Network Dynamics and Learning Homework II

Shannon Mc Mahon
Student id: s289958

***Abstract*—The following document contains the discussion of the solutions of the assigned exercises.**
Python implementation is available on *Github*. Note that simulation results will of course differ slightly from eachother on different runs.

I.

Consider the network in Figure I. We study a continuous-

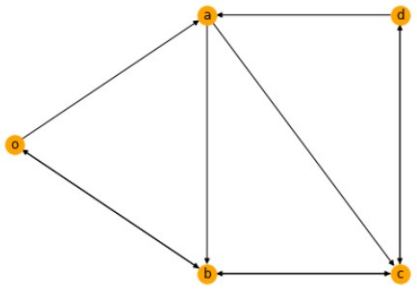

Fig. 1.

time random walk (i.e. Markov Chain) of a particle on such a graph. Specifically, the particle will move around according to the following transition rate matrix Λ:

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

(a) Firstly, we are interested in simulating the average time it takes a particle that starts in node *a* to leave the node and then return to it. To do so, we compute the matrix of transition probabilities, which is defined as follows

$$\bar{P}_{ij} = \frac{\Lambda_{ij}}{w^*} \quad i \neq j, \quad \bar{P}_{ii} = 1 - \sum_{j \neq i} \bar{P}_{ij} \tag{1}$$

where ω^* is our rate for the global Poisson clock and is defined as $\omega^* = \max \omega_i$, $\omega = \Lambda \mathbb{1}$. Whenever the clock ticks the particle moves, specifically the probability of it moving from node i to j is given by $\bar{P}_{ij}$. The time between two consecutive ticks instead is equal to $t_{next} = -\frac{ln(u)}{\omega^*}$, where $u \sim \mathcal{U}(0,\,1)$ is a random variable with uniform distribution. If we simulate the process 10^4 times, we find that on average it takes the particle 6.74 s before it returns to node *a*.

(b) We now compare the result of point (a) with the theoretical return-time, which is given by

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

where $\bar{\pi}$ is the Laplacian-invariant distribution of the graph and satisfies $L\bar{\pi} = 0$, $\mathbb{1}'\bar{\pi} = 1$, with $L = diag(\omega) - \Lambda$. In other words $\bar{\pi}$ is the normalised eigenvector corresponding to the eigenvalue 0, and from the Perron Frobenius Theorem we know it is unique.
By applying equation (2) we find that the vector of expected return times is given by [9, 6.75, 4.5, 4.5, 6.75]. Consequently, the expected return time for node a is 6.75 s (second position in the vector) which is very close to the estimate we had found with our simulation in point (a) as a consequence of the law of large numbers.

(c) By applying the same process explained in point (a) we can simulate the average time it takes to move from node *o* to node *d*. This is equal to 8.83 s.

(d) We now compare the result obtained with simulations in point (c) with the theoretical hitting-time $\mathbb{E}_o[T_d]$. From the theory we know that given a continous-time Markov chain $X(t)$, $D = diag(\omega)$, $P = D^{-1}\Lambda$ and a subset $S \subseteq V$, the hitting time T_S is defined as $T_S := inf\{t \geq 0 : X(t) \in S\}$. Furthermore the expected hitting time is given by:

$$\begin{aligned} &\mathbb{E}_i[T_s] = 0 \quad if \quad i \in S \\ &\mathbb{E}_i[T_s] = \frac{1}{w_i} + \sum_j P_{ij}\, \mathbb{E}_j[T_s] \quad if \quad i \notin S \end{aligned} \tag{3}$$

In our specific case the set S coincides with node *d*. By applying the above equations we find that the vector of expected hitting times is given by [8.786, 7.143, 7.071, 3.357, 0]. It follows that the expected hitting time for node *d* starting from *o* is 8.786 s (first entry). Once again the simulations gave a very good approximation.

(e) Consider now the graph $G = (V, E)$ with weight matrix Λ. Given the French-DeGroot learning model

$$x(t) = P^t x(0) \tag{4}$$

where $x(0)$ is an arbitrary initial condition, convergence to a consensus is guaranteed since the graph is strongly

connected and aperiodic. Specifically, we have

$$x(t) \to \sum_k \pi_k x_k(0) = \pi^{'} x(0) \mathbb{1} \tag{5}$$

where $\pi = P^{'}\pi$ is the unique invariant distribution of P. If for example we take $x(0) =$ [0.407, 0.946, 0.338, 0.614, 0.343], since $\pi =$[0.13, 0.174, 0.261, 0.261, 0.174] we find that the consensus value is 0.525. Such result is obtainable also by simulating the process by means of equation (4). After 500 iterations, we have $x =$[0.525, 0.525, 0.525, 0.525, 0.525].

(f) We now assume that the initial state of the dynamics for each node $i \in V$ is given by $x_i(0) = \xi_i$, where $\{\xi_i\}_{i \in V}$ are i.i.d random variables with variance σ^2. In our case we will use the continuous uniform distribution $\mathcal{U}(0,\, 1)$, so that $\mu = \frac{1}{2}$ and $\sigma^2 = \frac{1}{12} = 0.083$. Since aperiodicity and strong connectivity are satisfied, we can compute the variance of the consensus value σ_c^2, and compare the result with numerical simulations. Recall that

$$\sigma_c^2 = \sigma^2 \sum_k \pi_k^2 \tag{6}$$

which allows us to determine that $\sigma_c^2 = 0.018$. Notice how $\sigma_c^2 < \sigma^2$, in other words the wisdom of the crowd is greater than that of the single individual.
We can also use numerical simulation to illustrate the same result. By considering 10^4 simulations of the French-DeGroot model, each one starting from a randomly extracted initial condition $x(0)$ sampled from the uniform distribution and having a duration of 500 iterations, we can calculate the average of the variances found at the end of each simulation, and find an empirical variance of the consensus state equal to 0.018.

(g) We now consider the graph of Figure 2, obtained from G by removing edges (d,a) and (d,c).

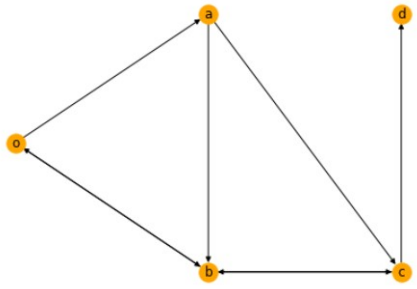

Fig. 2.

In this case, the graph is no longer strongly connected. However, by adding a self-loop to d we can identify a globally reachable and aperiodic component, coinciding with node *d*. These are the more general conditions under which convergence to a consensus is guaranteed. Our new transition rate matrix Λ is given by:

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

Once again we can calculate the consensus value by applying equation (5). For example, if $x(0) =$[0.327, 0.296, 0.058, 0.922, 0.46] we find that the opinion converges to the consensus 0.46. Notice how this value coincides with the initial opinion of node d. This is no coincidence: in fact $\pi =$[0, 0, 0, 0, 1], i.e. it has support only on d, meaning that only the opinions of people in the globally reachable component have an influence on the final result. It is possible to obtain the same result via numerical simulation: after 500 iterations we get $x =$[0.46, 0.46, 0.46, 0.46, 0.46].
We can also repeat what was done in point (f) for our new graph, and find according to equation (6) that the variance of the consensus value is $\sigma_c^2 = 0.083$. Once again node d is the only one with an influence, and in particular we have $\sigma_c^2 = \sigma^2$.

(h) Starting from the network of Figure I we remove edges (c,b) and (d,a). The resulting Λ transition rate matrix is given by

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

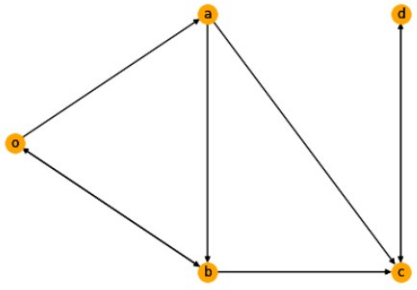

Fig. 3.

while the network is visible in Figure 3. If we look into the French-DeGroot dynamics on the new graph, we see that in this case although $C_0 = \{c, d\}$ is globally reachable, it is not aperiodic (since its period is 2), and as a result we do not reach consensus unless the initial opinions of c and d are equal. For example, with initial condition $x(0) =$[0.734 0.208 0.064 0.652 0.125]

we find x =[0.408 0.429 0.246 0.652 0.125] after 500 iterations, i.e. there is no common opinion, while if $x(0)$ =[0.734, 0.208, 0.064, 0.125, 0.125] then we reach x =[0.125, 0.125, 0.125, 0.125, 0.125], i.e. there is convergence to the initial opinion of c and d. Another way to understand this phenomenon is to visualise the condensation graph, in which we would have $C_0 = \{c, d\}$ and $C_1 = \{o, a, b\}$ with an edge from C_1 to C_0, i.e. C_1's opinion is influenced by that of C_0, but not vice versa.

II.

In this exercise we will use the network given in Figure I with weights given by its relative transition rate matrix. The aim is to simulate many particles moving around in the network in continuous time.

(a) We firstly focus on simulating the system from a particle perspective, i.e. "following the particle". In other words this is none other than the situation considered in Exercise I, with the difference that here there are many particles to follow. Given 100 particles, if we wish to calculate the average time necessary for a particle starting in node a to return to such node, all we need to do is apply the solution given in point (a) of Exercise I, but with a number of simulations equal to the number of particles, i.e. 100. Put differently, simulating a random walk with 100 particles is equal to simulating a random walk with one particle 100 times. It should not come as a surprise that the result of the simulation, in this case 6.08 s, is close to the previously found 6.74 s.

(b) We now simulate the system from the node perspective, i.e. we are not interested in the single particles, but in the number of particles at each node in time. Suppose we have 100 particles that start in node o, and that we are interested in determining the average number of particles in the various nodes at the end of the simulation. To solve this problem we define a system-wide Poisson clock with rate equal to ω^* times the number of particles. At each tick we randomly, and proportionally to the number of particles in the different nodes, select a node, and move a particle from such node according to the transition probability matrix $\bar{P}$.

Figure 4 shows the dynamics of such simulation: at the beginning of the process we have 100 particles all in node o, but as time goes by particles leave node o to move to other nodes in the graph. In fact, the blue curve of node o has an initial decreasing trend, unlike the other nodes which are receiving particles and thus have an increasing trend. The process then stabilises, as the particles are distributed throughout the network. After 60 s we have a distribution of the 100 particles among the different nodes given by [14, 15, 25, 15, 31], or in normalised form [0.14, 0.15, 0.25, 0.15, 0.31]. This is a quite appreciable approximation of the stationary probability vector $\bar{\pi}$ =[0.185, 0.148, 0.222, 0.222, 0.222], which contains the probabilities of a particle ending up in a specific node (e.g. 0.148 is the probability for node a).

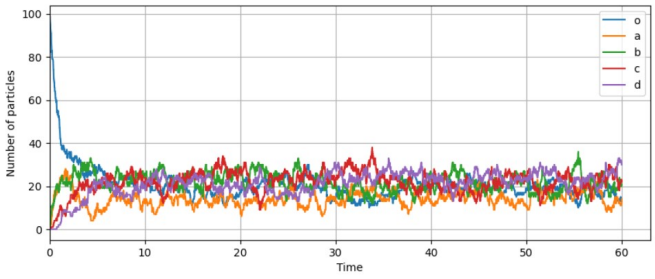

Fig. 4.

III.

In this final exercise we study how different particles affect each other when moving around in a network in continuous time. I. We will use the network in Figure 5 and a transition rate matrix Λ_{open} defined as follows:

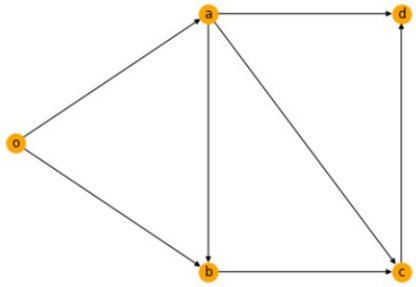

Fig. 5.

$$\begin{array}{c} \begin{matrix} o & a & b & c & d \end{matrix} \\ \begin{pmatrix} 0 & 2/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 2/4 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} o \\ a \\ b \\ c \\ d \end{matrix} \end{array}$$

We assume that particles enter the system at node o according to a Poisson process with input rate $\lambda = 1$, and that each node of the network passes particles along according to a second given rate. Since the input rate is constant we can precompute the times of entry for each particle. Also notice how node d does not have a node to send its particles to, so when the clock ticks for this node we will simply decrease the number of particles in the node by one.

We now consider two possible options for the rate at which nodes pass on particles.

(a) In the first case we chose a *proportional rate*, in which each node passes along particles according to a Poisson process with rate equal to the number of particles in such node. To simulate the system we use a global Poisson

clock with rate equal to ω^* times the number of particles in the system in that moment $n(t)$. At each iteration of the simulation we compute the time of the next tick: if it is prior to next entering time of a particle then we randomly and proportionally to the number of particles in each node, sample a node and move a particle from such node to another one using the matrix $\bar{P}$. Otherwise, we simply move in time to the next entry of a particle in the system.

Due to the proportional nature of the rate of the clock we have that an increase in the input rate of particles in the system leads to an increase in the speed of the clock (i.e. the rate). Therefore there is no upper bound for the values that $lambda$ can assume. In fact, figures 6 and 7 show the simulation results for 60 time units, respectively for $\lambda = 1$ and $\lambda = 10^4$, and as expected there is no blow-up effect.

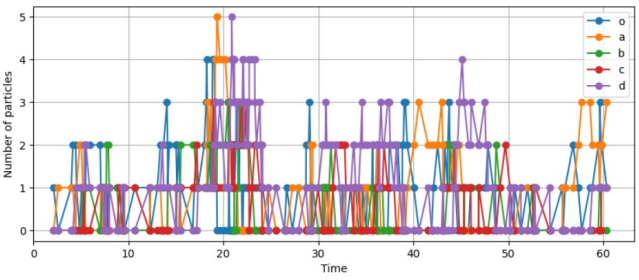

Fig. 6. Proportional rate scenario with $\lambda = 1$

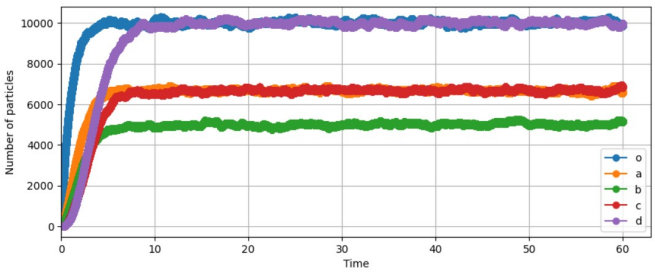

Fig. 7. Proportional rate scenario with $\lambda = 10^4$

(b) The second scenario we consider instead is the *fixed rate* one, in which each node passes along particles with a fixed rate of 1. We define a fixed rate for the internal global Poisson clock which is equal to ω^* times the number of nodes, so unlike the first case we saw there is no dependence on the number of particles. As far as time management is concerned, we simply let particles enter between consecutive ticks based on the precomputed entering times.

Unlike the previous setting, here we can find a critical value of the input rate, above which we observe a blow-up effect. Figures 8, 9, 10 and 11 show simulations of the system for 60 time units, with λ equal to 0.5, 0.7, 1 and 3 respectively. We observe that for values greater or equal to 0.7 particles accumulate in node o. In fact, by keeping the internal rate fixed no matter what the input rate λ of particles is, we do not allow the system to adapt accordingly as instead was the case with the proportional rate.

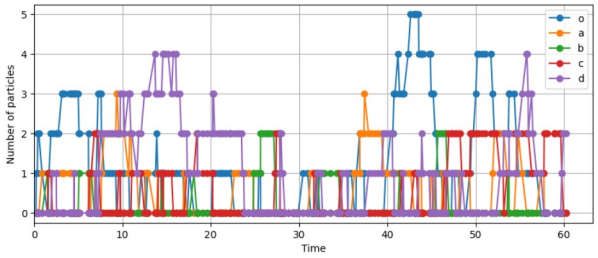

Fig. 8. Fixed rate scenario with $\lambda = 0.5$

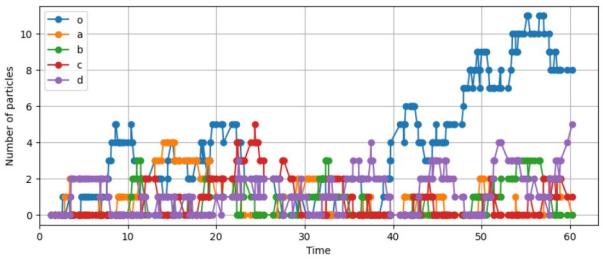

Fig. 9. Fixed rate scenario with $\lambda = 0.7$

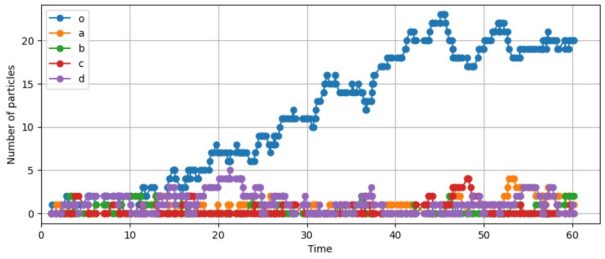

Fig. 10. Fixed rate scenario with $\lambda = 1$

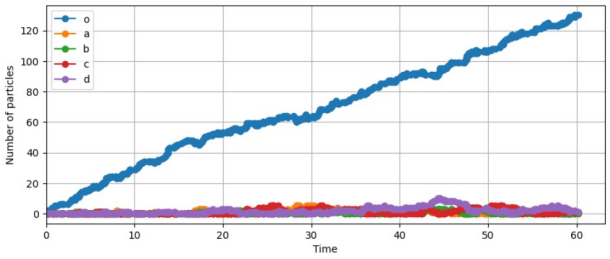

Fig. 11. Fixed rate scenario with $\lambda = 3$