CSE 5245: Network Science

Network Node Measures

Samuel Roth
Ohio State University
roth.375@osu.edu

Abstract

Centrality is a measure of how important a node is within a network. Analyzing central nodes can lead to novel insights in various fields such as economics, sociology, and information technology. When it comes to defining central nodes, however, there are a variety of different approaches that can sometimes provide conflicting results. In this project, we utilize datasets made public by the Stanford Network Analysis Project to demonstrate how these different measures can be computed and how their results can vary given an assortment of different network topologies.

Introduction

Network Terminology

Generally speaking, *networks* consist of two components: a set of *nodes*, which represent unique objects or data points, and a set of *edges*, which describe relationships amongst two or more nodes. Edges can either be *directed* or *undirected*, meaning that the relationship between nodes is either unidirectional or bidirectional, respectively. Figure 1 illustrates an undirected network that could represent a group of friends just as easily as it could represent a set of interdisciplinary research projects at a university.

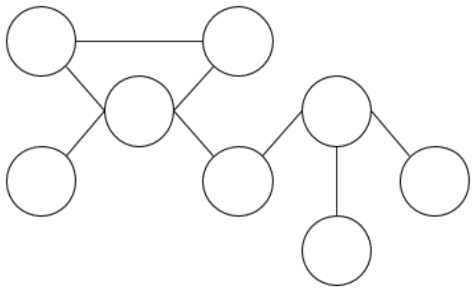

Figure 1. An example of an undirected network.

Figure 2, on the other hand, shows a directed network. As stated, directed networks convey unidirectional relationships, such as followers of a particular page on a social network or customers of some business.

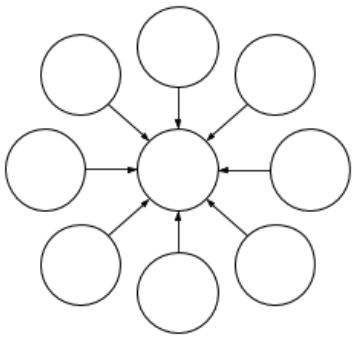

Figure 2. An example of a directed network.

Centrality

In addition to the type of edges between nodes, there is another important feature that distinguishes the two example networks. Specifically, one node in Figure 2 has a relationship with every other node, yet there is no such node in Figure 1. It can be inferred that the aforementioned node in Figure 2 is meant to be the focus of the network; in other words, it is more central, or more important than the others. What exactly does that mean? The answer to that question, of course, depends on the context of the network.

At first glance, it may appear that the nodes in Figure 1 are, for the most part, equally important. However, upon further inspection it can be observed that there is one node that has relationships with four other nodes, four nodes that have relationships with two other nodes, and finally, three nodes that have relationships with only one node. Thus, there exists a sort of implicit heirarchy that allows us to gauge the centrality of nodes in this network and others like it.

In these two examples, we utilize a measure called *degree centrality*, which is a rather straight-forward method of evaluating the importance of nodes in a network. There are many other ways of determining important nodes, which will be presented in this paper. To compare these different measures, we employ them on a series of different datasets made available by the Stanford Network Analysis Project [1].

Datasets

In order to develop insight into how the various centrality measures differ from one another, it is important to test them on real-world datasets. Thankfully, the rapid growth

of the Internet and connected devices has resulted in the availability of a staggering amount of data. Stanford's Network Analysis Project (SNAP) is just one of many places to find data published to help researchers. An example of a similar resource is the Network Repository [2].

For this experiment, we limit our analysis to four different datasets, all of which were provided by SNAP. Table 1 figures characteristics of these datasets; more information regarding these networks can be found on their respective websites.

Table 1. Overview of Tested Datasets

Dataset Name	Type	# Nodes	# Edges	Description
wiki-Vote [3]	Directed	7115	103689	Voting data from Wikipedia users.
ca-GrQc [4]	Undirected	5242	14496	Collaborations between authors for a specific Arxiv category.
p2p-Gnutella08 [5]	Directed	6301	20777	Snapshot of Gnutella peer-to-peer file-sharing network.
ego-Facebook [6]	Undirected	4039	88234	Anonymized social circles on Facebook.

Implementation

This project was conducted with Python on the Ohio State University student Linux system. We utilized various tools to build our analysis pipeline, which are elaborated upon below.

Managing Development Environments with Anaconda

Anaconda is a popular platform for managing Python development environments, particularly geared for those doing work in the data science industry [7]. This is because the standard Anaconda Python distribution comes bundled with many popular data science modules, including the ones that were used in this experiment, which are described in more detail below. Development environments help to isolate our experiments and ensure that they are reproducible across different platforms; we no

longer have to consider the complications associated with Python's fragmented versioning.

Analyzing Networks with networkx *and* matplotlib

networkx and **matplotlib** are two Python modules that are used in our analysis pipeline. The former provides a set of graph structures and algorithms, making it easy to transform the plaintext files distributed by SNAP into structures that can be quickly analyzed. Matplotlib was used to generate images corresponding to these different datasets, which are shown in the Results section.

Conducting Analysis in Jupyter Notebook

While the project is submitted as a consolidated Python script, the work was done in a Jupyter notebook, which allows for a much more flexible development experience. It allows us to show every stage of our analysis, from importing different modules to generating network diagrams. Figure 3 illustrates the Jupyter notebook interface.

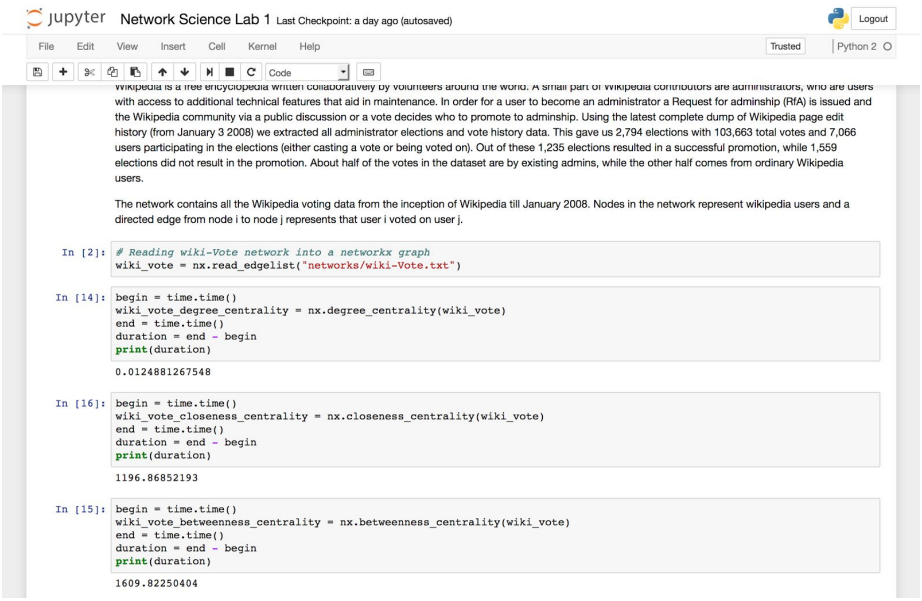

Figure 3. The Jupyter notebook interface.

Results

Topology Analysis

Network Attributes

We can already glean insight simply from the number of nodes and edges presented in Table 1. Consider two different types of *complete networks*, directed and undirected, where every node has relationships with every other node in the network. We can define a *completeness factor* for directed and undirected networks. This is a value between 0 and 1, defined as the ratio of the number of edges in some network to the number of edges required for that network to be complete. For both calculations, we will assume that any given node cannot have a directed nor undirected relationship with itself. We will also assume that all relationships are unique, meaning that one node cannot have multiple relationships with another node.

Thus, for an undirected network of N nodes, it must have $E_U = N * (N - 1) / 2$ edges to be considered complete. For a directed network of N nodes, on the other hand, each node must have an inbound and outbound relationship with every other node, thus making the number of edges required for completeness to be $E_D = 2 \times E_U = N * (N - 1)$. Table 2 shows the completeness factor for the four datasets.

Table 2. Completeness Factors for Tested Datasets

Dataset Name	Completeness Factor
wiki-Vote	0.002049
ca-GrQc	0.001055
p2p-Gnutella08	0.000523
ego-Facebook	0.010820

These values indicate that the Facebook network is the most connected, followed by the Wikipedia network, then the Arxiv network, and finally the Gnutella network. This seems to make sense because Facebook tends to have a higher emphasis on mutual friends (or friends-of-friends) than the other networks. Facebook's Suggested Friends feature also helps to connect these nodes which may otherwise go unconnected in other scenarios.

The next two networks, Wikipedia and Arxiv, are less connected it seems because their relationships imply a sort of endorsement, whether it be Wikipedia users approving a new piece of content or groups of academics working together to publish their research. These

two groups tend to be more isolated, usually by subject area. For example, those who are connected in the Arxiv network are collaborators who are likely part of the same research group. Finally, the Gnutella network is pretty dispersed, which makes sense because peer-to-peer sharing networks are built on the premise that not every node has to be connected.

Network Topologies

The matplotlib Python module allowed us to generate drawings depicting the tested networks. Those drawings are shown in Figure 4.

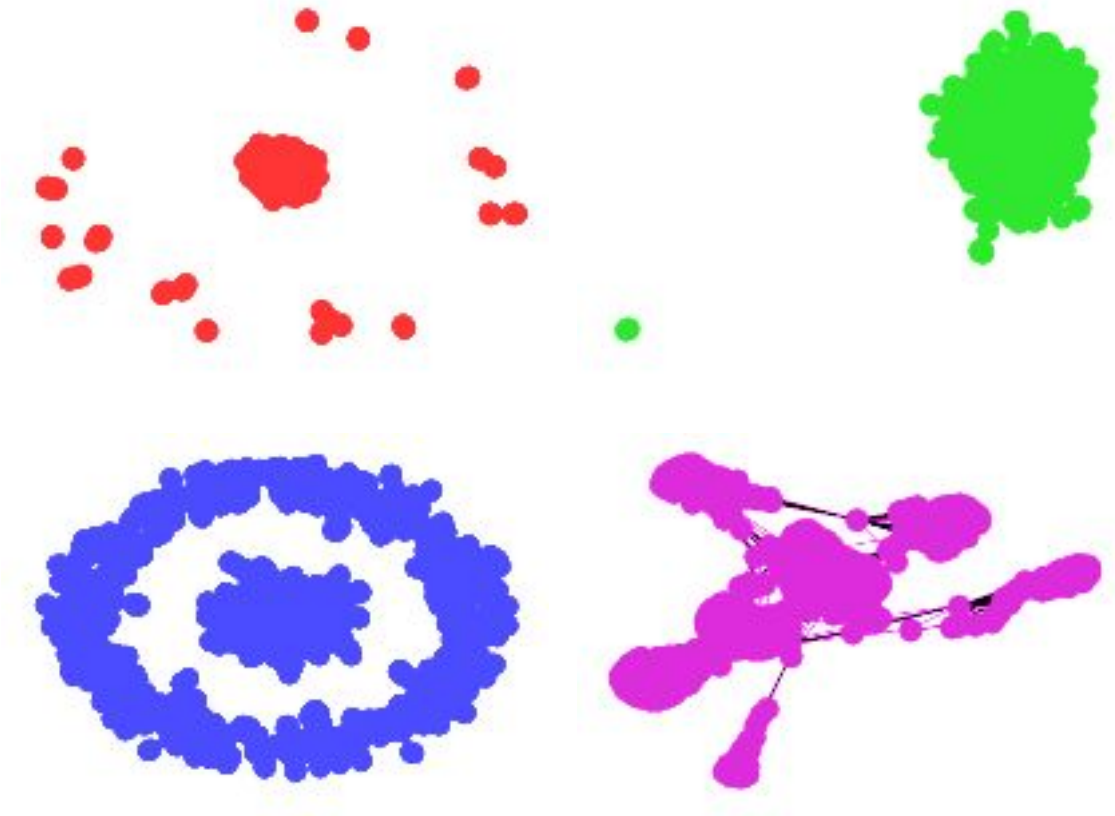

Figure 4. Network topologies drawn by matplotlib.
Clockwise from top left: wiki-Vote, p2p-Gnutella08, ego-Facebook, then ca-GrQc.

A quick glance at these topologies suggests that we are looking at four different types of networks. The wiki-Vote network contains a strongly-connected central node and a loosely-connected outer ring. The ca-GrQc network is similar, except its outer nodes are more connected (and there are more). The p2p-Gnutella08 network has a large group of nodes with another, much smaller group that is very isolated. Finally, the ego-Facebook network is fairly evenly-distributed amongst a series of similarly-sized groups.

Centrality Measures

An overview of the different centrality measures is presented in Tables 3 through 5. For each dataset, the value for normalized centrality measures are ordered (by ascending numeric value) in color: red, orange, yellow, green, blue, then purple. This allows us to visually identify differences in centrality measures amongst datasets. (Table 5 is not colored because the ordering of measure time across all datasets is the same.)

Table 3. Average of Centrality Measures for Tested Datasets

Measure (avg)	wiki-Vote	ca-GrQc	p2p-Gnutella08	ego-Facebook
Degree	0.003981	0.001055	0.001047	0.010820
Closeness	0.309213	0.106301	0.218031	0.276168
Betweenness	0.000312	0.000606	0.000578	0.000667
Eigenvector	0.005369	0.001615	0.003588	0.003864
Pagerank	0.000141	0.000191	0.000159	0.000248
Clustering coef.	0.140898	0.529636	0.010868	0.605547

Table 4. Standard Deviation of Centrality Measures for Tested Datasets

Measure (stdv)	wiki-Vote	ca-GrQc	p2p-Gnutella08	ego-Facebook
Degree	0.008093	0.001511	0.001356	0.012980
Closeness	0.049673	0.056590	0.025216	0.036120
Betweenness	0.001471	0.001975	0.001029	0.011645
Eigenvector	0.010570	0.013717	0.012076	0.015253
Pagerank	0.000238	0.000132	0.000149	0.000259
Clustering coef.	0.199439	0.428682	0.049695	0.214436

Table 5. Time Required of Centrality Measures for Tested Datasets

Measure (sec)	wiki-Vote	ca-GrQc	p2p-Gnutella08	ego-Facebook
Degree	0.012488	0.008931	0.010804	0.007697

Closeness	1196.868522	170.401973	450.171536	498.007318
Betweenness	1609.822504	254.810570	758.492588	525.946447
Eigenvector	10.035305	4.609458	5.246363	19.748623
Pagerank	11.042044	2.542989	3.555104	8.932661
Clustering coef.	6.726064	0.388206	0.466764	4.728789

Some observations that follow:

- Average clustering coefficient and closeness index were reversed for the Arxiv network.
- Standard deviation for degree and betweenness indices were reversed in the Arxiv network.
- Average degree and eigenvector indices were reversed in the Facebook network, as were the average clustering coefficient and closeness index.

Though these differences are subtle, it should follow that normalized centrality measures are consistent with each other within a particular network. The conflicts mentioned above can likely be attributed to the variety in network topologies that were analyzed.

Degree Centrality

As described earlier, degree centrality refers simply to the number of edges that connect to a particular node. The more edges a node has, the more important it is within the context of the network. This measure is rather easy to calculate because we just have to count the edges at a particular node.

Closeness Centrality

For a given node, its closeness corresponds to the shortest paths from itself to every other node. A node is more central than some other node if it has shorter paths to all other nodes in the graph. To compute this measure, we must calculate every possible path measure for a given network, which makes it one of the most expensive measures.

Betweenness Centrality

Similar to closeness centrality, betweenness involves computing shortest paths amongst nodes. It is somewhat of a compliment, and a measure of the number of times that a particular node serves as part of a *bridge* in the shortest path between some other two nodes. In this sense, a node is more central if many other pairs of nodes depend on it for their shortest path relationship. Considering the time to actually process the shortest-path

data for every pair of nodes, this measure ends up being even more expensive than closeness centrality.

Eigenvector and Pagerank Centrality

Eigenvector centrality is based on a calculated score rather than shortest paths amongst nodes in a network. Specifically, nodes with high-scoring neighbors are more important than those with low-scoring neighbors. This measure is substantially less expensive to calculate than closeness or betweenness. Pagerank is a derivative of the eigenvector centrality measure, developed by Larry Page and popularized by its use in the Google search engine. Logically, it states that a node is more central or important if it has many other nodes (prioritized by their own importance, or Pagerank) linking to it.

Clustering Coefficient

The clustering coefficient is a measure that can be conducted globally — across the entire network — or, in our case, locally. In the latter, it is a measure of how a particular node fits in amongst a group. Specifically, we are calculating how close a particular node's neighbors are to being complete (i.e. all connected with one another). The closer they are, the more important the particular node is. Next to simply counting the inbound and outbound links, this is the cheapest measure to calculate.

Conclusion

In this paper, we introduced centrality as a type of measure that can be conducted to determine the importance of nodes within a network. We analyzed four different datasets to better understand how different measures of centrality can portray nodes with varying levels of importance. This experiment was done in a Jupyter notebook, then consolidated into a standalone Python program, which was made available to the course instructor.

References

1. Stanford Network Analysis Project, http://snap.stanford.edu/
2. Network Repository, http://networkrepository.com/
3. Wikipedia Voting network, http://snap.stanford.edu/data/wiki-Vote.html
4. Arxiv General Relativity and Quantum Cosmology collection network, https://snap.stanford.edu/data/ca-GrQc.html
5. Gnutella p2p network, http://snap.stanford.edu/data/p2p-Gnutella08.html
6. Facebook Egonet, https://snap.stanford.edu/data/egonets-Facebook.html
7. Anaconda Data Science Platform, https://www.anaconda.com/