INSTITUTO SUPERIOR TÉCNICO | UNIVERSIDADE DE LISBOA
COMPLEX NETWORKS 2017/18

Solution of the 2nd Exam | Saturday, 27 January 2018 (2h)

This exam can be answered in Portuguese or in English in a standard IST exam sheet.
All exercises are equally valued.

1. Briefly explain the clustering coefficient of a graph and how we can compute it, by providing an algorithm (more efficient algorithms will be valorized).

R: The clustering coefficient of a graph G = (V, E) is given by three times the number of triangles in G divided by the number of connected triplets in G, and it evaluates how G is clustered. Alternatively, we can consider it as the average of the local clustering coefficient C_i over all vertices in G. The local clustering coefficient C_i of a vertex quantifies how close its neighbors are to be a clique (complete graph). For each node i with degree k_i, one can compute the local cluster coefficient C_i of i as the number of links e_i between vertices within the neighborhood of i divided by the total number links that could possible exist between them ($k_i(k_i-1)/2$). Hence, we are also counting triangles. To count triangles in a graph we can proceed link-by-link and node-by-node, where for each vertex we must iterate over its neighborhood, and then iterate again over the neighborhood of each neighbor to check if we have a triangle (we should use a hash table or an adjacency matrix to check link existence in constant time). Given a vertex, we can also check if each one of the $k_i(k_i-1)/2$ pairs of neighbors are linked (using again a hash table). In both cases, for each vertex, we have at most |V| neighbors and, hence, the overall running time is $O(|V|^3)$. For sparse graphs, this running time can be improved to $O(|E|^{3/2})$ by splitting vertices as heavy hitters and non-heavy hitters as discussed in theoretical classes.

2. Given a large network, a) how can we compute the distribution of the sizes of connected components in linear time on the graph size? b) Can you also do it in linear space on the number of vertices?

R: Connected components in a graph refer to a set of vertices that are connected. In other words, a set of vertices in a graph is a connected component if, by ignoring possible edge directions, each vertex in that set can be reached from every other node in that set. A straightforward way to compute the distribution of the sizes of connected components in linear time would be to use breadth-first search or depth-first search. In either case, a search that begins at some particular vertex v will find the entire connected component containing v before returning. To find all the connected components of a graph and their sizes, loop through its vertices, starting a new breadth first or depth first search whenever the loop reaches a vertex that has not already been included in a previously found connected component. Both breadth first or depth first search take linear time on the size of the graph, i.e., O(V+E), and computing the size distribution take linear time on the number of vertices. Since both breadth first or depth first search assume random access to vertices neighborhoods, these approaches assume that the graph is loaded in memory, requiring O(V+E) space. We can instead track the connected components of a graph (as vertices and edges are added) by using disjoint-set data structures. A disjoint-set data structure, also called a union–find data structure, is a data structure that keeps track of a set of elements partitioned into a number of non-overlapping (disjoint) subsets. It provides near-constant-time operations to add new sets, to merge existing sets, and to determine whether elements are in the same set. This model can then be used to determine whether two vertices belong to the same component, or whether adding an edge between them would result in a cycle. This structure is, for instance, used by Boost Graph Library to implement its Incremental Connected Components functionality. In this case, although we still take linear time on the size of the graph, we need only O(V) space.

3. Consider the modularity measure for evaluating the partitioning of graph in several non-overlapping clusters or communities. Assuming the classic underlying null model, i.e., the random network with the same degree distribution, explain what is evaluated by modularity and discuss its minimum and maximum value.

R: In a randomly wired network the connection pattern between the nodes is expected to be uniform, independent of the network's degree distribution. Consequently, these networks are not expected to display systematic local density fluctuations that we could interpret as communities. This is the basis of a modularity M used by several popular community finding algorithm: The modularity of a network with E edges and n communities is given by $M = \sum_{r=1}^{n}\left[\frac{E_r}{E} - \left(\frac{k_r}{2E}\right)^2\right]$*, where E_r is the total number of links within the community r and k_r is the total degree of the nodes in this community. By comparing the link density, E_r/E, within community with the link density obtained for the same group of nodes for a randomly rewired network, $(k_r/2E)^2$, we could assess if the original community corresponds to a dense subgraph, or its connectivity pattern emerged by chance. M sums this measure for the entire set of communities. Note that M cannot exceed one, and it is likely that the optimal partition does not reach this value. By taking the whole* *network as a single community we obtain M=0, as in this case the two terms in the parenthesis of are equal.* M £0 *is also the expected outcome for a random partition. The partition with the maximum modularity M offers the best community structure. However, it is also known that this measure has some limitations. In particular, networks often lack a clear modularity maximum, developing instead a modularity plateau containing many partitions with similar modularity. This plateau explains why numerous modularity maximization algorithms can rapidly identify a partition with high M: They identify one of the numerous partitions with close to optimal M.*

4. The Barabási-Albert (BA) model supports the idea that growth and preferential attachment are two fundamental principles underlying the emergence of power-law degree distributions in real systems. Contrary to networks generated through the BA model, most empirically observed networks also portray a significant saturation for low degrees and prominent exponential cutoffs for high degrees. Suggest one hypothesis for the origin of each of these features, and briefly describe how the BA model can be modified to test your hypothesis.

   R: In the BA model an isolated node cannot acquire links, as according to preferential attachment the likelihood that a new node attaches to a k=0 node is strictly zero. In real networks, even isolated nodes acquire links. To allow unconnected nodes to acquire links we may add a constant to the preferential attachment function. Such initial attractiveness adds a random component to preferential attachment. Consequently, the degree distribution develops a small-degree saturation, as observed in real-world networks. Furthermore, often nodes have a limited lifetime or a threshold above which cannot receive more ties. To cope with that, node and link removal, present in many real systems, can be added to the BA, inducing exponential high-degree cutoffs in the degree distribution. A similar effect is obtained if, in the BA model, one precludes nodes above a given degree to receive additional ties.

5. Network topology has a drastic impact on the dynamics of the spreading process, offering distinct predictions for the outbreak of diseases on random (Erdős-Rényi, ER) graphs and on scale-free networks. Resort to the SIS model and the concept of epidemic threshold to illustrate such differences.

   R: To predict when a pathogen persists in the population we may define the spreading rate $\lambda=\beta/\delta$ *which depends only on the biological characteristics of the pathogen, namely the transmission probability* β *and the recovery rate* δ*. A random ER network has a finite epidemic threshold* λ_c*, implying that a pathogen with a small spreading rate (*$\lambda<\lambda_c$*) must die out. If, however, the spreading rate of the pathogen exceeds* λ_c*, the pathogen becomes endemic and a finite fraction of the population is infected at any time. For a scale-free network we have* $\lambda_c=0$*. Hence even viruses with a very small spreading rate* λ *can persist in the population.*

6. Suggest a vaccination strategy for scale-free networks which *i)* is likely to perform better than random vaccination, and *ii)* does not require complete information on the network topology.

   R: The ineffectiveness of random immunization is rooted in the vanishing epidemic threshold of scale-free networks (see Question 5). Hub immunization would represent a better option for immunization protocols. The problem with a hub-based immunization strategy is that for most epidemic processes we lack a detailed map of the contact network. Indeed, for instance, we do not know the number of sexual partners each individual has in a population, nor can we accurately identify the super-spreaders during an influenza outbreak. As an alternative, it can be shown that a strategy based on immunization of acquaintances partially solves this problem. The idea is to vaccinate the acquaintances of a randomly selected individual, indirectly targeting the hubs without having to know precisely which individuals are hubs: 1) Choose randomly a p fraction of nodes (Group 0). Select randomly a link for each node in Group 0. Let us call Group 1 to this new set of nodes. For example, we ask each individual from Group 0 to nominate one of its acquaintance with whom he/she engaged in an activity that could have resulted in the transmission of the pathogen. For instance, in the case of HIV, ask them to name a sexual partner. Immunize the Group 1 individuals. Proceed like this creating a group 2 if needed.

7. Degree assortativity is a preference for network's nodes to attach to others that have a similar degree. Discuss the impact of high degree assortativity on the *diameter* and *average path length* of a random (ER) graph. If you do not know the answer, propose a model capable of finding one.

   R: High assortativity increases the number of low-degree nodes solely connected to other low-degree nodes, fostering chains of low-degree nodes. This will have a low impact on the Average Path Length but a significant impact on the diameter of the network. To show this, one may compute the APL and the diameter for a random network (ER model) with a high degree assortativiy created through the Xulvi-Brunet-Sokolov algorithm: 1) Start with a random network. 2) Randomly select two links connecting four different nodes. 3) Sort these nodes by their degrees. 4) Rewire the links in such a way that one link connects the two nodes with the smaller degrees and the other connects the two nodes with the larger degrees. If one or both of these links already existed in the network, the step is discarded and is repeated again. Repeat 2, 3 and 4 until a desired level of assortativity is achieved and measure the final APL and diameter. Average the APL and diameter over many simulations and initial ER networks. This algorithm allows keeping network's degree distribution unchanged when changing the value of the assortativity.

8. One of most important challenges in network science is to understand the impact networks have on various social dynamics, including opinion dynamics, individual preferences or strategic decisions. These processes can be modeled (at least, partially) through large-scale numerical simulations of population dynamics on graphs. Discuss how the representation of the network alters the running time of an evolutionary game theory simulation, where individuals' fitness result from interacting through 2-player games with all their neighbors, and individuals are also limited to be influenced by their closest peers.

R: At each time step, for each node X, one needs to access the strategies of each neighbor of X both for computing the fitness of X but also for choosing a partner to imitate (this part may depend on the update rule). If the network is represented by adjacency matrix, it is necessary to loop over each row of the matrix to find the states of each neighboring vertex, an operation which scales with the total number of vertices. Differently, an adjacency list allows an immediate access to the list of the neighbors of a given vertex, scaling with the degree of each node.

9. Imagine a couple that agreed to meet this evening, but cannot recall if they committed to go to a party or to stay at home. One prefers to stay at home, while the other would rather go to the party. Both would prefer to go to the same place rather than different ones. Where should they go? In this game both players prefer engaging in the same activity over going alone, but their preferences differ over which activity they should engage in. Resort to game theory to propose a *payoff matrix* that describes the conflict between the strategies *Home* and *Party*. Justify the proposed model, the variables used, and indicate the Nash equilibrium or equilibria depending on the parameters of your model.

R: This game corresponds to a simple version of the battle of the sexes without assigning a gender to each player. Please find below a possible payoff matrix from the point of view of the player that prefers to stay at home. The second payoff in each entry corresponds to one preferring to go for a party.

	Home	**Party**
Home	2,1	0,0
Party	0,0	1,2

Since the couple prefers to stay together, if they go separate ways, they will receive no utility (set of payoffs will be 0 for both). If they go either to the party or stay at home, both will receive some utility from the fact that they're together (at least 1), but one of them will actually enjoy the activity (get 1 point extra). This game has two pure Nash equilibria highlighted in blue, one where both go to the party and another where both stay at home. In both cases, no player has anything to gain by changing only their own strategy. One can also write a more general matrix in the form

	Home	**Party**
Home	*a,b*	*c,d*
Party	*d,c*	*b,a*

such that $a > b > c^{3}\ d$.

Remark 1: One may also argue that there is a mixed strategies Nash equilibrium, where the players go to their preferred event more often than the other. In this case, we're assuming the use of mixed strategies, in which we look at the probability of our opponent choosing one or the other strategy and balance our pay off against it.

Remark 2: The payoff matrices above exhibit on each side of the matrix the different players (say, players 1 (rows, preferring home) and player 2 (columns, preferring party)), each strategy or choice they can make (here strategies home and party) and sets of payoffs they will each receive for a given strategy (p1-Home, p2-Home; p1-Home, p2-Party, etc.). It allows us to quickly analyse each possible outcome of a game. In the first matrix, if player 1 chooses strategy Party and player 2 also chooses strategy Party, the set of payoffs given by the outcome would be 1 and 2, respectively.