STAT 4224 HW #4

Carlyle Morgan

11/11/2021

```
set.seed(1)
knitr::opts_chunk$set(echo = TRUE)
```

1.

a.

```
y <- c(95,70,80,62,108,62,61,74) / 10

J <- length(y)

n <- c(25,23,20,24,24,22,22,20)

sigma <- sqrt(14.3/n); rm(n);

Sim.size <- 10000

log.post.tau.fun <- function(tau, y, sigma)
{
 V.mu <- 1 / sum( 1/(sigma^2 + tau^2) )
 mu.hat <- V.mu * sum( y / (sigma^2 + tau^2) )
 log.post <- ( log(V.mu) - sum(log(sigma^2 + tau^2) ) ) / 2
 log.post <- log.post - 0.5 * sum( (y-mu.hat)^2 / (sigma^2 + tau^2) )
 log.post
}

log.tau <- seq(-2.5, 2.5, .01);

T <- length(log.tau);

log.post.tau <- rep(NA, T)

for(t in 1:T)
{
 log.post.tau[t] <- log.post.tau.fun(tau=exp(log.tau[t]), y, sigma)
}

maxie <- max(log.post.tau)

log.post.tau <- log.post.tau - maxie; rm(maxie);
```

```r
post.tau <- exp(log.post.tau)

S <- Sim.size

theta.sim <- matrix(NA, J, S)

delta <- (log.tau[2] - log.tau[1]) / 2

post.log.tau <- exp(log.tau) * post.tau

for(s in 1:S)
{
 t <- sample(T, 1, prob=post.log.tau)
 tau <- exp(log.tau[t] + runif(1, -delta, delta))
 V.mu <- 1 / sum( 1 / (sigma^2 + tau^2) )
 mu.hat <- V.mu * sum( y / (sigma^2 +  tau^2) )
 mu <- rnorm(1, mean=mu.hat, sd=sqrt(V.mu))
 V <- 1 / (1/sigma^2 + 1/tau^2)
 theta.hat <- V * (y/sigma^2 + mu/tau^2)
 theta.sim[,s] <- rnorm(J, mean=theta.hat, sd=sqrt(V))
}


CIs<-data.frame(seq(1,8,1), apply(theta.sim,1,function(x) quantile(x,0.025)),apply(theta.sim,1,
function(x) quantile(x,0.975)))
colnames(CIs)<-c("School", "95% CI Lower Bound", "95% CI Upper Bound")
CIs
```

```
##   School 95% CI Lower Bound 95% CI Upper Bound
## 1      1           7.774868          10.625420
## 2      2           5.663680           8.594743
## 3      3           6.417326           9.418949
## 4      4           5.012549           7.845976
## 5      5           8.705884          11.806266
## 6      6           4.985416           7.938029
## 7      7           4.909415           7.889974
## 8      8           5.963686           8.949002
```

b.

```r
mean(theta.sim[7,] < theta.sim[6,])
```

```
## [1] 0.5312
```

```r
mean(apply(theta.sim, 2, which.min)==7)
```

```
## [1] 0.3229
```

$P(\theta_7 < \theta_6|y) = 0.53$

$P(min(\theta_1, ..., \theta_8) = \theta_7|y) = 0.3229$

c.

```r
ytilde.sim <- matrix(rnorm(J*S, theta.sim, sd=sqrt(14.3)), J, S)

mean(ytilde.sim[7,] < ytilde.sim[6,])
```

```
## [1] 0.5024
```

```r
mean(apply(ytilde.sim, 2, which.min)==7)
```

```
## [1] 0.1805
```

$P(\tilde{y}_7 < \tilde{y}_6|y) = 0.5024$

$P(min(\tilde{y}_1, ..., \tilde{y}_8) = \tilde{y}_7|y) = 0.1805$

2.

a.

```r
x <- c(-3.3, +0.1, -1.1, +2.7, +2.0, -0.4)

y <- c(-2.6, -0.2, -1.5, +1.5, +1.9, -0.3)

n <- length(y)

Sxx <- sum(x^2);  Syy <- sum(y^2);  Sxy <- sum(x*y);

neg.log.q <- function(theta, Sxx, Syy, Sxy, n)
{
 n/2 * log(1 - theta^2) + 1/(2 * (1 - theta^2)) *
   ( Sxx - 2 * theta * Sxy + Syy )
}

foo <- optim(0.5, fn=neg.log.q, method="L-BFGS-B",
  lower=0.000001, upper=0.999999,
  Sxx=Sxx, Syy=Syy, Sxy=Sxy, n=n)

theta.hat.MAP <- foo$par;

theta.hat.MAP
```

```
## [1] 0.8468805
```

The MAP estimator approximates theta to be 0.8468.

b.

```r
theta.vals <- seq(.005, .995, .01);  length(theta.vals);
```

```
## [1] 100
```

```r
log.q <- -1 * neg.log.q(theta.vals, Sxx, Syy, Sxy, n)

maxie <- max(log.q);  log.q <- log.q - maxie; rm(maxie);

q <- exp(log.q);  rm(log.q);
```

```r
theta <- theta.vals;

sum(theta * q) / sum(q);   sum(theta^2 * q) / sum(q);
```

```
## [1] 0.7754085
```

```
## [1] 0.6135621
```

Using quadrature, we approximate $E[\theta|x, y]$ to be ~0.7754 and $E[\theta^2|x, y]$ to be 0.6134.

c.

```r
theta.sim <- sample(theta.vals, 1000, replace=T, prob=q)

theta.sim <- theta.sim + runif(1000, -.005, +.005) # 'jitter'

mean(theta.sim)
```

```
## [1] 0.7802546
```

```r
mean(theta.sim^2)
```

```
## [1] 0.6208263
```

Using MC Integration, we approximate $E[\theta|x, y]$ to be ~0.7721 and $E[\theta^2|x, y]$ to be ~0.60876.

d.

```r
S<-1000
neg.log.M <- neg.log.q(theta.hat.MAP, Sxx, Syy, Sxy, n)

theta.prop <- rep(NA, S);  accept <- rep(NA, S);

for(s in 1:S)
{
 theta.star <- runif(1)
 log.prob <- neg.log.M - neg.log.q(theta.star,Sxx,Syy,Sxy,n)
 accept[s] <- ( log(runif(1)) < log.prob )
 theta.prop[s] <- theta.star;  rm(theta.star);  rm(log.prob);
}

theta.RS <- theta.prop[accept]
length(theta.RS)
```

```
## [1] 196
```

```r
mean(theta.RS)
```

```
## [1] 0.7582437
```

```r
mean(theta.RS^2)
```

```
## [1] 0.5908891
```

Using Rejection Sampling with a uniform proposal distribution, we approximate $E[\theta|x, y]$ to be ~0.772 and $E[\theta^2|x, y]$ to be ~0.611. The effective sample size is 227.

e.

```r
theta.IS <- runif(S)
w <- exp(-neg.log.q(theta.IS, Sxx, Syy, Sxy, n))
sum(w * theta.IS) / sum(w)
```

```
## [1] 0.7708949
```

```r
sum(w * theta.IS^2) / sum(w)
```

```
## [1] 0.6065504
```

```r
w.tilde <- w / sum(w)
1 / sum(w.tilde^2)
```

```
## [1] 318.5525
```

Using Importance Sampling with a uniform proposal distribution, we approximate $E[\theta|x, y]$ to be ~0.771 and $E[\theta^2|x, y]$ to be ~0.607. The effective sample size is ~320.