



Show that the closeness centralities C_A and C_B of nodes A and B respectively, are related by

$$\frac{1}{C_A} + \frac{n_A}{n} = \frac{1}{C_B} + \frac{n_B}{n}.$$

Recall that, the closeness centrality is given by: $C_i = \frac{n}{\sum_j d_{ij}}$, where d_{ij} is the length of the geodesic path from i to j .

IV. Analyzing a Real Network

In the last part of the assignment, you will analyze the CA-GrQc collaboration network, examining several structural properties. The arXiv GR-QC (General Relativity and Quantum Cosmology) collaboration network has been extracted from the e-print arXiv (arxiv.org) and covers scientific collaborations between authors for papers submitted to General Relativity and Quantum Cosmology category. If an author i co-authored a paper with author j , the graph contains an undirected edge from i to j . Here we consider that the graph is unweighted.

The graph is stored in the `ca-GrQc.txt` file¹ as an edge list:

```
# Directed graph (each unordered pair of nodes is saved once): CA-GrQc.txt
# Collaboration network of Arxiv General Relativity category (there is an
# edge if authors coauthored at least one paper)
# Nodes: 5242 Edges: 28980
# FromNodeId ToNodeId
3466 937
3466 5233
...
```

For the following questions, feel free to use the graph library of your preference (we strongly encourage you to use Python's NetworkX, igraph or snappy), unless stated otherwise. For the figures, make sure to include axes titles and labels.

Question 7 [19 points]

- (a) [3 p] (*Basic properties of the network*). (1) Compute the following statistics: number of nodes, number of edges. (2) If the graph is not connected: (i) find the number of connected components (CCs); (ii) plot the distribution of their sizes (similar to MSN's connectivity shown in Lecture 1B, slide number 81). (3) Extract the largest (i.e., giant) connected component (GCC), and then (i) find the number of nodes and edges of GCC and (ii) the fraction of nodes and edges of the whole graph that belong to GCC (e.g., 5% of the number of nodes are part of GCC). Discuss briefly your observation (2-3 lines).

¹The data can be downloaded from the following link: <http://snap.stanford.edu/data/ca-GrQc.txt.gz>

- (b) [5 p] (*Analysis of the degree distribution*). (1) Find the minimum, maximum, median and mean degree of the nodes of the graph. What do you observe (please, discuss briefly in 1-2 lines)? (2) Visualize (appropriately) the degree distribution of the graph. What is the type of the degree distribution and what are the parameters? (You can use freely available software that was discussed in class).
- (c) [4 p] (*Triangles*). For this one and the following questions, we will focus on the GCC of the graph (in other words, consider that the GCC itself is the graph that we are interested to analyze). As we have discussed in the class, a triangle is a clique of three nodes, i.e., all nodes are connected to each other. Triangle subgraphs play a crucial role in the area of graph mining and social network analysis, since they are closely related to the existence of clustering structures in the graph. (1) Compute the total number of triangles in the GCC of the network. (2) Visualize (plot) the triangle participation distribution, i.e., a histogram that shows the number of triangles that each node participates in (i.e., how many nodes participate in one triangle, how many nodes participate in two triangles, etc). Note that, this process is similar to the computation of the degree distribution. Discuss briefly your observations.
- (d) [7 p] (*Spectral counting of triangles*). In Question 3, you have been asked to express the total number of triangles $\Delta(G)$ using information about the eigenvalues of the adjacency matrix of the graph. Here, you should compute the number of triangles in the GCC of the graph using the eigenvalues of the corresponding adjacency matrix (again, consider that your graph G is the GCC). In order to compute exactly $\Delta(G)$, we need to compute the whole spectrum of the adjacency matrix, which can be a computational bottleneck (why?). Here, we argue that we can approximate $\Delta(G)$ using the top- k ($k \ll |V|$) largest eigenvalues of the adjacency matrix, i.e., $\Delta(G) \approx \tilde{\Delta}_k(G)$, where $\tilde{\Delta}_k(G)$ is the spectral computation of the number of triangles, using only the top- k eigenvalues of the adjacency matrix. (1) Why is this happening? Explain your answer, giving specific arguments. [Hint: we can do low rank approximation of the adjacency matrix for the computation of triangles, taking advantage of some of the “power-law” properties of real networks that we have seen in class]. (2) Suppose that we approximate $\Delta(G)$ with $\tilde{\Delta}_k(G)$. Compute and visualize the error of approximation for various values of k (i.e., how the error behaves as we increase the number of eigenvalues used in the computation of $\tilde{\Delta}_k(G)$):

$$\text{error}_k = \frac{|\tilde{\Delta}_k(G) - \Delta(G)|}{\Delta(G)},$$

How many eigenvalues should we retain to achieve good approximation? In the extreme case where all the $|V|$ eigenvalues are used, $\Delta(G) = \tilde{\Delta}_{k=|V|}(G)$.

[**Note:** Depending on the memory of your computer, the computation of the full spectrum of \mathbf{A} may take significant amount of time and resources. Thus, in your experiments, restrict your attention only on the largest 1000 eigenvalues of \mathbf{A} , avoiding computing all the eigenvalues.]

Question 8 [6 points]

Generate an Erdős-Rényi random graph $G_{n,p}$ with $n = 1000$ nodes and $p = 0.009$. For this task, you can use the built-in function of NetworkX `nx.fast_gnp_random_graph(n, p)`.

- (a) [2 p] What is the mean degree of the graph? Justify theoretically your answer (max 1 line).
- (b) [2 p] Is the graph connected? Justify theoretically your answer (max 1 line).

- (c) [2 p] Compute the mean degree from the graph that you have generated and visualize the degree distribution of the graph. Discuss briefly your observations.

Question 9 [8 points]

In this question, you will examine the properties of the graphs produced by the *Kronecker model*, which was introduced as a simple generation model for real-world graphs based on the Kronecker product of matrices. More precisely, assuming an initiator adjacency matrix \mathbf{A}_1 of size $\ell \times \ell$, the Kronecker graph after k iterations is defined as the graph with the following adjacency matrix:

$$\mathbf{A}_k = \underbrace{\mathbf{A}_1 \otimes \mathbf{A}_1 \otimes \cdots \otimes \mathbf{A}_1}_{k \text{ iterations}} = \mathbf{A}_{k-1} \otimes \mathbf{A}_1.$$

In practice, a stochastic version of the Kronecker model is used, in the sense that the initiator matrix \mathbf{A}_1 is not the binary adjacency matrix itself but the probability matrix for the existence of an edge. For example, in the typical case of a 2×2 initiator matrix $\mathbf{A}_1 = [a \ b; c \ d]$, each value represents the probability of existence of the corresponding edge. Starting by such an initiator matrix and applying the Kronecker product for a desired number of iterations k , the resulting adjacency matrix of the graph corresponds to a realization of the matrix \mathbf{A}_k , i.e., each edge (i, j) is introduced to the graph with probability $A_k(i, j)$.

Consider the following 2×2 initiator matrix $\mathbf{A}_1 = [0.99, 0.26; 0.26, 0.53]$ (notice that the matrix is in the form $[a \ b; b \ c]$). The values of the initiator matrix are important, in the sense that they are related to the properties of the generated graph. In particular, the above matrix has been produced by the KRONFIT algorithm presented in class, fitting its parameters to the structure of the CA-GrQC network. Repeat the Kronecker product for k times and compute the \mathbf{A}_k adjacency matrix (you can use the `kron(A, B)` built-in function of *numpy* for Kronecker multiplication). Lastly, for each entry $\mathbf{A}_k(i, j)$ of matrix \mathbf{A}_k , include the edge (i, j) with probability $\mathbf{A}_k(i, j)$. This matrix will be the adjacency matrix of the final graph (consider that the produced Kronecker graph is *undirected*).

- (a) [2 p] (i) Is the produced Kronecker graph connected? (ii) Does the graph have a giant connected component of size $\Theta(n)$? Provide *theoretical justification* of your answers. You can also confirm your answers by examining the properties of the graph (e.g., computing the actual size of the GCC).
- (b) [6 p] Describe **three** structural properties which show that the produced graph looks similar (qualitatively or even quantitatively) to the CA-GrQC network. For each of those properties you should examine how they look like in both the original CA-GrQC network and produced Kronecker graph (e.g., compute/visualize property A of CA-GrQC and repeat the same for the Kronecker graph). Explain briefly your observations.

Question 10 [12 points]

The robustness of a network is related to the capability of the network to retain its structure and connectivity properties, after losing a portion of its nodes and edges. Let us focus our attention to the case where the nodes of a network are deleted according to their degree based on two strategies:

- (i) *Random deletion*: delete a randomly selected node.
- (ii) *Targeted deletion*: delete a node chosen among the ones with the highest degree in the network.

Depending on the type of networks, the above two strategies can simulate various scenarios. For example, in the case of the Internet graph (nodes correspond to routers and edges capture physical connections between routers), a random deletion can be interpreted as an *error* that occurred in the network, where a router switched off due to technical problems (e.g., electricity problems). On the other hand, the targeted removal of a high-degree node can simulate the case of an attack to the network, where the removal of nodes aims to cause big damage to the network.

How the structure of the network is affected after random and targeted deletions of nodes? In the class, we discussed that due to the existence of the heavy-tailed degree distribution, real-world networks tend to be robust under random removal of nodes (e.g., errors) and vulnerable under the attacks to high degree nodes. The notion of robustness can be quantified by structural characteristics of the network, such as the fragmentation of the network into disconnected components. For instance, we can examine how the fraction of nodes that belong to the largest connected component (GCC) and the rest isolated components is affected by the random/targeted removal of nodes. Here, your goal will be to examine the robustness of the CA-GrQC network, with respect to the above strategies.

For this question, work with the GCC of the graph (i.e., your initial graph is the GCC of the CA-GrQC network). To assess the robustness, you should examine how the size of GCC and the rest components is affected, after removing a fraction of nodes in the range of $[0\%, 20\%]$.

- Plot the size of GCC and the size of the rest components (sum of their sizes) vs. the fraction of deleted nodes, for both strategies (i.e., random and targeted) in the same figure (i.e., four different curves). Discuss briefly your observations.

Note: In order to have a “reference figure”, the goal is to reproduce Fig. 3 (c) or (d) of the paper *Error and attack tolerance of complex networks*, by R. Albert, H Jeong, and A.-L. Barabasi (Nature 406, 378-382, 2000)². Notice that in the beginning, the GCC contains the whole graph (thus the rest components are empty), but both curves start from the same point (1) in the vertical axis. Moreover, the produced figure may not be exactly the same as the one of the above paper, since we analyze a different network – but, overall, there should be some resemblance.

²<https://www.nature.com/articles/35019019>