Homework 6 - Markov chains

1) Discrete-time queueing model for a service station. Suppose customers can arrive to a service station at times $n = 0, 1, 2, \ldots$ In any given period, independent of everything else, there is one arrival with probability p, and there is no arrival with probability $1 - p$. Suppose customers are served one-at-a-time on a first-come-first-served basis. If at the time of an arrival, there are no customers present, then the arriving customer immediately enters service. Otherwise, the arrival joins the back of the queue.

In a time period n, events happen in the following order: (i) arrivals, if any, occur; (ii) service completions, if any, occur; and (iii) service begins on a new customer if there has been an arrival to an empty queue or a service has just finished and there is another customer present.

Assume that service times are i.i.d. geometric RVs (each with parameter q) that are independent of the arrival process. Note that a customer who enters service in time n can complete service, at the earliest, in time $n + 1$ (in which case his service time is 1). Let X_n denote the number of customers at the station at the end of time period n; i.e., after the time-n arrivals and services. Note that X_n includes both customers waiting as well as any customer being served.

A) Suppose that Y is a geometric RV with parameter q. Show that $\mathrm{P}\left[Y = i \,\middle|\, Y \geq i\right] = q$, for $i = 1, 2, \ldots$

B) Suppose that a particular customer begins service at time n, and does not complete service in periods $n + 1$, $n + 2$, or $n + 3$. Given this, what is the conditional probability that he will complete service in period in $n + 4$?

C) Explain why $X_{\mathbb{N}} = X_0, X_1, \ldots, X_n, \ldots$ is a Markov chain (MC).

D) Derive the transition probability matrix of $X_{\mathbb{N}}$.

2) Probability of the instant when a state is visited for the first time. Suppose that $X_{\mathbb{N}} = X_0, X_1, \ldots, X_n, \ldots$ is a MC with state space S, and for $i, j \in S$ define

$$g_{ij}^1 = \mathrm{P}\left[X_1 = j \,\middle|\, X_0 = i\right],$$
$$g_{ij}^n = \mathrm{P}\left[X_n = j, X_{n-1} \neq j, \ldots X_2 \neq j, X_1 \neq j \,\middle|\, X_0 = i\right], \;\; n \geq 2.$$

A) Derive a recursion that relates g_{ij}^n to g_{ij}^{n-1}. (Hint: condition on the first transition.)

B) The quantities g_{ii}^n represent the probabilities that starting from $X_0 = i$, the MC revisits i for the first time after exactly n transitions. Write an expression in terms of the g_{ii}^n, $n \geq 1$, for the probability f_i that the MC ever revisits state i, i.e.,

$$f_i = \mathrm{P}\left[\bigcup_{n=1}^{\infty} X_n = i \,\middle|\, X_0 = i\right].$$

Hence, derive yet another characterization of recurrence (or transience) of state i in terms of the g_{ii}^n, $n \geq 1$.

3) Expected discounted costs. Suppose that $X_{\mathbb{N}} = X_0, X_1, \ldots, X_n, \ldots$ is a MC with state space S, and transition probabilities P_{ij} for $i, j \in S$. Suppose that $c(\cdot)$ is a cost function so that we are charged cost $c(i)$ when we are in state $i \in S$. Suppose that $0 < \lambda < 1$ is a discount factor. Let

$$v(k, i) = \mathbb{E}\left[\sum_{n=0}^{k} \lambda^n c(X_n) \,\middle|\, X_0 = i\right].$$

Show that

$$v(k + 1, i) = c(i) + \lambda \sum_{j \in S} P_{ij} v(k, j), \quad k \geq 0.$$

4) Ranking of nodes in graphs. The most popular algorithms to rank pages in a web search are stochastic. Consider a web surfer that visits a page and clicks on any of the page's links at random. She repeats this process forever. What fraction of her time will be spent on a given page? The answer to this question is the rank assigned to the page. The same idea can be used to understand the structure of networks in different settings. For example, we can use this algorithm to extract connectivity information from a social graph. Say we choose a student and ask her to direct us to any of her friends selected randomly. We then go to this friend repeat the question and are directed to this new student. This is no different from the random web surfer model. Repeating this process forever we can therefore infer the degree of connectedness of students in the class from the average number of visits to each of them. This is not a pointless exercise. To market products, for example, it is worthwhile to concentrate the effort in the individuals that are most connected to other persons. The important insight here is that the network possesses knowledge that individuals do not. For this assignment we use a homework collaboration matrix that you can download from the class's webpage.

For a formal problem definition consider a network with J nodes. Describe connectivity by a directed graph $G(V, E)$ with V denoting the set of nodes and E the set of edges. An edge is an ordered pair $e := (i, j)$ representing a link from node i to node j. Further define $n(i)$ as the i-th node's neighborhood containing the indexes j to which i is pointing, i.e., $n(i) := \{j : (i, j) \in E\}$. Let N_i be the number of nodes in the neighborhood of i. Similarly, define the incoming neighborhood of i as the set of nodes that point to i, i.e., $n^{-1}(i) := \{j : (j, i) \in E\}$.

An outside agent approaches an arbitrary node A_0 at time $n = 0$. From there it jumps to one of the neighbors $A_1 \in n(A_0)$ of A_0 at time $n = 1$, then to one of the neighbors of this neighbor and so on. If the agent is visiting node A_n at time n it will visit a node in the neighborhood $n(A_n)$ at time $n+1$ with all neighbors chosen equiprobably. The fraction of time the agent spends visiting node i, is defined as the node's rank. To express this mathematically define the indicator function $\mathbb{I}\{A_n = i\}$ with value 1 when the agent visits i at time n and 0 otherwise. The rank r_i of node i is then given by

$$r_i(A_0) := \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} \mathbb{I}(A_m = i). \tag{1}$$

Since we are considering equal probabilities of jumping to any neighbor, the probability P_{ij} of the agent transitioning from node i to node j is

$$P_{ij} := \mathrm{P}\left[A_{n+1} = j \,\middle|\, A_n = i\right] = \frac{1}{N_i}, \quad j \in n(i) \tag{2}$$

where, we recall, N_i is the number of nodes in the neighborhood of i. The movement of the agent through the nodes in the graphs is called an equiprobable random walk in a graph.

A) Markov chain model. The random process of agent visits $A_{\mathbb{N}}$ is a MC. Explain. Give conditions for the following statements to be true:

- State i, meaning visit of agent to node i, of this MC is transient.
- All states of this MC are transient.
- All the states of this MC are recurrent.
- All states of this MC are aperiodic.
- All states are positive recurrent.
- All states are ergodic.
- The MC is irreducible.

Some of the above statements might be always true and some might be never true. Also notice that the agent here behaves somewhat different from the agent we covered in class. In class, we covered the ranking problem assuming that the random walk was recurrent, aperiodic, ergodic and irreducible. In the rest of the exercise you can assume that all states in this random walk are recurrent, aperiodic and ergodic, but you *cannot* assume that the MC is irreducible unless otherwise explicitly stated. If you wish, you are going to study the extent to which we can deal with lack of irreducibility. Thus, before proceeding make sure that you understand correctly the graph's aspect when the MC is not irreducible.

B) Implement the random walk on the graph. We are now ready to build an algorithm to compute r_i. We start with a randomly chosen node $A_0 = i$ and jump equiprobably to any of its neighboring nodes $j \in n(i)$. The probability of selecting any of this nodes is $P_{ij} = 1/N_i$. We repeat this process a large number of times n and keep track of the number of visits to each node. The rank r_i is then approximated as the ratio between the number of visits to node i and the total time n, i.e.,

$$r_i(A_0) \approx \frac{1}{n} \sum_{m=1}^{n} \mathbb{I}(A_m = i). \tag{3}$$

Write your Matlab code to implement this random walk. Use this code to compute the rankings as defined in (3) for $A_0 = 1$. What condition needs to be satisfied for the ranks in (1) to be independent of the initial state A_0? If this condition is *not* satisfied we can modify the algorithm by introducing an artificial node that is connected to all members of the graph. You can think of this node as the class' professor that knows, or should know, all students. Call these modified ranks r_i and approximate them as

$$r_i \approx \frac{1}{n} \sum_{m=1}^{n} \mathbb{I}(A_m = i). \tag{4}$$

The definitions in (3) and (4) look the same but recall that the agent visits are in different graphs. Compute your modified ranks. Convergence of this algorithm is admittedly very slow. Rough numerical approximations are acceptable as the answer to this question.

C) Probability update. The algorithm in Part *B* is certainly a possibility, but we can obtain a faster version by exploiting the fact that these random visits can be modeled as a MC. Let $p_i(n) := \mathrm{P}[A_n = i]$ denote the unconditional probability that the outside agent is at node i at time n. It is possible to express $p_i(n+1)$ in terms of the probabilities at time n of those nodes that can transition into i. Write this probability update. Define the vector $\mathbf{p}(n) := [p_1(n), \dots, p_J(n)]^T$ and write this update equation in matrix form.

D) Find ranks using the probability update. An interesting property of MCs is the existence of limit probabilities $\lim_{n\to\infty} p_i(n)$ under some conditions. State these conditions. These limit probabilities might depend on the initial probability distribution $\mathbf{p}(0)$. If this vector is chosen such that all the initial probability is on A_0, i.e., $p_{A_0}(0) = 1$ the rank can be equally computed as

$$r_i(A_0) = \lim_{n\to\infty} p_i(n). \tag{5}$$

Explain why this is the case. Write your Matlab code to compute ranks using the property stated in (5). Use this code to compute the rankings as defined in (3) for $A_0 = 1$. The ranks in (4) obtained from the modified graph can be computed as the limit

$$r_i = \lim_{n\to\infty} p_i(n) \tag{6}$$

for any initial probability distribution. Explain why this is the case. Modify your function to compute your ranks r_i. Run you function and show the rankings. A convenient initial probability distribution is $\mathbf{p}(0) = (1/J)\mathbf{1}$.

E) Recast as system of linear equations. Restrict attention to the modified graph containing the fully connected node. As you have already seen and explained, ranks are independent of the initial state A_0. Use your knowledge of MCs to recast the ranking problem as the solution of a system of linear equations. Solve this system of linear equations and compare with the results of Parts *B* and *D*.

F) Recast as eigenvalue problem. Restrict attention to the modified graph containing the fully connected node. Use your knowledge of MCs to recast the ranking problem as an eigenvector problem for a suitably chosen matrix. Compute this eigenvector and compare with the results of Parts *B*, *D* and *E*. The Matlab function to compute eigenvalues and eigenvectors is `eig`.

G) Discuss advantages of each method. Restrict attention to the modified graph containing the fully connected node. All four methods yield the same results but have particular advantages that make them suitable for different applications. Discuss. Most of these advantages were discussed in class. But there is a particular advantage of the method in Part *D* that we did not discuss and you should now be able to appreciate.