Homework 2

Andrea Rubeis, s290216

December 12, 2021

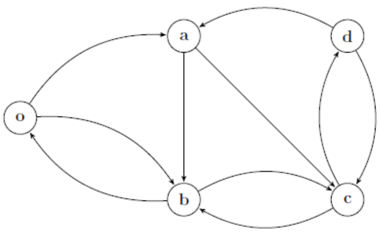
Abstract

Here are reported the solutions of the second homework of Network Dynamics and Learning course. Code here.

Problem 1

The first part of this assignment consists in studying a single particle performing a continuous-time random walk in the network described by the graph in figure with the transition matrix below.

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



Your task is to simulate the particle moving around in the network in continuous time according to the transition rate matrix (1). To help you with this we have provided some hints below. You should then use these simulations to answer the following questions:

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

First of all, let us briefly intoduce some theory concepts that will be exploited during the resolution of the following problems.
In this case, the random walk treated has 2 sources of randomness:

- **Neighbor's choice**: it follows an *uniform distribution.*
- **Time spent before jump**: since the time is a continous quantity, we need to model our random walk with a *continous Markow Chain*. For this reason we need to define a *Poisson process* defined as follows: for a given rate $r > 0$ let $S_1, S_2, ...$ be a sequence of indipendent random variables with exponential distribution. In our case the exponential random variable used is the following:

  $$t_{next} = \frac{-\ln(u)}{r}, \quad u \sim \mathcal{U}(0,1)$$

  The sequence below is defined as rate-r *Poisson clock*:

  $$T_0 = 0, \quad T_k = \sum_{1 \leq j \leq k} S_j, \quad k = 1, 2...$$

  The *Poisson process* is defined as:

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

After brefly summarizing the theory behind, we are ready to solve the exercise.

First of all we denote with x_0 and x_d the starting and end nodes of the random walk. In this case $x_0 = x_d = a$ and we simulate 1000 times the particle's movement with a random walk by a continous Markow chain model.

We consider a global poisson clock with a rate $\omega^* = \max_i(\omega_i)$, where $\omega_i = \sum_{ij} \Lambda_{ij}$ and we build the transition probabilities matrix Q defined as:

$$Q_{ij} = \frac{\Lambda_{ij}}{\omega^*}, \quad i \neq j \qquad Q_{ii} = 1 - \sum_{i \neq j} Q_{ij} \tag{1}$$

The particle jumps from node i to node j with probability Q_{ij} according to a global rate $\omega^* = 1$. We get an average time equals to 6.96.

(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.)**

The theoretical return time $\mathbb{E}_a[T_a^+]+$ is defined as:

$$\mathbb{E}_a[T_a^+] = \frac{1}{\omega_i \bar{\pi}_i} \tag{2}$$

Where $\omega = \Lambda\mathbf{1}$ and $\bar{\pi}$ is the probability distribution of the CTMC. After computing all the necessary vectors and matrices we find a value of

$\mathbb{E}_a[T_a^+] = 6.75$. The relative error with respect to what we found in 1.a is 3.16 %.

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

The answer to this question is the same of point a with $x_0 = o$ and $x_d = d$ and we get an average time equals to 8.80 after 1000 simulations.

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

Instead of simulate the average time between nodes o and d as done in point 1.c), here we compute the theoretical expected return time which has to satisfy the *recursive relations for the expected hitting times*:

$$\begin{aligned} \mathbb{E}_i[T_{\mathcal{S}}] &= 0, \quad i \in \mathcal{S} \\ \mathbb{E}_i[T_{\mathcal{S}}] &= 1 + \sum_{j \in \mathcal{V}} P_{ij} \mathbb{E}_j[T_{\mathcal{S}}] \quad i \notin \mathcal{S} \end{aligned} \tag{3}$$

In which we have $\mathcal{S} \subseteq \mathcal{V}$ where $\mathcal{V}$ are the states o, a, b, c, d and $\mathcal{S}$ corresponds to the set d. We get $\mathbb{E}_o[T_d] = 8.79$ with a relative error equal to 0.15%

(e) **Interpret the matrix Λ as the weight matrix of a graph $\mathcal{G} = (\mathcal{V}; \mathcal{E}; \Lambda)$, and simulate the French-DeGroot dynamics on $\mathcal{G}$ with an arbitrary initial condition $x(0)$. Does the dynamics converge to a consensus state? Motivate your answer**.

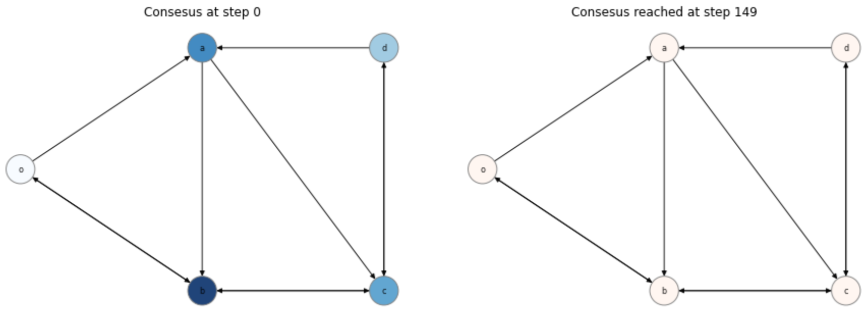
First of all we need to see if the *French-DeGroot* dynamics's conditions are satisfied:

- **Strong connectivity**: each node is reacheable from each other so the graph is strongly connected.
- **Aperiodicity**: the graph is aperiodic since the *gcd* of the number of cycles involving each node is equal to 1.

Since both conditions are satisfied the dynamics converge to a consensus state for sure. Again we simulate dynamic's evolution by multiplying the initial state $x(0)$ =[0.825 0.052 0.442 0.945 0.792] by the transition probability matrix P for 1000 times according to the French-DeGroot learning model:

$$x(t+1) = P'x(t) \tag{4}$$

We report below two pictures that better explain the evolution of the dynamics.



Consesus at step 0

Consesus reached at step 149

As we can see we have on the left the initial state, while on the right we report the consensus 0.635 reached after 149 steps by computing the equation (4) at each step.

(f) **Assume that the initial state of the dynamics for each node $i \in \mathcal{V}$ is given by $x_i(0) = \xi_i$, where $\{\xi_i\}_{i\in\mathcal{V}}$ are i.i.d random variables with variance σ^2. Compute the variance of the consensus value, and compare your results with numerical simulations.**

We compute the variance of the consensus (*crowd*) first by simulations and then by theoretical results.

Regarding the simulation phase,we start from an initial state uniformely distributed on the $[0, 1]$ interval.This means that each agent has expected value equal to $\frac{1}{2}$ and variance equal to $\frac{1}{12} = 0.0833$. To simulate the variance of the consensus we will simulate 1000 times again and apply (4) as before to reach the consensus. We get a simulated variance equal to $\sigma_s^2 = 0.0192$.

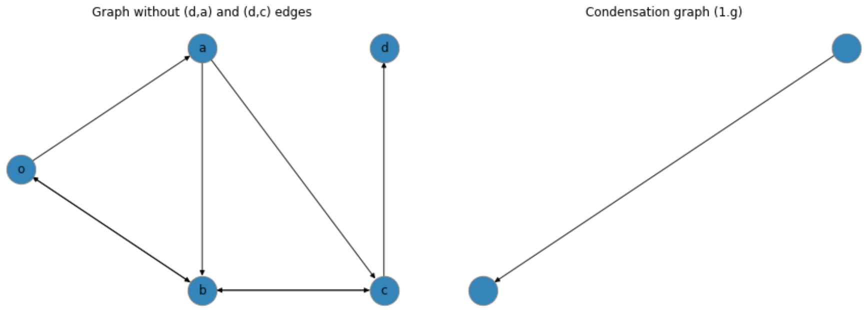
The theoretical consensus' variance is defined as:

$$\sigma_t^2 = \sigma^2 \sum \pi_k^2 \tag{5}$$

Where σ^2 is the variance of the individual and π is the invariant distribution. We get $\sigma_t^2 = 0.0187$, which is lower than the variance of the single agent (0.833) as expected (*wisdom of the crowd*).

(g) **Remove the edges (d; a) and (d; c). Describe and motivate the asymptotic behaviour of the dynamics. If the dynamics converges to a consensus state, how is the consensus value related to the initial condition $x(0)$? Assume that the initial state of the dynamics for each node $i \in \mathcal{V}$ is given by $x_i(0) = \xi_i$, where $\{\xi_i\}_{i\in\mathcal{V}}$ are i.i.d random variables with variance σ^2. Compute the variance of the consensus value. Motivate your answer.**

After removing the edges (d, a) and (d, c) we report the new graph and its condensation graph.



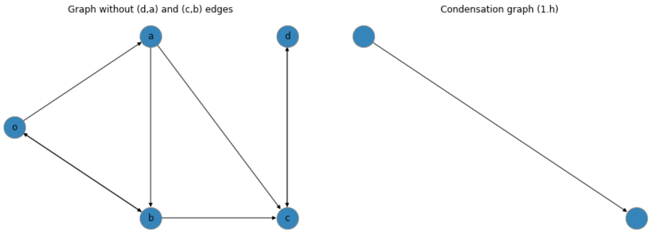
To identify a potential consensus we check the *French-DeGroot* theorem's conditions once more. In this case we find out that after removing (d, a) and (d, c) edges the graph is aperiodic but not strongly connected as we can see from the figure above. So FDG theroem cannot be applied. However, from theory we know that there is a more general result which tells us that, given a graph $\mathcal{G}$, if it posses a *globally reachable* component $\mathcal{C}_o$ (i.e. $\mathcal{C}_o$ has a single sink), then we have:

$$\lim_{t\to\infty} x_i(t) = \pi' x(0) \quad \forall i, \quad \pi = P'\pi \tag{6}$$

In our case we have a single sink d which is not aperiodic, so to reach a consensus we need to add a self-loop on it to transform it into an aperiodic component. We start from the initial state $x(0) = [0.064\ 0.524\ 0.347\ 0.822\ 0.771]$ The consensus we reach both with 1000 simulations and theoretical result (6) is equal to 0.771 which is exactly the initial state of d. This is not a coincidence, indeed having a single sink means that all the nodes have edges pointing towards to the sink, which implies that the sink influences all of them.

(h) **Consider the graph $\mathcal{G} = (\mathcal{V}; \mathcal{E}; \Lambda)$, and remove the edges (c; b) and (d; a). Analyse the French-DeGroot dynamics on the new graph. In particular, describe and motivate the asymptotic behaviour of the dynamics in terms of the initial condition $x(0)$.**

After removing the edges (d, a) and (c, b) we report the new graph and its condensation graph:



In this case we have again a single sink given by the c and d nodes. For the same reasoning made before, we can reach the consensus only if c and d have the same intial value, otherwise it is not possible. Also in this case the sink c, d is not aperiodic so we apply a self-loop on d again according to the same reasoning done in point g). The consensus we reach both with 1000 simulations and theoretical result (6) is equal to 0.069.

Problem 2

In this part we will again consider the network of Fig. 1, with weights according to (1). However, now we will simulate many particles moving around in the network in continuous time. Each of the particles in the network will move around just as the single particle moved around in Problem 1: the time it will stay in a node is exponentially distributed, and on average it will stay $1/w_i$ time-units in a node i before moving to one of its outneighbors. The next node it will visit is based on the probability matrix $P = diag(\omega)^{-}1\Lambda$, where $\omega = \Lambda 1$.
Your task is to simulate this system from two different perspectives: the *particle perspective*, i.e. "follow the particle", and the *node perspective*, i.e. "observe from the node".
Simulating the system from a particle perspective is exactly as in Problem 1, but here you have to follow many particles instead.
To simulate the system from the node perspective you instead have to observe the particles from the node. When doing this you do not have to care about each single particle, but only about the number of particles in the node. Note that at node i, each particle in that node will stay there on average $1/\omega_i$ time units. Therefore, the node will pass along particles at a rate proportional to the number of particles in the node. In fact, if at time t the number of particles in node i is $n_i(t)$, it will pass along particles at a rate of $n_i(t)\omega_i$. The departure times of the node can thus be seen as a Poisson process with rate $n_i(t)\omega_i$. At each tick of the Poisson clock of the node, it will move a particle to a neighboring node. The node to which the particle will move is again based on the normalized transition rate matrix P.
Simulate the system from the two perspectives, and then answer the following questions (we have again provided a hint below):

(a) **Particle perspective**:

- If 100 particles all start in node a, what is the average time for a particle to return to node a?
- How does this compare to the answer in Problem 1?

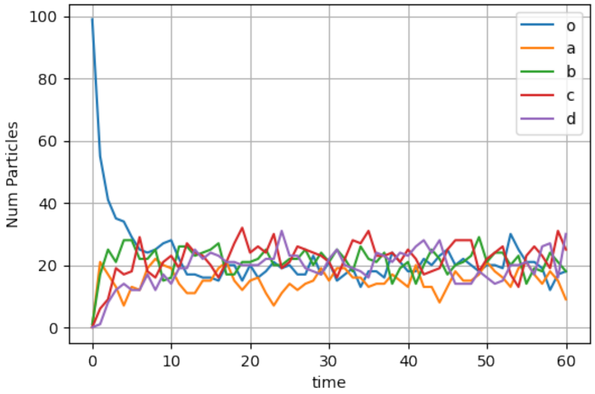
(b) **Node perspective**:

- 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?
- Illustrate the simulation above with a plot showing the number of particles in each node during the simulation time.
- Compare the simulation result in the first point above with the stationary distribution of the continuous-time random walk followed by the single particles.

The question 2.a can be easily solved by considering the same approach used in 1.a in which we evaluated the random walk of a single particle for a number of simulations equal to 1000. Since the particles are indipendent one another, computing the average time required to execute a random walk from a to a by 100 particles is equivalent to simulate the same random walk for 100 times. Also here the particles move from one node to another one according to a global Poisson clock of rate $\omega^* = 1$ as before and the same transition-probability matrix Q used in 1.a. The average time we get is 7.01, as you notice, this is really close to 6.96 found in 1.a, this proves our reasoning.

To answer the 2.b question we consider a Poisson clock of rate equal to ω^* times n, where n is equal to 100 (number of particles) and the same transition probability matrix Q in 1.a..

We report below the graph representing the number of particles in each node during the simulation.



As we can see from the graph above, since all the particles start in o, few time instants later they start to leave the origin node in order to reach an equilibrium that has an oscillation of approximately 10-30 particles per node. The last configuration of the system is [21 11 22 24 22], by normalizing we get [0.21, 0.11, 0.22, 0.24, 0.22] which is similar to the theoretical invariant distribution

π =[0.185, 0.148 0.222 0.222 0.222] which represent also the average number of particles in every node.

Problem 3

In this part we study how different particles affect each other when moving around in a network in continuous time. We consider the open network and transition rate matrix Λ_{open} according to the picture below.

$$\Lambda_{\text{open}} = \begin{array}{cc} \begin{array}{ccccc} o & a & b & c & d \end{array} & \\ \begin{pmatrix} 0 & 2/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 2/4 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} o \\ a \\ b \\ c \\ d \end{array} \end{array}$$

For this system, particles will enter the system at node o according to a Poisson process with rate $\lambda = 1$. Each node will then pass along a particle according to a given rate, similar to what you did in Problem 2 with the "node perspective". You will simulate two different scenarios that differ by what rate the nodes will pass along particles: *i*) proportional rate, and *ii*) fixed rate. In scenario *i*), each node will pass along particles according to a Poisson process with rate equal to the number of particles in the node. In scenario *ii*), each node will instead pass along particles with a fixed rate of 1.

Note that since node d does not have a node to send its particles to, when the Poisson clock ticks for this node you could simply decrease the number of particles in the node by one (if there are any particles in the node). Equivalently think of another node d' connected to node d, such that at every tick of the Poisson clock of d, it sends a particle to node d'.

The goal is to simulate the two systems and answer the following questions:

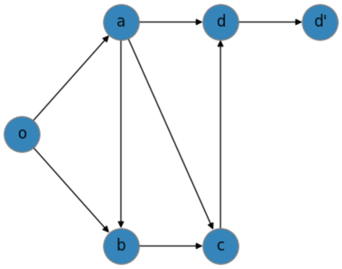
(a) **Proportional rate**: The rate of the Poisson clock of each node is equal to the number of particles in it.

- Simulate the system for 60 time units and plot the evolution of the number of particles in each node over time.
- What is the largest input rate that the system can handle without blowing up?

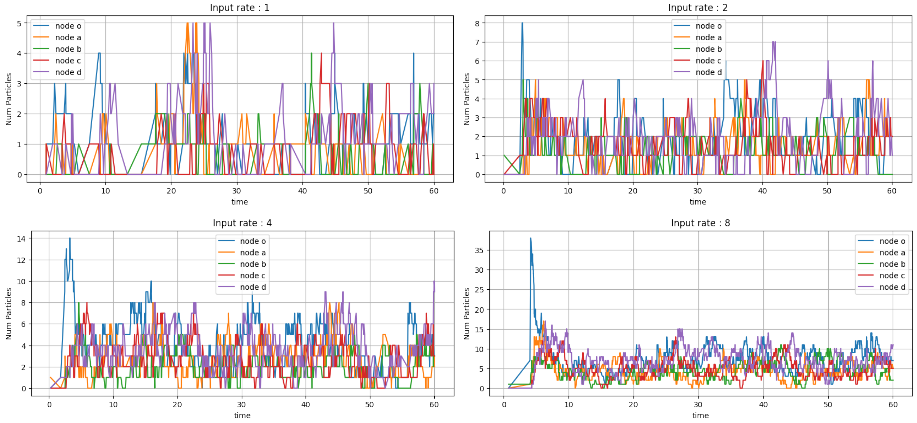
(b) **Fixed rate**:

- Simulate the system for 60 time units and plot the evolution of number of particles in each node over time.
- What is the largest input rate that the system can handle without blowing up? Why is this different from the other case?

In 3.a case we have 2 Poisson clocks: the usual global clock with rate proportional to the number of particles inside the system (scenario i)) and another one with rate $\lambda = 1$ which ticks every time a new particle enters in the system (scenario ii)). This is equivalent to have one clock whose rate is the minimum between the two.
According to the new transition rate matrix Λ_{open} we get the following graph:

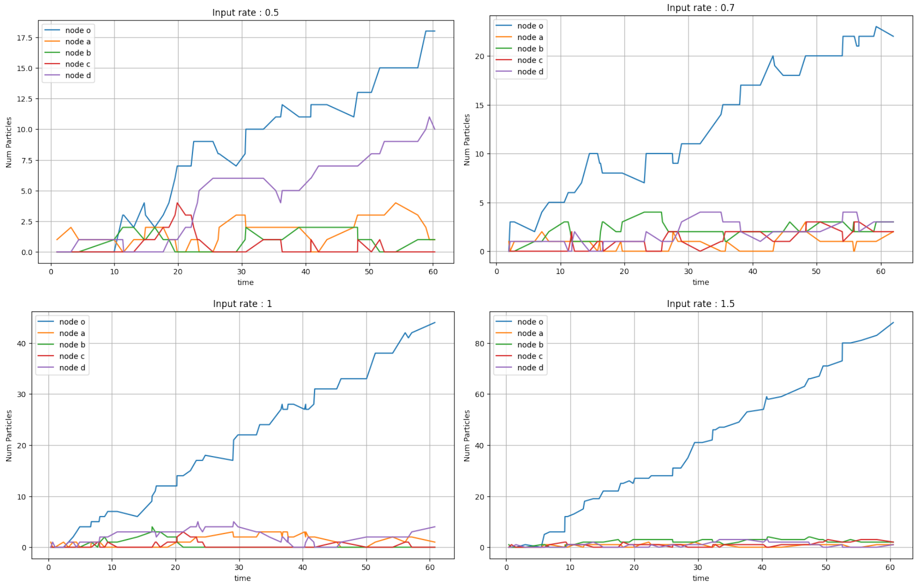


We simulate for 60 time units as requested with the following input rates: 1, 2, 4, 8.



As you can see by the figures, the system does not blow up for any input rate simiulated, This is due to the global Poisson clock with proportional rate, that allows the exchange of particles that each node makes with its neighbors. So none of the nodes accumulates an abundant number of particles compared to the other nodes. Indeed in the graphs above, we can see that each node has around from 0 to 20 number of particles inside for the whole simulation.

Instead case 3.b) is the same of 3.a) where the global clock in this case has a fixed rate equals to ω^* times the nodes and it is not proportional anymore as before. In this case we simulate for input rates: $0, 5$, 0.7, 1 and 1.5.



We can see that the amount of particles in o is much higher compaerd to the other nodes for all the input rates simulated, so the system in this case quickly blows up, in particular this happens for input rates $\lambda \geq 0.7$. So the system is not able to scale up as it was instead in case 3.a).