MTH422A Assignment1

Vijay Soren

2024-01-31

1.

Answer

Y_N will follow $Binomial(N, \theta)$ With $E(Y_N) = N\theta$, $Var(Y_N) = N\theta*(1-\theta)$ and $\bar{Y}_N$ will follow $Binomial(N, \theta)$ with $E(\bar{Y}_N) = \theta$, $Var(\bar{Y}_N) = (\theta * (1 - \theta))/N$ and Support of $\bar{Y}_N$ is $[0, 1]$

```r
set.seed(1)

# Define the parameters
theta <- 0.4
N_values <- c(10, 50, 500)

# Function to generate samples of Y_bar_N
Y_bar_N <- function(N, theta) {
  X <- rbinom(N, 1, theta)  # Generate N Bernoulli samples
  Y <- sum(X)
  Y_bar <- Y / N
  return(Y_bar)
}

# Function to compute the approximate distribution based on CLT
approximate_CLT <- function(N, theta) {
  mu <- theta  # Mean of Y_bar_N
  sigma <- sqrt((theta * (1 - theta))/(N))  # Standard deviation Y_bar_N
  Y_values <- seq(0, 1, length.out = 1000)  # Possible values of Y_bar_N
  CLT_approx <- pnorm(Y_values, mean = mu, sd = sigma)  # CDF of normal distribution
  return(CLT_approx)
}

# Compare CDFs using Kolmogorov-Smirnov distance
for (N in N_values) {
  # Generate samples of Y_bar_N
  Y_bar_N_samples <- replicate(1000, Y_bar_N(N, theta))
  emp_CDF <- ecdf(Y_bar_N_samples)             # Compute empirical CDF
  CLT_CDF <- approximate_CLT(N, theta)    # Compute approximate CDF based on CLT

  # Compute Kolmogorov-Smirnov distance
  ks_distance <- ks.test(Y_bar_N_samples, CLT_CDF)$statistic

  cat("For N =", N, "KS distance:", ks_distance, "\n")  # Print KS distance
}
```

```
## Warning in ks.test.default(Y_bar_N_samples, CLT_CDF): p-value will be
## approximate in the presence of ties

## For N = 10 KS distance: 0.503

## Warning in ks.test.default(Y_bar_N_samples, CLT_CDF): p-value will be
## approximate in the presence of ties

## For N = 50 KS distance: 0.585

## Warning in ks.test.default(Y_bar_N_samples, CLT_CDF): p-value will be
## approximate in the presence of ties

## For N = 500 KS distance: 0.601
```

2.

Answer:

$\bar{X}_N$ will follow $Gamma(N\alpha, N\lambda)$

3

Answer:

```r
library(pracma)
fx <- function(x) 2 * besselK(2 * sqrt(x), nu = 0)
result <- integrate(fx, lower = 0, upper = Inf) # Numerical integration
print(result)
```

```
## 1 with absolute error < 8.1e-06
```

```r
print("Given fx is valid pdf")
```

```
## [1] "Given fx is valid pdf"
```

6

Answer

The conditional distribution of X given $Y_1, ..., Y_T$ will follow $Gamma(\sum_{i=1}^{T} Y_i + a, T + b)$.

7

Answer

Time taken for draw 10000 samples:

```r
library(tictoc)
```

```
##
## Attaching package: 'tictoc'

## The following objects are masked from 'package:pracma':
##
##     clear, size, tic, toc
```

```r
set.seed(1)

Uniform_exp <- function()
{
  accept <- 0
  while(accept == 0)
  {
    X <- runif(1, min = -1.14, max = 1.14) #X~ U(-1.14,1.14)
    Y <- runif(1,min = -1,max = 1.23)  #Y~U(-1,1.23)
    prop <- c(X,Y)

    if( ((prop[1]^2 + prop[2]^2 - 1)^3 - prop[1]^2 *prop[2]^3) <= 0  )
    {
      accept <- 1
      return(prop)
    }
  }
}
# Simulation 10^4 samples from (X^2 + Y^2 - 1)^3 <= X^2*Y^3
N <- 1e4
samp <- matrix(0, ncol = 2, nrow = N)
tic()
for(i in 1:N)
{
  foo <- Uniform_exp()
  samp[i,] <- foo[1:2]
}
toc()
```

```
## 0.11 sec elapsed
```

```r
# Plotting the obtained samples

plot(samp[,1], samp[,2], xlab = "x", ylab = "y",
     main = "Uniform samples from a (X^2 + Y^2 - 1)^3 <= X^2*Y^3" , asp = 1, ylim = c(-2,2))
```

Uniform samples from a (X^2 + Y^2 − 1)^3 <= X^2*Y^3