Centrality Metrics via NetworkX, Python

pj
Nov 17, 2018 · 4 min read

Zachary's karate club is a widely used dataset [1] which originated from the paper "An Information Flow Model for Conflict and Fission in Small Group" that was written by Wayne Zachary [2]. The paper was published in 1977.

This dataset will be used to explore four widely used ***node*** centrality metrics (Degree, Eigenvector, Closeness and Betweenness) using the python library NetworkX.

Warning: *This social network is not a directed graph. Computing directed graph centrality metrics will not be covered here.*

```
import networkx as nx

G = nx.karate_club_graph()

## #nodes: 34 and #edges: 78
print('#nodes:', len(G.nodes()), 'and', '#edges:', len(G.edges()))
```

Building a Text Generator based on the most influential authors in the r/Siacoin…

Building an LSTM model using Keras based on the content produced by the most influential authors in the r/Siacoin…

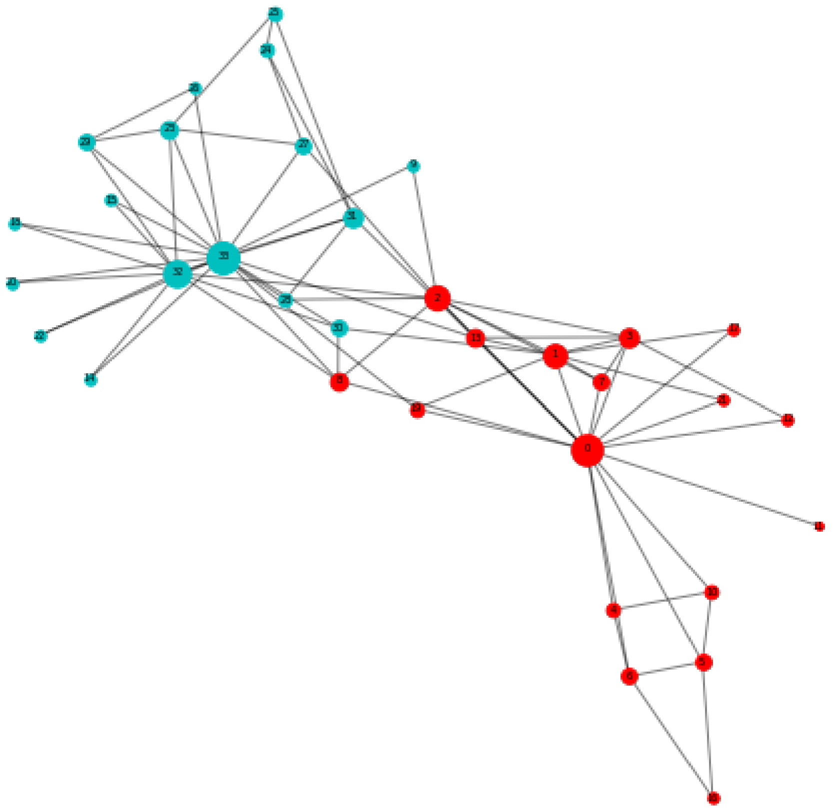
medium.com

Degree Centrality

The degree of a node is simply defined as the number of connecting edges that it has. The node '33' has 17 edges connecting it, to other nodes in the network. This results in a degree of 17. To determine the degree centrality, the degree of a node is divided by the number of other nodes in the network (n-1). To continue with computing the degree centrality for node '33', 17 / (34-1) results in 0.5152. Remember from above, the number of nodes in the dataset is 34.

An interpretation of this metric, ***Popularity***.

```
degree_centrality = nx.degree_centrality(G)
```



Degree Centrality — Karate Club

```
[
   (33, 0.5152),
   (0, 0.4848),
   (32, 0.3636),
   (2, 0.303),
   (1, 0.2727),
   (3, 0.1818),
```

```
    (31, 0.1818),
    (8, 0.1515),
    (13, 0.1515),
    (23, 0.1515),
    (5, 0.1212),
    (6, 0.1212),
    (7, 0.1212),
    (27, 0.1212),
    (29, 0.1212),
    (30, 0.1212),
    (4, 0.0909),
    (10, 0.0909),
    (19, 0.0909),
       ...
]
```

For the full notebook, go here.

. . .

Eigenvector Centrality

The adjacency matrix allows the connectivity of a node to be expressed in matrix form. So, for non-directed networks, the matrix is symmetric.

```
nx.adjacency_matrix(G).todense()

matrix([
   [0, 1, 1, ..., 1, 0, 0],
   [1, 0, 1, ..., 0, 0, 0],
   [1, 1, 0, ..., 0, 1, 0],
             ...,
   [1, 0, 0, ..., 0, 1, 1],
   [0, 0, 1, ..., 1, 0, 1],
   [0, 0, 0, ..., 1, 1, 0]], dtype=int32)
```

Eigenvector centrality uses this matrix to compute its largest, most unique eigenvalues. The resulting eigenvector is used as the metric. The basic idea behind this metric revolves around a nodes neighbors and how connected they are. To score higher, a node needs to be well connected (high degree centrality) but it also needs to be connected to others that are well connected.

An interpretation of this metric, ***Influence***.

```
eigenvector_centrality = nx.eigenvector_centrality(G)
```


Eigenvector Centrality — Karate Club

```
[
   (33, 0.3734),
   (0, 0.3555),
   (2, 0.3172),
   (32, 0.3087),
   (1, 0.266),
   (8, 0.2274),
   (13, 0.2265),
   (3, 0.2112),
   (31, 0.191),
   (30, 0.1748),
   (7, 0.171),
   (23, 0.1501),
   (19, 0.1479),
   (29, 0.135),
```

```
    (27, 0.1335),
    (28, 0.1311),
    (9, 0.1027),
    (14, 0.1014),
    (15, 0.1014),
        ...
]
```

For the full notebook, go here.

. . .

Closeness Centrality

This metric involves computing the shortest path between a node and all the other nodes in the network. Closeness centrality of a node is just the average of all of those shortest paths. This can also be computed using the previously mentioned adjacency matrix. See this post if curious.

An interpretation of this metric, ***Centralness***.

```
closeness_centrality = nx.closeness_centrality(G)
```

Closeness Centrality — Karate Club

```
[
   (0, 0.569),
   (2, 0.5593),
   (33, 0.55),
   (31, 0.541),
   (8, 0.5156),
   (13, 0.5156),
   (32, 0.5156),
   (19, 0.5),
   (1, 0.4853),
   (3, 0.4648),
   (27, 0.4583),
   (30, 0.4583),
   (28, 0.4521),
   (7, 0.44),
   (9, 0.4342),
   (23, 0.3929),
   (5, 0.3837),
   (6, 0.3837),
   (29, 0.3837),
       ...
]
```

For the full notebook, go here.

. . .

Betweenness Centrality

This metric revolves around the idea of counting the number of times a node acts as a *bridge*. A bridge in a social network is someone who connects two different social groups. This allows for the dissemination of information between two social groups to occur [3].

For each node (v) in the network, do the following for each pair of nodes (s, t) where (s) and (t) is not (v),

1. Compute the number of shortest paths where the two nodes (s, t) are the ends.
2. Out of all those paths, figure out how many of those shortest paths have (v) in them.
3. Compute the fraction of (step 2 / step 1).
4. Sum all of these fractions up across all the pairs of nodes.

Intuition: A high value for a node indicates that it is situated in the middle of a number (higher amount) of shortest paths.

An interpretation of this metric, ***Bridge***.

```
betweenness_centrality = nx.betweenness_centrality(G)
```


Betweenness Centrality — Karate Club

```
[
   (0, 0.4376),
   (33, 0.3041),
   (32, 0.1452),
   (2, 0.1437),
   (31, 0.1383),
   (8, 0.0559),
   (1, 0.0539),
   (13, 0.0459),
   (19, 0.0325),
   (5, 0.03),
   (6, 0.03),
   (27, 0.0223),
   (23, 0.0176),
   (30, 0.0144),
   (3, 0.0119),
   (25, 0.0038),
   (29, 0.0029),
   (24, 0.0022),
   (28, 0.0018),
       ...
]
```

For the full notebook, go here.

. . .

Building a Centrality Metrics author filter for the r/Siacoin Subreddit Community

Using centrality metrics to filter authors to just the most influential (top=n) to aid in noise reduction.

medium.com

References:

[1]. https://en.wikipedia.org/wiki/Zachary%27s_karate_club

[2]. http://www1.ind.ku.dk/complexLearning/zachary1977.pdf

[3]. https://en.wikipedia.org/wiki/Centrality

Data Visualization Social Network Analysis Centrality Metrics Statistical Analysis