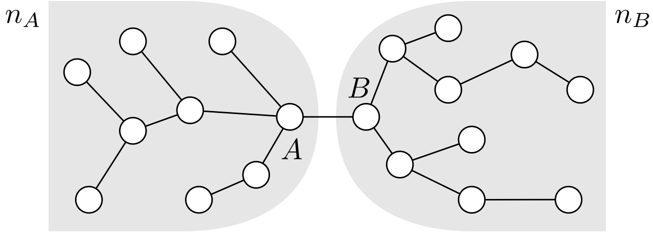
CSCI 5352 Network Analysis and Modeling Fall 2020
Prof. Daniel Larremore
Problem Set 4
Student: Alex Book; Collaborators: None

1. (15 pts) Consider an undirected, unweighted network of n vertices that contains exactly two subnetworks of size n_A and n_B, which are connected by a single edge (A, B), as sketched here:



Show that the closeness centralities C_A and C_B of vertices A and B, as defined by Eq. (7.21) in *Networks* [v1: 7.29], are related by

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

$$C_A = \frac{n}{\sum_j d_{Aj}},\ C_B = \frac{n}{\sum_j d_{Bj}} \text{ (from Eq. (7.21))}$$

Let N_A and N_B represent the halves of the network containing A and B, respectively.

$$C_A = \frac{n}{\sum_{j \in N_A}(d_{Bj}-1)+\sum_{j \in N_B}(1+d_{Bj})}$$

$$C_A = \frac{n}{\sum_{j \in N_A} d_{Bj}-\sum_{j \in N_A} 1+\sum_{j \in N_B} 1+\sum_{j \in N_B} d_{Bj}}$$

$$C_A = \frac{n}{\sum_j d_{Bj}-n_A+n_B}$$

$$\frac{1}{C_A} = \frac{\sum_j d_{Bj}-n_A+n_B}{n} = \frac{\sum_j d_{Bj}}{n} - \frac{n_A}{n} + \frac{n_B}{n}$$

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

2. (15 pts) Consider an undirected (connected) tree of n vertices. Suppose that a particular vertex in the tree has degree k, so that its removal would divide the tree into k disjoint regions, and suppose that the sizes of those regions are $n_1, \ldots, n_k$. Show that the betweenness centrality b of the vertex is

$$b = n^2 - \sum_{i=1}^{k} n_i^2 \ .$$

The betweenness centrality of vertex v can be interpreted as follows: When v is removed, splitting the tree into k disjoint regions, sum the number of shortest paths between two nodes in different regions (as only these paths needed to pass through v). This quantity is equal to the total number of shortest paths between two nodes in any region minus the total number of shortest paths between two nodes in the same region. The simplification is as follows.

Let $N_1, \ldots, N_k$ be the regions of size $n_1, \ldots, n_k$.

$$\begin{aligned} b = x_i &= \sum_{st} \frac{n_{st}^i}{g_{st}} \text{(from Eq. (7.24))} \\ &= \sum_{\substack{s \in N_i \\ t \in N_j \\ i \neq j}} 1 \\ &= \sum_{st} 1 - \sum_{\substack{s \in N_i \\ t \in N_j \\ i = j}} 1 \\ &= n^2 - \sum_{i=1}^{k} n_i^2 \end{aligned}$$

3. (30 pts) The Medici family was a powerful political dynasty and banking family in 15th century Florence. The classic network-based explanation of their power, offered by Padgett and Ansell in 1993, claims that they established themselves as the most central players within the network of prominent Florentine families (shown in Fig. 1 below).

   Visit the *Index of Complex Networks* (ICON) at `icon.colorado.edu` and obtain a copy of the `Medici network` data file, under the "Padgett Florentine families" ICON entry.

   Conduct the following two tests of their structural importance hypothesis:

   (a) Construct a table with four columns, each containing a list of (family, score) pairs for the following centrality score functions:

   - degree centrality,
   - harmonic centrality (Eq. 7.22 in *Networks* [v1: 7.30]),
   - eigenvector centrality (Eq. 7.2 in *Networks* [v1: 7.6]),
   - betweenness centrality[1] (Eq. 7.25 in *Networks* [v1: 7.38]).

   Within each centrality, sort the pairs in decreasing order of importance; three decimal places is sufficient detail.

   Discuss (i) the degree to which these scores agree with Padgett and Ansell's claim that the Medicis occupied a structurally important position in this network, and (ii) what the scores say about the second most important family.

Degree Centrality	**Harmonic Centrality**	**Eigenvector Centrality**	**Betweenness Centrality**
(Medici, 0.4)	(Medici, 0.633)	(Medici, 0.43)	(Medici, 0.452)
(Guadagni, 0.267)	(Guadagni, 0.539)	(Strozzi, 0.356)	(Guadagni, 0.221)
(Strozzi, 0.267)	(Ridolfi, 0.533)	(Ridolfi, 0.342)	(Albizzi, 0.184)
(Albizzi, 0.2)	(Albizzi, 0.522)	(Tornabuoni, 0.326)	(Salviati, 0.124)
(Castellani, 0.2)	(Strozzi, 0.522)	(Guadagni, 0.289)	(Ridolfi, 0.098)
(Bischeri, 0.2)	(Tornabuoni, 0.522)	(Bischeri, 0.283)	(Bischeri, 0.09)
(Peruzzi, 0.2)	(Bischeri, 0.48)	(Peruzzi, 0.276)	(Strozzi, 0.089)
(Tornabuoni, 0.2)	(Barbadori, 0.472)	(Castellani, 0.259)	(Barbadori, 0.081)
(Ridolfi, 0.2)	(Castellani, 0.461)	(Albizzi, 0.244)	(Tornabuoni, 0.079)
(Barbadori, 0.133)	(Peruzzi, 0.452)	(Barbadori, 0.212)	(Castellani, 0.048)
(Salviati, 0.133)	(Salviati, 0.439)	(Salviati, 0.146)	(Peruzzi, 0.019)
(Acciaiuoli, 0.067)	(Acciaiuoli, 0.394)	(Acciaiuoli, 0.132)	(Acciaiuoli, 0)
(Ginori, 0.06)	(Lamberteschi, 0.358)	(Lamberteschi, 0.089)	(Ginori, 0)
(Lamberteschi, 0.06)	(Ginori, 0.356)	(Ginori, 0.075)	(Lamberteschi, 0)
(Pazzi, 0.067)	(Pazzi, 0.318)	(Pazzi, 0.045)	(Pazzi, 0)
(Pucci, 0)	(Pucci, 0)	(Pucci, 0)	(Pucci, 0)

[1] To properly compute betweenness, you must handle multiple geodesics with equal length between a pair i and j. See Lectures 2 nodes, footnote 16 for one way to do this.

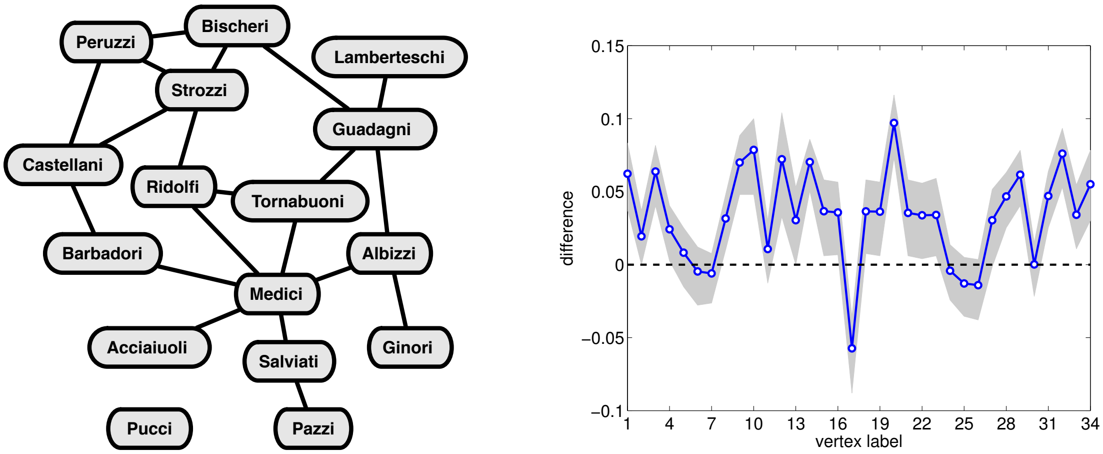


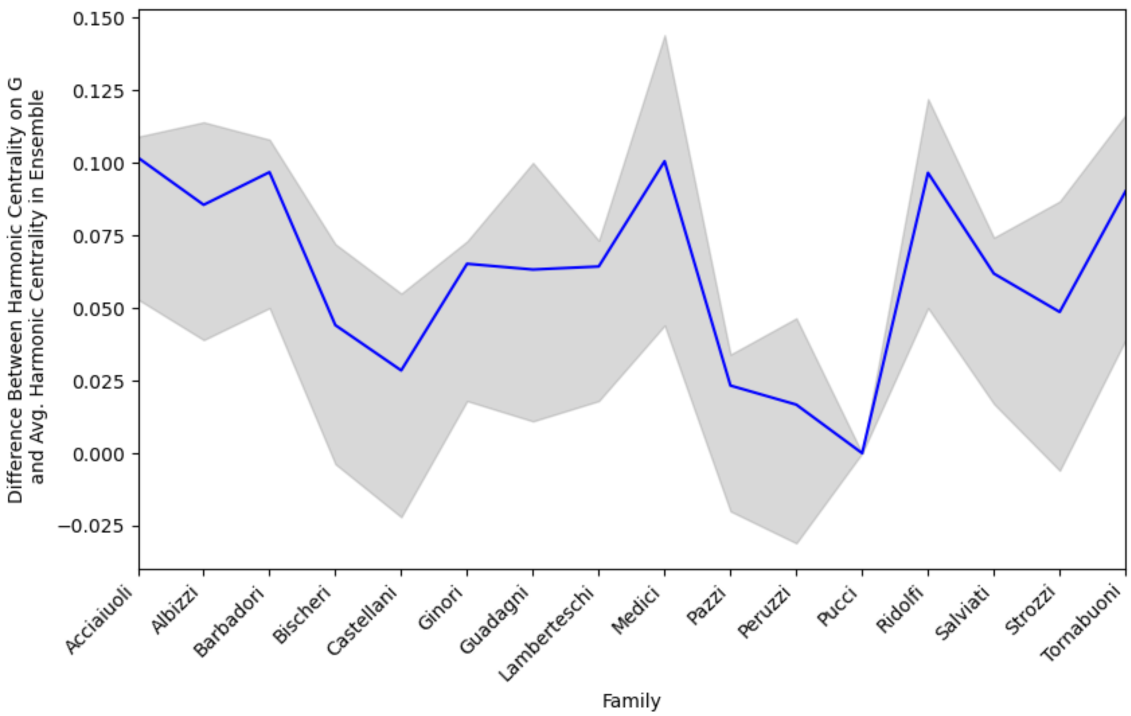
Figure 1: (left) The Medici family alliance network, from Padgett and Ansell (1993). (right) The difference between the observed centrality and the mean centrality under the configuration model, for the karate club network.

i. These scores completely agree with Padgett and Ansell's claim that the Medici family occupied a structurally important position in this network, as they place at the top of every measured centrality metric by a fairly sizeable margin.

ii. These scores show that, while the Guadagni family may appear to be 2nd in 3 of the 4 metrics, the competition is fierce. It is plain that not only was the Medici family the important family in the network, but they dominated other families in terms of these centrality metrics.

(a) Determine whether the relative structural importance of the Medicis can be explained solely by the degree sequence $\{k_i\}$ of the network G. To do this, use the configuration model to produce an ensemble of 1000 random graphs with the same degree sequence $\{k_i\}$ as G, and extract a distribution of harmonic centrality scores for each node.

Make a figure showing the difference between a node's harmonic centrality on G and its average harmonic centrality in the ensemble. On this figure, include lines showing the 25% and 75% quantiles of the distribution around the mean for all vertices (as in the figure below, for the Karate club).



Finally, (i) discuss your results here in terms of Padgett and Ansell's story of the Medici family, and (ii) comment on whether your results from part (3a) have changed.

i. Keeping the same degree centrality allowed differences in edge endpoints, but not the number of edges itself. In essence, creating an ensemble of 1000 random graphs was used to see if *who* a family is connected to matters for importance, rather than simply *how many* families a family is connected to. The results seen in the plot above show that no family/node has egregious differences between the random graphs and the harmonic centrality calculated on the original graph G.

ii. With that being said, my perspective on the tabled results from part (3a) has changed, as it seems now that the dominance of the Medici family isn't necessarily dependent upon the centralities measured (some of which are dependent upon *what* connections each node has), but moreso on the degree sequence of the network (which controls only *how many* connections each node has). Adding or subtracting the differences shown in (3b) from the harmonic centralities shown in (3a) doesn't change the hierarchy in a remarkable way, so the degree sequence alone is enough to explain the relative structural importance of the Medici family.

4. (25 pts *extra credit*) Consider the task of designing a network in which distinct vertices hold the status of highest centrality for some set $\mathcal{S}$ of centrality measures. (Ties are prohibited.) If the network is relatively small compared to the number of centrality measures, this can be a very difficult task.

   - As a warm up, design by hand a small network ($n \leq 7$) that has this property with respect to $\mathcal{S} = \{$ degree, betweenness $\}$ centralities. Include a visualization of the network (label all the vertices, and indicate which ones win at which centrality measure) and produce a table in which the columns show the ranked vertices and scores for these centrality measures.
   - Now, write a program to search the space of all possible connected, non-isomorphic networks of a given size n to find a network with this property for $\mathcal{S} = \{$ degree, harmonic, betweenness, eigenvector $\}$ centralities, or report that no such network exists for that size. Run this program for $n = 3, 4, 5, \ldots$ and report the smallest value of n for which at least one such network exists. Visualize that network, label all the vertices, and indicate which ones win at which centrality measure; and produce a table in which the columns show the ranked vertices and scores for these centrality measures. Comment briefly about what insights you gained about how to build networks that have distinct winners for different measures of centrality.

Code used in question 3:

```
import networkx as nx
import matplotlib
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np

if __name__ == '__main__':
    with open('medici_edge_list.txt') as f:
        G = nx.read_edgelist(f, delimiter=',')
    G.remove_edge('11','11')

    relabelDict = {
        '0' : 'Acciaiuoli',
        '1' : 'Albizzi',
        '2' : 'Barbadori',
        '3' : 'Bischeri',
        '4' : 'Castellani',
        '5' : 'Ginori',
        '6' : 'Guadagni',
        '7' : 'Lamberteschi',
        '8' : 'Medici',
        '9' : 'Pazzi',
        '10' : 'Peruzzi',
        '11' : 'Pucci',
        '12' : 'Ridolfi',
        '13' : 'Salviati',
        '14' : 'Strozzi',
        '15' : 'Tornabuoni'
    }
    G = nx.relabel_nodes(G, relabelDict)
    # nx.draw_networkx(G, with_labels = True)
    # plt.show()

    # degree centrality
    degreeDict = {}
    for node in G.nodes:
        degreeDict[node] = round(G.degree(node)/(len(G.nodes)-1),3)

    # harmonic centrality
    harmonicDict = {}
    for node1 in G.nodes:
        sum = 0
        for node2 in G.nodes:
```

```
        if (node1 != node2) and nx.has_path(G, node1, node2):
            sum += 1/nx.algorithms.shortest_paths.shortest_path_length(G, source=node1,
                                                                       target=node2)
    harmonicDict[node1] = round(sum/(len(G.nodes)-1),3)

# eigenvector centrality
eigenvectorDict = nx.eigenvector_centrality(G)
for key in eigenvectorDict.keys():
    eigenvectorDict[key] = round(eigenvectorDict[key], 3)

# betweenness centrality
betweennessDict = nx.betweenness_centrality(G, k=len(G.nodes))
for key in betweennessDict.keys():
    betweennessDict[key] = round(betweennessDict[key], 3)

dfD = pd.DataFrame.from_dict(degreeDict, orient=’index’,
            columns = [’Degree Centrality’]).sort_values(’Degree Centrality’,
                                                         ascending=False)
dfH = pd.DataFrame.from_dict(harmonicDict, orient=’index’,
            columns = [’Harmonic Centrality’]).sort_values(’Harmonic Centrality’,
                                                           ascending=False)
dfE = pd.DataFrame.from_dict(eigenvectorDict, orient=’index’,
            columns = [’Eigenvector Centrality’]).sort_values(’Eigenvector Centrality’,
                                                              ascending=False)
dfB = pd.DataFrame.from_dict(betweennessDict, orient=’index’,
            columns = [’Betweenness Centrality’]).sort_values(’Betweenness Centrality’,
                                                              ascending=False)
df = pd.concat([dfD, dfH, dfE, dfB], axis=1).sort_values(’Degree Centrality’,
                                                         ascending=False)

# config model stuff
thousandDict = {
    ’Acciaiuoli’ : [],
    ’Albizzi’ : [],
    ’Barbadori’ : [],
    ’Bischeri’ : [],
    ’Castellani’ : [],
    ’Ginori’ : [],
    ’Guadagni’ : [],
    ’Lamberteschi’ : [],
    ’Medici’ : [],
    ’Pazzi’ : [],
    ’Peruzzi’ : [],
    ’Pucci’ : [],
    ’Ridolfi’ : [],
    ’Salviati’ : [],
    ’Strozzi’ : [],
    ’Tornabuoni’ : []
}
for i in range(1000):
    G = nx.Graph(nx.configuration_model([1,3,2,3,3,1,4,1,6,1,3,0,3,2,4,3]))
    relabelDict = {
        0 : ’Acciaiuoli’,
        1 : ’Albizzi’,
        2 : ’Barbadori’,
        3 : ’Bischeri’,
        4 : ’Castellani’,
        5 : ’Ginori’,
        6 : ’Guadagni’,
        7 : ’Lamberteschi’,
        8 : ’Medici’,
        9 : ’Pazzi’,
        10 : ’Peruzzi’,
        11 : ’Pucci’,
        12 : ’Ridolfi’,
        13 : ’Salviati’,
        14 : ’Strozzi’,
        15 : ’Tornabuoni’
    }
    G = nx.relabel_nodes(G, relabelDict)
    for node1 in G.nodes:
        sum = 0
        for node2 in G.nodes:
            if (node1 != node2) and nx.has_path(G, node1, node2):
                sum += 1/nx.algorithms.shortest_paths.shortest_path_length(G,
                                                                           source=node1,
                                                                           target=node2)
        thousandDict[node1].append(round(sum/(len(G.nodes)-1),3))
percentile25dict = {}
percentile75dict = {}
diffDict = {}
for key, value in thousandDict.items():
    avgEnsembleCentrality = np.average(value)
    diff = harmonicDict[key] - avgEnsembleCentrality
    diffDict[key] = diff

    diffList = []
    for i in range(len(value)):
        diffList.append(harmonicDict[key]-value[i])
    percentile25dict[key] = np.percentile(diffList, 25)
    percentile75dict[key] = np.percentile(diffList, 75)

harmonicDict = dict(sorted(harmonicDict.items()))
fig, ax = plt.subplots()
ax.plot(list(diffDict.keys()), list(diffDict.values()), color=’blue’)
ax.plot(list(percentile25dict.keys()), list(percentile25dict.values()), ’r-’, alpha=0)
ax.plot(list(percentile75dict.keys()), list(percentile75dict.values()), ’r-’, alpha=0)
ax.fill_between(range(len(list(diffDict.values()))), list(percentile25dict.values()),
                list(percentile75dict.values()), alpha=.3, color=’gray’)
fig.autofmt_xdate(rotation=45)
ax.set_xlabel(’Family’)
ax.set_ylabel(’Difference Between Harmonic Centrality on G\n and Avg. Harmonic Centrality
                                                                             in Ensemble’)
ax.set_xlim(0, len(list(diffDict.values()))-1)
plt.show()
```