Modul Praktikum: Limiting Probabilities dan Time-Reversibility

Tim Pengampu Mata Kuliah Pemodelan Stokastik

Pertemuan 9

Contents

BAB 1. Pendahuluan

1.1 Latar Belakang

Dalam analisis proses stokastik kita sering tertarik bukan hanya pada perilaku sementara suatu sistem, tetapi juga pada perilaku jangka panjangnya. **Limiting probabilities** (atau steady-state distributions) memberikan probabilitas bahwa sistem berada pada suatu state tertentu bila waktu berjalan menuju tak hingga. Konsep ini penting dalam sistem antrian, jaringan komputer, ekonomi, dan epidemiologi.

Time-reversibility (reversibilitas waktu) adalah sifat khusus dari beberapa proses Markov di mana proses yang diamati saat waktu berjalan mundur memiliki sifat probabilistik yang sama dengan proses saat waktu berjalan maju. Sifat ini memudahkan perhitungan dan pembuktian untuk berbagai sistem antrian.

1.2 Tujuan Pembelajaran

Setelah menyelesaikan modul praktikum ini, mahasiswa diharapkan mampu:

- Menjelaskan definisi limiting probabilities.
- Memverifikasi keberadaan steady-state untuk proses kelahiran-kematian.
- Memahami definisi dan implikasi time-reversibility.
- Mengimplementasikan simulasi rantai Markov dan antrian M/M/1 di R.
- Membandingkan hasil simulasi numerik dengan solusi teoritis.

BAB 2. Limiting Probabilities

2.1 Definisi

Untuk rantai Markov kontinu, limiting probability
pi_j didefinisikan sebagai

$$\pi_j = \lim_{n\to\infty} P(X_n = j)$$

jika limit tersebut ada dan tidak bergantung pada kondisi awal.

Interpretasi dari *limiting probabilities* adalah sebagai berikut: Sebagai ilustrasi, jika $\pi_0 = 0.8$, ini berarti dalam jangka panjang, sistem akan menghabiskan 80% dari waktunya dalam keadaan 0, terlepas dari keadaan awal sistem tersebut. Ini menunjukkan stabilitas dan distribusi waktu yang dapat diprediksi dari suatu proses stokastik.

Untuk menghitung limiting probabilities, kita menyelesaikan sistem persamaan yang diturunkan dari prinsip keseimbangan: aliran masuk sama dengan aliran keluar.

$$\sum_{k\neq j} q_{kj}\pi_k = v_j\pi_j$$

untuk setiap state j, dan juga syarat normalisasi

$$\sum_j \pi_j = 1.$$

Dengan menyelesaikan SPL tersebut kita akan mendapatkan limiting probabilities dari sistem tersebut untuk seluruh state.

Contohnya, pada sistem antrean M/M/1 untuk keadaan n (n pelanggan dalam sistem), persamaan keseimbangan diberikan oleh:

- Keadaan $n = 0$: $\mu\pi_1 = \lambda\pi_0$
- Keadaan $n \geq 1$: $\lambda\pi_{n-1} + \mu\pi_{n+1} = (\lambda + \mu)\pi_n$

Dengan menyelesaikan SPL di atas yang bersifat rekursif, kita peroleh rumus limiting probabilities untuk sistem M/M/1 sebagai berikut:

$$\pi_n = (1-\rho)\rho^n, \quad n = 0, 1, 2, \ldots.$$

2.2 Contoh Perhitungan

Sistem dengan 2 state: Tinjau mesin dengan laju kegagalan $\alpha = 0.1$/jam dan laju perbaikan $\beta = 0.4$/jam. Berapa proporsi waktu mesin tersebut bekerja dalam jangka panjang?

Solusi: Dari soal kita ketahui $q_{01}=0.1, v_0=0.1 $ dan $q_{10} = 0.4, v_1 = 0.4$. Kita peroleh SPL

$$q_{10}\pi_1 = q_0\pi_0 q_{01}\pi_0 = q_1\pi_1\pi_0 + \pi_1 = 1$$

atau

$$0.4\pi_1 = 0.1\pi_0\pi_0 + \pi_1 = 1.$$

Sehingga $\pi_0 = 4/5$ dan $\pi_1 = 1/5$. Kita dapatkan kesimpulan bahwa mesin tersebut bekerja 80% dari toal waktu dan rusak 20% dari waktu dalam jangka panjang.

2.3 Simulasi Limiting Probabilities

Pada bagian ini kita akan melakukan simulasi dari proses M/M/1 untuk jangka waktu yang panjang. Selanjutnya kita akan melihat bagaimana peluang waktu yang dihabiskan oleh proses ini pada suatu state dan akan kita bandingkan dengan nilai teoretik dari limiting probabilities proses M/M/1.

Kode R dibawah ini digunakan untuk melakukan simulasi proses M/M/1.

```r
library(tidyverse)
library(gridExtra)
library(ggplot2)
set.seed(2025)

simulate_mm1 <- function(lambda, mu, T = 10){
  t <- 0    # time variable
  n <- 0    # jumlah pelanggan
  # records is a dataframe containing when customer arrive or depart
  records <- tibble(time = 0, n = n, event = "start")
  next_arrival <- t + rexp(1, rate = lambda)  # inter-arrival time is exponential
  next_departure <- Inf
  while(t < T){
    if(next_arrival <= next_departure){
      t <- next_arrival
      n <- n + 1
      records <- add_row(records, time = t, n = n, event = "arrival")
      next_arrival <- t + rexp(1, rate = lambda)
      if(n == 1) next_departure <- t + rexp(1, rate = mu)
    } else {
      t <- next_departure
      n <- max(0, n - 1)
      records <- add_row(records, time = t, n = n, event = "departure")
      if(n > 0) next_departure <- t + rexp(1, rate = mu) else next_departure <- Inf
    }
    if(t > T) break
  }
  # ensure final time T recorded for plotting
  if(t < T) records <- add_row(records, time = T, n = records$n[nrow(records)],
                               event = "end")
  records
```

```
}

# contoh simulasi
# parameters
lambda <- 10
mu <- 15
T <- 1000      # long run time for steady state approximation
rho <- lambda/mu

# simulation
sim <- simulate_mm1(lambda, mu, T)
head(sim)
```

```
## # A tibble: 6 x 3
##      time     n event
##     <dbl> <dbl> <chr>
## 1 0           0 start
## 2 0.0465      1 arrival
## 3 0.0839      0 departure
## 4 0.150       1 arrival
## 5 0.151       2 arrival
## 6 0.156       1 departure
```

Kode dibawah ini digunakan untuk menghitung proporsi dari waktu yang dihabiskan di setiap state.

```
# compute proportion of time in each state
sim <- sim %>%
  mutate(dt = lead(time, default = T) - time) %>%
  group_by(n) %>%
  summarise(time_in_state = sum(dt)) %>%
  mutate(sim_prob = time_in_state / sum(time_in_state))

# theoretical probabilities
max_state <- max(sim$n)
theory <- tibble(
  n = 0:max_state,
  theoretical = (1 - rho) * rho^(0:max_state)
)

compare <- left_join(sim, theory, by = "n")
head(compare)
```

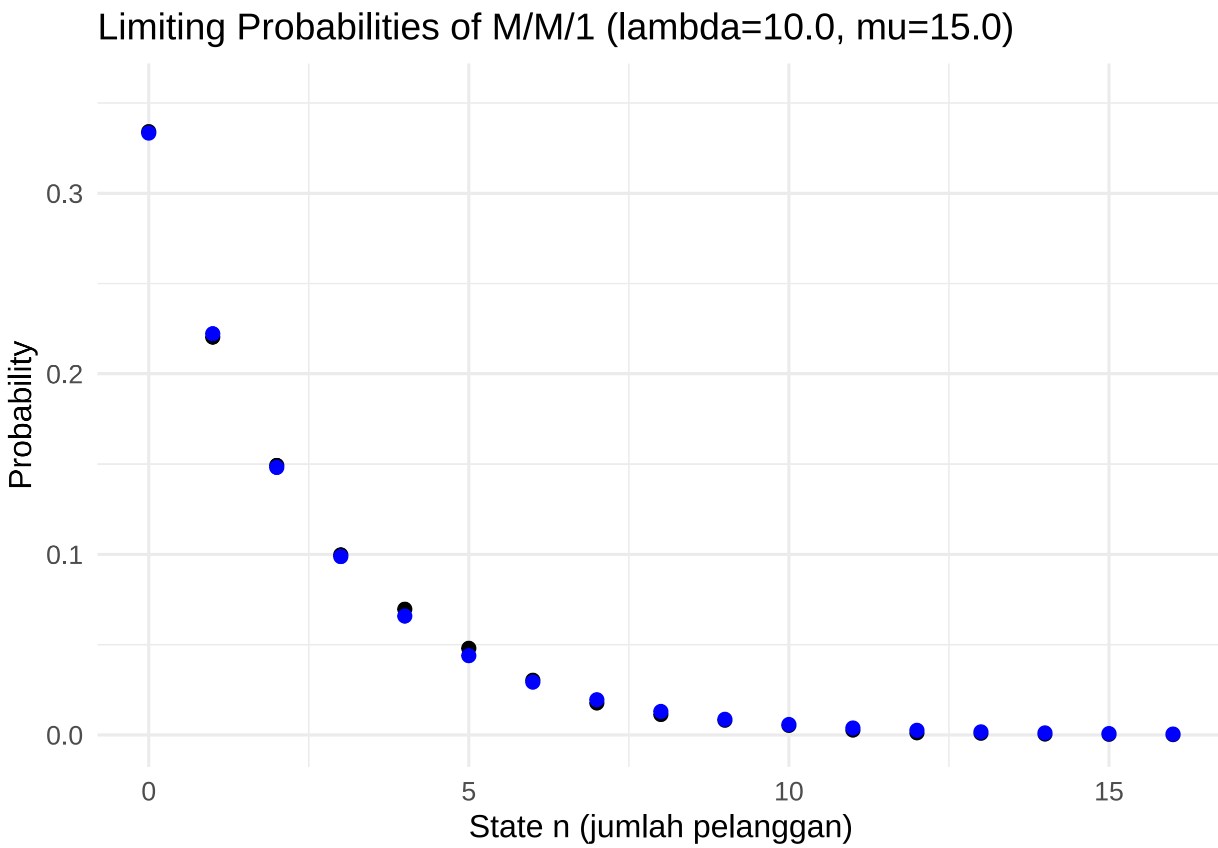
```
## # A tibble: 6 x 4
##         n time_in_state sim_prob theoretical
##     <dbl>         <dbl>    <dbl>       <dbl>
```

```
## 1     0          334.    0.334        0.333
## 2     1          220.    0.220        0.222
## 3     2          149.    0.149        0.148
## 4     3           99.8   0.0998       0.0988
## 5     4           69.7   0.0697       0.0658
## 6     5           48.0   0.0480       0.0439
```

Table di atas adalah tabel yang menyatakan berapa proporsi waktu yang dihabiskan proses pada suatu state berdasarkan simulasi dan juga berdasarkan perhitungan teoritis. Dapat kita lihat bahwa hasil simulasi sangat mendekati nilai berdasarkan teori.

Gambar di bawah ini adalah perbandingan dari nilai-nilai tersebut, dimana titik biru menyatakan nilai teoritik dan titik hitam menyatakan nilai hasil simulasi. Perhatikan bahwa titik-titik tersebut cenderung saling tindih dan nilainya sangat mendekati satu-sama-lain.

```r
# plot comparison
ggplot(compare, aes(x = n)) +
  geom_point(aes(y = sim_prob), size = 2) +
  geom_point(aes(y = theoretical), size=2, color="blue") +
  labs(title = sprintf("Limiting Probabilities of M/M/1 (lambda=%.1f, mu=%.1f)",
                       lambda, mu),
       x = "State n (jumlah pelanggan)",
       y = "Probability") +
  theme_minimal(base_size = 12) +
  ylim(0, max(compare$sim_prob, compare$theoretical) + 0.02)
```


Limiting Probabilities of M/M/1 (lambda=10.0, mu=15.0)
Probability
0.3
0.2
0.1
0.0
0
5
10
15
State n (jumlah pelanggan)

BAB 3. Time-Reversibility

3.1 Definisi

Rantai Markov dengan distribusi stasioner π dikatakan time-reversible jika memenuhi *detailed balance equation*:

$$\pi_i q_{ij} = \pi_j q_{ji} \quad \text{untuk semua } i, j.$$

dimana q_{ij} merupakan laju transisi dari state i ke state j. Persamaan ini menjamin bahwa aliran probabilitas dari i ke j sama dengan aliran dari j ke i. Jika proses bersifta time-reversible, maka melihat proses dalam arah waktu terbalik tidak mengubah sifat statistik transisi ketika distribusi keadaan mengikuti limiting probabilities π.

Reversibilitas waktu memberikan keuntungan signifikan saat menganalisis Rantai Markov Waktu-Kontinu (CTMC), terutama dengan menyederhanakan perhitungan distribusi stasioner.

- Detailed Balance Equations melibatkan penyelesaian persamaan

  $$\pi_i q_{ij} = \pi_j q_{ji}$$

  untuk semua keadaan i, j. Setiap persamaan ini pada hanya melibatkan dua probabilitas stasioner π_i dan π_j.
- Sedangkan Global Balance Equations membutuhkan penyelesaian SPL

  $$v_j \pi_j = \sum_{i \neq j} \pi_i q_{ij}$$

  dimana setiap persamaan melibatkan π_j dan π_i untuk setiap i dimana terdapat laju tak-nol dari state i ke state j. Ini bisa jadi intensif secara komputasi dan kompleks, terutama untuk ruang keadaan yang besar.

3.2 Contoh Perhitungan

Contoh: Misalkan sebuah CTMC memiliki 3 state $\{1, 2, 3\}$ dan laju transisi sebagai berikut:

- $q_{12} = 2$, $q_{13} = 4$
- $q_{21} = 1$, $q_{23} = 2$
- $q_{31} = 2$, $q_{32} = 2$ Tentukan probabilitas stasioner (π) dengan menggunakan Global Balance Equations. Apakah CTMC ini time-reversible?

Solusi: Kita peroleh laju keluar dari masing-masing state sebagai berikut

- $v_1 = q_{12} + q_{13} = 2 + 4 = 6$
- $v_2 = q_{21} + q_{23} = 1 + 2 = 3$
- $v_3 = q_{31} + q_{32} = 2 + 2 = 4$

Sehingga kita peroleh Global Balance Equations

- State 1: $6\pi_1 = \pi_2 + 2\pi_3$
- State 2: $3\pi_2 = 2\pi_1 + 2\pi_3$
- State 3: $4\pi_3 = 4\pi_1 + 2\pi_2$
- Normalisasi: $\pi_1 + \pi_2 + \pi_3 = 1$

Dengan menyelesaikan sistem persamaan di atas, kita mendapatkan:

$$\pi_1 = \frac{1}{5}, \quad \pi_2 = \frac{2}{5}, \quad \pi_3 = \frac{2}{5}$$

Cek Detailed Balance Equations:

- $\pi_1 q_{12} = (\frac{1}{5})(2) = \frac{2}{5}$ dan $\pi_2 q_{21} = (\frac{2}{5})(1) = \frac{2}{5}$
- $\pi_1 q_{13} = (\frac{1}{5})(4) = \frac{4}{5}$ dan $\pi_3 q_{31} = (\frac{2}{5})(2) = \frac{4}{5}$
- $\pi_2 q_{23} = (\frac{2}{5})(2) = \frac{4}{5}$ dan $\pi_3 q_{32} = (\frac{2}{5})(2) = \frac{4}{5}$

Karena semua persamaan keseimbangan detail terpenuhi, CTMC ini time-reversible.

3.3 Simulasi Proses Time-Reversible

Pada bagian ini kita akan melakukan simulasi dari proses M/M/1 dan membalikan proses tersebut terhadap waktu dan membandingkan proses saat waktu berjalan maju dan berjalan mundur. Kita gunakan fungsi *simulate_mm1()* yang sudah kita buat sebelumnya dengan terlebih dahulu membuang 100 satuan waktu pertama karena kita asumsi pada 100 satuan waktu pertama, proses belum stabil.

```r
lambda <- 4; mu <- 5
sim <- simulate_mm1(lambda, mu, T = 500)

# Remove initial transient
burnin <- 100
sim_ss <- sim %>% filter(time > burnin)

# Reverse time process
reverse_sim <- sim_ss %>%
  mutate(time = max(time) - time) %>%
  arrange(time) %>%
  mutate(event = ifelse(event=="arrival","departure",
                 ifelse(event=="departure","arrival","start")))

# Estimate empirical transition rates
forward_rates <- sim_ss %>%
  mutate(next_n = lead(n)) %>%
  filter(!is.na(next_n) & n != next_n) %>%
  count(change = next_n - n) %>%
  mutate(rate_est = n / max(sim_ss$time))

reverse_rates <- reverse_sim %>%
```

```r
  mutate(next_n = lead(n)) %>%
  filter(!is.na(next_n) & n != next_n) %>%
  count(change = next_n - n) %>%
  mutate(rate_est = n / max(reverse_sim$time))

forward_rates
```

```
## # A tibble: 2 x 3
##   change     n rate_est
##    <dbl> <int>    <dbl>
## 1     -1  1592     3.18
## 2      1  1600     3.20
```

```r
reverse_rates
```

```
## # A tibble: 2 x 3
##   change     n rate_est
##    <dbl> <int>    <dbl>
## 1     -1  1600     4.00
## 2      1  1592     3.98
```

Tabel di atas menyatakan laju kedatangan pelanggan dan juga laju pelayanan. Dapat kita lihat bahwa kedua laju tersebut saling bertukar saat kita ubah proses dari waktu maju menjadi waktu mundur.

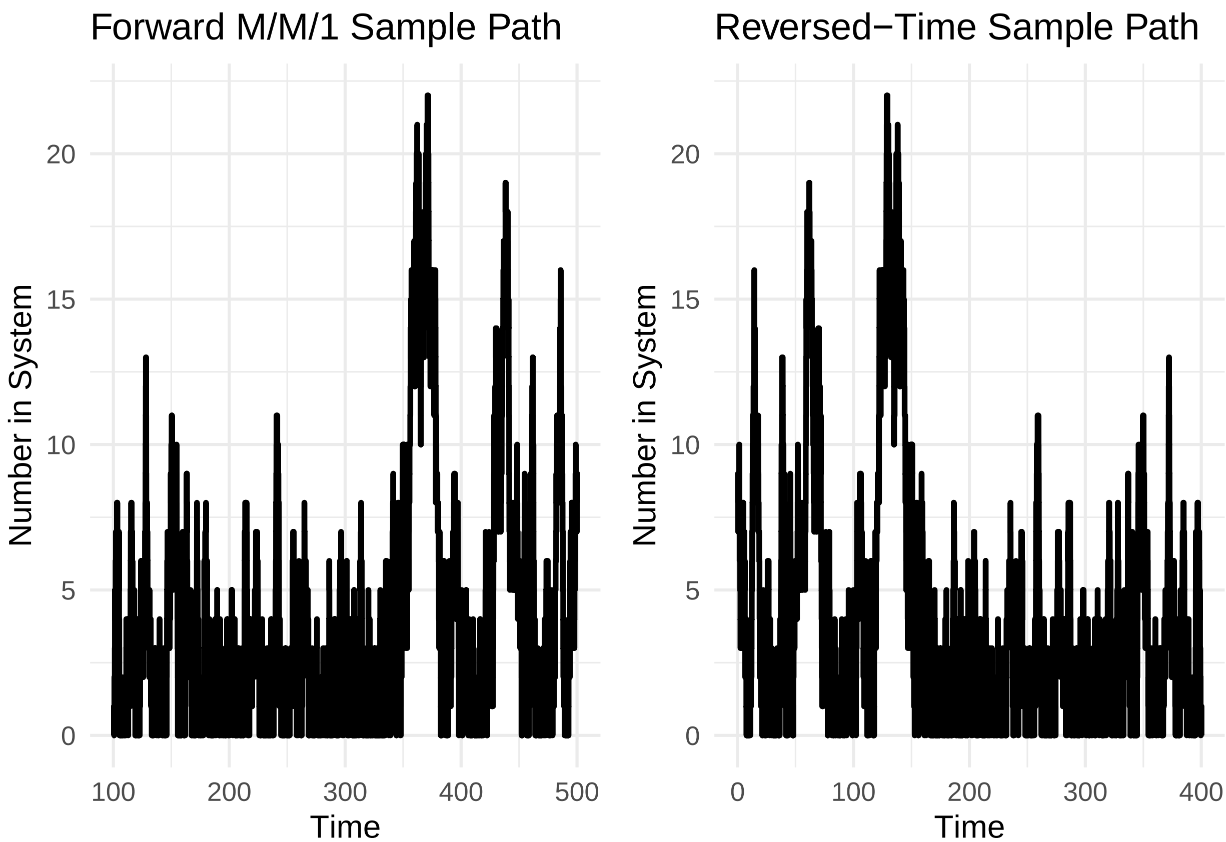
Berikut ini adalah perbandingan plot antara proses pada waktu maju dan waktu mundur.

```r
library(tidyverse)
library(gridExtra)

# Forward process step plot
p_forward <- ggplot(sim_ss, aes(x = time, y = n)) +
  geom_step(linewidth = 1) +
  labs(title = "Forward M/M/1 Sample Path",
       x = "Time", y = "Number in System") +
  theme_minimal(base_size = 12)

# Reverse process step plot
p_reverse <- ggplot(reverse_sim, aes(x = time, y = n)) +
  geom_step(linewidth = 1) +
  labs(title = "Reversed-Time Sample Path",
       x = "Time", y = "Number in System") +
  theme_minimal(base_size = 12)

# Combine plots
grid.arrange(p_forward, p_reverse, ncol = 2)
```


Forward M/M/1 Sample Path
Number in System
Time
0
5
10
15
20
100
200
300
400
500
Reversed−Time Sample Path
Number in System
Time
0
5
10
15
20
0
100
200
300
400

BAB 4. Soal Latihan

Berikut soal praktikum beserta petunjuk singkat implementasi di R.

Soal 1

Simulasikan M/M/1 $\lambda = 8, \mu = 12$ selama T=1000 satuan waktu.

- Buat tabel yang membandingkan limiting probabilities hasil simulasi dan berdasarkan teori.
- Berapa proporsi bahwa dalam sistem tidak ada pelanggan untuk jangka waktu panjang?
- Berapa peluang bahwa dalam sistem terdapat kurang dari atau sama dengan 5 pelanggan?

Soal 2

Simulasikan M/M/1 $\lambda = 12, \mu = 15$ selama T=1000 satuan waktu.

- Buat tabel yang membandingkan limiting probabilities hasil simulasi dan berdasarkan teori.
- Hitung Total Variation Distance (TVD) dari peluang limit yang diperoleh berdasarkan simulasi dan teori.

$$\delta(P, Q) = \frac{1}{2} \sum_{x \in \Omega} |P(x) - Q(x)|$$

  dimana x di atas menyatakan state yang ada pada simulasi, P menyatakan peluang hasil simulasi dan Q menyatakan peluang teori.
- Buat CDF (Cumulative Distribution Function) dari peluang limit berdasarkan hasil simulasi.

Soal 3

Simulasikan M/M/1 $\lambda = 8, \mu = 12$ selama T=1000 satuan waktu.

- Berikan tabel yang yang berisi laju pada proses maju dan mundur (pada proses mundur, buang terlebih dahulu 100 satuan waktu pertama).
- Plot kedua proses tersebut.

Referensi

- Ross, S. (2014). *Introduction to Probability Models.*
- Norris, J. R. (1998). *Markov Chains.*
- Gross, D. dan Harris, C. M. (1998). *Fundamentals of Queueing Theory.*