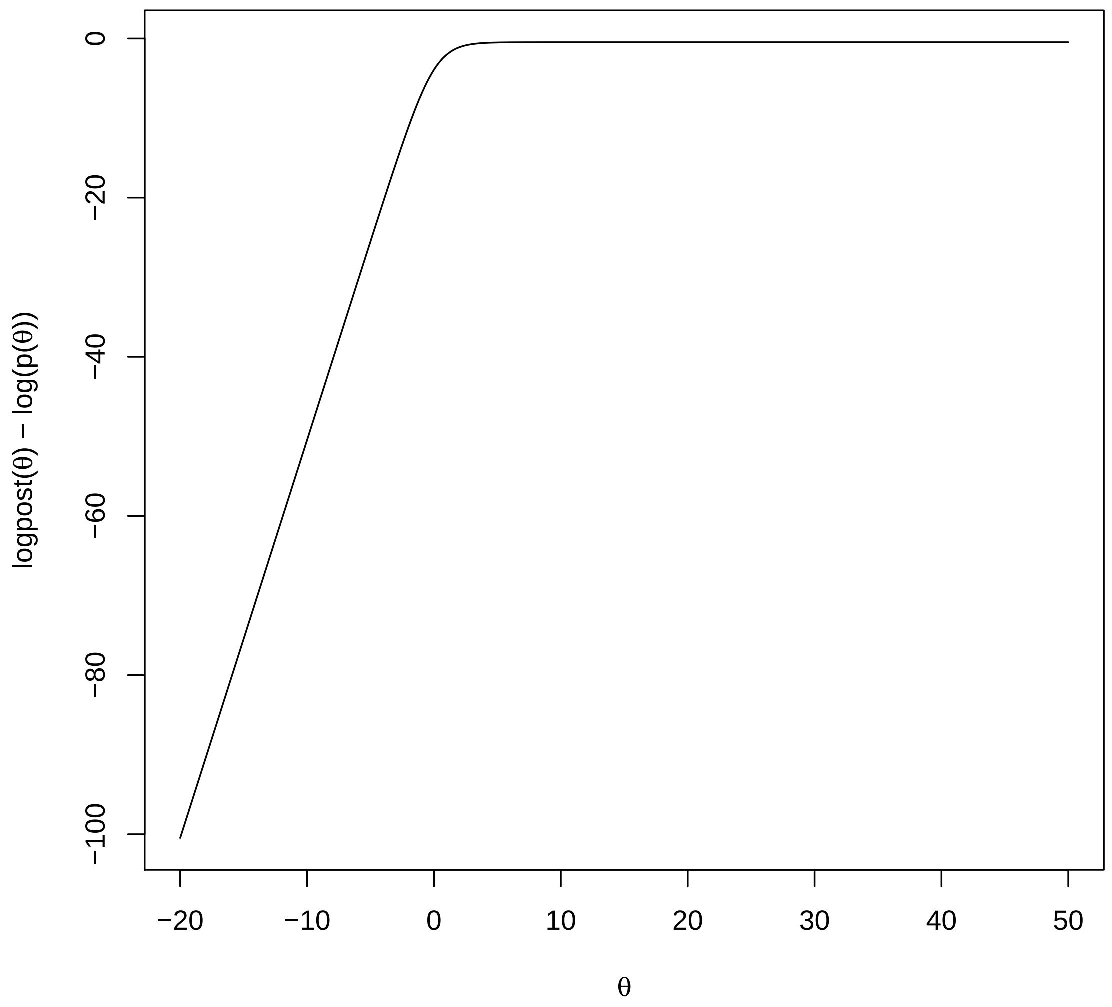
```
mu <- 0
sigma2 <- 0.25^2
n <- 5
y <- 5
logpost <- function(theta, y, n, mu, sigma2){
        theta*y - n*log(1+exp(theta)) - 0.5*(theta-mu)^2/sigma2
}
diff <- function(theta,y,n,mu,sigma2){
        logpost(theta,y,n,mu,sigma2) -
        dnorm(theta,mean=mu,sd=sqrt(sigma2),log=TRUE)
}

# find M' by maximizing the diffence function
(out <- optim(par=10, fn=diff,control=list(fnscale=-1),method="Brent",
              lower=0,upper=100,y=y,n=n,mu=mu,sigma2=sigma2))

## $par
## [1] 32.62379
##
## $value
## [1] -0.4673558
##
## $counts
## function gradient
##       NA       NA
##
## $convergence
## [1] 0
##
## $message
## NULL

logM <- out$value
```

Alternatively,

$$\begin{aligned} D(\theta) &= \ell(\theta) - \log p(\theta) \\ &= \theta y - n\log(1+\mathrm{e}^{\theta}) + 0.5\log(2\pi) + 0.5\log(\sigma^2) \\ &= 5[\theta - \log(1+\mathrm{e}^{\theta})] + 0.5\log(2\pi) + 0.5\log(0.25^2). \\ D'(\theta) &= 5\left(1 - \frac{\mathrm{e}^{\theta}}{1+\mathrm{e}^{\theta}}\right) > 0 \end{aligned}$$

Note that $D(\theta)$ is strictly increasing. Since $\mathrm{e}^{\theta} < 1+\mathrm{e}^{\theta} \Rightarrow \theta - \log(1+\mathrm{e}^{\theta}) < 0$, we have $D(\theta) < 0.5\log(2\pi) + 0.5\log(0.25^2) = -0.4673558$.

```
# plot of the difference function
theta.grid <- seq(from=-20,to=50,by=0.1)
plot(theta.grid, diff(theta.grid,y,n,mu,sigma2),type="l",
        ylab=expression(paste("logpost(",theta,") - log(p(",theta,"))")),
        xlab=expression(theta))
```



```
# Alternatively, by analytical calculation
(logM <- 0.5*log(2*pi)+0.5*log(sigma2))

## [1] -0.4673558
```

We can design the rejection algorithm as follows.

(a) Generate $\theta^{(s)} \sim \mathrm{N}(0, 0.25^2)$.

(b) Generate a random variable U from Uniform[0,1].

(c) If $U < \exp\{\ell(\theta) - M' \log p(\theta)\}$ with $M' = -0.4673558$, accept $\theta^{(s)}$; otherwise reject $\theta^{(s)}$.

(d) Repeat steps 1–3 until the desired sample $\{\theta^{(s)} | s = 1, \ldots, S\}$ is obtained. The members of this sample will then be random variables from $p(\theta|y)$.

The rejection algorithm can be implemented using the following R-code.

```r
# rejection sampling function
rej_sampling <- function(S, y, n, mu, sigma2, logM){
  theta.samples <- rep(0,S)
  s <- 0
  T <- 0
  while (s < S){
    T <- T+1
    theta <- rnorm(1, mean=mu, sd=sqrt(sigma2))
    U <- runif(1)
    if (log(U) < diff(theta, y, n, mu, sigma2) - logM){
      s <- s+1
      theta.samples[s] <- theta
    }
  }
  accept.rate <- S/T
  list(accept.rate=accept.rate, theta.samples=theta.samples)
}

set.seed(4234)
S <- 10^4
output <- rej_sampling(S, y, n, mu, sigma2, logM)
output$accept.rate

## [1] 0.03556833

# estimate P(theta>0|y)
mean(output$theta.samples>0)

## [1] 0.7238
```

An estimate of the probability that the coin is biased towards head using samples form the rejection algorithm is 0.724. The acceptance rate is 3.6%. This low acceptance rate is caused by the mismatch between the shapes of densities in the prior and the posterior.