Analysis of Tesla Friends Network Based on Twitter Data

Muhao Chen
A20456889
mchen69@hawk.iit.edu

Abstract: Nowadays, social media data analysis is becoming more and more important, and valuable visualizations and algorithms is gradually appearing. This paper focuses on analysis of Tesla friends network through graph algorithms based on Twitter data. This paper crawl 130 nodes through Twitter API and visualize data with networkX. Then do some research about the indegree and outdegree distribution of nodes. Besides, analyze the directed graph's clustering, centrality and similarity with different algorithms to find the features of Tesla friends network.

1. Introduction

1.1. Twitter Data Analysis

More and more scientist start to research the social media data through different algorithms, because social media become more significant in our daily life. A common analysis method is to study the network of friends through graph algorithms, which is showed in this paper.

1.2. Main Libraries and Tools

tweepy: Twitter API to get data of nodes.
pandas, numpy: analyze and compute the data matrix.
networkx: visualize the graph and apply graph algorithms.
matplotlib and seaborn: visualize the graph and data.

2. Data Collection

2.1. Method of Collection

All data is crawled from Twitter through official API, tweepy. Tweepy is an official library which provides limited public information of Twitter's users, including users' name, friends, tweets, comments and so on. In this article, we have crawled users' screen name, friends and followers, through Algorithm 1.

Algorithm 1. Crawl Algorithm

```
import tweepy
# Get Authorization
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
# Get Information/Data
api.get_user(screen_name)
api.friends(screen_name,count = friends_count)
api.followers(screen_name,count = followers_count)
```

Finally, We got 130 users/nodes' information. There is an example showed in Table 1.

2.2. Find Friends and Followers

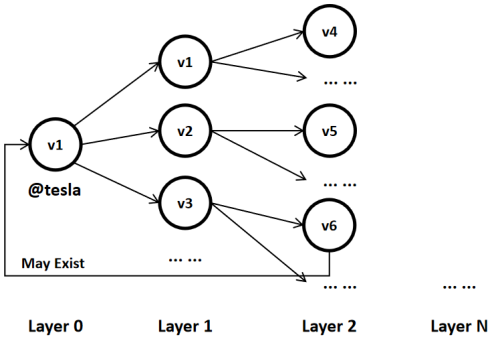
Friends and followers are the edges in the directed graph, which show the relation between the entities/nodes. In Twitter, it's easy to find the friends but difficult to search followers due to the limitation of access. In order to solve this problem, we only consider the outdegree of nodes, because it's possible that $vertex_A$'s pointed nodes could be the pointing nodes of $vertex_B$.

Tabela 1. Data Example

Screen Name	Friends	Followers
@Tesla	[@SpaceX,@elonmusk...]	[@TeslaCharging...]
@elonmusk	[@OpenAI,@teslacn...]	[@Tesla...]
@SpaceX	[@AstroBehnken,@NASA...]	[@elonmusk...]

Therefore, we set a initial node, @Tesla. Then find its' pointed nodes and repeat step by step, which showed in figure 1. In actual process, we select the pointed nodes, which connect as many other nodes as possible. Because, too many nodes easily lead to difficulties in visualization and too few nodes have greater randomness. Show in Figure 1.

Figura 1. Search Rules



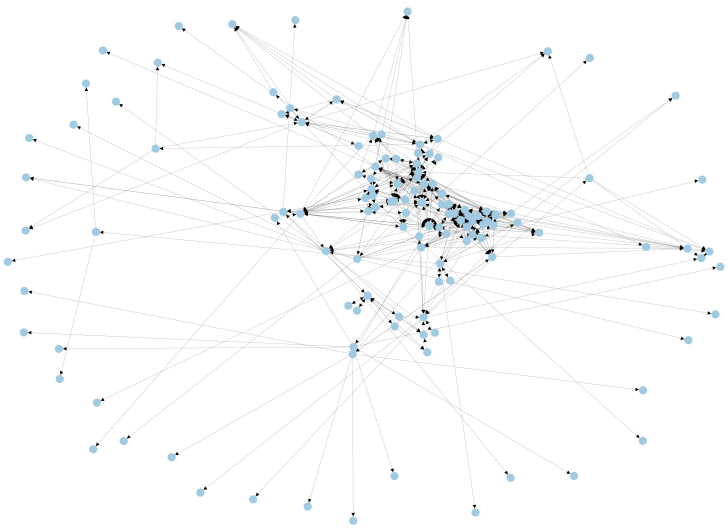
3. Data Visualization

3.1. Overall Visualization

In this sector, we use networkx package to visualize the data of nodes and edges. And we choose two visualization layouts:spring layout and shell layout.

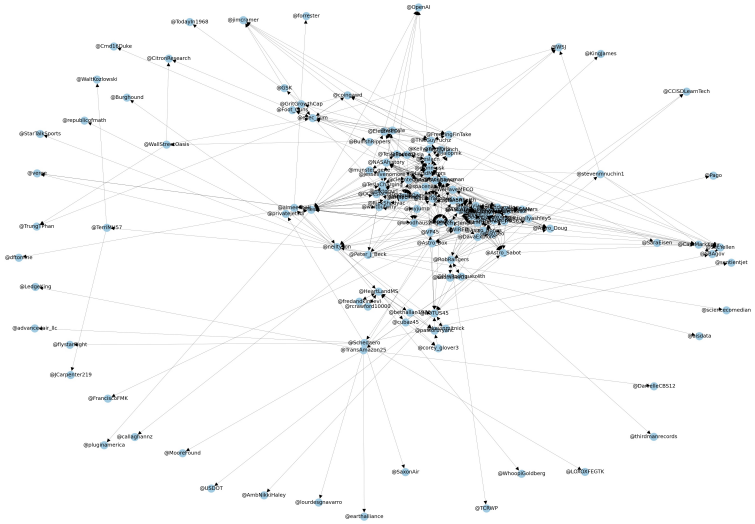
spring layout: Logically organize the locations of different nodes, which is convenient for us to learn the relation between nodes through different clusters. Show in Figure 2.

Figura 2. Spring Layout Visualization Without Labels



It’s obvious that there are one center node with several secondary center nodes, and most of nodes are far from the center nodes.”@Tesla”is the only center node, which connect the ”SpaceX”, ”TeslaCharging”and so on. These secondary center nodes connect astronauts, aviation bureaus and subsidiaries. Show in Figure 3.

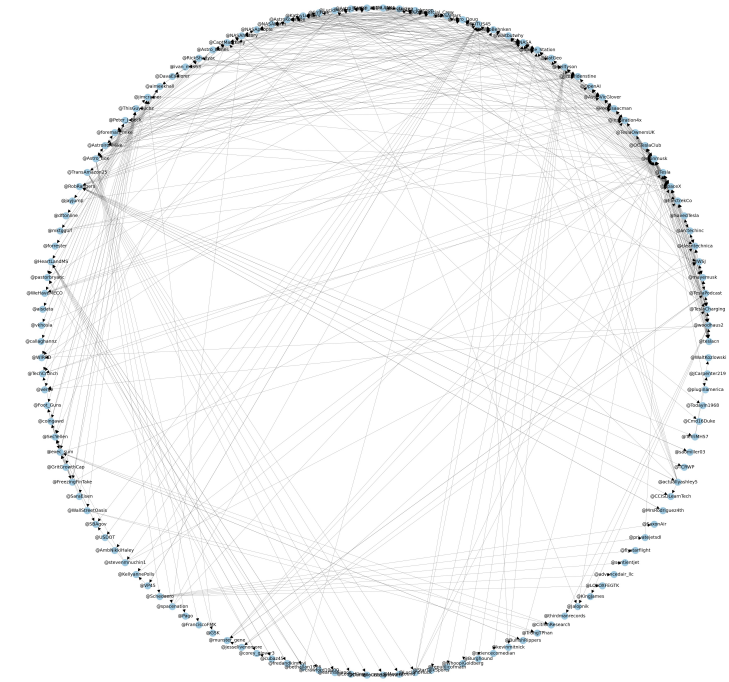
Figura 3. Spring Layout Visualization With Labels



shell layout: The layout clearly shows all nodes and the relationship between each node.

Almost one third nodes has complex connection between each other and the other nodes only has one or two edges. Show in Figure 4.

Figura 4. Shell Layout Visualization With Labels

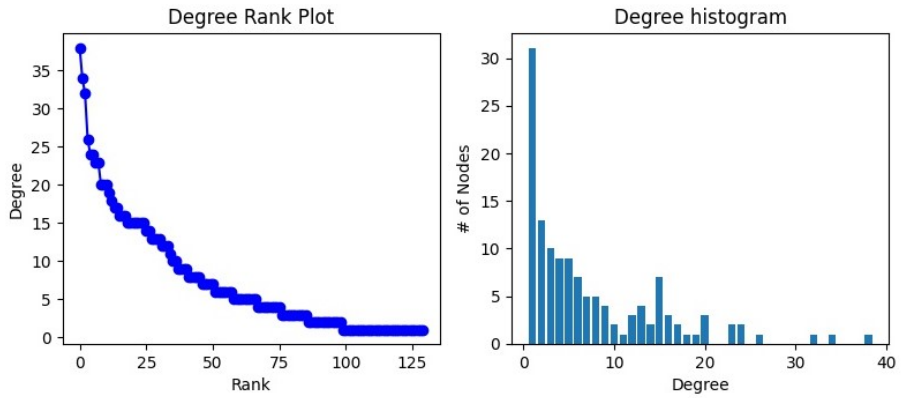


3.2. Degree Visualization

It's difficult to find the situation of edges from spring or shell layout visualization. Therefore, we use degree graph to show the overall situation of the edges.

Most of the nodes have a degree less than 10, and a small number of nodes even exceed 30. Among them, there are close to 30 nodes with degree 1, which proves that the circle of friends of most Twitter users do not overlap. Show in Figure 5.

Figura 5. Degree Rank Plot And Histogram

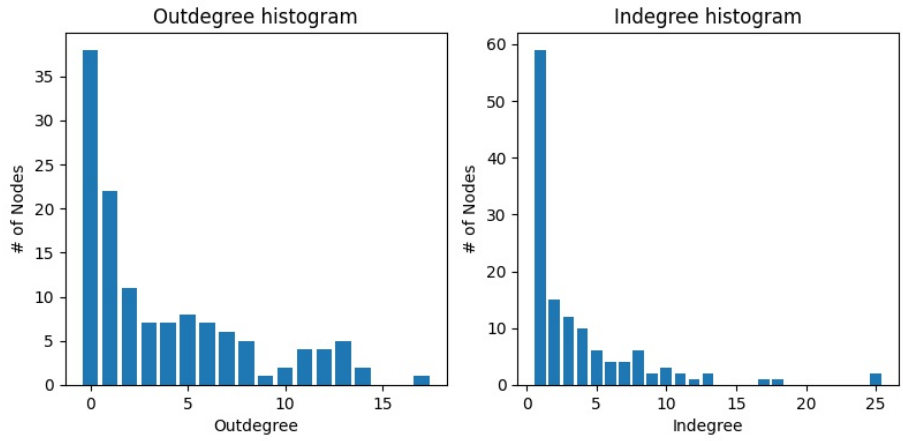


4. Network Measures Calculation

4.1. Indegree and OutDegree

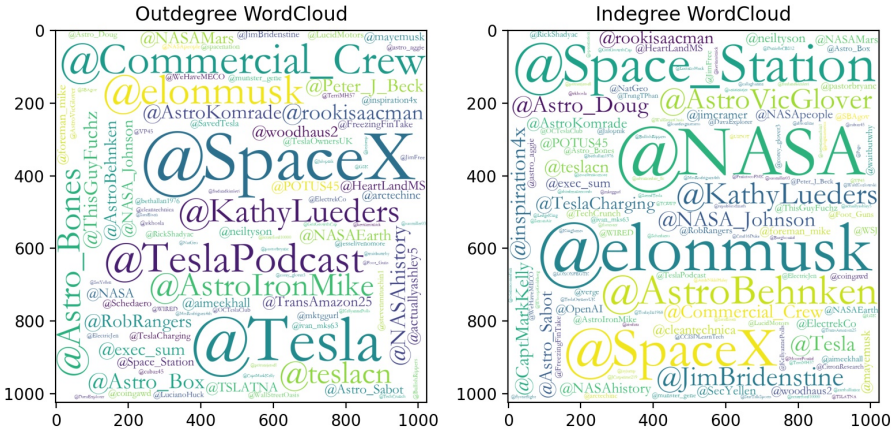
Because Tweeter friendship graph is directed, the degree of nodes consist of outdegree and indegree. Firstly, we generated the adjacency matrix from graph and computed the outdegree and indegree of nodes. Finally, we found that most of nodes have higher indegree than outdegree. Besides, The sum of indegree or outdegree is 488. The average of indegree or outdegree is 3.75. Show in Figure 6.

Figura 6. Outdegree and Indegree Histogram



In order to know which node has higher indegree than outdegree, we use WordCloud library to show the degree of different nodes. From the angle of outdegree, "@SpaceX", "@Tesla", "@TeslaPodcast"and "Commercial Crew"have higher outdegree. From the angle of indegree, "@NASA", "@elonmusk", "@Space Station"and "SpaceX"have higher indegree. Show in Figure 7.

Figura 7. Outdegree and Indegree WordCloud



4.2. *Clustering Coefficient*

Obviously, the Tesla Friends Network is a directed graph, which not applies to the global clustering coefficient calculation. However, through *Clustering in complex directed networks* written by *G. Fagiolo* in 2007, the local clustering is similarly defined as the fraction of all possible directed triangles or geometric average of the subgraph edge weights for directed graph respectively.

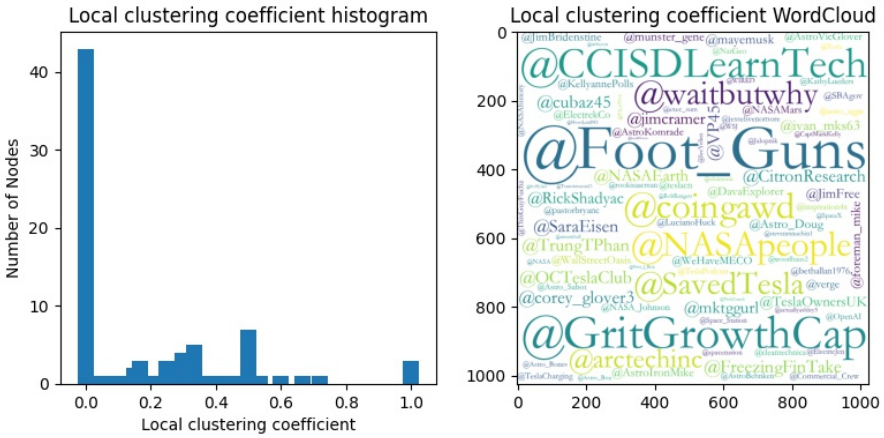
The formula for calculating directed graph's loacal clustering coefficient is:

$$c_u = \frac{2}{deg^{tot}(u)(deg^{tot}(u)-1)-2deg^{\leftrightarrow}(u)} T(u) \tag{1}$$

where $T(u)$ is the number of directed triangles through node u,$deg^{tot}(u)$ is the sum of in degree and out degree of u and $deg^{\leftrightarrow}(u)$ is the reciprocal degree of u.

Finally, we get the histogram and WordCloud of local clustering coefficient through library networkx. Show in Figure 8.

Figura 8. Histogram and WordCloud of Local Clustering Coefficient



In result, we found that most of nodes have zero local clustering coefficients and only a few nodes' local clustering coefficients are higher than 0.6. Besides, the average of local clustering coefficients is 0.21. Through analysis, we believed that the directed graph of the Tesla friend network is extremely discrete. Only a few neighbors of nodes are connected, which proved that the connection between "@Tesla"'s friends' relationship is alienated.

4.3. Centrality

4.3.1. indegree centrality and outdegree centrality

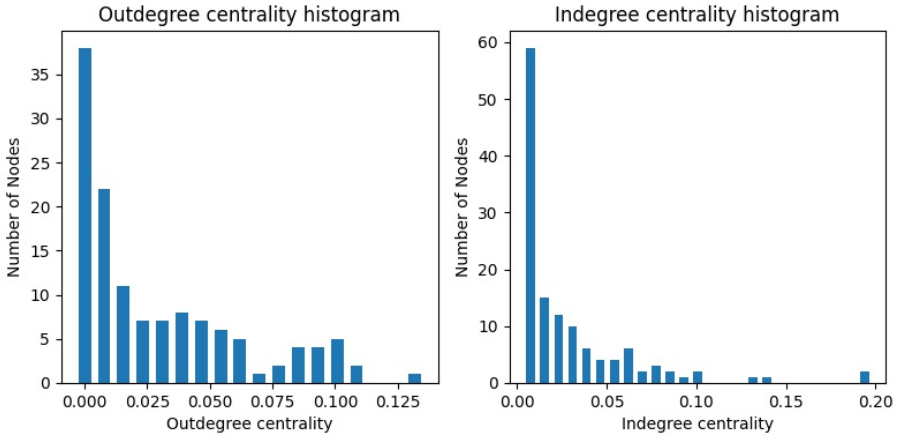
In directed graph, we usually calculate indegree centrality and outdegree centrality which show the significance and the degree of centrality of nodes. Different from the degree distribution, normalization is necessary, preventing the influence of the maximum probability of the edges. We use the following formulas to calculate:

$$C_j^{OUT} = \frac{1}{n-1}\sum_{i=1}^{n} a_{ij} \tag{2}$$

$$C_j^{IN} = \frac{1}{n-1}\sum_{i=1}^{n} a_{ij} \tag{3}$$

In result, similar to the 4.1 conclusion, nodes of graph have higher centrality of indegree than outdegree. But there is fewer differences of centrality of outdegree between nodes. Show in Figure 9.

Figura 9. Indegree Centrality and Outdegree Centrality Histogram



4.3.2. Katz Centrality

Katz centrality computes the centrality for a node based on the centrality of its neighbors. The magic of Katz centrality algorithm is that using unknow centrality value of nodes' neighbors to estimate the unknow centrality value of nodes. Before we apply Katz centrality algorithm, we should calculate the eigenvector centrality. The Katz centrality formula is:

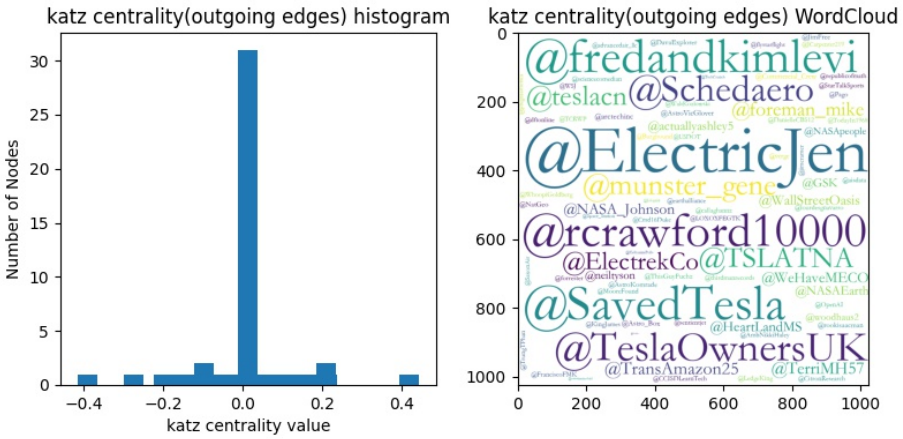
$$C_{Katz}(v_i) = \alpha \sum_{j=1}^{n} A_{j,i} C_{Katz}(v_j) + \beta \tag{4}$$

Where α and β are hyperparameters. α should be smaller than reciprocal of maximum of lambda/eigenvector maxtrix. And β is a bias parameter. After calculation, $\lambda_{\max}$ is 0.337. Therefore, α should smaller than 2.97.

$$\alpha < \frac{1}{\lambda_{\max}} = 2.97 \tag{5}$$

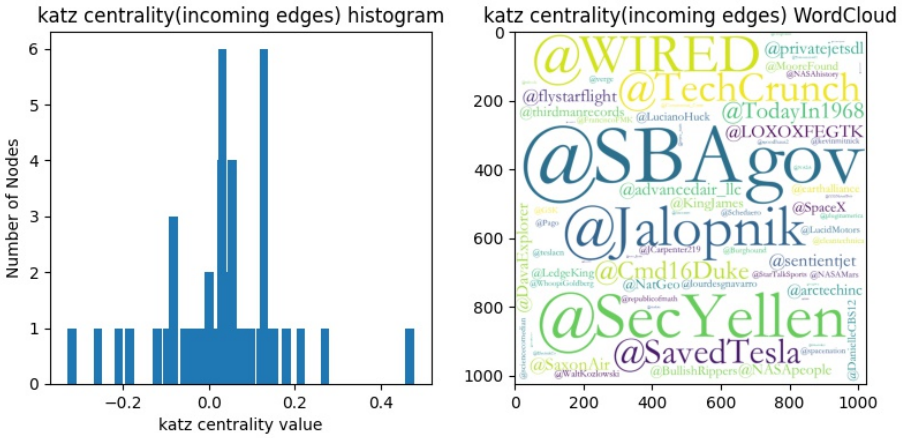
Initially, we set $\alpha = 2.9$ and $\beta = 1.0$. Then use Katz centrality algorithm to get the results of outgoing edges and incoming edges. The centrality of nodes of outgoing edges show in Figure 10. The centrality of nodes of incoming edges show in Figure 11.

Figura 10. Histogram and WordCloud of katz centrality(outgoing edges)



After calculating outgoing edges, we found that most of nodes are less significant, which shows that there are some marginalized node without outgoing edges. Surprisingly, the "@ElectricJen"and "SavedTesla"have higher centrality value. Through analysis, these important nodes connect with lots of nodes. Although these nodes are not initialized node, such as "@Tesla".

Figura 11. Histogram and WordCloud of katz centrality(incoming edges)



After calculating incoming edges, we found that the result is different from the outgoing edges but reasonable actually. For example, "@SBAgov"is a platform empowering small businesses to start, grow, expand or recover. "@Jalopnik"is a car platform. The common feature is that the nodes have plenty of followers, nearly one million followers.

4.3.3. Pagerank

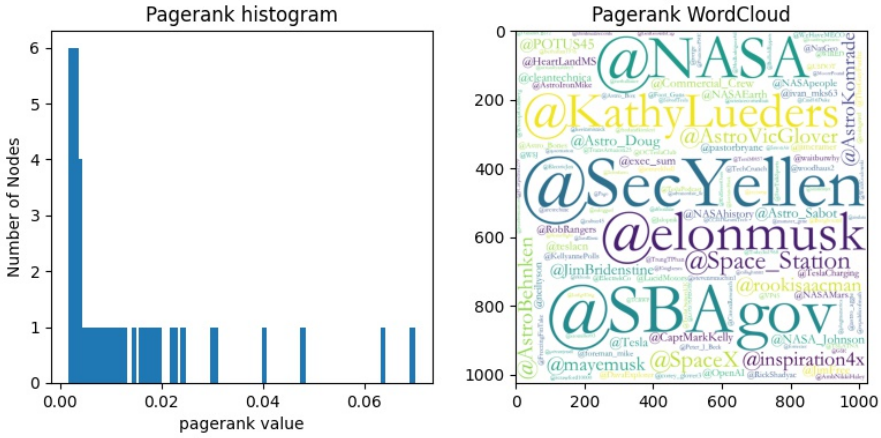
PageRank is a link analysis algorithm, which has similar function to Katz Centrality, computing the significance of nodes in another way. Initially, the node is assigned a equal weight. And the weight would be changed due to the different indegree. After many iterations, the weight of each node gradually converges to a relatively fixed value. Practically, different from content taught in class, there is a damping factor α, because the theory believe that the relation will decrease with continuous steps. The PageRank formula is:

$$PR(A) = 1 - \alpha + \alpha(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}) \tag{6}$$

In the experiment, we set $\alpha = 0.85$ and $maxiter = 100$. In result, we found the similar result, that most nodes have a small PageRank value and only a few nodes' values are higher than 0.04. The nodes "@NASA", "@SBA-

gov"and "@elonmusk"are more significant. Because the distance between these nodes is shorter, the transferred weights are larger.

Figura 12. Histogram and WordCloud of PageRank



4.4. *Closeness*

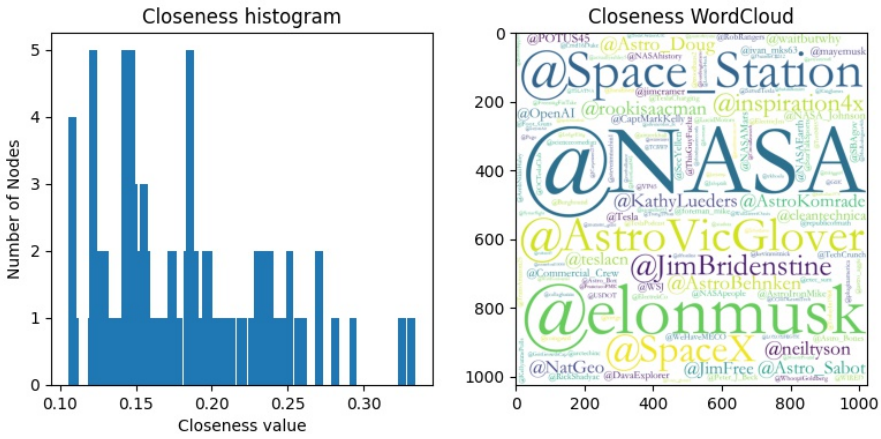
In directed graph, we can calculate the closeness to research the centrality based on the shortest path. Through *Social Network Analysis: Methods and Applications* written by *Wasserman and Faust* in 1994, we use an improved formula to calculate closeness of nodes:

$$C_{WF}(u) = \frac{n-1}{N-1}\frac{n-1}{\sum_{v=1}^{n-1} d(v,u)} \tag{7}$$

Where $d(v,u)$ is the shortest path from $vertex_v$ to $vertex_u$, $n-1$ is the number of nodes that can reach u.

Through calculation, it's obvious that "@NASA"and "@elonmusk"have higher value of closeness, which means that plenty of nodes could easily and quickly reach them. Therefore, in practice, we could firstly find these nodes with high value of closeness, and it's easy for us to reach other nodes. Show in Figure 13.

Figura 13. Histogram and WordCloud of Closeness



4.5. *Simrank Similarity*

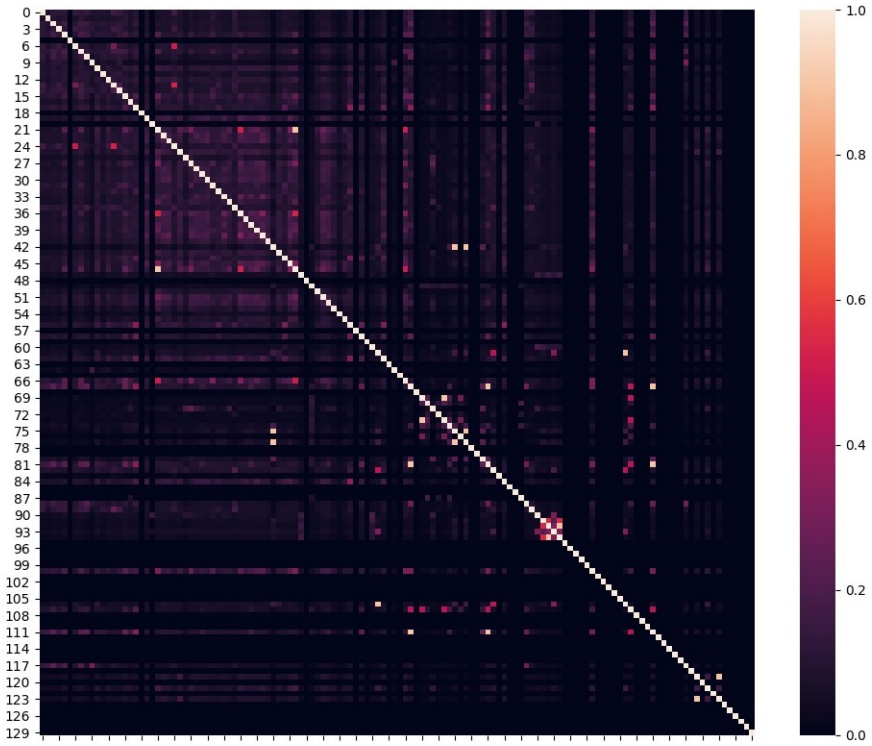
Similarity computation is a common algorithm in the field of data analysis. In this sector, we apply simrank similarity algorithm to analyze each node's similarity to other nodes.

Firstly, denote the similarity between objects a and b by $s(a,b)$. If $a = b$ then $s(a,b)$ is defined to be 1. Otherwise,

$$s(a,b) = \frac{C}{|I(a)|\,|I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \tag{8}$$

After many iterations, finally get the similarity matrix. Then visualize the similarity matrix with Heat Map. We found that most of the area is close to black, but a few area is red. Therefore, most nodes have low similarity except a few nodes,which have strong correlation. Show in Figure 14.

Figura 14. Similarity Heat Map



Referências

1. G. Fagiolo, *Clustering in complex directed networks* (Physical Review E, 76(2), 026107, 2007).
2. Leo Katz, *A New Status Index Derived from Sociometric Index* (Psychometrika 18(1):39–43, 1953).
3. pg. 201 of Wasserman, S. and Faust, K., *Social Network Analysis: Methods and Applications* (Cambridge University Press, 1994).
4. G. Jeh and J. Widom, S. and Faust, K., *SimRank: a measure of structural-context similarity* (International Conference on Knowledge Discovery and Data Mining, pp. 538–543. ACM Press,, 2002).
5. The document of networkx
   https://networkx.org/documentation/stable/reference/introduction.html
6. The document of tweepy
   https://docs.tweepy.org/en/stable/
7. Libraries: tweepy, pandas, numpy, networkx, matplotlib and seaborn.