STAT 200B 2019 Week08

Soyeon Ahn

1 Monte Carlo Integration

Based on chapter 24 Simulation Methods, the related R code is presented.

1.1 Basic Monte Carlo Integration

Suppose we want to evaluate the integral

$$I = \int_a^b h(x)dx$$

for some function h.

$$I = \int_a^b h(x)(b-a)1/(b-a)dx = \int_a^b w(x)f(x)dx$$

where $w(x) = h(x)(b-a)$ and $f(x) = 1/(b-a)$. Notice that f is the probability density for a uniform random variable over (a, b). Hence,

$$I = E_f(w(X))$$

when $X \sim \text{Unif}(a, b)$. If we generate $X_1, \ldots, X_N \sim \text{Unif}(a, b)$ then by the law of large numbers

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} w(X_i) \xrightarrow{P} E(w(X)) = I$$

We can also compute the standard error of the estimate

$$\hat{se}^2(\hat{I}) = \frac{1}{N}\frac{\sum_{i=1}^{N}(Y_i - \hat{I})^2}{N-1}$$

, where $Y_i = w(X_i)$.

1.2 Posterior mean

Suppose we have $\theta_1, \ldots, \theta_B \overset{iid}{\sim} f(\theta|x^n)$. The basic Monte Carlo approximation to the posterior mean of any function $q(\theta)$ is

$$\begin{aligned} E[q(\theta)|x^n] &= \int q(\theta)f(\theta|x^n)d\theta \\ &\approx \frac{1}{B}\sum_{i=1}^{B} q(\theta_i) \end{aligned}$$

1.3 Example: integration of a simple function

Let $h(x) = x^3$. $I = \int_0^1 x^3 dx = 1/4$. The Monte Carlo approximation of the integral is Ih = 0.2524212

```
set.seed(1)
h <- function(x){ x^3 }
N <- 10000
x <- runif(N,0,1)
Ih <- mean(h(x))
Ih
```

1.4 Example: CDF of the standard normal

The PDF of the standard normal distribution is

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp^{-x^2/2}$$

Suppose we want to compute the cdf at some point x:

$$I = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp^{-s^2/2} ds = \Phi(x) = \int h(s)f(s)ds$$

where $h(s) = \mathbf{1}(s < x)$.

Now we generate $X_1, \ldots, X_N \sim N(0, 1)$ and set

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} h(X_i) = \frac{\text{number of observations} \leq x}{N}$$

When x = 1.96, $\Phi(1.96) = 0.975$. Using $N = 100,000$, the estimate is 0.974.

```
set.seed(1)
h <- function(s, x1){ as.numeric(s < x1) }
N <- 100000
x <- rnorm(N)
Ih <- h(x, 1.96)
mean(Ih)
```

1.5 Example: Bayesian Inference for Two Binomials

Let $X \sim Binomial(n, p_1)$ and $X \sim Binomial(m, p_2)$. We would like to estimate $\delta = p_2 - p_1$. Suppose we use the prior $f(p_1, p_2) = f(p_1)f(p_2) = 1$, an independent flat prior on (p_1, p_2). The posterior is Beta distribution, where $p_1|X \sim Beta(X + 1, n - X + 1)$ and $p_2|Y \sim Beta(Y + 1, m - Y + 1)$.

For example, suppose that $n = m = 10$, X = 8 and Y = 6. From a posterior sample of size 1000, a 95 percent posterior interval is (-0.52, 0.20).

```
set.seed(10)
N <- 1000
x1 <- rbeta(N, 8+1, 10-8+1)
y1 <- rbeta(N, 6+1, 10-6+1)
d1 <- y1 - x1
quantile(d1, 0.025)
quantile(d1, 0.975)
hist(d1, br=100)
```

2 Imporatance sampling

Consider the integral

$$I = \int h(x)f(x)dx$$

where f is a probability density. We may not know how to sample from f. Importance sampling is generalized Monte Carlo approximation. Let g be a probability density (candidate) that we know how to simulate from.

$$I = \int h(x)f(x)dx = \int h(x)\frac{f(x)}{g(x)}g(x)dx = E_g Y$$

where $Y = \frac{h(x)f(x)}{g(x)}$. $\frac{f(x)}{g(x)}$ is called an importance weight.

We can simulate $X_i \sim g$ and estimate I by

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} Y_i = \frac{1}{N}\sum_{i=1}^{N} h(X_i)\frac{f(X_i)}{g(X_i)}$$

Please note that we should choose g to be similar in shape to f but with thicker tails. The standard error of $\hat{I}$ could be infinite.

2.1 Posterior mean

The posterior mean is

$$\bar{\theta} = \int \theta \frac{\mathcal{L}_n f(\theta)}{\int \mathcal{L}_n f(\theta)}$$

To obtain an approximation to the posterior mean of any function $q(\theta)$, $E[q(\theta)|x^n]$ is as follows: sample from the prior: $\theta_1, \ldots, \theta_B \overset{iid}{\sim} f(\theta)$, then for each $i = 1, \ldots, B$, calculate

$$w_i = \frac{\mathcal{L}_n(\theta_i)}{\sum_{i=1}^{B} \mathcal{L}_n(\theta_i)}$$

Then $E[q(\theta)|x^n] \approx \sum_{i=1}^{B} q(\theta_i)w_i$.

The posterior mean of any function $q(\theta)$ is

$$\int q(\theta)\frac{\mathcal{L}_n f(\theta)}{\int \mathcal{L}_n f(\theta)}$$

Using the importance sampling, sample from the known function g: $\theta_1, \ldots, \theta_B \overset{iid}{\sim} g(\theta)$, calculate

$$w_i = \frac{\frac{\mathcal{L}_n(\theta_i)f(\theta_i)}{g(\theta_i)}}{\sum_{i=1}^{B}\frac{\mathcal{L}_n(\theta_i)f(\theta_i)}{g(\theta_i)}}$$

Then $E[q(\theta)|x^n] \approx \sum_{i=1}^{B} q(\theta_i)w_i$.

To handle numerical problems, the summation of the log posterior is often used and factor out the largest value to prevent numerical underflow.

2.2 Example: Tail Probability

$I = P(Z > 3) = 1 - \Phi(3) = 0.0013$ when $Z \sim N(0, 1)$ Write $I = \int h(x)f(x)dx$ where $f(x)$ is the standard Normal density and $h(x) = 1$ if $x > 3$, and 0 otherwise. The basic Monte Carlo estimator is $\hat{I} = \frac{1}{N}\sum_{i=1}^{N} h(X_i)$. Notice that most observations are wasted in the sense that most are not near the right tail. Now we will estimate this with importance sampling taking g to be a $N(4, 1)$. We draw values from g and the estimate is now $\hat{I} = \frac{1}{N}\sum_{i=1}^{N} f(X_i)h(X_i)/g(X_i)$.

```
set.seed(1)
1-pnorm(3) #[1] 0.001349898
N <- 100000
h <- function(s, x1){ as.numeric(s > x1) }
x1 <- rnorm(N)
Ih1 <- h(x1, 3)
mean(Ih1) #[1] 0.00132
var(Ih1) #[1] 0.001318271
x2 <- rnorm(N,4,1)
Ih2 <- dnorm(x2)*h(x2,3)/dnorm(x2,4,1)
mean(Ih2) #[1] 0.001346459
var(Ih2) #[1] 9.536561e-06
```

2.3 Example: Posterior mean of normal

For example, suppose that $X_i \sim N(\theta, \sigma^2)$ and $\theta \sim N(a, b^2)$. Note that $w_j = \frac{f(\theta_j|y)}{g(\theta_j)} \propto \frac{f(\theta_j)\prod_{i=1}^{n-1} f(y_i|\theta)}{g(\theta_j)} f(y_n|\theta_j)$ When a new data point arrives, we simply update the weights as $w_j^{n+1} = w_j^n * f(y_{n+1}|\theta)$

```
theta0 <- 0
sigma <- 1
a <- 0
b <- 10
n <- 100
#candidate, g: unif(-5,5)
N <- 10000
theta <- runif(N, -5, 5)
#Set the weights based on the prior
logw <- dnorm(theta,a,b,log=TRUE) -
dunif(theta,-5,5,log=TRUE)
w <- exp(logw-max(logw))
w <- w/sum(w)
plot(theta,w,type="h")
par(ask=TRUE)

for(i in 1:n){
# more observations
y <- rnorm(1,theta0,sigma)
# update the weights
logw <- logw+dnorm(y,theta,sigma,log=TRUE)
w <- exp(logw-max(logw))
w <- w/sum(w)
postmean<-sum(w*theta)
if(i%%10==0){
plot(theta,w,type="h")
}
}
```

3 Rejection sampling

Suppose we can easily sample from some density $g(\theta)$, but what we want is a sample from $h(\theta)$, and we know $h(\theta)$ up to some proportionality constant. That is, suppose we know $k(\theta)$, where $h(\theta) = k(\theta)/\int k(\theta)d\theta$.

Moreover, suppose that we can find $M > 0$ such that

$$k(\theta) \leq Mg(\theta)\ \forall\theta \quad \text{(envelope condition)}$$

Then the following algorithm produces B iid draws from $h(\theta)$.

1. Draw $\theta^{cand} \sim g(\theta)$.
2. Generate $u \sim Unif(0,1)$.

3. If $u \leq k(\theta^{cand})/Mg(\theta^{cand})$, accept θ^{cand}; otherwise reject it.

Repeat 1-3 until B values of θ^{cand} have been accepted.

3.1 posterior density

Rejection sampling takes samples from a distribution that resembles the posterior, and then thins these samples to obtain draws from the posterior. The approximate density M is called the envelope function.

1. Draw $\theta^{cand} \sim g(\theta)$.
2. Generate $u \sim Unif(0, 1)$.
3. If $u \leq f(\theta^{cand}|x^n)/Mg(\theta^{cand})$, accept θ^{cand}; otherwise reject it.

4 The Metropolis-Hastings Algorithm

We introduce Markov chain Monte Carlo (MCMC) methods. The idea is to construct a Markov chain $X_1, \ldots,$ whose stationary distribution is f.

The Metropolis-Hastings algorithm is a specific MCMC method that works as follows. Let $q(y|x)$ be an arbitrary, friendly distribution. The conditional density $q(y|x)$ is called the proposal distribution. The Metropolis-Hastings algorithm

1. Choose X_0 arbitrarily.
2. Generate a proposal or candidate value $Y \sim q(y|X_i)$
3. Evaluate $r \sim r(X_i, Y)$ where
$$r(x, y) = \min\left\{\frac{f(y)q(x|y)}{f(x)q(y|x)}, 1\right\}$$
4. Set
$$X_{i+1} = \begin{cases} Y & \text{with probability } r \\ X_i & \text{with probability } 1 - r \end{cases}$$

By construction, $X_0, X_1, \ldots$ is a Markov chain.

- f is a stationary distribution if $f(x) = \int f(y)p(y, x)dy$
- detailed balance holds for f if
$$f(x)p(x, y) = f(y)p(y, x)$$

where $p(x, y)$ the probability of making a transition from x to y. (A stationary distribution of a Markov chain is a probability distribution that remains unchanged in the Markov chain as time progresses.)

Detailed balance implies that f is a stationary distribution. since,

$$\int f(y)p(y,x)dy = \int f(x)p(x,y)dy = f(x)\int p(x,y)dy = f(x)$$

Our goal is to show that f satisfies detailed balance which will imply that f is a stationary distribution for the chain. Consider two points x and y. Without loss of generality, assume that $f(x)q(y|x) > f(y)q(x|y)$. Then,

$$r(x,y) = \frac{f(y)q(x|y)}{f(x)q(y|x)}$$

and $r(y, x) = 1$. Now $p(x, y)$ is the probability of jumping from x to y. This requires two things: (i) the proposal distribution must generate y, and (ii) you must accept y. Thus,

$$p(x,y) = q(y|x)r(x,y) = \frac{f(y)}{f(x)}q(x|y)$$

Therefore,

$$f(x)p(x,y) = f(y)q(x|y)$$

Similarly, we can show $p(y, x) = q(x|y)r(y, x) = q(x|y)$, and hence,

$$f(y)p(y,x) = f(y)q(x|y)$$

. and it leads to detailed balance.

4.1 Example: Cauchy distribution

$$f(x) = \frac{1}{\pi}\frac{1}{1+x^2}$$

Our goal is to simulate a Markov chain whose stationary distribution is f. We take $q(y|x) \sim N(x, b^2)$. (the proposal density q is symmetric)

$$r(x,y) = \min\left\{\frac{f(y)}{f(x)}, 1\right\} = \min\left\{\frac{1+x^2}{1+y^2}, 1\right\}$$

```
set.seed(1)
N <- 10000
Cauchy=function(x){
1/(1+x^2)
}
x <- rep(runif(1), N) #initial value
for(i in 1:(N-1)){
```

```
u1 <- runif(1)
y1 <- rnorm(1, 0, 2)
r1 <- min( (Cauchy(y1)*dnorm(x[i],0,2))/(Cauchy(x[i])*dnorm(y1,0,2)), 1)
if(u1 < r1){
x[i+1] <- y1}else{
x[i+1] <- x[i]
}
}
xs2 <- x
par(mfrow=c(1,2))
plot(dcauchy, -10, 10)
hist(xs2,xlim=c(-10,10))
```

5 Gibbs Sampling

The Gibbs sampler is a special case of Metropolis-Hastings where the proposal distributions loops over the conditional distribution

1. $X_{n+1} \sim f_{X|Y}(x|Y_n)$
2. $Y_{n+1} \sim f_{Y|X}(y|X_{n+1})$.

5.1 Example: Binomial-beta distribution

$X \sim Bin(n, \theta)$ and $\theta \sim Beta(a, b)$. The posterior distribution is Binomial-beta $Beta(a + x, n - x + b)$ Our tool is Gibbs sampling: generate $\theta_{(i)} \sim Beta(\theta|x_{(i)}$ and $x_(i) \sim f(x|\theta_{(i)})$.

```
set.seed(1)
a <- 2
b <- 4
n <- 16
N <- 100
x= rep(0,N); th=rep(0,N)
# initial value
x[1]=1; th[1]=0.5
# Gibbs iterations
for (i in 2:N)
{
x[i] =rbinom(1,size=n,prob=th[i-1])
th[i] = rbeta(1,a+x[i],b+n-x[i])
}
```

reference

https://www4.stat.ncsu.edu/~reich/st740/Computing1.pdf