STAT201 HW3

Sylvia Song

2023-11-26

1.Simulation of Markov Process.

a) Write the Markov process in matrix representation, that is, define the matrix $P \in \mathbb{R}^{3Ö3}$ such that P_{ij} is the probability of transitioning from the node i to j.

```
# Define the transition matrix P
P <- matrix(c(0.2, 0.7, 0.1,  # Transitions from state 1 to states 1, 2, 3
              0.2, 0.5, 0.3,  # Transitions from state 2 to states 1, 2, 3
              0.2, 0.4, 0.4), # Transitions from state 3 to states 1, 2, 3
            byrow = TRUE, nrow = 3)

print(P)
```

```
##      [,1] [,2] [,3]
## [1,]  0.2  0.7  0.1
## [2,]  0.2  0.5  0.3
## [3,]  0.2  0.4  0.4
```

b) Simulate one single realization of the chain, that is, starting from $X_0 = 1$, update the value of X_i using the probabilities defined by the process.

```
# Function to simulate one step of the Markov chain
simulate_step <- function(current_state, P) {
  transition_probs <- P[current_state, ]
  next_state <- sample(1:3, size = 1, prob = transition_probs)
  return(next_state)
}

# Initialize the starting state
X_0 <- 1

# Simulate 10 steps of the Markov chain
chain <- numeric(10) # store the states
chain[1] <- X_0

for (i in 2:10) {
  chain[i] <- simulate_step(chain[i-1], P)
}

print(chain)
```

```
##  [1] 1 2 1 2 3 2 3 1 2 1
```

2.Stationary Distribution.

a) Calculate theoretically the stationary state of the process by finding the vector $\pi_\infty \in \mathbb{R}^3$ such that $\pi_\infty^T = \pi_\infty^T P$. Notice that this is the same as finding the eigenvector with eigenvalue equals one of the matrix P^T. This is the same as solving $(P^T - I)\pi_\infty = 0$. You can solve the linear system of equation numerically or analytically.

```
eigen_values <- eigen(t(P))$values
eigen_vectors <- eigen(t(P))$vectors

# Find the position of the eigenvalue that is closest to 1
index <- which.min(abs(eigen_values - 1))

# Extract the corresponding eigenvector and normalize it to sum to 1
stationary_distribution <- eigen_vectors[,index] / sum(eigen_vectors[,index])

# Since we want a row vector, we transpose the stationary distribution
stationary_distribution <- t(stationary_distribution)

print(stationary_distribution)
```

```
##      [,1]      [,2]      [,3]
## [1,]  0.2 0.5111111 0.2888889
```

Calculate it by hand

$$P = \begin{pmatrix} 0.2 & 0.7 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{pmatrix}$$

We want to find the stationary distribution, we need to solve the equation $\pi_\infty^T(P - I) = 0$:

$$P - I = \begin{pmatrix} -0.8 & 0.7 & 0.1 \\ 0.2 & -0.5 & 0.3 \\ 0.2 & 0.4 & -0.6 \end{pmatrix}$$

The system of linear equations becomes:

$$\begin{aligned} -0.8\pi_1 + 0.7\pi_2 + 0.1\pi_3 &= 0 \\ 0.2\pi_1 - 0.5\pi_2 + 0.3\pi_3 &= 0 \\ 0.2\pi_1 + 0.4\pi_2 - 0.6\pi_3 &= 0 \\ \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned}$$

Solve this equation, we get:

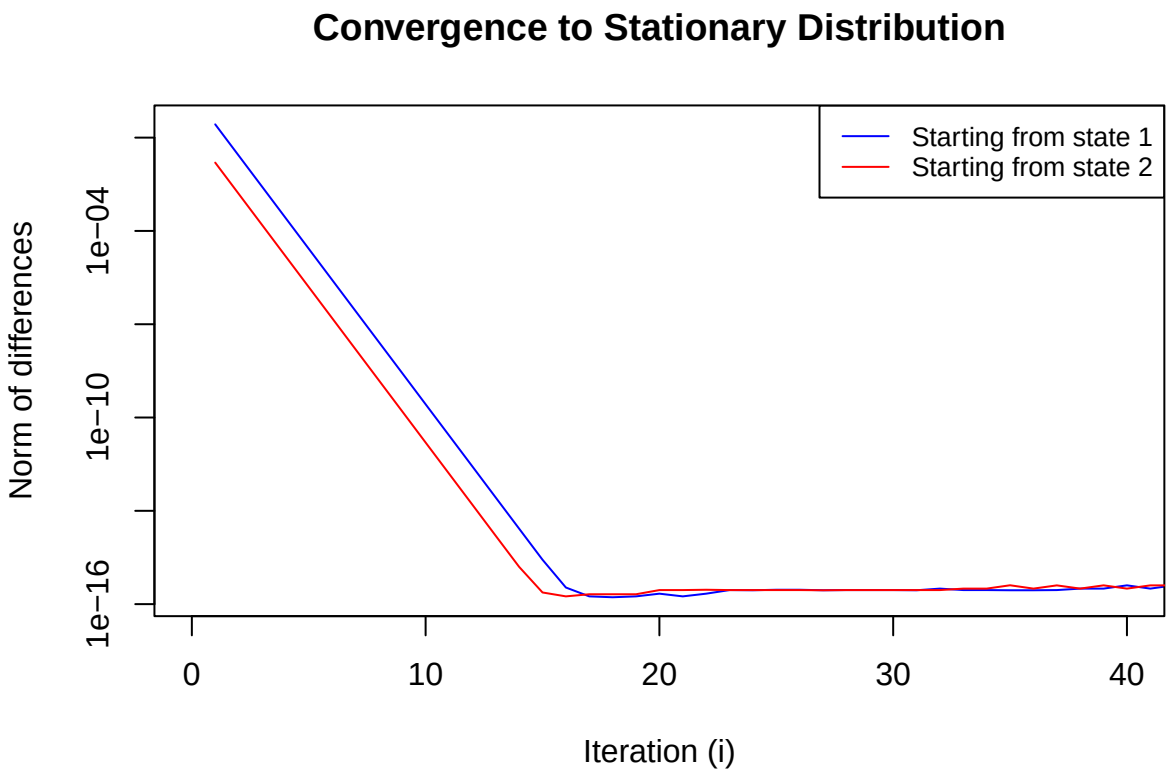
$$\pi_\infty = \begin{pmatrix} 0.2 \\ 0.51111111 \\ 0.28888889 \end{pmatrix}$$

b) Starting now from an initial probability distribution π_0 on the nodes, compute the value of $\pi_i^T = \pi_0^T P^i$ the probability distribution at time i. Show that $\pi_i \to \pi_\infty$ and make a plot of i vs $\|\pi_i - \pi_\infty\|_2^2$. Generate this plot for at least two different initial conditions π_0 and compare.

```
pi_0_a <- c(1, 0, 0) # Starting from state 1
pi_0_b <- c(0, 1, 0) # Starting from state 2 for comparison

# Function to compute the probability distribution after i steps
compute_pi_i <- function(pi_0, P, i) {
  return (pi_0 %*% (P %^% i))
}

iterations <- 100

# Initialize a matrix
norm_differences_a <- numeric(iterations)
norm_differences_b <- numeric(iterations)

# Calculate the norm differences for each step
for (i in 1:iterations) {
  pi_i_a <- compute_pi_i(pi_0_a, P, i)
  pi_i_b <- compute_pi_i(pi_0_b, P, i)

  norm_differences_a[i] <- sqrt(sum((pi_i_a - stationary_distribution)^2))
  norm_differences_b[i] <- sqrt(sum((pi_i_b - stationary_distribution)^2))
}

# Plot
plot(1:iterations, norm_differences_a, log = "y",type='l', col='blue',xlim = c(0,40),
     xlab='Iteration (i)', ylab='Norm of differences',
     main='Convergence to Stationary Distribution')
lines(1:iterations, norm_differences_b, log = "y",col='red',xlim = c(0,40))
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "log" is not a graphical
## parameter
```

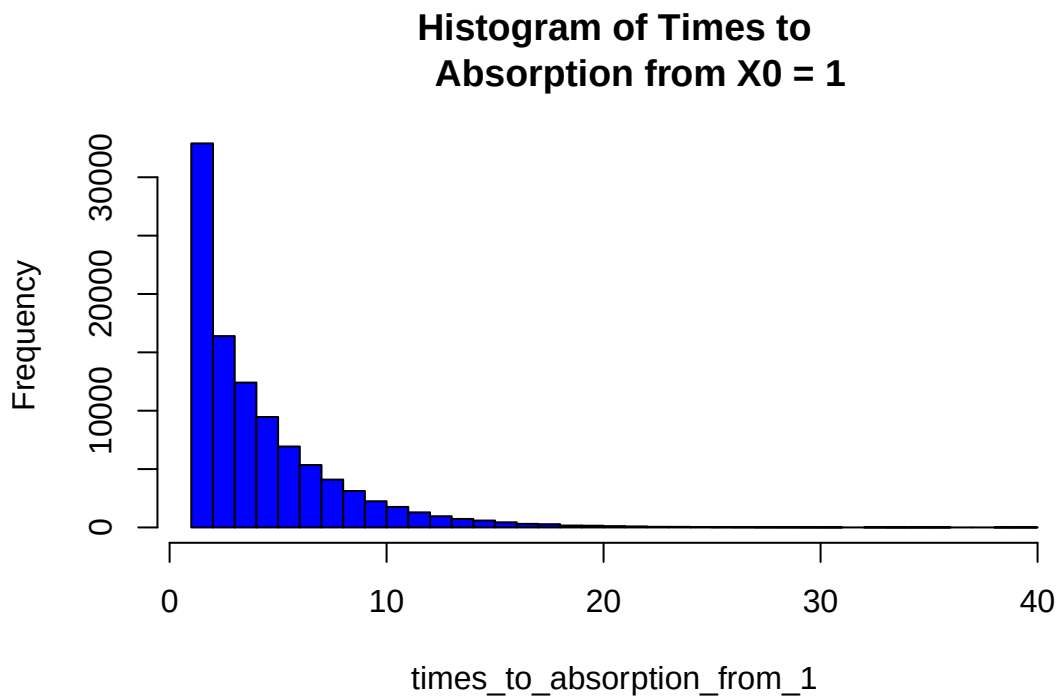
```
legend("topright", legend=c("Starting from state 1", "Starting from state 2"),
       col=c("blue", "red"), lty=1, cex=0.8)
```


Convergence to Stationary Distribution
Starting from state 1
Starting from state 2
Norm of differences
1e−04
1e−10
1e−16
0
10
20
30
40
Iteration (i)

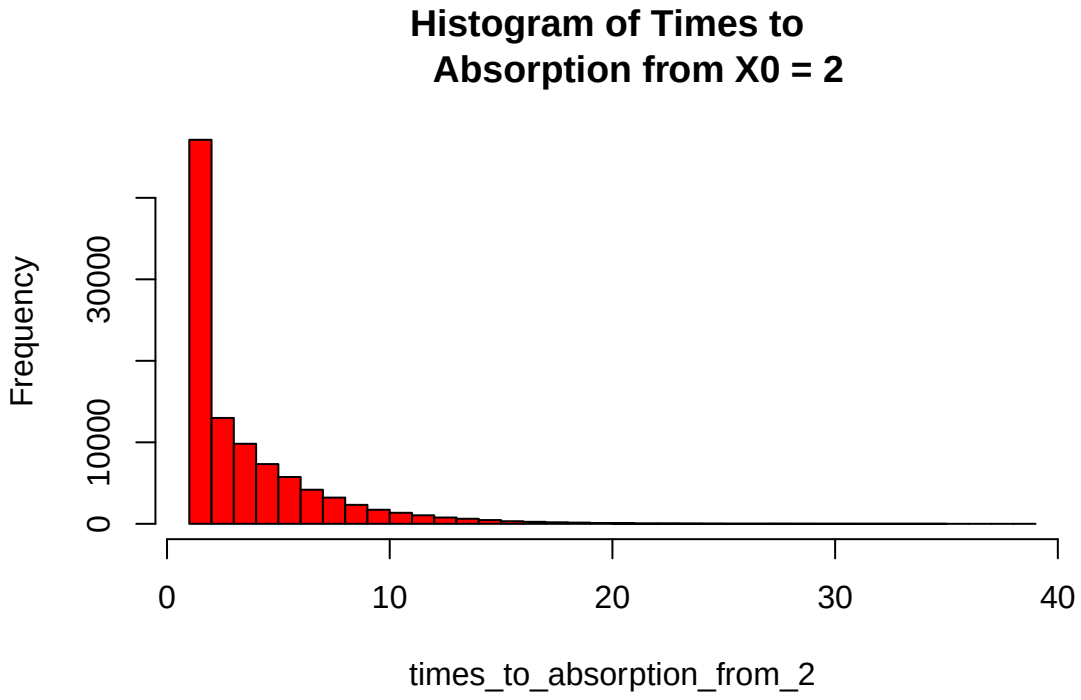
3.Absorbing State

a) Starting from each one of $X_0 = 1$ and $X_0 = 2$, run multiple simulation of the Markov chain (Problem 1, part b) until $X_i = 3$ and store the arrival time until this happens. Make a histogram of the arrival time for both $X_0 = 1$ and $X_0 = 2$ and compute the mean.

```
simulate_to_absorption <- function(P, start_state) {
  state <- start_state
  steps <- 0
  while(state != 3) {
    state <- sample(1:3, size = 1, prob = P[state, ])
    steps <- steps + 1
  }
  return(steps)
}

# Number of simulations
num_simulations <- 100000
times_to_absorption_from_1 <- replicate(num_simulations, simulate_to_absorption(P, 1))
times_to_absorption_from_2 <- replicate(num_simulations, simulate_to_absorption(P, 2))

# Plot histograms
hist(times_to_absorption_from_1, breaks = 50, col = 'blue', main = 'Histogram of Times to
     Absorption from X0 = 1')
```




```
hist(times_to_absorption_from_2, breaks = 50, col = 'red', main = 'Histogram of Times to
     Absorption from X0 = 2')
```




```
# Compute means
mean_time_from_1 <- mean(times_to_absorption_from_1)
mean_time_from_2 <- mean(times_to_absorption_from_2)

print(mean_time_from_1)
```

```
## [1] 4.6436
```

```
print(mean_time_from_2)
```

```
## [1] 3.84365
```

b) Compute theoretically the mean arrival time to the absorbing state and compare it with part a. To do so, notice that if T_i denotes the random variable associated to the arrival time starting from $X_0 = i$, then

$$\mu_i = 1 + \sum_{j=1}^{3} p_{ij}\mu_j;$$

with $\mu_i = \mathbb{E}[T_i]$. This is a linear system of equations that you can solve. Notice $T_3 = 0$.

```r
# Define the transition matrix P, modified to have 0s for transitions out of the absorbing state
P <- matrix(c(0.2, 0.7, 0.1,    # State 1 transitions
              0.2, 0.5, 0.3,    # State 2 transitions
              0,   0,   1),     # State 3 transitions (absorbing state)
            byrow = TRUE, nrow = 3)

# Define the system of equations based on the provided formula
# Since T3 = 0, we only need to calculate T1 and T2

# Coefficients matrix for T1 and T2
A <- matrix(c(1 - P[1, 1], -P[1, 2],
              -P[2, 1], 1 - P[2, 2]), ncol = 2)

b <- c(1, 1)

# Solve for T1 and T2
mean_times <- solve(t(A), b)

# Add T3 which is 0
mean_times <- c(mean_times, 0)
mean_times
```

```
## [1] 4.615385 3.846154 0.000000
```

Calculate by hand

$$\begin{aligned}
\mu_1 &= 1 + p_{11}\mu_1 + p_{12}\mu_2 + p_{13}\mu_3, \\
\mu_2 &= 1 + p_{21}\mu_1 + p_{22}\mu_2 + p_{23}\mu_3, \\
\mu_3 &= 0, \text{ since state 3 is absorbing.}
\end{aligned}$$

$$\begin{aligned}
0.8\mu_1 - 0.7\mu_2 &= 1, \\
-0.2\mu_1 + 0.5\mu_2 &= 1.
\end{aligned}$$

$$\mu = \begin{pmatrix} \mu_1 = 4.615 \\ \mu_2 = 3.846 \\ \mu_3 = 0 \end{pmatrix}$$