Lund University
Faculty of Engineering (LTH)

FRTN 30

Network Dynamics

Markov chains and networks, Hand-In 2

Author:
Marcel Attar, 941127-2173

The report was handed in on: May 8, 2019

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Single-particle random walk

We were given the following transition rate matrix

$$\Lambda = \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{ccccc} o & a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 2/5 & 1/5 & 0 & 0 \\ 0 & 0 & 3/4 & 1/4 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/3 & 0 & 2/3 \\ 0 & 1/3 & 0 & 1/3 & 0 \end{pmatrix} \end{array}\begin{array}{c} \\ o \\ a \\ b \\ c \\ d \end{array} \tag{1}$$

with the corresponding graph

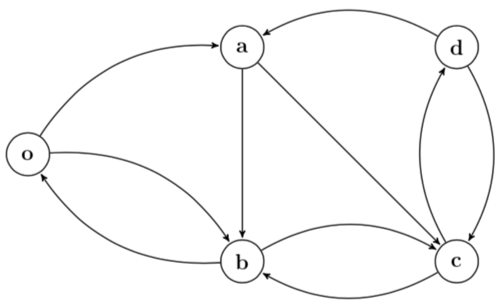

Figure 1: Closed network in which particles move according to the transition rate matrix 5

The time here is continuous and the time S a particle stays in a state i is a stochastic variable according to

$$\mathbb{P}(S \geq t) = e^{-rt}, \quad t \geq 0 \tag{2}$$

where $r = w_i = \sum_j \Lambda_{ij}$. We can reformulate the problem in terms of a transition probability matrix $\bar{\mathrm{P}}$, with elements

$$\bar{P}_{ij} = \frac{\Lambda_{ij}}{w_*}, \quad w_* = \max_i w_i \tag{3}$$

$$\bar{P}_{ij} = 1 - \sum_{j \neq i} \bar{P}_{ij} \tag{4}$$

When we calculate this we get

$$
\bar{P} = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}
\begin{array}{c}
\begin{array}{ccccc} o & a & b & c & d \end{array} \\
\begin{pmatrix}
2/5 & 2/5 & 1/5 & 0 & 0 \\
0 & 0 & 3/4 & 1/4 & 0 \\
1/2 & 0 & 0 & 1/2 & 0 \\
0 & 0 & 1/3 & 0 & 2/3 \\
0 & 1/3 & 0 & 1/3 & 1/3
\end{pmatrix}
\end{array}
\begin{array}{c} \\ o \\ a \\ b \\ c \\ d \end{array}
\tag{5}
$$

Since I now have the $\bar{P}$ matrix I can use a global poisson clock, as opposed to having one poisson clock for each node.

Questions:

- a) *What is, according to the simulations, the average time it takes a particle that starts in node a to leave the node and then return to it?*

See attached code for exact implementation. I used the $\bar{P}$ matrix and a global poisson clock and looked at how the particle moved in the system for each time step (and noted down how long that time step was). When the particle had returned "home" I noted the total time it took. The average return time was $\mathbf{6.7318}$ after one million iterations.

- b) *How does the result in a) compare to the theoretical return-time $\mathbb{E}_a[T_a^+]$? (Include a description of how this is computed.)*

By theorem 4.2 in the lecture notes, the expected return-times satisfy

$$
\mathbb{E}_i[T_i^+] = \frac{1}{w_i \bar{\pi}_i}, \quad i \in \mathcal{X} \tag{6}
$$

for $T_i^+ = inf\{t \geq 0 : \ X(t) = i \text{ and } X(s) \neq i \text{ for some } s \in (0,t)\}$, where $\bar{\pi}$ is the invariant probability distribution, given by

$$
\bar{\pi} = \bar{P}' \bar{\pi}
$$

We therefore need the invariant distribution to solve $\mathbb{E}_a[T_a^+]$, solving the eigenvalue problem in MatLab and calculating the out-degree vector w_i we get

$$
\mathbb{E}_i[T_i^+] = 6.75
$$

This is close to the empirical value that was calculated in the previous task.

- c) *What is, according to the simulations, the average time it takes to move from node o to node d?*

See attached code for exact implementation. Once again I used the $\bar{P}$ matrix and a global poisson clock and looked at how the particle moved through the system. The only difference was that I instead reset the time when the particle hit node d instead of node a. The average hitting time for node o to node d is $\mathbf{8.7637}$ after one million iterations.

- d) *How does the result in c) compare to the theoretical hitting-time $\mathbb{E}_o[T_d]$? (Describe also how this is computed.)*

The expected hitting time $\mathbb{E}_i(T_s)$ for node s from node i is given by

$$\mathbb{E}_i(T_s) = 0, \quad \text{if} \quad i = s \tag{7}$$

$$\mathbb{E}_i(T_s) = 1 + \sum_j P_{ij}\mathbb{E}_j(T_s), \quad \text{if} \quad i \neq s \tag{8}$$

This holds for discrete time, we can modify it to hold for continuous time if we instead use $\bar{P}$, we then get the expected number of time steps. To translate this into time units when then have to divide by the mean rate, which is $w^* = 1$. I solved the above equation, with four unknowns and four equations, with the MatLab function 'linsolve', yielding the result $\mathbb{E}_i(T_s) = 8.7857$. My result in c) reflects this fairly well.

Multi-particle random walk – a matter of perspective

a) Particle perspective:

- *If 100 particles all start in node a, what is the average time for a particle to return to node a?*

See attached code for implementation. Since all the particles move independently of each other, there is no difference in "releasing" the particles all at the same time or on at a time. I therefore could use my code from task 1a, but instead of running one million iterations I only ran 100 iterations. The average return time is **6.8552**.

- *How does this compare to the answer in Part 1, why?*

This is close to the same result in part 1, however, since we only run fewer iterations there is greater variance in this part.

b) Node perspective:

- *If 100 particles start in node o, and the system is simulated for 60 time units, what is the average number of particles in the different nodes at the end of the simulation?*

See attached code for implementation.

Node	# of particles
o	18.6
a	14.8
b	22.2
c	22.1
d	22.3

We know from theorem 4.2 in the lecture notes that $\lim_{t \to +\infty} \bar{\pi}(t) = \bar{\pi}$, and this $\bar{\pi}$ was used in part 1b to calculate the theoretical return-time.

Node	**Probability distribution**
o	0.1852
a	0.1481
b	0.2222
c	0.2222
d	0.2222

Multiplying these numbers with the amount of particles in the system (100) we get very similar results as the simulation. Therefore, the theory and empirical results match.

- *Illustrate this with a plot showing the number of particles in each node during the simulation time.*

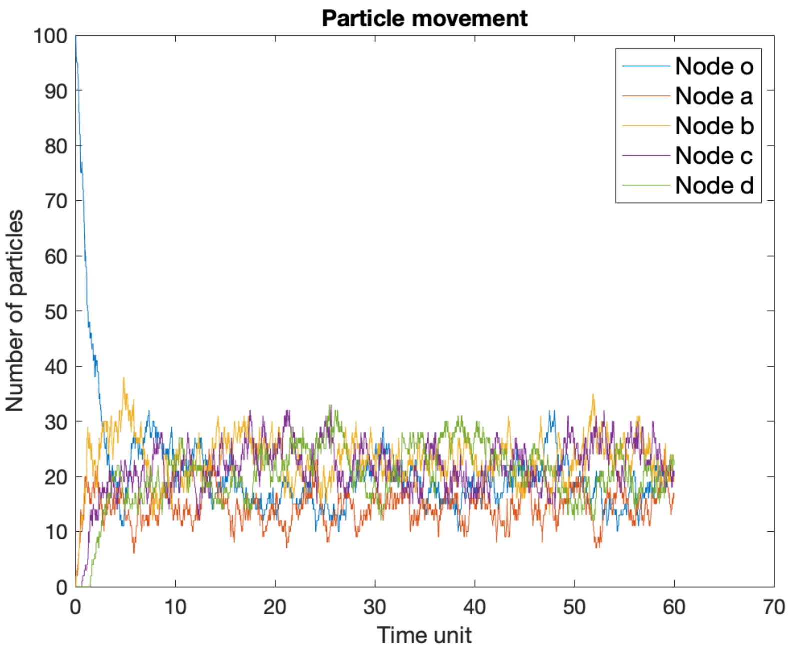

Figure 2: Plot of the particle movement through time. In the begining all particles starts in node o, eventually they spread out over all the nodes. The particles doesn't seem to converge in only 60 time units.