Finding Top-k Central Nodes in a Diffusion Network Using Various Methods

Seojun Yang

Concord Academy

ABSTRACT

We created a synthetic undirected graph of disease diffusion network that expresses the disease infectee as a node and their relation to other infectees as an edge. To figure out the infectee who is influential the most in spreading the disease, we used various methods to compare each infectee›s influence across the network: degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, PageRank, and Katz centrality. After calculating each infectee›s centralities and PageRank in the diffusion network, we concluded that betweenness centrality is the ideal method for diffusion network since the similarity between the infectees with high betweenness centrality and the infectees whose substantial influence is intuitively noticed is high. Also, we discussed future work to get the most central nodes in a graph more accurately.

I. INTRODUCTION

A disease's diffusion network is a graph that expresses infected people as nodes and routes of virus transmission as edges. The diffusion network is an effective way of showing the interactions between people that transmits the virus and which individual or group of people are the most influential in transmitting the virus. As mentioned above, each node in the diffusion network represents each individual, and the edges that connect the nodes show the route the virus took to infect one person from another. In other words, a node that has the most number of edges has the most number of routes that carry the node's virus to others and is thus the most influential node in the diffusion networks.

It is important to use the diffusion network to find the most influential node since knowing how might the whereabouts and actions of the most influential infectee from the network have affected spreading the disease can be a key to the cessation of the disease's current epidemic. Not only does taking an in-depth look at the diffusion network help to accelerate the process of fighting disease, but it can also function as a landmark that data scientists can begin with when another virus breaks out in the future, which set up a basis for this research project.

In this paper, we will use centrality [4] to find the most influential nodes. The number of nodes affected by a node is directly proportional to the node's centrality score [5]. Thus, we need to find the node with the highest centrality score.

In chapter II, methods for calculating centralities for this experiment were described with equations and examples. In chapter III, the experiment's result is analyzed. Chapter IV provides a conclusion and proposes future works.

II. RELATED WORKS

There are methods to analyze the importance of a node across a network of a graph such as degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, PageRank centrality, and so on[6],[7],[9]. In the following sections, these methods will be described with equations and examples.

Degree Centrality

The degree centrality [1] ranks a node's importance by the number of edges [1]. If a node has many edges that connect to other nodes, then the node has a high degree centrality. In contrast, if a node has a few edges, then the node has a low degree centrality.

In an undirected graph, the degree centrality C_d of node v_i is defined as

$$C_d(v_i)=d_i \quad \textbf{(1.1)}$$

In the equation above,d_i represents node v_i's the degree, or the number of adjacent edges.

There are three types of degree centrality in directed graphs: the in-degree centrality, the out-degree centrality, and both.

$C_d(v_i)=d_i^{in}$
(1.2)

$C_d(v_i)=d_i^{out}$
(1.3)

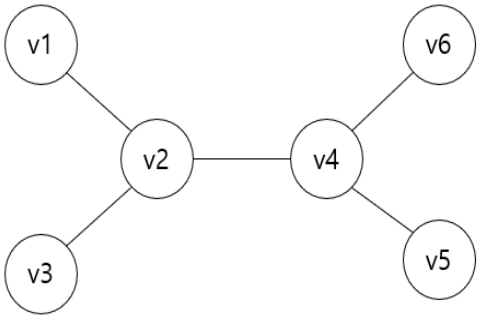
$C_d(v_i)=d_i^{in}+ d_i^{out}$
(1.4)

The in-degree centrality measures a node's prominence, or how many other nodes are connected to the node. The out-degree centrality measures a node's gregariousness, or how many other nodes the node is connected to. Using both the in-degree and the out-degree centrality doesn't take direction into account, which is the same as equation 1.1.

Normalization of Degree Centrality

<Fig 1 >

A node's degree centrality does not measure the node's importance in comparison to other nodes. To compare the degree centralities of nodes, the degree centrality values have to be normalized.



The degree centrality of node v_i can be normalized:

$C_d^{norm}(v_i)=d_i/(n-1)$
(1.5)

Also, maximum degree can be used to normalize the degree centrality:

$C_d^{max}(v_i)=d_i/(max_j d_j)$
(1.6)

Lastly, it is possible to normalize the degree centrality by the degree sum:

$C_d^{sum}(v_i)=d_i/(\sum_j d_j)=d_i/2|E| =d_i/2m$
(1.7)

Example 1 - Consider Figure 1

In the case of the graph in figure 1, a node's degree equals the number of edges that the node is connected to. For instance, the degrees of the nodes in the graph in figure 1 are:

j = 1: 2

j = 2: 3

j = 3: 1

j = 4: 3

j = 5: 1

j = 6: 1

In the graph in figure 1, n, the total number of nodes is 6. To normalize degree centrality, we need to divide a node's degree centrality by n-1:

j = 1: 2/5

j = 2: 3/5

j = 3: 1/5

j = 4: 3/5

j = 5: 1/5

j = 6: 1/5

Nodes v_2 and v_4 have the highest degree centrality. Thus, when we normalize the degree centrality by maximum degree, v_2 and v_4 become 1 as $max_j d_j = 3$.

Betweenness Centrality

Betweenness centrality [8] determines the node's importance by measuring how many shortest paths between other nodes include the node.

$C_b(v_i)=\sum_{s\neq t\neq v_i} \sigma_{st}(v_i)/\sigma_{st}$
(2.1)

In the equation above, σ_{st} is the number of shortest paths between node s and node t, and $\sigma_{st}(v_i)$ is the number of shortest paths between node s and node t that includes node v_i .

Betweenness centrality also needs to be normalized to be compared to other nodes across the network. To normalize betweenness centrality, the maximum value of $C_b(v_i)$ needs to be found. When $C_b(v_i)$ is its maximum, $\sigma_{st}(v_i)$, the number of every shortest path except for those that include v_i as s or t, is equal to σ_{st} .

$C_b(v_i)=\sum_{s\neq t\neq v_i} \sigma_{st}(v_i)/\sigma_{st} =\sum_{s\neq t\neq v_i} 1=2\ (n-1)/2=(n-1)(n-2)$
(2.2)

Example 2 - Consider Figure 1

Take node v_2 in figure 1 as an instance. v_2 is between nodes v_1 and v_3 on the shortest path between v_1 and v_3. There is only one shortest path that connects v_1 and v_3, thus there is only one instance when v_2 is a part of shortest path. The number of instances when v_2 is a part of shortest path between v_1 and v_3 is divided by the number of shortest path between v_1 and v_3. The same needs to be done for shortest paths between other combinations of two nodes and summed. Then, the summed number needs to be multiplied by 2 since one edge has two possible directions.

j = 1: 0

j = 2: 2×((1/1)+(1/1)+(1/1)+(1/1)+(1/1)+(1/1)+(1/1)+0+0) = 14

j = 3: 0

j = 4: 2×((1/1)+(1/1)+(1/1)+(1/1)+(1/1)+(1/1)+(1/1)+0+0) = 14

j = 5: 0

j = 6: 0

Closeness Centrality

Closeness centrality [3] measures how quickly a node can get to other nodes. In other words, a node with high closeness centrality is less in average shortest path length to other nodes.

$$C_c(v_i)=1/(l_{v_i})$$

(3.1)

In the equation above, $C_c(v_i)$ represents the closeness centrality of a node v_i. l_{v_i} represents the average shortest path length that connects v_i to other nodes.

Example 3 - Consider Figure 1

Take node v_1 in figure 1 as an instance. The length of shortest path between v_1 and v_2, between v_1 and v_3, between v_1 and v_4, between v_1 and v_5, and between v_1 and v_6 are 1, 2, 2, 3, and 3 respectively. The average of these lengths is l_{v_i}. Dividing 1 by l_{v_i} results in the closeness centrality of the node v_1.

j = 1: 1/((1+2+2+3+3)/5) = 0.45454545454

j = 2: 1/((1+1+1+2+2)/5) = 0.71428571428

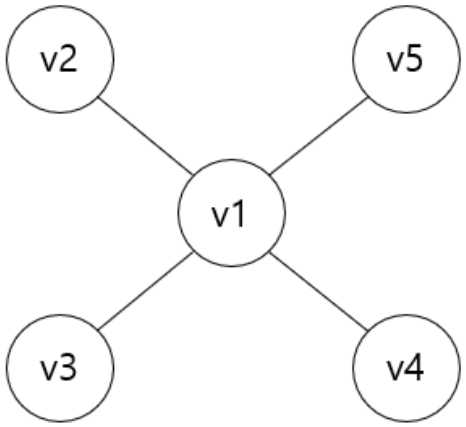
j = 3: 1/((2+1+2+3+3)/5) = 0.45454545454

j = 4: 1/((2+1+2+1+1)/5) = 0.71428571428

j = 5: 1/((3+2+3+1+2)/5) = 0.45454545454

j = 6: 1/((3+2+3+1+2)/5) = 0.45454545454

Eigenvector Centrality



<Fig 2>

Eigenvector centrality puts more importance on how many influential nodes a node is connected to than on the length of the shortest path that connects the node to other nodes or the betweenness of the node. The equation below represents eigenvector centrality.

$$C_e(v_i)=1/\lambda \sum_{j=1}^{n} A_{j,i}C_e(v_j)$$

(4.1)

In the equation, λ is a constant. The equation can be rewritten as:

$$\lambda C_e=A^T C_e$$

(4.2)

In undirected graph, A equals A^T. Thus the equation above can be expressed as:

$$\lambda C_e=A\ C_e \Leftrightarrow (\lambda-A)C_e=0$$

(4.3)

Example 4 - Consider Figure 2

C_e is not 0, so we need to consider λ-A as 0. However, λ is a constant, while A is a matrix. Therefore, we need to multiply λ by an identity matrix I, and we get the equation below.

$$(A-\lambda I)C_e=0$$

(4.4)

The adjacency matrix of the graph in figure 2 is:

	v1	v2	v3	v4	v5
v1	0	1	1	1	1
v2	1	0	0	0	0
v3	1	0	0	0	0
v4	1	0	0	0	0
v5	1	0	0	0	0

<Table 1>

Once the adjacency matrix is substituted to equation 4.4, we get the equations below.

$$det(A-\lambda I) = det\left(\begin{bmatrix}0&1&1&1&1\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\end{bmatrix} - \begin{bmatrix}\lambda&0&0&0&0\\0&\lambda&0&0&0\\0&0&\lambda&0&0\\0&0&0&\lambda&0\\0&0&0&0&\lambda\end{bmatrix}\right) = det\left(\begin{bmatrix}-\lambda&1&1&1&1\\1&-\lambda&0&0&0\\1&0&-\lambda&0&0\\1&0&0&-\lambda&0\\1&0&0&0&-\lambda\end{bmatrix}\right)$$

(4.5)

As we calculate the determinant of A-λI, we can see that$(-\lambda)^5=0$. Thus, λis 0, and A-λI equals the adjacency matrix. Assuming that C_e is $[u_1,u_2,u_3,u_4,u_5]^T$, we can replace A-λI with the adjacency matrix in equation n, which results in the equation below.

$$\begin{bmatrix}0&1&1&1&1\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\end{bmatrix}\begin{bmatrix}u_1\\u_2\\u_3\\u_4\\u_5\end{bmatrix} = 0$$

(4.6)

The equations above means that C_e equals 0, which means the eigenvector centrality of each node in the graph of figure 2 equals zero. The eigenvector centrality of every node is the same because the node that has the most number of edges is v_1, and every node in the graph is connected to v_1. Since eigenvector centrality measures how many influential nodes a node is connected to, the eigenvector centralities for all of the nodes are the same.

While the eigenvector centrality isn't useful in situations in which every node is connected to the most central node, PageRank [2], [10], and Katz centrality can distinguish the nodes with their features by using two constants, α and β:

$$C_p=\alpha A^T D^{-1} C_p+\beta 1,\ C_{Katz}=\beta(I-\alpha A^T)^{-1}\cdot 1$$

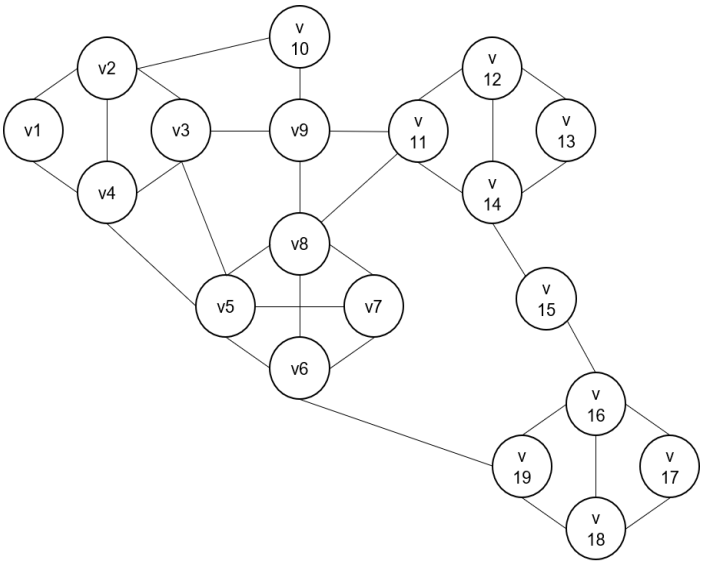
(4.7)

Example 5

The table 3 shows centralities of the nodes in the graph in figure 3 with various methods: degree centrality, betweenness centrality, closeness centrality, and PageRank. Consider the nodes in the First and Second column. v_5, v_6, v_8, v_{11} in the two columns are nodes that are connected to a larger community of nodes, which increases these nodes, centrality.

	First	Second	Third	Fourth
Degree Centrality	{v5, v8}	{v2, v3, v4, v6, v9, v11, v14, v16}	{v7, v12, v18, v19}	{v1, v10, v13, v15, v17}
Betweenness Centrality	{v11}	{v6}	{v5}	{v8}
Closeness Centrality	{v8}	{v6}	{v5, v11}	{v9}
PageRank	{v5}	{v8}	{v16}	{v14}

***<Table 2:** Centralities of a graph in Fig 3>*



<Fig 3 >

III. EXPERIMENT

The experiment was done on a computer of which the cpu is Intel(R) Core(™) i7-8750H and the ram is 16.0 GB.

Setting

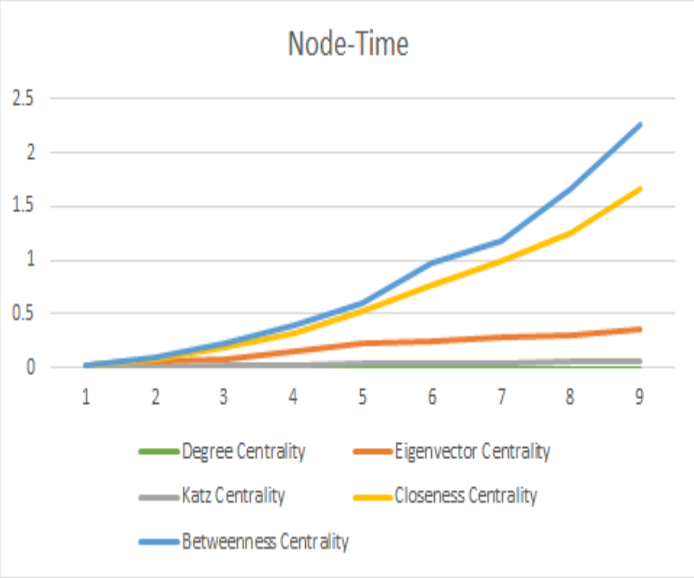
The codes for this experiment were written in Python. A Python library called Networkx was also used for this experiment.

{node,edge}	{100, 130}, {200, 260}, {300, 390}, {400, 520}, {500, 650}, {600, 780}, {700, 910}, {800, 1040}, {900, 1170}
methods	degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, katz centrality

<Table 3>

We used synthetic graphs with 100, 200, 300, 400, 500, 600, 700, 800, and 900 nodes respectively. In each graph, the number of

total edges was determined by the equation G(n,e), e=1.3×n since doing so reveals top-k in the results of the various calculations in an ideal way. If the set of edges gets larger, the graph G won't be able to take the feature of diffusion network of disease into account completely. In contrast, if E is too small, then the calculated centralities won't be able to represent the nodes' influence.



<Fig 4>

Analysis

The amount of time required to calculate each type of centrality increases as the number of nodes increases since there are more nodes to take into calculation. The degree centrality almost stays the same throughout all number of nodes because getting degree centrality is done by reading the number of edges a node has, not calculating a ton of data.

Calculating betweenness centrality took the most time since getting betweenness centrality follows getting every shortest path in the graph, which takes a lot of time. The time it takes to calculate betweenness centrality and the time it takes to calculate closeness centrality increases exponentially because the time complexity of both betweenness centrality and closeness centrality is $O(n^2)$

IV. CONCLUSION

Out of all the methods for calculating centrality, betweenness centrality suits this diffusion network case based on the result of the experiment. Even though betweenness centrality takes a lot of time to calculate, the similarity between the nodes whose substantial influence is intuitively noticed and the most central nodes based on betweenness centrality is high.

V. FUTURE WORKS

Although the similarity between the intuitively most influential nodes and the nodes that have high betweenness centrality is high, centralities cannot return top-k most accurately. To get top-k in a more accurate way, methods other than centralities calculation need to be incorporated: bridge detection and community detection. Bridge detection and community detection traces the most influential nodes in each community of nodes, which helps us figure out the nodes that are influential the most in spreading the disease within a group of people. In other words, we would be able to deal with suppress diffusion more effectively.

VI. REFERENCE

1. X. Zhao, S. Guo, and Y. Wang, "The Node Influence Analysis in Social Networks Based on Structural Holes and Degree Centrality," in IEEE International COnference on CSE and IEEE International Conference on EUC, 2017, pp. 708-711.
2. L. Lv, K. Zhang, T. Zhang, D. Bardou, J. Zhang, and Y. Cai, "PageRank centrality for temporal networks," in Physics Letters A, 2019, pp. 1-8.
3. E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Computing classic closeness centrality, at scale," in COSN, 2014, pp. 37–50.Freeman, L.C., "A set of measures of centrality based on betweeness", Sociometry, Vol. 40, pages 35-41, 1977.
4. F. Cadini, E. Zio, and C. Petrescu, "Using centrality measures to rank the importance of the components of a complex network infrastructure," in CRITIS, 2008, pp. 155–167.
5. S. Gao, J. Ma, Z. Chen, G. Wang, C. Xing, "Ranking the spreading ability of nodes in complex networks based on local structure," in Physica A 403 (6), 2014, pp. 130–147.
6. G. Sabidussi, "The centrality index of a graph," in Psychometrika 31 (4), 1966, pp. 581–603.
7. L.C. Freeman, "Centrality in social networks conceptual clarification," in Soc. Netw. 1 (3), 2008, pp. 215–239.
8. M.E.J. Newman, "A measure of betweenness centrality based on random walks," in Soc. Netw. 27 (1), 2003, pp. 39–54.
9. P. Bonacich, "Power and centrality: a family of measures," in Am. J. Sociol. 92 (5), 1987, pp. 1170–1182.
10. L. Page, S. Brin, R. Motwani, T. Winograd, "The Pagerank Citation Ranking:Bringing Order to the Web," in Technical report, Stanford InfoLab, 1999, http://ilpubs.stanford.edu:8090/422/.