Cassandra Spath

Problem Set 1

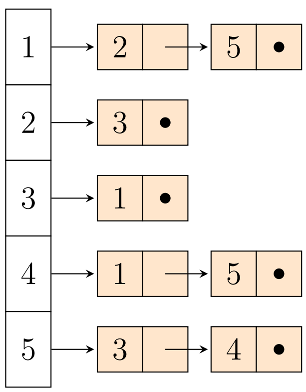
CSCI 5352
September 11, 2019

1. (12 pts) Consider the following two networks:

   (a) (3 pts) Give the adjacency matrix for network (A).

$$
\begin{array}{c c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}
$$

   (b) (3 pts) Give adjacency list for network (A).



graphic source

   (c) (6 pts) Give adjacency matrices for both one-mode projections of network (B).

$$
\begin{array}{c c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{bmatrix}
0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0
\end{bmatrix}
\end{array}
\qquad
\begin{array}{c c}
 & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} &
\begin{bmatrix}
0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

2. (15 pts) Let A be the adjacency matrix of a simple graph (unweighted, undirected edges with no self-loops) and 1 be the column vector whose elements are all 1. In terms of these quantities, multiplicative constants and simple matrix operations like transpose and trace, write expressions for:

   (a) (3 pts) the vector k whose elements are the degrees k_i of the vertices

   $$k = A * 1$$

   (b) (3 pts) the number m of edges in the network

   $$m = \frac{(A1)^T 1}{2}$$

   Since there are no self-loops, each edge is connected to two different nodes, thus the sum of all the degrees of the nodes will be even. By dividing the sum of all the degrees by 2, the number of edges can be obtained.

   (c) (5 pts) the matrix N whose elements N_{ij} is equal to the number of common neighbors of vertices i and j

   $$N = A * A^T$$

   The diagonals or N_{ii} will equal the number of nodes that node i is connected to.

   (d) (4 pts) the total number of triangles in the network, where a triangle means three vertices, each connected by edges to both of the others.

   $$t = \frac{\text{trace}(A * A * A)}{6}$$

   Multiple the adjacency matrix by itself twice. The diagonals M_{ii} are the number of triangles that node i has. Thus, we need to do the trace of the matrix. However, this counts both directions since the graph is undirected meaning it counts $A-B-C-A$ and $A-B-C-A$. Therefore, the trace must be divide by 2. Additionally, the triangle is counted for all three nodes that make it up, so the result must be divided by another 3 to get the correct total.

3. (10 pts) Consider a bipartite network, with its two types of vertices, and suppose there are n_1 vertices of type 1 and n_2 vertices of type 2. Show that the mean degrees c_1 and c_2 of the two types are given by

$$c_2 = \frac{n_1}{n_2} c_1$$

Proof. Assume that the network has m edges.
Since each edge must connect type 1 to type 2, the sum of the degrees of all the nodes for one type is equal to m.
Thus, the mean degree for vertices of type 1 is equal to the sum of the degrees divided by the number of nodes:

$$c_1 = \frac{m}{n_1} \implies m = c_1 * n_1$$

Additionally, the mean degree for vertices of type 2 is equal to the sum of the degrees divided by the number of nodes:

$$c_2 = \frac{m}{n_2} \implies m = c_2 * n_2$$

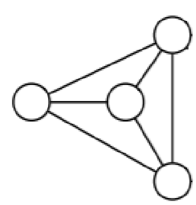
Therefore, we can combine the expressions for m and rearrange:

$$c_2 * n_2 = c_1 * n_1 \implies c_2 = \frac{n_1}{n_2} c_1$$

□

4. (13 pts) Consider the following three networks:

   (a) (4 pts) Find a 3-core in network (A).



   (b) (5 pts) What is the reciprocity of network (B)?

$$r = \frac{1}{m} \sum_{ij} A_{ij} * A_{ji} = \frac{1}{8} * (1 + 1 + 1 + 0 + 0 + 0) = \frac{3}{8}$$

(c) (4 pts) What is the cosine similarity of vertices A and B in network (C)?

$$A = [1\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0]$$
$$B = [0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1]$$
$$A^T B = 1*0+0*1+1*1+1*1+1*0+0*1+0*1 = 2$$
$$||A|| = \sqrt{1*1+0*0+1*1+1*1+1*1+0*0+0*0} = 2$$
$$||B|| = \sqrt{0*0+1*1+1*1+1*1+0*0+1*1+1*1} = \sqrt{5}$$
$$sim(A,B) = \frac{A^T B}{||A||*||B||} = \frac{2}{2*\sqrt{5}} = \frac{1}{\sqrt{5}}$$

5. (15 pts) A Cayley tree is a symmetric regular tree in which each vertex is connected to the same number k of others, until we get out to the leaves, like the figure below, with $k = 3$.

- Show that the number of vertices reachable in d steps from the central vertex is $k*(k-1)^{d-1}$ for $d \geq 1$.

Base Case: d = 1

$k*(k-1)^{d-1} = k*(k-1)^0 = k$

For the center node, it is connected to k other nodes as defined by the definition of the Cayley tree. Thus, the formula works for the base case.

Recursive Step: Assume that for $d = i$ the formula $k*(k-1)^{i-1}$ holds. Prove that the formula $k*(k-1)^i$ holds for $d = i+1$.

At the ith depth, $k*(k-1)^{i-1}$ vertices can be reached from the center. For each of these vertices, $k-1$ vertices can be reached which are 1 step further from the center or $i+1$ steps from the center. Thus, there are $k*(k-1)^{i-1}*(k-1) = k*(k-1)^i$ vertices $i+1$ steps from the center.

- Give an expression for the diameter of the network in terms of k and the number of vertices n.

Since the diameter will be twice the length of the center to an edge, using the previous formula, the total number of vertices are:

$$
\begin{aligned}
n &= 1 + \sum_i d/2k * (k-1)^{i-1} \\
n - 1 &= \frac{k * ((k-1)^{\frac{d}{2}} - 1)}{k-2} \\
\frac{(n-1)*(k-2)}{k} + 1 &= (k-1)^{\frac{d}{2}} \\
\log\left(\frac{(n-1)*(k-2)}{k} + 1\right) &= \frac{d}{2}\log(k-1) \\
\frac{2\log\left(\frac{(n-1)*(k-2)}{k} + 1\right)}{\log(k-1)} &= d
\end{aligned}
$$

This formula can be approximated by ignoring the constants which become unnecessary when n gets larger. Additionally for smaller values of k canceling out the terms in the numerator log as they will be a small constant. For a larger value of k, $k-2 \approx k$. Thus, the two k terms in the numerator log can be canceled out.

$$
d = \frac{2\log\left(\frac{(n-1)*(k-2)}{k} + 1\right)}{\log(k-1)} = \frac{2\log\left(\frac{(n)*(k)}{k}\right)}{\log(k)} = \frac{2\log(n)}{\log(k)}
$$

- State whether this network displays the small-world effect, defined as having a diameter that increases as $O(\log n)$ or slower.

The diameter formula has been found above to be

$$
d = \frac{2\log(n)}{\log(k)}
$$

For a small k, $d = C * log(n)$ for some constant C. Thus $d = O(log(n)$. For a large k, where $k = n/C_1$ for some constant C_1, $d = C_2 * log(n)/log(n)$. Thus, $d = O(1)$.

Since the diameter will increase as $O(\log n)$ or slower regardless of the value of k, this network displays the small-world effect.

6. (35 pts total) We will investigate several properties of online social networks by analyzing the Facebook100 (FB100) data set. Each of the 100 plaintext ASCII files in the FB100 folder contains an edge list for a 2005 snapshot of a Facebook social network among university students and faculty within some university. Interpret this edge list as a simple graph.

   (a) (5 pts) In most social networks, we observe a surprising phenomenon called the friendship paradox. Let k_u denote the degree of some individual u, and let some edge $(u, v) \in E$. The paradox is that the average degree of the neighbor $\langle k_v \rangle$ is greater than the average degree $\langle k_u \rangle$ of the vertex. That is, on average, each friend of yours has more friends than you.

   The mean neighbor degree (MND) of a network is defined as

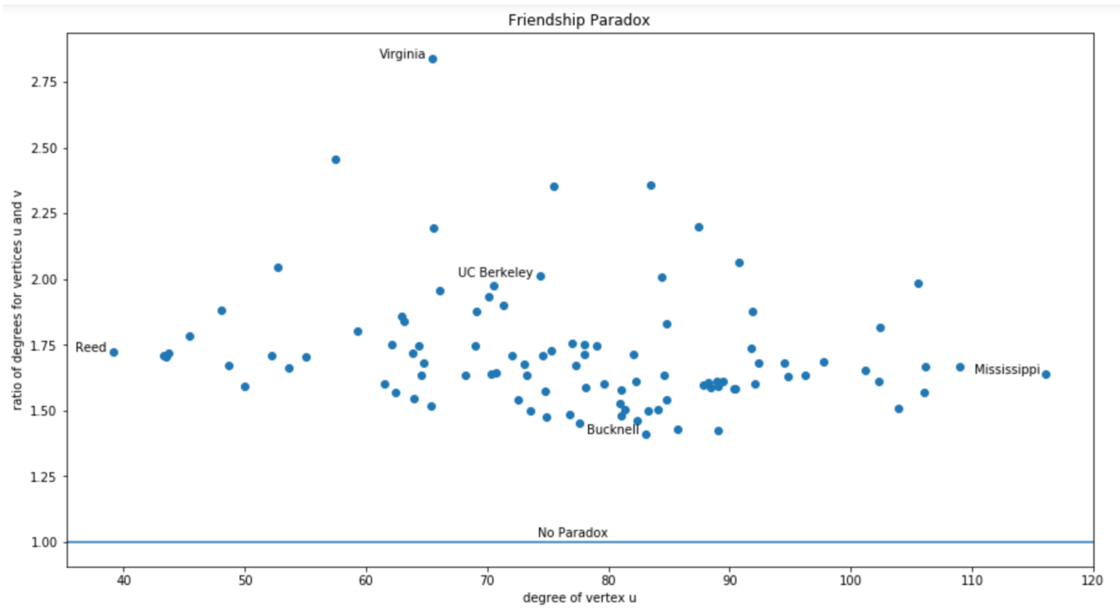
$$\langle k_v \rangle = \frac{1}{2m} \sum_{u=1}^{n} \sum_{v=1}^{n} k_v A_{uv}$$

   Derive an expression for $\langle k_v \rangle$ in terms of the average squared-degree $\langle k^2 \rangle$ and the average degree $\langle k \rangle$.

$$\begin{aligned}
\langle k_v \rangle &= \frac{1}{2m} \sum_{u=1}^{n} \sum_{v=1}^{n} k_v A_{uv} \\
&= \frac{1}{2 * \frac{1}{2} \sum_{i=1}^{n} k_i} \sum_{u=1}^{n} \vec{k} \vec{A_u} \\
&= \frac{1}{\sum_{i=1}^{n} k_i} \sum_{u=1}^{n} \vec{k} \vec{A_u} \\
&= \frac{\sum_{u=1}^{n} k_u^2}{\sum_{i=1}^{n} k_i} \\
&= \frac{\frac{1}{n} \sum_{u=1}^{n} k_u^2}{\frac{1}{n} \sum_{i=1}^{n} k_i} \\
&= \frac{\langle k^2 \rangle}{\langle k \rangle}
\end{aligned}$$

   Thus, the mean neighbor degree (MND) is the average squared-degree divided by the average degree.

(b) (15 pts) Now, using all 100 of the FB100 networks, make a figure showing a scatterplot of the ratio $\langle k_v \rangle$ / $\langle k_u \rangle$ as a function of the mean degree $\langle k_u \rangle$. Include a horizontal line representing the line of no paradox, and label the nodes corresponding to Reed, Bucknell, Mississippi, Virginia, and UC Berkeley. (Remember: figures without axes labels will receive no credit.)



- Comment on the degree to which we do or do not observe a friendship paradox across these networks as a group.
- Comment on whether there is any dependency between the size of the paradox (the MND value) and the networks mean degree. A few points of extra credit will be awarded to an explanation of why we should, in fact, expect to see a friendship paradox in these networks, and that identifies the conditions under which we should expect to see no paradox.

The friendship paradox can be seen across the networks since they all fall above the no paradox line. Every network has a ratio of $\langle k_v \rangle$ greater than 1. This graph doesn't show any dependency between the size of the paradox and the network's mean degree. These networks have ratios between 1.5 and 1.75. Since these networks all involve human interactions, it makes sense that they have this friendship paradox. For you to be connected to another person it's more likely they are connected to more people than you. This paradox would not be in a network with natural interactions rather than human interactions. For example a biological network such as proteins and bindings.

(c) (15 pts) A related phenomenon in social networks is the *majority illusion*. Let $x \in 0, 1$ be a binary-valued vertex-level property, and let $q = \frac{1}{n}\sum_u x_u$ be the fraction of vertices that exhibit this property. If we set $q < 0.5$, then this property appears only in a minority of nodes. The majority illusion occurs when $q < 0.5$, but the majority of a nodes neighbors, on average, exhibit that property, that is, $\langle x_v \rangle > 0.5$. Explain in words and mathematics how this can be possible.

This property will happen when the nodes with the highest degrees have the property. The nodes with smaller degrees will be connected to the nodes with the property more that nodes without the property since the nodes without the property don't have as many edges. Thus, even though a minority of nodes have the properties, a majority of the edges are connected to these nodes since they have a higher degree. Since the nodes with the property and connected to each other as well as nodes without the property, about half of their connections will have the property. Additionally, the nodes without the property will have more than half their connections with the property. Since these nodes are the majority, the average of node neighbors with the property will be greater than 0.5.

Assume approximately one-fourth of the nodes have the properties that are nearly fully connected. The remaining three-fourths of the nodes are connected to several nodes with the property and a couple of nodes without the property. Thus, on average for each of these nodes, 0.6 of the neighbor nodes have a the property. For the one-fourth of the nodes with the property, they will be connected to almost all the other nodes with the property and several nodes without the property. This will balance out to have close to 0.4 of the neighbors having the property. However, this will result in an average of $0.6 * 0.75 + 0.4 * 0.25 = 0.55$. Thus, even though only one-fourth of the nodes have the property on average 0.55 of the neighbors have the property.

(d) (20 pts extra credit) Another common property of social networks is that they have very small diameters relative to their total size. This property is sometimes called the small-world phenomenon and is the origin of the popular phrase six degrees of separation.

- For each FB100 network, compute (i) the diameter l_{max} of the largest component of the network and (ii) the mean geodesic distance $\langle l \rangle$ between pairs of vertices in the largest component of the network. Make two figures, one showing l_{max} versus network size n and one showing $\langle l \rangle$ versus the size of the largest component n.
  Comment on the degree to which these figures support the six-degrees of separation idea.

  The processing time took too long for me to get completed graphs. However, from the partial graphs, the networks support six-degrees of separation.

- Briefly discuss whether and why you think the diameter of Facebook has increased, stayed the same, or decreased relative to these values, since 2005. (Recall that Facebook now claims to have roughly 10^9 accounts.)

  The diameter of Facebook has likely increased since 2005. Since there are more accounts, there are likely more components to the graph especially as Facebook expands to more areas and regions. These groups will form connected components in the graph. However, by having some of these larger components connected by only a couple of edges, the diameter will increase. Although the diameter will still be $O(log(n))$, the diameter has likely increased since 2005.

Algorithm 1: Anaconda Python Code for Problem 6

```
import numpy as np
import matplotlib.pyplot as plt
import os
import networkx as nx

# intialize variables
directory = "facebook100txt"
edge_list = []
node_list = []
plot_data_x = []
plot_data_y = []
diameter = []
network_size = []
geo_dist = []
largest_n = []

# loop through all networks
for filename in os.listdir(directory):
    if not filename.endswith("_attr.txt"):
        node_list = []
        # get node list
        with open(directory+"/"+filename.split(".txt")[0]+"_attr.txt", "r") as f:
            data = f.readline()
            data = f.readline()
            while data:
                vals = data.split("\n")[0].split("\t")
                node_list.append(int(vals[0]))
                data = f.readline()
        f.close()

        edge_list = []
        # get edge list
        with open(directory+"/"+filename, "r") as f:
            data = f.readline()
            while data:
                vals = data.split("\n")[0].split("\t")
                edge_list.append((int(vals[0]), int(vals[1])))
                data = f.readline()
        f.close()

        # create graph
        G = nx.from_edgelist(edge_list)
        G.add_nodes_from(node_list)

        # calculate network size, mean neighbor degree, and average degree
        n = len(G.nodes)
        network_size.append(n)
        aveND = nx.average_neighbor_degree(G)
        MND = 0
```

```
        k = 0
        for val in aveND:
            MND += aveND[val]
            k += G.degree(val)
        MND /= n
        k /= n
        plot_data_y.append(MND/k)
        plot_data_x.append(k)

        # get largest component
        largest_cc = max(nx.connected_components(G), key=len)
        sub = G.subgraph(largest_cc)

        # calculate size, diameter, and average geodesic path length
        largest_n.append(len(largest_cc))
        geo_dist.append(nx.average_shortest_path_length(sub))
        diameter.append(nx.diameter(sub))

# create graph for 6b
labels = (("Reed", 50), ("Bucknell", 11), ("Mississippi",37),
          ("Virginia", 91), ("UC Berkeley", 5))
fig, ax = plt.subplots(figsize=(15, 8))
ax.scatter(plot_data_x, plot_data_y)
# label axis
ax.set_ylabel("ratio of degrees for vertices u and v")
ax.set_xlabel("degree of vertex u")
ax.set_title("Friendship Paradox")
# add no paradox line
ax.axhline(1)
ax.annotate("No Paradox", (80, 1.02),horizontalalignment='right')
# label points
for (txt, i) in labels:
    ax.annotate(txt, (plot_data_x[i]-0.5, plot_data_y[i]),horizontalalignment='right')

# create graph for 6d
fig, ax = plt.subplots(figsize=(15, 8))
ax.scatter(network_size, diameter)
ax.set_ylabel("diameter")
ax.set_xlabel("network size")
ax.set_title("Network Diameter")

# create second graph for 6d
fig, ax = plt.subplots(figsize=(15, 8))
ax.scatter(largest_n, geo_dist)
ax.set_ylabel("mean geodesic distance")
ax.set_xlabel("largest component size")
ax.set_title("Geodesic Distance")
```