Bayesian Computing

Name :- Preksha A. Patel
Sapid :- 60004210126
Branch :- Computer Engineering
Div :- C2

Experiment no. 7

Aim :- Estimation for the two-parameter exponential distribution: using simulated values from the posterior, find the posterior mean and posterior standard deviation.

Theory :-

Sampling Importance Resampling (SIR) Algorithm :-

The Sampling Importance Resampling (SIR) algorithm is a Monte Carlo method used for approximating complex distributions by drawing samples from an importance distribution & then resampling to assign higher weights to more relevant samples. The key steps include :-

1. Sampling :-
   Generate a set of N samples (x_i) from an importance distribution $q(x)$. Compute the importance weights (w_i) proportional to the target distributions $p(x)$:- $w_i = \frac{p(x_i)}{q(x_i)}$

2. Normalization :-
   Normalize the weights to make their sum to 1 :-
   $\hat{w}_i = \frac{w_i}{\sum_{i=1}^{N} w_i}$

3. Resampling :- Draw N samples with replacement from the set of x_i with probabilities given by the normalized weights $\hat{w}_i$

Usage of SIR simulation for Estimation of Two-parameter Exponential Distribution :- Use SIR simulation to generate samples from the posterior distribution of the two-parameter exponential distribution. Then, calculate the posterior mean & posterior standard deviation using the simulated values.

1. Two - Parameter Exponential Distribution : The PDF for 2-parameter exponential distribution is given by : $f(x \mid \lambda, \theta) = \lambda e^{-\lambda(x-\theta)}$

2. Posterior Distribution : Assume prior distribution $p(\lambda, \theta)$ and use Baye's theorem to obtain posterior distribution.

$$P(\lambda, \theta \mid \text{data}) \propto f(\text{data} \mid \lambda, \theta) \times p(\lambda, \theta)$$

3. SIR ~~Sa~~ Simulation :

Generate samples (λ_i, θ_i) using the SIR algorithm from the posterior distribution.

4. Posterior Mean and standard Deviation :

$$\text{Posterior Mean} = \frac{\sum_{i=1}^{N} \lambda_i}{N}$$

$$\text{Posterior standard deviation} = \sqrt{\frac{\sum_{i=1}^{N} (\lambda_i - \text{Posterior Mean})^2}{N}}$$

Conclusion :

In this experiment, we successfully implemented the SIR algorithm for estimating the parameters of the experimental distribution

SAP ID.: 60004210126 Name: Preksha Ashok Patel

Batch: C2-1

Subject: Bayesian Computing Laboratory Semester: VII

Experiment No. 7

Aim:

Estimation for the two-parameter exponential distribution: Using your simulated values from the posterior, find the posterior mean and posterior standard deviation.

Code:

Import Libraries:

```
library(MASS) # for multivariate t-distribution
```

Define the parameters for the exponential distribution:

```
lambda <- 1 # rate parameter for the exponential distribution
theta <- 1 # scale parameter for the exponential distribution
```

Define the proposal density function (multivariate t-distribution):

```
proposal_density <- function(x, df, mu, Sigma) {

dmvt(x, df = df, mu = mu, Sigma = Sigma)

}
```

SIR Step 1: Generate a sample from the proposal density:

```
n <- 1000 # number of samples df <- 2 # degrees of
freedom for the t-distribution mu <- c(lambda, theta) #
mean vector Sigma <- diag(c(1, 1)) # covariance matrix
proposal_sample <- rmvnorm(n, mean = mu, sigma =
Sigma)
```

SIR Step 2: Calculate the weights for each sample:

```
weights <- dexp(proposal_sample[, 1], rate = lambda) * dexp(proposal_sample[, 2], rate =
1/theta) weights <- weights / sum(weights) # normalize
the weights
```

SIR Step 3: Perform the resampling step:

```
resampled_indices <- sample(1:n, size = n, replace = TRUE, prob = weights)
resampled_sample <- proposal_sample[resampled_indices, ]
```

Calculate the posterior mean and standard deviation:

```
posterior_mean <- colMeans(resampled_sample) posterior_sd <-
apply(resampled_sample, 2, sd) print(paste("Posterior mean
(lambda): ", posterior_mean[1])) print(paste("Posterior mean (theta):
", posterior_mean[2])) print(paste("Posterior standard deviation
(lambda): ", posterior_sd[1])) print(paste("Posterior standard
deviation (theta): ", posterior_sd[2]))
```

Visualize posterior sample:

```
plot(resampled_sample[, 1], resampled_sample[, 2], xlab = "Lambda", ylab = "Theta",
main
= "Posterior sample")
```

Output:

Calculate posterior mean and standard deviation:

```
[1] "Posterior mean (lambda):  0.743075390295787"
```

```
[1] "Posterior mean (theta):  0.76260856601354"
```

```
[1] "Posterior standard deviation (lambda):  0.587616872814157"
```

```
[1] "Posterior standard deviation (theta):  0.611449920236162"
```

Visualize the Posterior Sample: