

# Bayesian Inference and Computation for Data Scientists

## Assessment 2: Monte Carlo algorithms project: DNA Methylation of cytosines

Rushmila Islam

2024-04-06

### Contents

1	Plotting the density function $f(x)$	1
2	Implementation of the Accept-Reject algorithm	2
3	Comparison of the observed acceptance rate with the theoretical acceptance rate	4
4	Implementation of the Importance sampling algorithm	5
5	Comparison with the target density	6
6	Comparison of the two algorithms	7

### 1 Plotting the density function $f(x)$

When analysing Methylation data we may need to simulate values from mixtures of Beta distributions. In this assessment, we simulate values from a mixture of Beta distributions using Accept/Reject sampling and importance sampling.

Let  $X$  be a random variable describing the frequency of Methylation and  $X_1, X_2 \dots X_N$  are Independent and identically distributed random variables according to the density function  $f(x)$ . A mixture of the three Beta distributions as follows:

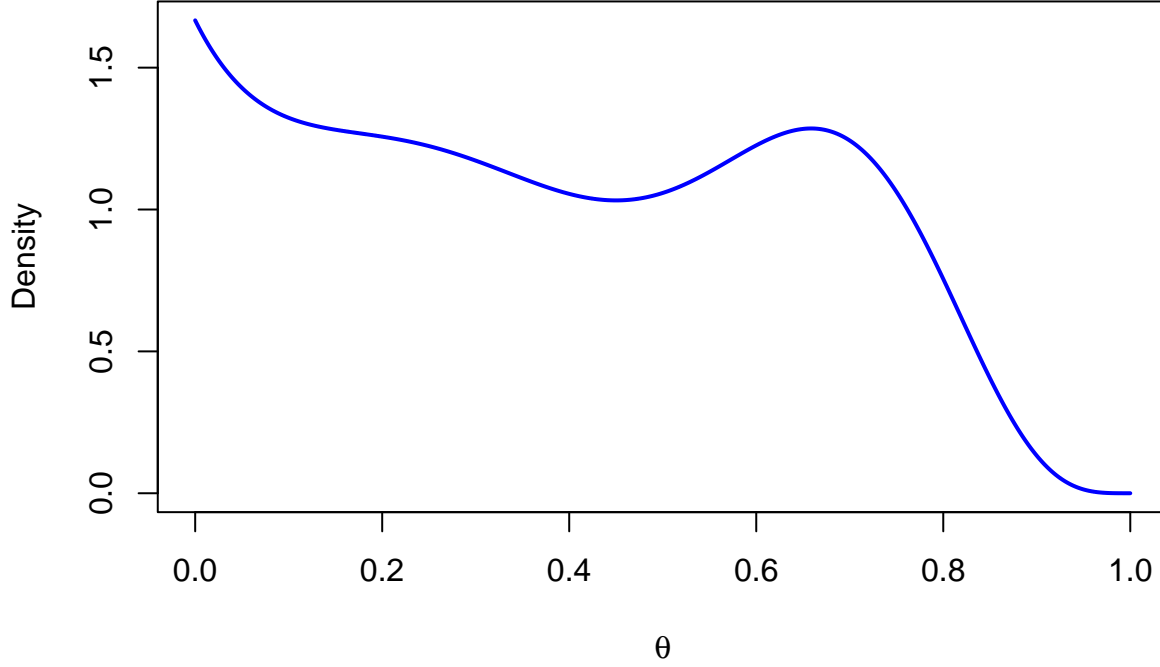
$$f(x) = \frac{1}{3}Be(\alpha_1 = 1, \beta_1 = 5) + \frac{1}{3}Be(\alpha_2 = 3, \beta_2 = 5) + \frac{1}{3}Be(\alpha_3 = 10, \beta_3 = 5)$$

Plotting the density function  $f(x)$ :

```
# Define the density function
f <- function(x) {
  1/3 * dbeta(x, 1, 5) + 1/3 * dbeta(x, 3, 5) + 1/3 * dbeta(x, 10, 5)
}

# Q1: Plot the density function f(x)
par(mfrow=c(1,1))
th=seq(0,1,length=10000)
plot(th, f(th), type = "l", lwd = 2,
      xlab = expression(theta), ylab = "Density",
      main = "Density of mixture Beta distributions",
      col='blue')
```

## Density of mixture Beta distributions



## 2 Implementation of the Accept-Reject algorithm

There are two versions of the Accept-Reject algorithm. As we are asked to use the acceptance probability that is used in version 1 of the algorithm, so here we implement the version of the Accept-Reject algorithm.

---

### Accept/Reject algorithm (version 1)

1. Generate  $\theta$  from  $g(\theta)$ .
  2. Accept  $\theta$  with probability  $\frac{\pi(\theta|\mathbf{x})}{Kg(\theta)}$  ( generate  $u$  from  $U(0, 1)$  and accept if  $u \leq \frac{\pi(\theta|\mathbf{x})}{Kg(\theta)}$  )
- 

Let, the posterior distribution of  $\theta$  :

$$\pi(\theta|X) = \pi(\theta|X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$$

where  $x = (x_1, x_2, \dots, x_n)$  .

To implement Accept-Reject algorithm we need a proposal distribution. We need a sampling distribution  $g(\theta)$ . We are given the proposal distribution is  $Unif(0, 1)$ .

$$g(\theta) = Unif(0, 1)$$

We need to compute the  $K$  in order to gain the maximum efficiency.

$$K = \max_{\theta} \frac{\pi(\theta|\mathbf{x})}{g(\theta)}$$

As in this case,  $g(\theta)$  is a  $Unif(0, 1)$  so,  $K$  will be the maximum value of the density  $\pi(\theta|\mathbf{x}) = f(x)$  .

We define value  $K$ , i.e the maximum of the density and associate that with corresponding  $x$  value. From the distribution plot, we can see that the maximum of the density is  $\theta_{max} = 0$ , when  $\theta = 0$ . We are setting the seed value in  $R$  at 1234 to ensure the reproducibility of random number generation.

```
set.seed(1234)
```

Now, we simulate 10,000 values from  $Unif(0, 1)$  distribution.

```
N=10000
theta=runif(N)
```

We have implemented below is the code to accept or reject the simulated values according to the acceptance probability. Here we have used algorithm version 1 of accept-reject algorithm.

```
set.seed(1234)
N=10000
theta = runif(N,0,1) #generate samples from proposal distribution
u <- runif(N,0,1) #generate u; 10,000 values from a uniform distribution
theta_max=0.0 #value of theta where density point is maximum

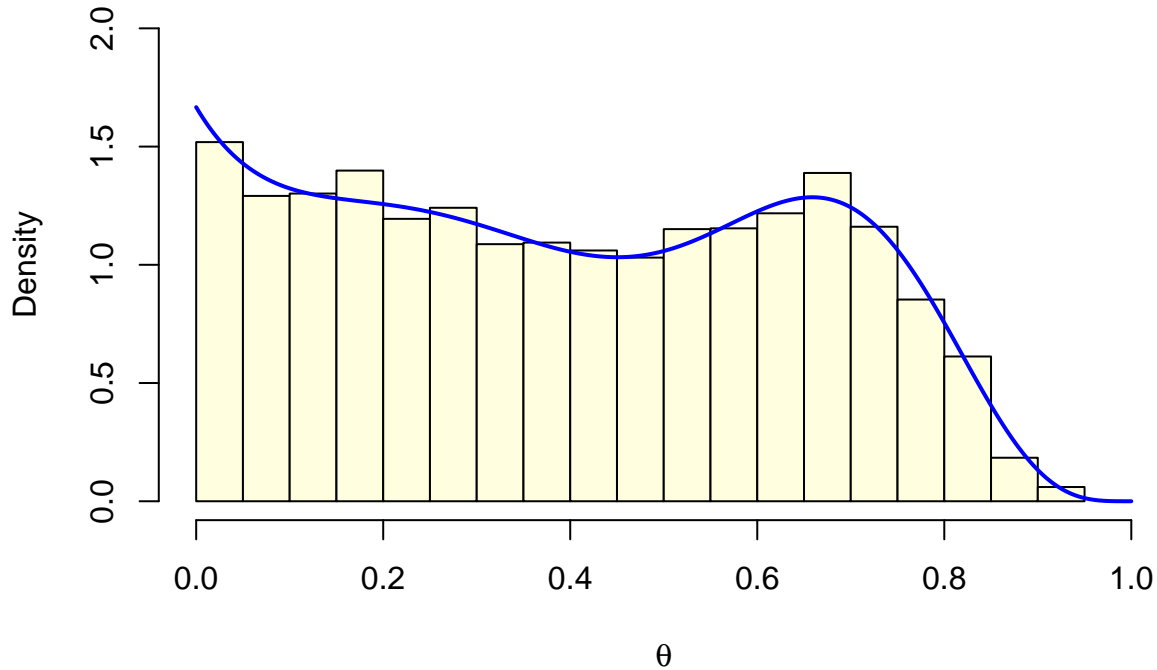
K <- f(theta_max) #maximum value of the density
K

## [1] 1.666667

#####
# A/R Algorithm 1
#####
theta2=c() #theta accepted
for(i in 1:N){
  #theta_prop <- runif(1,0,1)
  acc_prob <- f(theta[i])/K
  #u <- runif(1,0,1)
  if(u[i]<acc_prob){
    theta2 <- c(theta2, theta[i])
  }
}

# Plotting accept-reject sampling from algorithm 1 data against f(x)
ymax <-2
par(mfrow=c(1,1))
hist(theta2, probability=T, xlab=expression(theta), ylab = "Density",
      main = "Accept-Reject sampling (Algorithm 1)", xlim=c(0,1), ylim=c(0,ymax), col='lightyellow')
lines(th, f(th), col="blue", lwd=2)
```

## Accept-Reject sampling (Algorithm 1)



Acceptance rate for the Accept-Reject algorithm (version 1):

```
# Acceptance rate for algorithm 1
acceptance_rate_1 <- 1/K #length(theta2)/N
cat("Acceptance rate (Algorithm 1) = ", acceptance_rate_1, "\n") #0.6
```

```
## Acceptance rate (Algorithm 1) = 0.6
```

### 3 Comparison of the observed acceptance rate with the theoretical acceptance rate

At this point we have to calculate the acceptance probability theoretically using the main density function.

From the distribution we get  $\theta_{max} = 0$

And the maximum beta density at this value is equal to:

$$\frac{1}{3} \frac{\Gamma(1+5)}{\Gamma(1)\Gamma(5)} (0)^{(1-1)} (1-0)^{(5-1)} + \frac{1}{3} \frac{\Gamma(3+5)}{\Gamma(3)\Gamma(5)} (0)^{(3-1)} (1-0)^{(5-1)} + \frac{1}{3} \frac{\Gamma(10+5)}{\Gamma(1)\Gamma(5)} (0)^{(10-1)} (1-0)^{(5-1)}$$

```
theta_max <- 0.0
part1 <- gamma(6)/(gamma(1)*gamma(5))
part2 <- gamma(8)/(gamma(3)*gamma(5))
part3 <- gamma(15)/(gamma(10)*gamma(5))
final <- (1/3) * (1-theta_max)^4 * (part1 + ((theta_max)^2*part2) + ((theta_max)^9*part3))
final
```

```
## [1] 1.666667
```

Therefore,  $Kg(\theta) = 1.666667 \cdot Unif(0, 1)$

Accept the values with probability -

$$\frac{\pi(\theta|\mathbf{x})}{1.666667.Unif(0,1)}$$

The probability of acceptance is  $\frac{1}{K}$  ; therefore

$$\frac{1}{K} = \frac{1}{1.666667} = 0.5999999$$

So the probability of acceptance rate is approximately 60%.

$$\frac{1}{K} = \frac{1}{1.666667} = 0.5999999$$

So the probability of acceptance rate is approximately 60% which is similar to the computed acceptance rate by algorithm 1 (60%).

## 4 Implementation of the Importance sampling algorithm

Here we use the importance sampling method for sampling.

---

### Importance sampling algorithm

1. Generate  $\theta^{(i)}$  from  $g(\theta)$ .
2. Give  $\theta^{(i)}$  a weight  $w^{(i)} \propto \frac{\pi(\theta^{(i)}|\mathbf{x})}{g(\theta^{(i)})}$

Then we obtain samples  $(\theta^{(i)}, w^{(i)}), \dots, (\theta^{(N)}, w^{(N)})$  , which are weighted samples from  $\pi(\theta|\mathbf{x})$ .

---

To implement this algorithm we are using the the same 10,000 values generated from the  $Unif(0,1)$  for the implementation of the Accept-Reject algorithm.

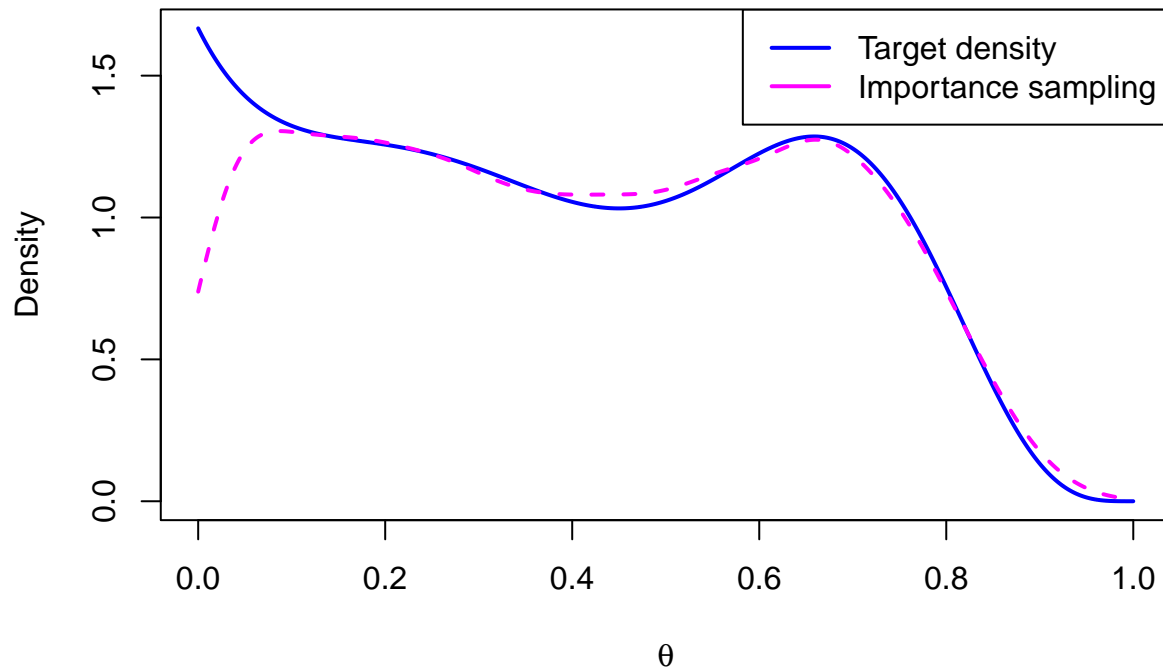
Below code shows the implementation of Importance sampling with the weights:

```
w=f(theta)
W=w/sum(w)
d=density(theta, weights=W, from=0, to=1)

## Warning in density.default(theta, weights = W, from = 0, to = 1): Selecting
## bandwidth *not* using 'weights'

# Plotting IS
par(mfrow=c(1,1))
plot(th,f(th), type="l", lwd=2, col="blue",
      xlab=expression(theta), ylab="Density",
      main="Importance sampling")
lines(d, col="magenta", lwd=2, lty=2)
legend("topright", legend = c("Target density", "Importance sampling"),
      col = c("blue", "magenta"), lty = 1, lwd = 2)
```

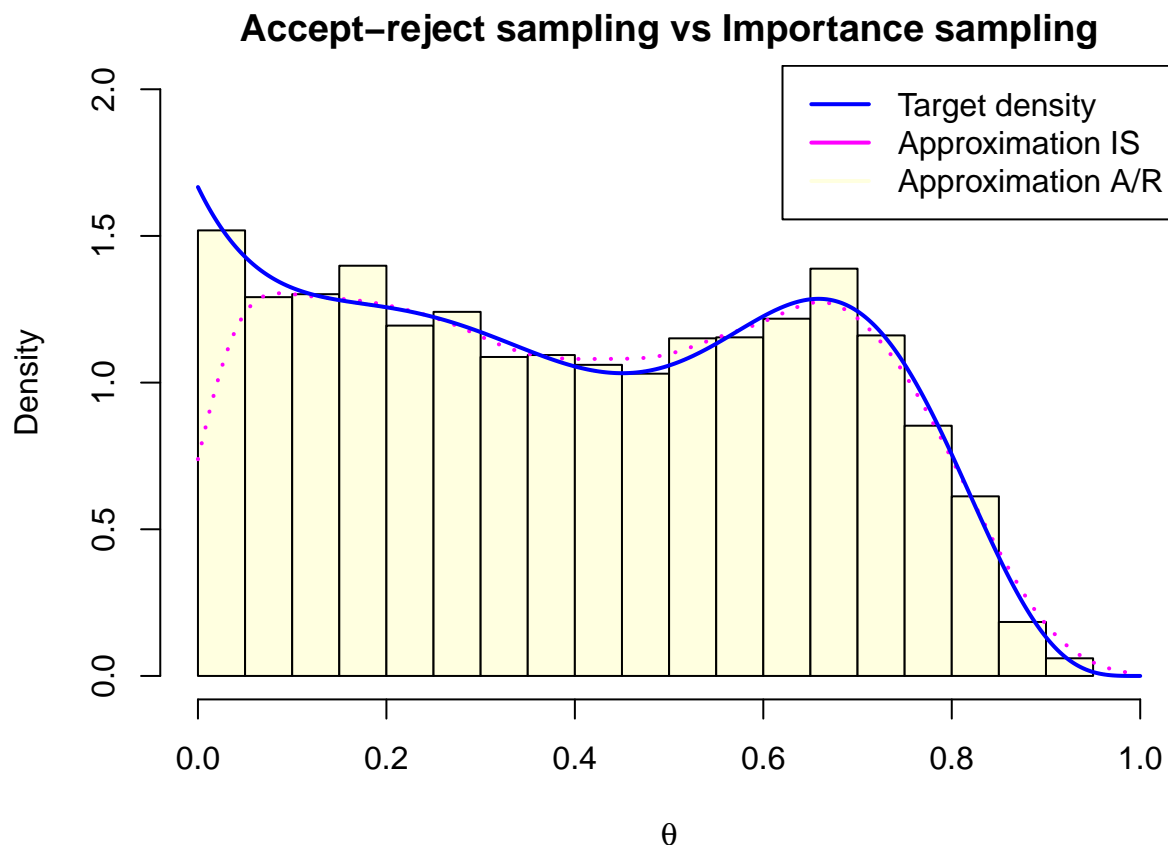
## Importance sampling



## 5 Comparison with the target density

In this section we compare on the same plot the target density, the density of the accepted values for the accept/reject algorithm, and the density of the values weighted by Importance sampling.

```
par(mfrow=c(1,1), mar=c(4,4,2,2))
hist(theta2, probability=T,
      xlab=expression(theta), ylab="Density", main="Accept-reject sampling vs Importance sampling",
      xlim=c(0,1),ylim=c(0,ymax), col="lightyellow")
lines(d, col="magenta", lwd=2, lty=3) #IS
lines(th,f(th),col="blue",lwd=2) #Target density
legend("topright", legend = c("Target density", "Approximation IS", "Approximation A/R"),
      col = c("blue", "magenta", "lightyellow"), lty = 1, lwd = 2)
```



## 6 Comparison of the two algorithms

Which of the two algorithms seems to approximate the target distribution better?

According to the strong law of large numbers, an infinite sequence  $X_1, X_2, \dots$  of independent and identically distributed (i.i.d) random variables with expected value  $E[X_1] = E[X_2] = \dots = \mu$ . The strong law of large numbers states that the sample average converges almost surely to the expected value which is population mean.

To understand the performance, we compute the approximation or the mean for the two sampling methods and compare with the actual mean of the sample data:

```
sample_true_mean_ <- sum(theta)/length(theta)
sample_true_mean_
```

```
## [1] 0.5003123
```

```
ar_approximation <- mean(theta2) #A/R approximation: 0.4017751
ar_approximation
```

```
## [1] 0.4017751
```

```
is_approximation <- sum(w * theta)/sum(w) #IS approximation: 0.4048396
is_approximation
```

```
## [1] 0.4048396
```

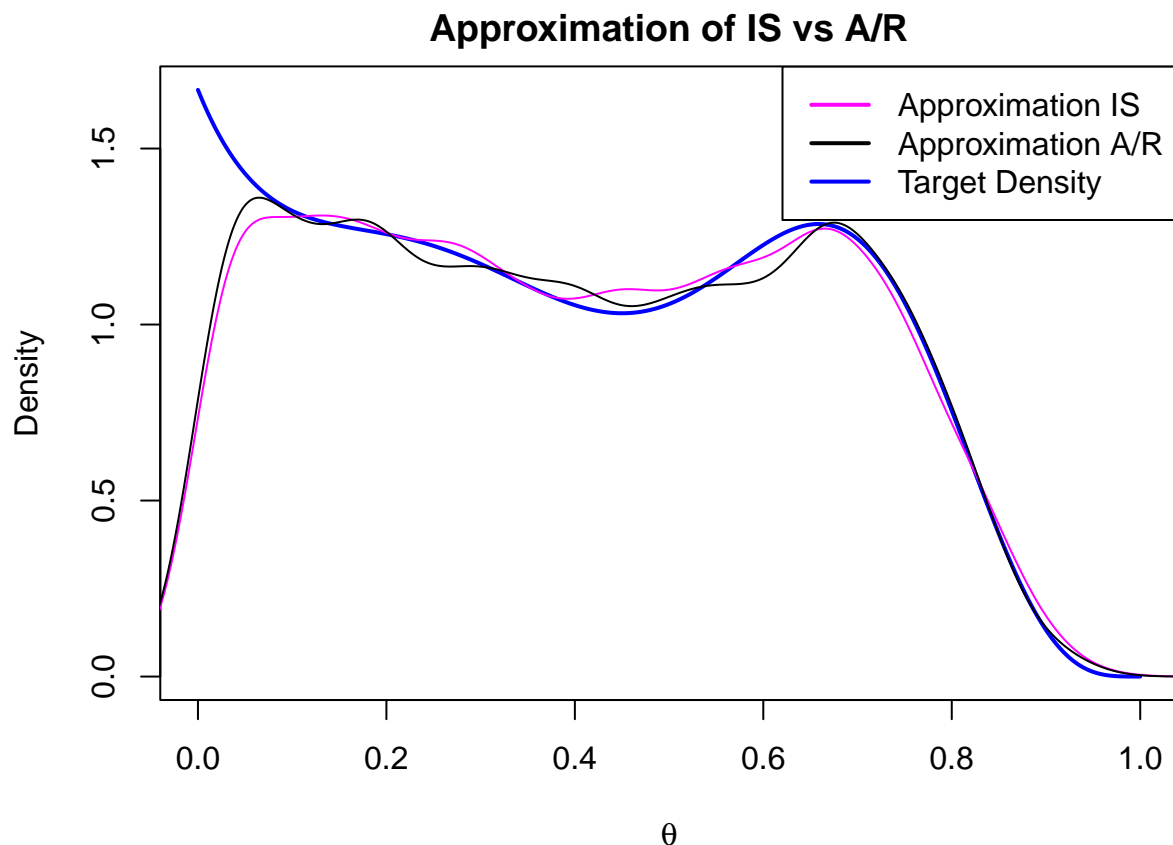
Quantile calculation for the two sampling methods:

```
# Quantile
par(mfrow=c(1,1), mar=c(4,4,2,2))
plot(th,f(th),col="blue",type="l", lwd=2,
      xlab=expression(theta), ylab="Density",
      main="Approximation of IS vs A/R") #Target density

theta_IS <- sample(theta, size=N, prob=W, replace=T)
lines(density(theta_IS), col="magenta")
quantile(theta_IS, probs=c(0.25, 0.50, 0.75))

##      25%      50%      75%
## 0.1851954 0.3967037 0.6202554

theta_AR <- sample(theta2, size=N, replace=T)
lines(density(theta_AR), col="black")
legend("topright", legend = c("Approximation IS", "Approximation A/R", "Target Density"),
      col = c("magenta", "black", "blue"), lty = 1, lwd = 2)
```



```
quantile(theta_AR, probs=c(0.25, 0.50, 0.75))
```

```
##      25%      50%      75%
## 0.1802941 0.3941427 0.6227623
```

We can observe from the quantile data and the mean calculation that, both methods results very close samples for the target distribution. Their mean approximation is also very close to each other ( $\sim 0.40$ ) and almost close the actual sample mean ( $\sim 0.50$ ).

**Which one you would use and why?**



If we have to choose one method over another, importance sampling would be the choice for this target distribution as its generating slightly more sample data according to the quantile information. Another reason is we have observed that the efficiency of the Accept-Reject sampling is quite low, only 60% acceptance rate. This means 40% of the sample data will be discarded if we use this method. One more advantage for choosing the importance sampling is we do not need to compute the K value, which could be very difficult to find if the distribution has many maximum points whereas Importance sampling uses weighted samples that is proportional to the the target posterior distribution.