1. Suppose that you have only have access to a routine that produces samples U from the Uniform [0, 1] distribution. Most packages used for statistical computing, such as R or MATLAB have access to such a routine.

(a) Write a program that uses the inversion method to sample from a Binomial (10, 1/3) distribution. Run the program and plot a histogram of 1, 000 samples. Recall that here you will need to make use of the pseudo-inverse of the cumulative distribution function.

Solution:

To sample from Binomial (n, p) distribution using inversion method we need to consider CDF of binomial distribution. The CDF is:

$$F(x) = P(X \leq x) = \sum_{i=0}^{x} \binom{n}{i} p^{i} \, q^{(n-i)}$$

For a given Uniform random sample U ~ Uniform (0,1) and transforming it using the inverse of the CDF of binomial dstribution. We simulate U and find the smallest x such that the cumulative probability P(X ≤ x) is greater than or equal to U.

Result from R code:

The histogram represents the frequency distribution of 1000 samples drawn from Binomial (10, 1/3) distribution. The values are mostly clusters around 2 to 4 where the mean of the distribution lies.

(b) Now, using what you know about Binomial distributions, write a program that uses a combination of the inversion and transformation methods to sample from the same Binomial (10, 1/3) distribution. Run the program and plot a histogram of 1, 000 samples. For this part, your answer should include a short proof of the correctness of your approach as well as the program itself.

Solution:

Here, we use the combination of both inversion and transformation methods. This combinantion is useful when we want to reduce computational effort of the target distribution.

Inversion method: as explained in (a), the inversion method relies on the CDF of Binomial distribution. By generating U ~ Uniform (0,1) and finding the smallest x such that F(x) ≥ U.

Transformation method: It involves using a different related distribution here it is Bernouli (p) distribution to get sample from Binomial (n, p) distribution.

For Binomial distribution, we can transform a sum of Bernoulli trials with each probability of success is (p). For Binomial (n=10, p=1/3), We can simulate the sum of 10 Bernoulli trials each with probability p=1/3.

Proof of Correctness:

Each Bernoulli trial is an independent random variable with probability p = 1/3. The sum of these independent Bernoulli trials gives the Binomial distribution.

That means sampling from Binomial distribution can be done by generating n Bernoulli samples. This transformation method works well because it has binary outcomes and can be compared with Uniform.

Result from R code:

The plotted histogram closely matches the Binomial (10, 1/3) distribution.

(c) We know that the expectation of a Binomial (10, 1/3) distribution is exactly 10/3. However, let's pretend that we don't know its true value and want to estimate it using elementary Monte Carlo integration. Use 100 samples generated with one of the methods above to estimate the expectation of a Binomial (10, 1/3) distribution and provide the standard error of the estimate and approximate confidence bounds as per the Central Limit Theorem. You should briefly restate the theory behind this approximation before applying it: simply putting the answer with confidence bounds will not earn full marks.

Solution:

We know the expectation of Binomial (n, p) is n * p. Here, it is 10 * 1/3 = 10/3.

We can also estimate the expectation of a Binomial distribution using Monte Carlo simulation.

Monte Carlo integration: It is a technique to estimate the expected value of a random variable by generating samples from that distribution and computing the mean. In this case, we approximate the expected value of binomial distribution by taking the average of simulated random samples from it.

Central Limit Theorem: For sufficiently large sample sizes, the sample mean follows an approximately normal distribution. We will generate 100 samples from binomial distribution using the **inversion** method and compute the sample mean as an estimator for the expectation. The standard error of the mean is given by:

$SE = \sigma / \sqrt{n}$

Where σ is the sample standard deviation and n is the number of samples.

an approximate 95% confidence interval for the expectation: $\hat{\mu} \pm z_{0.025} \cdot \hat{\sigma}/\sqrt{n}$

where, $\hat{\mu}$ is the mean and $z_{0.025} = 1.96$ is the critical value for a 95% confidence interval.

Explanation:

- The mean of the samples is our estimate of the expectation.
- We calculate the standard error and 95% confidence interval using the Central Limit Theorem.

Restatement of the Theory: The Central Limit Theorem states that for large enough sample sizes, the sampling distribution of the sample mean will be approximately normally distributed, regardless of the underlying distribution. The mean of this distribution is the true mean, and the standard deviation is the standard error. The 95% confidence interval is then computed as mean ± 1.96 * SE.

Result from R code:

Estimated expectation: 3.34
Standard error: 0.144404
95% confidence interval: (3.056968, 3.623032)

2. Write a program that draws samples of Poisson(t) random variables using Uniform[0, 1] variables as input. Use the transformation method to do so. Hint: recall that if Xi are i.i.d. Exponential(1) random variables and Sn = Pn i=1 Xi , then P(Sn ≤ t ≤ Sn+1) = e −t t n n! . You may use this result without proof. Run the program with t = 1 and plot a histogram of 1000 samples. Use 10, 100, 1000 and 10000 samples to estimate the mean of the Poisson(1) distribution and provide the standard error and associated confidence bounds on the estimate for each case.

Solution:

The transformation method for generating samples of Poission (t) random variables:

The pmf of Poission (t) distribution,

$P\ (X = n) = \frac{e^{-t}t^{n}}{n!}$, n = 0, 1, 2, …

For t=1, $P\ (X = n) = \frac{e^{-1}}{n!}$

Poission (t) random variables can be generated from U ~ Unifrom (0, 1) distribution based on:

If X_1, X_2, …, X_n are i.i.d. Exponential (1) random variables, and $S_n = \sum_{i=1}^{n} X_i$, then

P (Sn ≤ t <S_{n+1}) = $\frac{e^{-t}t^{n}}{n!}$ Follows Poisson(t) distribution.

The Exponential (λ) can be generated from uniform random variables using the transformation:

If U ~ Uniform(0,1) , $X = -\frac{\log(U)}{\lambda}$ will be an Exponential (λ) random variable.

Here, we use λ =1 for Exponential (1).

To generate Poission (t) random variables:

1. Generate X_1, X_2, …, X_n as i.i.d. Exponential (1) random variables using the transformation method on uniform random variables.
2. Sum the exponential random variables until the sum exceeds t.
3. The number of terms summed before exceeding t is a Poisson random variable.

Result from R code:

Sample Size: 10

Estimated Mean: 1.5

Standard Error: 0.4772607

95% Confidence Interval: 0.564569 2.435431

Sample Size: 100

Estimated Mean: 0.9

Standard Error: 0.09795897

95% Confidence Interval: 0.7080004 1.092

Sample Size: 1000

Estimated Mean: 1.026

Standard Error: 0.03095613

95% Confidence Interval: 0.965326 1.086674

Sample Size: 10000

Estimated Mean: 0.9906

Standard Error: 0.009896512

95% Confidence Interval: 0.9712028 1.009997

3. Finally, let us consider rejection sampling. We saw in class that rejection sampling is a generalpurpose sampling method that needs to be used judiciously in order to make it work efficiently. A poorly chosen proposal distribution will result in a high rejection rate.

(a) Suppose we have a mixture of a N(1, 0.5) and a N(2, 0.1) distribution with weights α1 = 0.2 and α2 = 0.8. Use a rejection sampling method to draw from this mixture with a single normal distribution as a proposal. Choose the best parameters for this distribution (with some justification why you have done so). What is the acceptance rate of your rejection sampler? As a sanity check, you can try implementing the composition method to check if you are getting the right answer.

Solution:

We need to sample from a mixture of two normal distributions, N (1, 0.5) and N (2, 0.1),

with weights $\alpha_1 = 0.2$ and $\alpha_2 = 0.8$

The target distribution is defined as,

$$f(x) = 0.2\,N\,(1, 0.5) + 0.8\,N\,(2, 0.1)$$

Where, N (μ, σ^2) is the normal density function.

We have to choose a single normal distribution g (x) as the proposal distribution. Let the reasonable choice to use a normal distribution:

Mean (μ) should be somewhere between the means of two mixture components.

Standard deviation (σ) should be large enough to cover both normal components.

We can empirically choose μ = 1.5 and σ = 0.8 for the proposal distribution.

To find the acceptance of a sample we need to compare the ratio of target density f (x) and the proposal density g (x).

The acceptance ratio = $\frac{f(x)}{M.g(x)}$ for all x. We have to estimate M such that

M. g(x) ≥ f (x) for all x. This means the proposal density dominates the target density.

Result from R code:

Acceptance rate: 0.6176652

Sanity Check:

The composition method is implemented as a sanity check to ensure the rejection sampling works correctly. The histogram generated from the composition method matches the rejection sampling result which confirms that the rejection sampling is correctly used here.

(b) Consider Y with an Exponential(λ) distribution and X = Y + a where a > 0. Devise a rejection sampler for X using draws of Y as a proposal. What happens to the efficiency of your sampler when a grows? Why does this happen?

Solution:

Givern, Y ~ Exponential(λ)

The pdf of Y is,

$f_Y(y) = \lambda e^{-\lambda y}$, $y \geq 0$ which is the proposal distribution.

Now, X = Y + a, where a > 0. This shifts the distribution of Y by a

So the PDF of X is shifted to the right by a,

$f_X(x) = \lambda e^{-\lambda(x-a)}$, $x \geq$ a This is the target distribution.

Rejection sampling:

In rejection sampling, we want to sample from X using Y as a proposal.

We need to find a constant M such that $f_X(x) \leq M\, f_Y(y)$ for all $x \geq$ a.

Here, the choice of M can be, $M = e^{\lambda a}$ this can be the smallest possible value that satisfies the inequality.

This can be found as, $\frac{f_X(y)}{M\, f_Y(y)} = \frac{\lambda\, e^{-\lambda\,(y-a)}}{M\, \lambda\, e^{-\lambda y}} = \frac{e^{\lambda a}}{M}$

Increasing a:

As a increases, the constant M increases, that means the acceptance of proposed samples will be reduced thus reduces the efficiency of the sampler. As a increases, the target distribution $f_X(x)$ shifts further to the right compared to the proposal distribution $f_Y(y)$.

Efficiency of the sampling:

The efficiency can be affected by the choice of a. As a increases, the acceptance rate is expected to decrease because of the increasing distance of the samples from the target distribution.

Result from R code:

a= 0.5 Acceptance Rate= 0.6071645

a= 1 Acceptance Rate= 0.3616637

a= 2 Acceptance Rate= 0.1418239

a= 5 Acceptance Rate= 0.006565126

Hence, the result for different a values shows that as a increases the acceptance rate decreases.