# POLYTECHNIC OF TURIN

**Master's Degree**
**in Data Science and Engineering**

Network Dynamics and Learning

## Homework 2


Riccardo Bosio 291299

Anno Accademico 2021-2022

# 1 Introduction

This document contains my solution to the Homework 2's exercises of the Network Dynamics and Learning course.

The work is organized as follows. The first section is a summary of the theory used. Then there is one section for each exercise.

At this link https://github.com/riccardobosio/NDL, it is possible to see the text of exercises and the code used.

I have discussed the solution with Beatrice Macchia and Edoardo Fantolino.

# 2 Theory used

Random walks are Markov chains whose state space is identified with the node set of a graph: the links of this graph in turn represent the possible state transitions.

A Poisson clock is characterized by the property that the time elapsed between any two of its consecutive ticks is an independent random variable with rate-r exponential distribution.

A discrete-time Markov chain $\mathcal{X}(t)$ with finite state space $\mathcal{X}$ can be associated to any pair of a stochastic matrix $P$, i.e., a nonnegative square matrix whose entries are labeled by the corresponding elements of $\mathcal{X}$ and whose rows all sum up to 1, and an initial probability distribution $\pi(0)$, i.e., a probability vector with entries corresponding to the elements of $\mathcal{X}$ such that

$$P(\mathcal{X}(0) = i) = \pi_i(0), i \in \mathcal{X}.$$

Given a finite state space $\mathcal{X}$ and a square nonnegative matrix $\Lambda = (\Lambda_{ij})_{i,j\in\mathcal{X}} \neq 0$ with entries indexed by elements of $\mathcal{X}$ and diagonal entries $\Lambda_{ij} = 0$, we define a continuous-time Markov chain $X(t)$ with transition rate matrix $\Lambda$ in the following way. Consider

$$w_i = \sum_{j\neq i} \Lambda_{ij}, i \in \mathcal{X}$$

$$w_* = max_{i\in\mathcal{X}} w_i$$

and a $rate - w_*$ Poisson clock $T_0 \leq T_1 \leq ....$ The associated Poisson process $N_t$ is

$$N_t = sup\{k \geq 0 : T_k \leq t\}, t \geq 0.$$

Consider also an independent discrete-time Markov chain $U(k)$, for $k = 0, 1, ...,$ with transition probabilities

$$\bar{P}_{ij} = \frac{\Lambda_{ij}}{w_*}, i \neq j, \bar{P}_{ii} = 1 - \sum_{j\neq i} \bar{P}_{ij}.$$

Then, we can define

$$X(t) = U(N_t), t \geq 0.$$

The discrete-time Markov chain $U(k)$ is called the jump chain associated to the continuous-time Markov chain $X(t)$. The hitting time on a given node $j \in \mathcal{X}$ is $T_j = inf\{t \geq 0 : X(t) = j\}$, while the return time is $T_j^+ = inf\{t \geq 1 : X(t) = j\}$.

Let $X(t)$ be a Markov chain with finite state space $\mathcal{X}$, transition probability matrix $P$, and initial state $i \in \mathcal{X}$. Let $S \subseteq \mathcal{X}$ be a subset of states that is reachable from every state $i \in \mathcal{X}$. The expected hitting times $\{\mathbf{E}_i[T_S]\}_{i \in \mathcal{X}}$ are the unique solution of the following linear system:

$$\mathbf{E}_i[T_S] = \begin{cases} 0, & \text{if } i \in S, \\ 1 + \sum_{j \in \mathcal{X}} P_{ij} \mathbf{E}_j[T_S], & \text{if } i \notin S. \end{cases}$$

Then for any state $i \in \mathcal{X}$, the expected return times satisfy

$$\mathbf{E}_i[T_i^+] = 1 + \sum_{j \in \mathcal{X}} P_{ij} \mathbf{E}_j[T_i].$$

Let $G = (V, E, W)$ be a weighted graph, and $x(t) \in R^V$ denote the state of the nodes of the graph.

The dynamics of $x(t)$ reads

$$x(t+1) = Px(t),$$

where P is the normalized adjacency matrix. In opinion dynamics, $x_i$ indicates the opinion of node i. This dynamics is known as French - De Groot and we assume by convention that the opinion of node $i$ is influenced by the opinion of node $j$ if $P_{ij} > 0$, i.e., the link $(i, j)$ is to be interpreted as $i$ watching $j$ and updating her opinion based on opinion of j.

Assuming that G has the condensation graph with one sink and the sink component of the graph is aperiodic, then

$$\lim_{t \to +\infty} x(t) = \alpha 1.$$

The consensus is a configuration of states all equal to each other.

If we now consider a graph and assume that the state of each node represents a noisy estimate of the real state $\mu$, that is

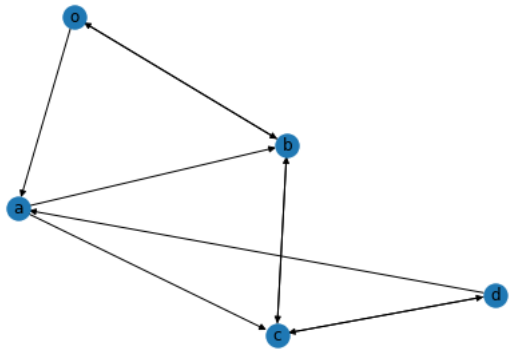
$$x_i = \mu + y_i$$

with variance $\sigma^2(y_i) = \sigma^2$ for each $i$. Assuming that the graph is connected and aperiodic, then $\lim_{t \to +\infty} x(t) = \alpha 1$. Since $\alpha = \pi'(\mu 1 + y)$, then $\sigma_\alpha^2 = \sigma^2 \sum_i \pi_i^2$ .

# 3 Exercise 1

The closed network used in this exercise is reported in figure 1.

Figure 1: Exercise 1 graph.




### 3.1 Point a

In this simulation the node $a$ is both the starting node and the ending node. I set the number of step to 10000, in order to get a balance between precision and computational time. The average time obtained according to simulations is about 6.70.

### 3.2 Point b

Now I compute the theoretical return time, obtaining 6.75. To do this I have used the formula

$$\mathbf{E}_i[T_i^+] = 1 + \sum_{j \in \mathcal{X}} P_{ij} \mathbf{E}_j[T_i],$$

reported in the theory section. Comparing it with the previous result I can observe that the error is about 0.05, so the simulation is approximately correct.

### 3.3 Point c

The starting node is now $o$ and the ending node is $d$. I compute the average time through 10000 simulations and I obtain 8.86.

### 3.4 Point d

The theoretical hitting-time is equal to 8.78. To compute it I have used the formula

$$\mathbf{E}_i[T_j] = 1 + \sum_{j \in \mathcal{X}} P_{ij} \mathbf{E}_j[T_i],$$

reported in the theory part. Comparing it with the previous result I notice that the error is approximately 0.08.

### 3.5 Point e

First of all I set the initial condition $X(0)$ to $[1, 0, 1, 0, 1]$. Simulating French - De Groot dynamics on $G$, it converges to the consensus state $[0.565, 0.565, 0.565, 0.565, 0.565]$. As a matter of fact, the graph $G$ is strongly connected and the condensation graph is composed of a single sink which is aperiodic. For this reason

$$\lim_{t \to +\infty} x(t) = \alpha 1.$$

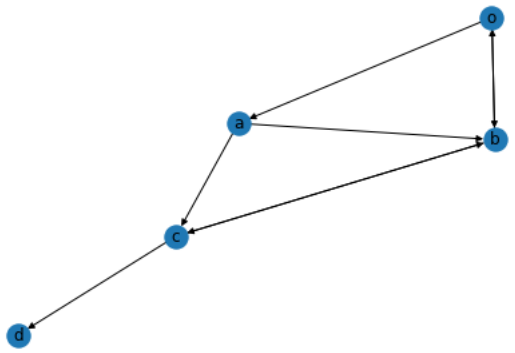
### 3.6 Point f

Now I assume that all the components of the initial state are i.i.d. random variables with variance $\sigma^2$. I compute 200 numerical simulations to approximate the variance of the consensus state, obtaining 0.018 as a result. I also calculate it analytically through $\sigma_\alpha^2 = \sigma^2 \sum_i \pi_i^2$ obtaining 0.0178 as a result.

### 3.7 Point g

Now I remove from the graph $G$ the edges $(d, a)$ and $(d, c)$. The obtained graph is represented in figure 2. The condensation graph contains a single aperiodic

Figure 2: Exercise 1 point G.




sink, that is the node $d$. As a result $\lim_{t \to +\infty} x(t) = \alpha 1$.

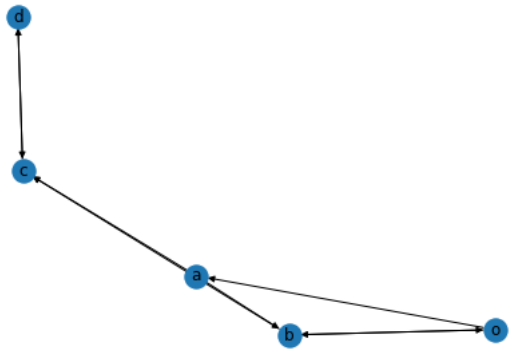
Moreover I observe that the asymptotic state is given by the fifth component of the initial state $x(0)$.

Then I suppose that each component of the initial state is given by an i.i.d random variable and I perform multiple simulations to compute the variance of the consensus state, getting 0.08 as a result.

### 3.8 Point h

Now I consider the initial graph, removing the edges $(c, b)$ and $(d, a)$ and I obtain the graph in figure 3.

Figure 3: Exercise 1 point H.




I simulate the French-De Groot dynamics on the new graph and I see that it does not converge to a consensus state.

The reason is the following. The subset $\{c, d\}$ is a trapping set and it is periodic. If there is a periodic trapping set, the system cannot reach the consensus state. Moreover, if I try to add a self loop in $c$ or $d$, the trapping set becomes aperiodic and the system converges.

# 4 Exercise 2

In this exercise I consider again the network in figure 1, but now we have many particles moving around in the network in continuous time. I simulate the system from two different perspectives: the particle perspective and the node perspective. I suppose that 100 particles start in node $o$ and the system is simulated for 60 times units.

## 4.1 Point a

From a particle perspective I can work in two ways.

### 4.1.1 Moving particles individually

I simulate the attachment of a Poisson clock to each of the particles and move them individually. I get 6.74 as average return time, which is, as predicted, more precise with respect to the theoretical return-time computed in 3.2. This is due to the fact that in the system there are 100 particles, each one with a Poisson clock: the error is about 0.019.

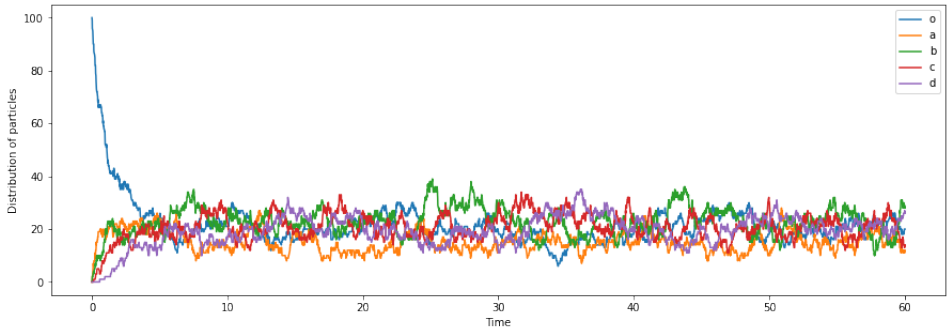
### 4.1.2 Using global rate

I consider a single system-wide Poisson clock with rate equal to the number of particles and move this particle to a neighbor node according to pr obability matrix Q. This leads to lower precision and computational speed: running it multiple times I always get very different results: 7,00 , 6,02 , 8,13...

## 4.2 Point b

By the node perspective each node has a rate proportional to the number of particles, in particular $w_i * n_{p(i)}$. So $\hat{\tau}_i = \frac{-ln(u)}{n_i w_i}$. The average of particles over time converges to $100\hat{\pi}$. The average of node $o$ is a little bit higher because it starts with every particle inside, but for longer periods it should converge. In figure 4 we can see the number of particles in each node during the simulation.

Figure 4: Exercise 2 point b: number of particles in each node during simulation.




# 5 Exercise 3

In figure 5 there is the network used in this exercise.

## 5.1 Point a

The results of the simulation with input rate equal to 1 and node rates proportional to the number of particles can be seen in figure 6. I also try with input rate equal to 100 and the results are in figure 7. In the first 8 time units there is a spike that makes me think the system can diverge. Instead, it stabilizes to his steady configuration.

## 5.2 Point b

For the fixed rate I create the NodeClock class, in order to have a gate for each node. If a particle finds two consecutive open gates at the same time, it can through both. Then I also have the *input_gate* which determines how many

Figure 5: Exercise 3 graph.

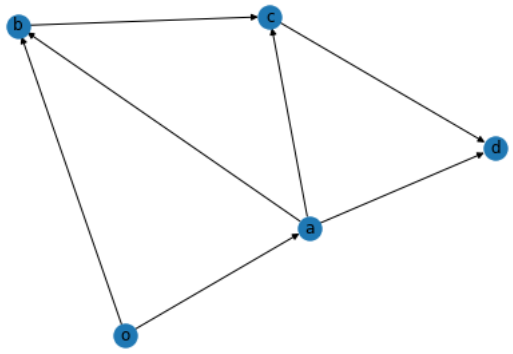
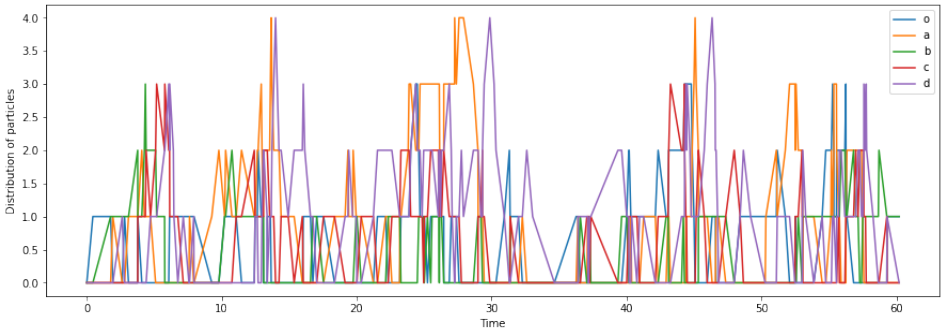



Figure 6: Simulation result with $input\_rate = 1$ and node rates proportional to the number of particles.




particles per time can enter the system in node $o$. For input rate equal to 1 the system seems to be stable. The resulting plot is available in figure 8. The system is not as stable as the proportinal rate one. It is sufficient to set the input rate equal to 2 to obtain a divergent system, as we can see in figure 9.

Figure 7: Simulation result with $input\_rate = 100$ and node rates proportional to the number of particles.

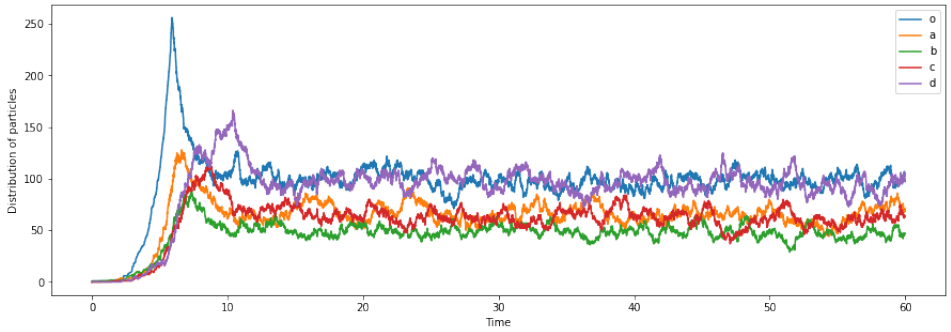



Figure 8: Simulation result with $input\_rate = 1$ and fixed node rates.

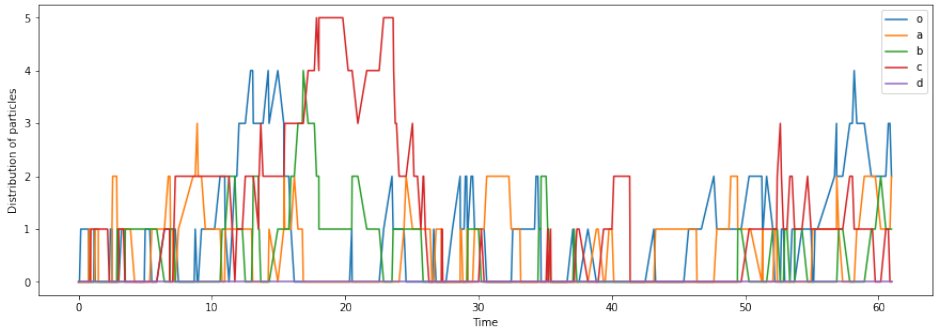



Figure 9: Simulation result with $input\_rate = 2$ and fixed node rates.