Spring 2022 I606 Assignment 7 (Centralities)

David Terner

Februrary 22nd, 2022
Click here for the Github repository.

Preliminaries

For this assignment we will be exploring several centralities to get an intuitive sense of what the various centrality metrics tell us about the nodes in the graph.

Before messing with data, let's think about one of the foundational centralities called eigenvector centrality. The idea is an extension of the degree centrality. Instead of defining the centrality of a node as its degree, eigenvector centrality defines the centrality of a node as the sum of eigenvector centrality of the neighbors. In other words, an important node is not just a node with many neighbors, but the one with many *important* neighbors.

Mathematically this idea can be translated into finding the eigenvector of the adjacency matrix that corresponds to the largest (and positive) eigenvalue. According to the Perron-Frobenius theorem, there is only one non-negative eigenvector and it is the leading eigenvector with the largest eigenvalue.

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

Here is a question: consider an undirected k-regular graph with only one connected component. That means that everyone can be reached from everyone else and every node's degree is . What would be the eigenvector centrality vector of this graph?

Solution

Let $\mathbf{A}$ be the adjacency matrix of the k-regular graph such that:

$$\mathbf{A} = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}_{k\times k}$$

Clearly $\text{rank}(\mathbf{A}) = 1$ since $[1, \dots, 1]_{k\times 1}$ is the only linearly independent column (or row) vector in $\mathbf{A}$. By the Rank–Nullity Theorem, we know that:

$$\dim(\mathbf{A}) = \text{rank}(\mathbf{A}) + \text{null}(\mathbf{A})$$

Since the dimension of $\mathbf{A}$ is k, this implies that $\mathbf{A}$'s nullity is:

$$\text{null}(\mathbf{A}) = k - 1$$

The size of $\mathbf{A}$'s null space tells us the multiplicity (read: number) of $\mathbf{A}$'s eigenvalues equal to zero (0), namely $k-1$. Moreover, it is well known that:

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^{k} \lambda_k$$

Since $\text{tr}(\mathbf{A}) = k$, then:

$$k = \lambda_1 + \underbrace{\Big((\lambda_2 = 0) + \ldots (\lambda_k = 0) \Big)}_{k-1 \text{ times}} \implies k = \lambda_1 = \lambda$$

Now that we know that our largest eigenvalue is $\lambda = k$, we can find λ's eigenvector:

$$\mathbf{A}\nu = \lambda\nu \iff (\mathbf{A} - \lambda \mathrm{I})\nu = \mathbf{0}$$

We can expand this last to make matrix row-operations easier to follow:

$$\begin{bmatrix} 1-k & 1 & \ldots & 1 \\ 1 & 1-k & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 1-k \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

which can be written more compactly as:

$$\left[\begin{array}{cccc|c} 1-k & 1 & \ldots & 1 & 0 \\ 1 & 1-k & \ldots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \ldots & 1-k & 0 \end{array}\right] \sim \left[\begin{array}{ccccc|c} -k & k & 0 & \ldots & 0 & 0 \\ 0 & -k & k & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 \\ 1 & 1 & \ldots & 1 & 1-k & 0 \end{array}\right]$$

Looking to the first row of our modified matrix, we find that $-\lambda\nu_1 + \lambda\nu_2 = 0 \implies \nu_1 = \nu_2$. We can apply symmetric reasoning to the first $k-1$ rows and conclude that $\nu_1 = \nu_2 = \nu_3 = \cdots = \nu_{k-1} = \nu_k$ Using the last row our modified matrix implies:

$$\begin{aligned} \underbrace{\nu_k + \nu_k + \cdots + \nu_k}_{k-1 \text{ times}} + (1-k)\nu_k = 0 &\iff \\ (k-1)\nu_k + (1-k)\nu_k = 0 &\iff (k-1)\nu_k = (k-1)\nu_k \\ &\implies \nu_k = 1 \end{aligned}$$

Thus,**the eigenvector centrality of the** k-regular **graph is** $\boxed{\nu = [1, \ldots, 1]_{k\times 1}.}$

Picking the right Dolphins

All network visualizations may be found in the Appendix. The Appendix also includes Table 2 with all dolphin rankings across centrality measures.

Popularity contest

We want to know who the top dolphins are in the network, the real centers of attraction. Using what you learned about centrality from the readings and videos, choose an appropriate centrality measure that will tell us who those dolphins are. Justify your decision and list who the important dolphins are.

Solution

I chose *eigenvector centrality* as a measure of dolphin popularity. Whereas degree centrality might be a tempting measure to capture a dolphin's social position, node degree is limited to a dolphin's first degree influence (e.g. how many direct neighbors does a dolphin have?). By contrast, eigenvector centrality reflects a dolphin's degree and the the degree of the dolphin's neighbors.
The top three (3) eigenvector dolphins are: `Grin` , `SN4`, and `Topless`. See Figure 1 for more.

Relay

Dolphins like passing information around efficiently along the shortest-paths. Among their neighbors who are the most important message relayers in the network? Justify your centrality choice for finding these dolphins.

Solution

I chose *closeness centrality* as the best measure of a dolphin's communication efficiency. Closeness centrality for node v is defined as:

$$closeness(v) = \frac{1}{\frac{1}{n-1}\sum_{i \neq v} d_{vi}} \tag{1}$$

which is the same thing as the inverse of node v's average shortest path. If v, on average, is only a few "hops" away from all other nodes, the denominator of (1) converges to zero, which in turn increases v's closeness score. Dolphins with large closeness scores will be efficient at spreading information.
The top three (3) close dolphins are: `SN100`, `SN9`, and `SN4`. See Figure 2 for more.

Gossip

There is a lot smack going around the pod and everyone wants to know if Flipper will be inviting them to the party next week. But gossip takes time travel. Which dolphins are in the best position for getting all the best gossip from around the pod? Justify your centrality choice for finding these dolphins.

Solution

I choose *betweeness* as the best measure of "gossip centrality." Node v's betweeness is defined as:

$$Betweeness(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v.

The idea behind choosing betweeness for optimizing gossip flow (e.g. minimizing gossip spread time) is that nodes with high betweeness scores connect the network together: efficient communication between all nodes flows through these nodes. While betweeness and closenss are highly correlated with each other, betweeness is marginally better at accounting for rumor spread.

As a thought experiment, suppose a rumor starts a node/dolphin v. With certainty, the rumor spreads to all of v's neighbors; at the next time step, the rumor spreads to neighbors of neighbors etc. Define the *rumor time* of node v as the total number of time steps necessary to spread the rumor to the entire network. I compute the rumor of time (e.g. the largest shortest path length) of each node and correlate this measure with each centrality measure.[1] Table 1 reports the correlations. Sure enough, betweeness has the strongest association with rumor time.

Table 1: Rumor Time vs Centrality Correlation

Centrality Measure	Kendall Rank Correlation
Betweenness	-0.562
Closeness	-0.490
Degree	-0.354
Eigenvector	-0.080

The top three (3) betweeness dolphins are: `SN100`, `Beescratch`, and `SN9`. See Figure 3 for more.

[1] I use the kendall rank correlation which measures ordinal (e.g. rank ordering) relationships between variables.

Appendix

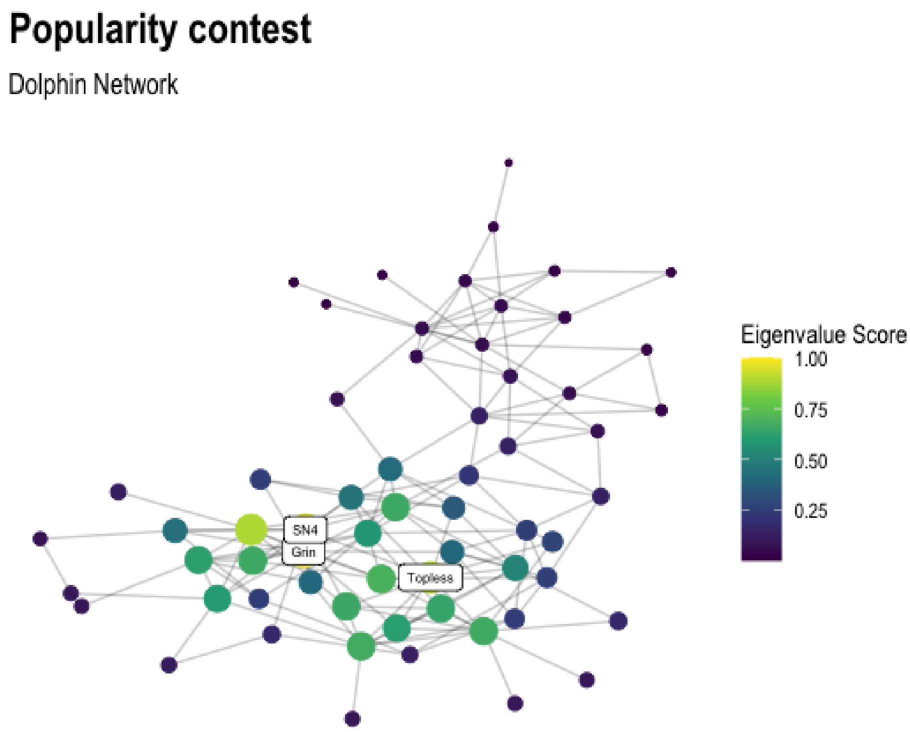

Figure 1: Popular Dolphins

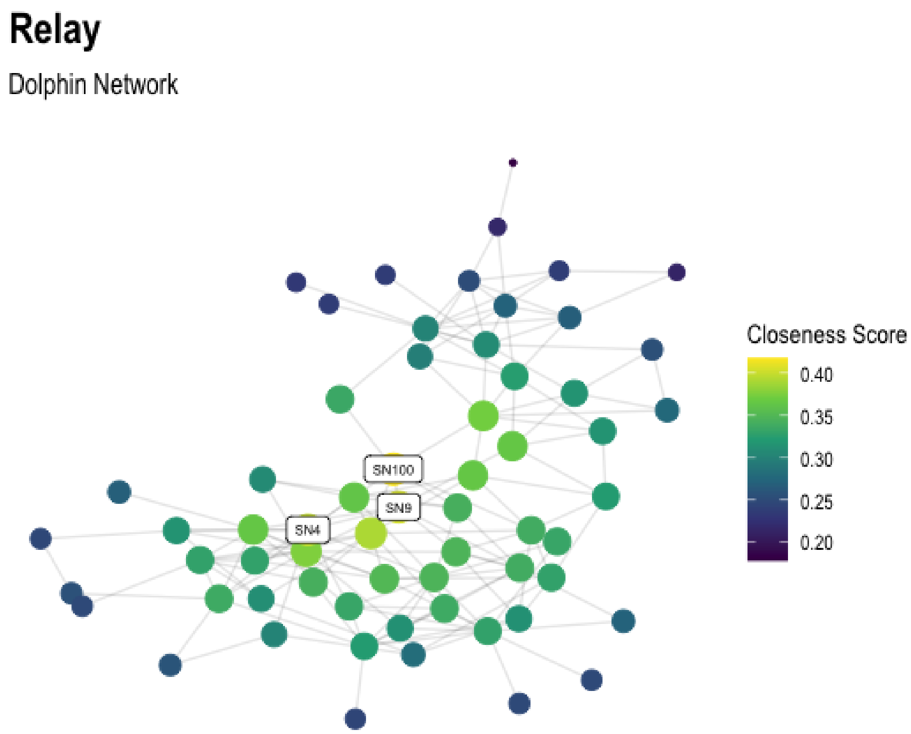

Figure 2: Relay Dolphins
Note: Dolphin closeness scores are scaled by $n-1$ to enhance visual differences (e.g. size + color) in this chart.

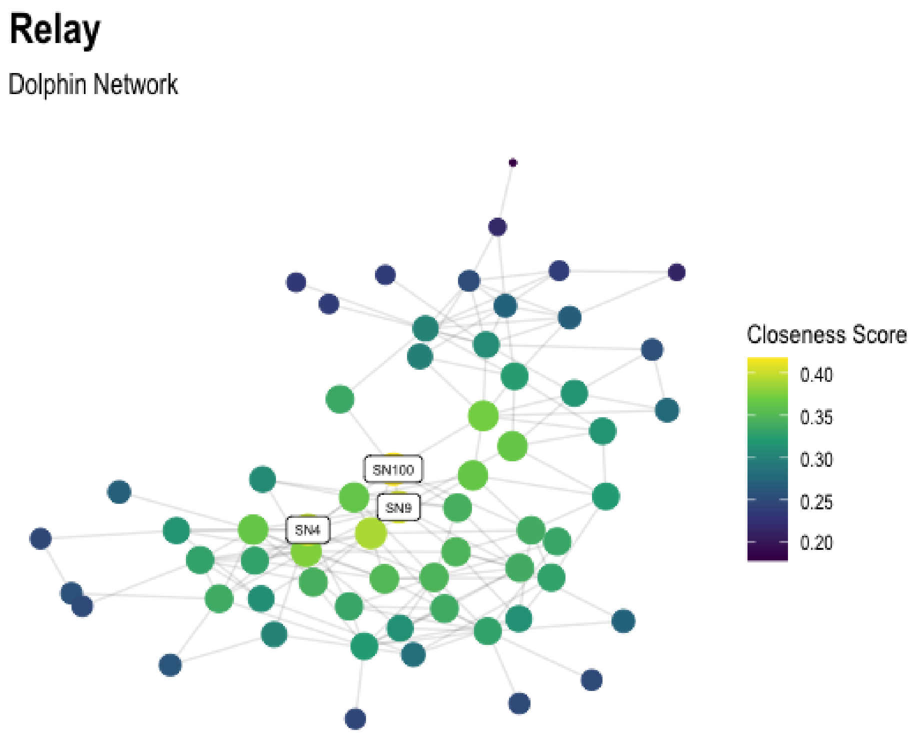

Figure 3: Gossip Dolphins

Table 2: All Dolphin Centrality Rankings

Rank	Dolphin	Eigen	Dolphin	Closeness	Dolphin	Betweenness
1	Grin	1.000	SN100	0.007	SN100	454.274
2	SN4	0.952	SN9	0.007	Beescratch	390.384
3	Topless	0.903	SN4	0.007	SN9	261.964
4	Scabs	0.890	Kringel	0.006	SN4	253.583
5	TR99	0.689	Grin	0.006	DN63	216.377
6	Patchback	0.671	Beescratch	0.006	Jet	209.169
7	Trigger	0.667	DN63	0.006	Kringel	187.842
8	Hook	0.659	Oscar	0.006	Upbang	181.393
9	SN9	0.658	Scabs	0.006	Trigger	154.959
10	MN105	0.657	Double	0.006	Web	154.095
11	Jonah	0.641	TR99	0.006	SN89	129.046
12	SN63	0.623	Beak	0.006	Oscar	122.165
13	MN83	0.612	Topless	0.006	Patchback	119.919
14	Stripes	0.603	TSN103	0.006	Stripes	114.980
15	Kringel	0.584	Zap	0.006	Grin	113.409
16	Haecksel	0.520	Haecksel	0.006	Scabs	104.615
17	Double	0.453	TR77	0.006	Gallatin	96.709
18	Shmuddel	0.440	Jonah	0.006	SN63	82.995
19	SN100	0.420	Stripes	0.006	MN60	77.194
20	TSN103	0.410	SN89	0.005	Topless	74.427
21	Beak	0.407	MN105	0.005	TR99	61.142
22	Zap	0.354	MN60	0.005	Haecksel	60.925
23	MN60	0.277	Hook	0.005	PL	60.482
24	SN96	0.256	SN63	0.005	Ripplefluke	60.000
25	TR77	0.255	SN96	0.005	Shmuddel	59.831
26	CCL	0.251	Trigger	0.005	DN21	53.752
27	Thumper	0.246	Upbang	0.005	Number1	53.503
28	Fish	0.238	Patchback	0.005	SN96	53.359
29	Oscar	0.216	PL	0.005	SN90	42.550
30	Vau	0.165	Knit	0.005	TR77	42.459
31	Zipfel	0.165	Number1	0.005	Double	40.929
32	DN63	0.136	Shmuddel	0.005	Feather	38.237
33	Beescratch	0.133	Fish	0.005	Zap	37.209
34	PL	0.129	MN83	0.005	TSN103	35.199
35	Bumper	0.126	Thumper	0.005	Beak	34.921
36	Fork	0.124	Jet	0.005	Fish	29.448
37	TSN83	0.107	CCL	0.005	Jonah	27.184
38	TR120	0.094	Web	0.005	Zipfel	25.977
39	SMN5	0.093	Zipfel	0.005	Knit	24.365
40	Five	0.093	SN90	0.005	MN105	23.242

Table 2: All Dolphin Centrality Rankings

Rank	Dolphin	Eigen	Dolphin	Closeness	Dolphin	Betweenness
41	Cross	0.093	Bumper	0.005	Thumper	22.029
42	Whitetip	0.087	Notch	0.005	Bumper	16.603
43	TR88	0.074	Gallatin	0.004	MN83	13.511
44	Upbang	0.073	Vau	0.004	DN16	8.016
45	SN89	0.066	Fork	0.004	Notch	7.983
46	Knit	0.065	DN21	0.004	Hook	6.048
47	Jet	0.055	TSN83	0.004	TR120	5.505
48	Web	0.055	TR120	0.004	CCL	4.344
49	Number1	0.052	Mus	0.004	Mus	3.009
50	SN90	0.048	Feather	0.004	TSN83	2.183
51	Gallatin	0.048	Cross	0.004	TR88	1.700
52	Feather	0.039	Five	0.004	Vau	1.606
53	DN21	0.039	Whitetip	0.004	Wave	0.250
54	Notch	0.028	TR88	0.004	Cross	0.000
55	DN16	0.021	SMN5	0.004	Five	0.000
56	Mus	0.019	DN16	0.004	Fork	0.000
57	Ripplefluke	0.012	MN23	0.004	MN23	0.000
58	Wave	0.008	Quasi	0.004	Quasi	0.000
59	MN23	0.008	TR82	0.004	SMN5	0.000
60	Quasi	0.008	Ripplefluke	0.004	TR82	0.000
61	TR82	0.008	Wave	0.003	Whitetip	0.000
62	Zig	0.002	Zig	0.003	Zig	0.000