Homework 1

Dinis Abranches

September 13, 2022

Problem 1:

Let $\mathbf{A}$ be the adjacency matrix of graph G containing N nodes, with A_{ij} being the value of its i^{th} row and j^{th} column. The Katz centrality of the i^{th} node of G (c_i) is defined as:

$$c_i = \alpha \sum_{j=1}^{N} (A_{ij} c_j) + \beta \tag{1.1}$$

where α and β are positive parameters.

Rewriting Equation 1.1 in matrix notation yields:

$$\mathbf{c} = \beta(\mathbf{I}_N - \alpha\mathbf{A})^{-1}\vec{\mathbf{1}}_N \tag{1.2}$$

where $\mathbf{c}$ is the Katz centrality vector of G, $\mathbf{I}_N$ is the identity matrix of size NxN, and $\vec{\mathbf{1}}_N$ is a 1-vector of size N.

When selecting a value for parameter α, two conditions should be observed. The first is that α should not be significantly smaller than β ($\alpha << \beta$ should be avoided). Otherwise, the β term in Equation 1.1 becomes dominant and the centrality of all nodes becomes approximately constant (and equal to β). This, however, does not lead to a diverging $\mathbf{c}$; it merely leads to a non-useful metric.

The second consideration is that α must not be so large as to make the inverse of $(\mathbf{I}_N - \alpha\mathbf{A})$ diverge. Because $(\mathbf{I}_N - \alpha\mathbf{A})$ is a square matrix, it becomes a singular matrix (and its inverse diverges thereafter) when its determinant becomes zero. To avoid this, the following condition is imposed:

$$\det(\mathbf{I}_N - \alpha\mathbf{A}) \neq 0 \tag{1.3}$$

Multiplying both sides of Equation 1.3 by $1/\alpha$ (and noting the scalar multiplication property of determinants) leads to:

$$\det\left(\frac{1}{\alpha}\mathbf{I}_N - \mathbf{A}\right) \neq 0 \tag{1.4}$$

The equality that would otherwise be represented by Equation 1.4 is the characteristic polynomial of $\mathbf{A}$ with indeterminate $1/\alpha$ and whose roots are its eigenvalues. Thus, the smallest α value that satisfies Equation 1.3 must be smaller than the reciprocal of the largest eigenvalue of $\mathbf{A}$. In other words, **the values of α that guarantee the convergence of the Katz centrality** are those that satisfy:

$$\alpha < \frac{1}{\lambda_{max}} \tag{1.5}$$

where λ_{max} is the largest eigenvalue of $\mathbf{A}$.

Problem 2:

Let $\mathbf{A}$ be the adjacency matrix of graph $G = (V, E)$ containing N nodes, with A_{ij} being the value of its i^{th} row and j^{th} column. For any given node v_i, $N(v_i)$ represents the list of all its neighbors. Within the concept of walks or paths (they differ by the fact that the former can revisit nodes, but can be used interchangeably within the scope of this problem), a node v_j belongs to $N(v_i)$ if there exists a path of size 1 between v_j and v_i. As such, the total number of neighbors of v_i ($|N(v_i)|$) is simply:

$$|N(v_i)| = \sum_{j=1}^{N} A_{ij} \tag{2.1}$$

By definition, node v_k is a common neighbor of nodes v_i and v_j if it is simultaneously a neighbor of both v_i and v_j. Thus, the set of common neighbors of v_i and v_j corresponds to the intersection between the sets $N(v_i)$ and $N(v_j)$ (i.e., $N(v_i) \cap N(v_j)$). From this, and noting the path-based definition of neighbors provided above, it can be inferred that, for every common neighbor, v_k, of v_i and v_j, there exists a path of size two that connects them passing through v_k. The total number of such paths ($N_{ij}^{(2)}$) is given by:

$$N_{ij}^{(2)} = \sum_{k=1}^{N} A_{ik}A_{kj} = [\mathbf{A}^2]_{ij} \tag{2.2}$$

where $[\mathbf{A}^2]_{ij}$ is the value of the i^{th} row and j^{th} column of $\mathbf{A}^2$.

As mentioned above, the number of common neighbors of $N(v_i)$ and $N(v_j)$ ($|N(v_i) \cap N(v_j)|$) is also the number of paths of size two between v_i and v_j. Thus, and considering Equation 2.2, $|N(v_i) \cap N(v_j)|$ is simply:

$$|N(v_i) \cap N(v_j)| = N_{ij}^{(2)} = [\mathbf{A}^2]_{ij} \tag{2.3}$$

Problem 3:

To solve part A of the problem, a function named computeJaccardMatrix() was developed, which computes the Jaccard's Similarity matrix of a graph. This function has two available algorithms. The first uses a vectorized approach, where matrices of common and total neighbors are calculated from the adjacency matrix of G using Equations 2.1 and 2.2, followed by element-wise division to obtain the Jaccard's matrix. The second algorithm uses a slower, double loop approach, iterating over all nodes in the graph and calculating their Jaccard's Similarity in a step-wise fashion. To this end, a second function, computeJaccardSimilarity(), was developed that computes the Jaccard's Similarity between any two nodes. This second function compares its results against the native implementation of NetworkX. Thus, the main purpose of the second algorithm of computeJaccardMatrix(), other than its educational value, is simply to compare the results of the first algorithm against those provided by NetworkX.

To solve part B, the original code was altered to provide two plots, one containing edges representing the Jaccard's similarity of the Ginori family with all other families and the other representing only non-zero similarities. These are depicted below.

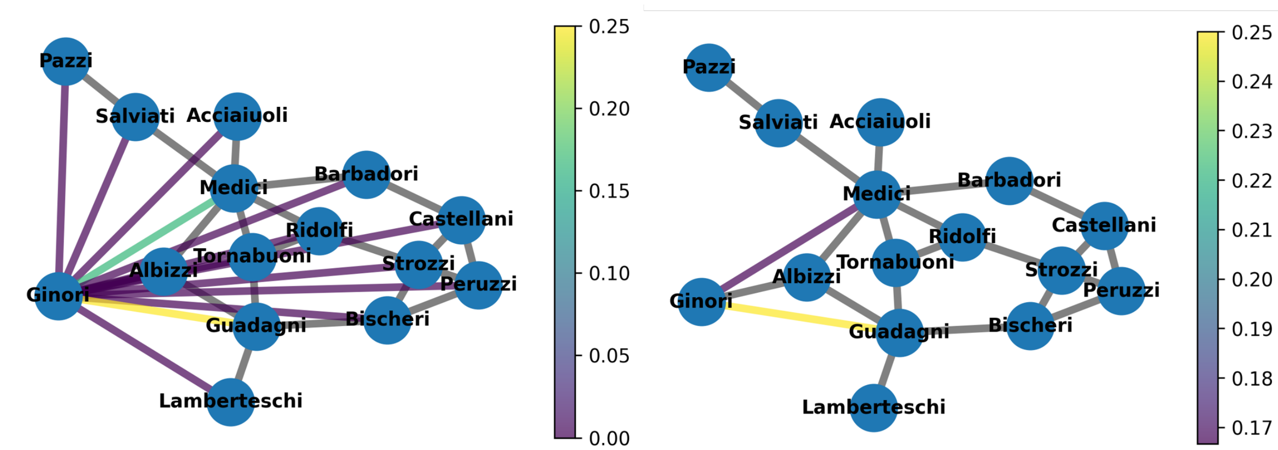

Figure 1: Florentine Families Graph with colored artificial edges representing the Jaccard's Similarity between the Ginori family and all other families (left) or representing only Ginori-based, non-zero Jaccard's similarities (right).