HW 7 M362M, Fall 2024

Wei Xuan Lee (wl22963)

2024-11-06

Problem 1

P_1 : There is only 1 communication class, because

$$1 \to 2 \to 3 \to 2 \to 1.$$

If $X_n = i$, then we must have $X_{n-1} \neq i$. So

$$p_{ii}^{(n)} = \left(1 - p_{ii}^{(n-1)}\right) \times \frac{1}{2}.$$

Since we know $p_{ii}^{(0)} = 1$, by expanding $p_{ii}^{(m)}, 0 \leq m < n$ successively and summing up the geometric sequence with ratio $-1/2$, we have

$$p_{ii}^{(n)} = \frac{1}{3}\left[1 - \left(\frac{-1}{2}\right)^{n-1}\right].$$

Note that $p_{ii}^{(n)}$ does not converge to 0 as $n \to \infty$, therefore

$$\sum_{n=1}^{\infty} p_{ii}^{(n)} = +\infty.$$

Since i was arbitrary, by the recurrent criterion, all states are recurrent. There are no absorbing states.

```r
# State space
S = c(1, 2, 3)

# transition matrix
P = matrix(c(0,   1/2,   1/2,
             1/2, 0,   1/2,
             1/2,   1/2, 0),
           byrow=TRUE, ncol=3)

T = 100 # number of time periods
nsim = 1000 # number of simulations

# simulate the next position of the chain
draw_next = function(s) {
  i = match(s, S) # the row number of the state s
  sample(S, prob = P[i, ], size = 1)
}
```

```r
# simulate a single trajectory of length T
# from the initial state
single_trajectory = function(initial_state) {
  path = numeric(T)
  last = initial_state
  for (n in 1:T) {
    path[n] = draw_next(last)
    last = path[n]
  }
  return(path)
}

# simulate the entire chain
simulate_chain = function(initial_state) {
  data.frame(X0 = initial_state,
             t(replicate(
               nsim, single_trajectory(initial_state)
             )))
}

# Long Term Behavior DOES NOT DEPEND on initial state

df_1 = simulate_chain(1)
c(mean(df_1$X100 == 1), mean(df_1$X100 == 2), mean(df_1$X100 == 3))
## [1] 0.347 0.340 0.313
df_2 = simulate_chain(2)
c(mean(df_2$X100 == 1), mean(df_2$X100 == 2), mean(df_2$X100 == 3))
## [1] 0.334 0.342 0.324
df_3 = simulate_chain(3)
c(mean(df_3$X100 == 1), mean(df_3$X100 == 2), mean(df_3$X100 == 3))
## [1] 0.335 0.323 0.342
```

P_2 : There is only 1 communication class because

$$1 \to 4 \to 3 \to 2 \to 4 \to 3 \to 1.$$

Recall that if a Markov chain has finite state space then it must have at least one recurrent state. Also, states in the same communication classes are either all recurrent or all transient. Since we only have 1 communication class, if there exists some state that is transient, then all states are transient, a contradiction. We conclude that all states are recurrent. There are no absorbing states.

```r
# State space
S = c(1, 2, 3, 4)

# transition matrix
P = matrix(c(0, 0, 0, 1,
             0, 0, 0, 1,
             1/2, 1/2, 0, 0,
             0, 0, 1, 0),
           byrow=TRUE, ncol=4)

T = 100 # number of time periods
nsim = 1000 # number of simulations
```

```r
# simulate the next position of the chain
draw_next = function(s) {
  i = match(s, S) # the row number of the state s
  sample(S, prob = P[i, ], size = 1)
}

# simulate a single trajectory of length T
# from the initial state
single_trajectory = function(initial_state) {
  path = numeric(T)
  last = initial_state
  for (n in 1:T) {
    path[n] = draw_next(last)
    last = path[n]
  }
  return(path)
}

# simulate the entire chain
simulate_chain = function(initial_state) {
  data.frame(X0 = initial_state,
             t(replicate(
               nsim, single_trajectory(initial_state)
             )))
}

# Long Term Behavior DEPENDS on initial state

df_1 = simulate_chain(1)
c(mean(df_1$X100 == 1), mean(df_1$X100 == 2), mean(df_1$X100 == 3), mean(df_1$X100 == 4))
## [1] 0 0 0 1
df_2 = simulate_chain(2)
c(mean(df_2$X100 == 1), mean(df_2$X100 == 2), mean(df_2$X100 == 3), mean(df_2$X100 == 4))
## [1] 0 0 0 1
df_3 = simulate_chain(3)
c(mean(df_3$X100 == 1), mean(df_3$X100 == 2), mean(df_3$X100 == 3), mean(df_3$X100 == 4))
## [1] 0.507 0.493 0.000 0.000
df_4 = simulate_chain(4)
c(mean(df_4$X100 == 1), mean(df_4$X100 == 2), mean(df_4$X100 == 3), mean(df_4$X100 == 4))
## [1] 0 0 1 0
```

P_3 : we have 3 communication classes.

The first is $\{2\}$ because for all $i \neq 2$

$$p_{i2} = 0.$$

And 2 is a transient state because it does not satisfy the recurrent criterion:

$$\sum_{n=1}^{\infty} p_{22}^{(n)} = \sum_{n=1}^{\infty} \left(\frac{1}{2}\right)^n = 1 < +\infty.$$

The second class is $\{1, 3\}$ because

$$1 \to 3 \to 1.$$

They are both transient states because we can look at the states $1, 3$ only and they form a Markov chain. With the same reasoning as P_2, they have to be both transient.

The third class is $\{4, 5\}$ because

$$4 \to 5 \to 4.$$

And they are both transient, for the exact same reason as above.

```r
# State space
S = c(1, 2, 3, 4, 5)

# transition matrix
P = matrix(c(1/2, 0, 1/2, 0, 0,
             1/4, 1/2, 1/4, 0, 0,
             1/2, 0, 1/2, 0, 0,
             0, 0, 0, 1/2, 1/2,
             0, 0, 0, 1/2, 1/2),
           byrow=TRUE, ncol=5)

T = 100 # number of time periods
nsim = 1000 # number of simulations

# simulate the next position of the chain
draw_next = function(s) {
  i = match(s, S) # the row number of the state s
  sample(S, prob = P[i, ], size = 1)
}

# simulate a single trajectory of length T
# from the initial state
single_trajectory = function(initial_state) {
  path = numeric(T)
  last = initial_state
  for (n in 1:T) {
    path[n] = draw_next(last)
    last = path[n]
  }
  return(path)
}

# simulate the entire chain
simulate_chain = function(initial_state) {
  data.frame(X0 = initial_state,
             t(replicate(
               nsim, single_trajectory(initial_state)
             )))
}

# Long Term Behavior DEPENDS on initial state

df_1 = simulate_chain(1)
c(mean(df_1$X100 == 1), mean(df_1$X100 == 2),
mean(df_1$X100 == 3), mean(df_1$X100 == 4), mean(df_1$X100 == 5))
## [1] 0.493 0.000 0.507 0.000 0.000
df_2 = simulate_chain(2)
```

```
c(mean(df_2$X100 == 1), mean(df_2$X100 == 2),
mean(df_2$X100 == 3), mean(df_2$X100 == 4), mean(df_2$X100 == 5))
## [1] 0.516 0.000 0.484 0.000 0.000
df_3 = simulate_chain(3)
c(mean(df_3$X100 == 1), mean(df_3$X100 == 2),
mean(df_3$X100 == 3), mean(df_3$X100 == 4), mean(df_3$X100 == 5))
## [1] 0.492 0.000 0.508 0.000 0.000
df_4 = simulate_chain(4)
c(mean(df_4$X100 == 1), mean(df_4$X100 == 2),
mean(df_4$X100 == 3), mean(df_4$X100 == 4), mean(df_4$X100 == 5))
## [1] 0.000 0.000 0.000 0.511 0.489
df_5 = simulate_chain(5)
c(mean(df_5$X100 == 1), mean(df_5$X100 == 2),
mean(df_5$X100 == 3), mean(df_5$X100 == 4), mean(df_5$X100 == 5))
## [1] 0.000 0.000 0.000 0.524 0.476
```

Problem 2

(1)

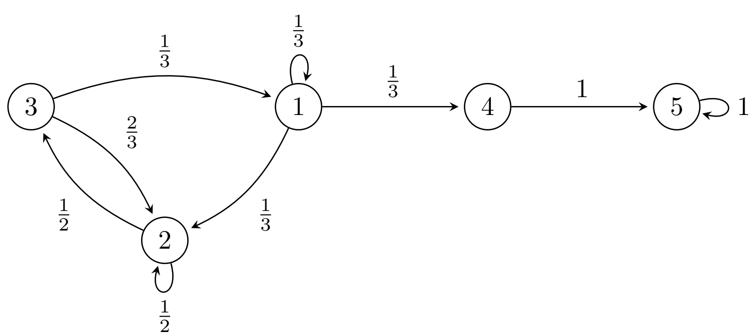

Figure 1: Transition Graph for Markov Chain

(2)

Communication Classes: $\{1, 2, 3\}, \{4\}, \{5\}$.

(3)

Closed sets: $\{5\}, \{4, 5\}, \{1, 2, 3, 4, 5\}$.

(4)

Return sets:

1. $R(1) = \mathbb{N}$ because $p_{11}^{(n)} \geq (p_{11})^n > 0$ for all $n \in \mathbb{N}$. Similarly, $R(2), R(5) = \mathbb{N}$.
2. $R(3) = \mathbb{N} \setminus \{1\}$ because
   - For even n, $p_{33}^{(n)} \geq (p_{32} \cdot p_{23})^{n/2} > 0$

- For odd $n \geq 3$, $p_{33}^{(n)} \geq (p_{32} \cdot p_{22} \cdot p_{23}) \cdot (p_{32} \cdot p_{23})^{(n-3)/2} > 0$

3. $R(4) = \emptyset$ because $p_{45}^{(n)} = 1$ for all $n \in \mathbb{N}$, thus $p_{44}^{(n)} = 0$.

(5)

Paths from 2 to 5 in

- 1 step, 2 steps, 3 steps: $\emptyset$

  Thus $p_{25}^{(n)} = 0, n = 1, 2, 3$
- 4 steps:
  $$2 \to 3 \to 1 \to 4 \to 5$$
  Thus $p_{25}^{(4)} = \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times 1 = \frac{1}{18}$
- 5 steps:
  $$\begin{aligned} &2 \to 2 \to 3 \to 1 \to 4 \to 5 \\ &2 \to 3 \to 1 \to 1 \to 4 \to 5 \\ &2 \to 3 \to 1 \to 4 \to 5 \to 5 \end{aligned}$$

  Thus
  $$\begin{aligned} p_{25}^{(5)} = &\frac{1}{2} \times \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times 1 \\ &+ \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \times 1 \\ &+ \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times 1 \times 1 = \frac{11}{108} \end{aligned}$$

```r
# State space
S = c(1, 2, 3, 4, 5)

# transition matrix
P = matrix(c(1/3, 1/3, 0, 1/3, 0,
             0, 1/2, 1/2, 0, 0,
             1/3, 2/3, 0, 0, 0,
             0, 0, 0, 0, 1,
             0, 0, 0, 0, 1),
           byrow=TRUE, ncol=5)

T = 5 # number of time periods
nsim = 10000 # number of simulations

# simulate the next position of the chain
draw_next = function(s) {
  i = match(s, S) # the row number of the state s
  sample(S, prob = P[i, ], size = 1)
}

# simulate a single trajectory of length T
# from the initial state
```

```r
single_trajectory = function(initial_state) {
  path = numeric(T)
  last = initial_state
  for (n in 1:T) {
    path[n] = draw_next(last)
    last = path[n]
  }
  return(path)
}

# simulate the entire chain
simulate_chain = function(initial_state) {
  data.frame(X0 = initial_state,
             t(replicate(
               nsim, single_trajectory(initial_state)
             )))
}

# Theoretical Probability for 2->5 in 1, 2, 3, 4, 5 step(s)
c(0, 0, 0, 1/18, 11/108)
## [1] 0.00000000 0.00000000 0.00000000 0.05555556 0.10185185
df = simulate_chain(2)

# Simulated Probability for 2->5 in 1, 2, 3, 4, 5 step(s)
c(mean(df$X1==5), mean(df$X2==5), mean(df$X3==5), mean(df$X4==5), mean(df$X5==5))
## [1] 0.0000 0.0000 0.0000 0.0539 0.0997
# Theoretical and Simulated Probabilities agree.
```

End of Homework