Network Analysis of Wikipedia Dataset with Spark

Ng Ka Fong Tang Ka Fu

1. Abstract

Networks are omnipresent in real life, from traffic nets, social relations and interactions to natural biology. In wikipedia, articles are joined with hyperlinks, forming a knowledge network. Understanding this type of network gives us insights into how knowledge is correlated, how it is spread and how to enhance learning.

Real world networks are large in size while graph algorithms are often computationally difficult. As an example, finding all shortest paths takes $O(n^3)$ by Floyd-Warshall algorithm and this algorithm is operated in a sequential manner, hence lacks scalability. Utilizing parallelism in Spark is a way to mitigate the scalability issue of graph search algorithms.

In this project, we will study the statistics of a specified wikipedia topic network using Spark. We will investigate three types of nodes centrality - degree centrality, closeness centrality, PageRank centrality and clustering coefficient. To investigate the latter three require a parallel BFS algorithm, a Spark PageRank algorithm and a parallel clustering coefficient algorithm respectively.

2. Background

2.1 The Dataset

In this project, due to the environment constraint as listed in 2.2, we decided to work on a specified topic dataset instead of a global one.

The dataset is obtained from SNAP [2], collected by the MUSEA project [2]. The datasets represent page-page networks on specific topics (chameleons, crocodiles and squirrels), in our project, we specifically work on the squirrels dataset. It contains 5201 nodes, 217073 undirected edges. Monthly average traffic of each node is also provided.

2.2 The Environment

This project is run on a cluster provided by databricks community edition, with runtime version 8.1. The cluster provided contains 15.3GB of memory, 2 CPU cores with 8 threads.

3. Methods and Results

3.1 Analysis of Node Traffic

Average monthly traffic of each article is provided in target.csv. As provided in Part 1.2 of ***network_analysis_deg.ipynb***, we group the traffic values according to their logarithmic value

in intervals of 1 decimal place, and count the number of nodes in each group. The counting suggested that most of the values lie center on logarithmic scale. The frequency count is converted to probabilistic value (pdf) and is converted to cumulative probabilistic value (cdf) by using a cumulative sum of pdf after sorting.

We obtain the median (3.96) and the standard deviation (0.767) of the logarithmic traffic data. According to non-parametric statistics, we use sample median instead of sample mean as a better estimator of the population mean. As provided in ***Figure 1,*** we fit the traffic of each article into a log-normal distribution with mean $10^{3.96}$ and standard deviation $10^{0.767}$.

$$T \sim log_{10}Norm(3.96,\ 0.767)$$

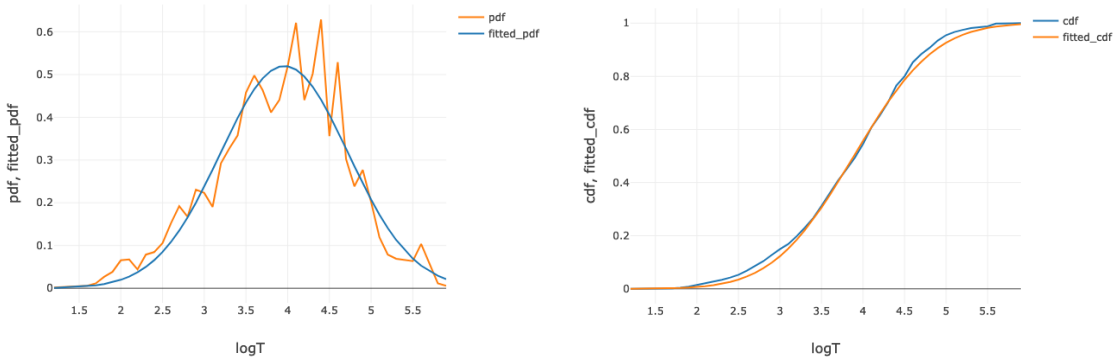


Figure 1) actual and fitted distribution of node traffic in probabilistic (left) and cumulative (right)

In fact, traffic following a log-normal distribution deviates from our expectation that it would be power-law distributed. This means that most articles actually have medium traffic size, with only minorities having comparatively small or large traffic. The underlying meaning might be articles of minority topics are never created, hence articles with small traffic are rare.

3.2 Analysis of Node Degree Centrality

As provided in Part 1.1 of ***network_analysis_deg.ipynb***, we reformat the data from edge list form to adjacency list format. The total degree of each node is hence the length of each adjacency list. Likewise in 3.1, acquiring the uneven scatter plot drives us to put the degree data into logarithmic scale.

Retrieving the probabilistic and cumulative distribution of the degree count this time gives us a heavily left-sided distribution. As shown by blue points in Figure 2.1), plotting pdf against degree on a log-log scale yield a typical pattern of power law with exponential cutoff, $p(x) \propto x^{-\alpha} e^{-\lambda x}$.

In order to determine the power-law degree exponent α, we perform a linear regression on the Complementary CDF (CCDF), $P(X \geq x)$. Estimating α from pdf or ccdf are both feasible, however, the latter is monotonous and more resilient to error due to fluctuation. The linear regression is only performed on the linear point where the power law term dominates over the cutoff term.

Result of Linear regression is $log(1 - P(X > x)) = 0.146 - 0.393\, log(x)$with Root Mean Squared Error (rmse) at 0.03 and r2 at 0.97.

Therefore, Slope of ccdf $\alpha' = 0.393$. By integration, if $p(x) \propto x^{-\alpha}$, $P(X > x) \propto x^{-(\alpha-1)}$, so $\alpha = 1 + \alpha'$. This gives us power-law exponent $\alpha = 1.393$. Calculate the x-intercept gives us $log(x) = 0.37$, hence x > 2 is the valid range of this power law distribution.

After estimating α, estimating slope of $\frac{P(X>x)}{x^{-1.393}}$ on semi-log scale yield λ=0.0029. Hence, we fit the distribution of node degree as shown in Figure2 by

$$p(x) \propto x^{-1.393} e^{-0.0029\, x}$$

where $x > 2$ is the total degree of a node.

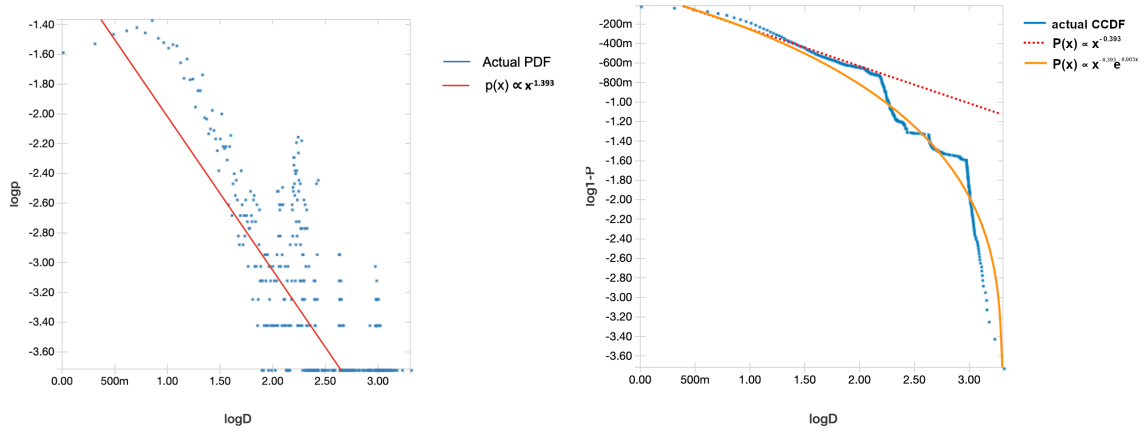


Figure 2) actual and fitted distribution of node degree in probabilistic (left) and cumulative (right)

Notice p(x) is a scale-free function, implying the network is a ***scale-free network***. In a scale-free network, the network pattern of a sampled smaller network would be similar to a global network. Considering this squirrel network as a sample of the global Wikipedia article network, this discovery would suggest our finding $p(x) \propto x^{-\alpha} e^{-\lambda x}$ is potentially applicable to the global network as well.

3.3 Analysis of Node Closeness Centrality

To determine the distance between nodes, we implement a parallel BFS algorithm as shown in ***network_analysis_closeness.ipynb.*** In this program, we use a distance array to represent the shortest distance of a node to any other node. For each BFS operation, the map operation is done with array addition and the reduction operation is done by element-wise minimum operation.

We also implement the landmarks approach for quick estimation and a procedure to convert the distance to landmarks to distance to all nodes. Using nodes with top 100 total degrees as landmarks, Table below is a comparison of the performance of the two.

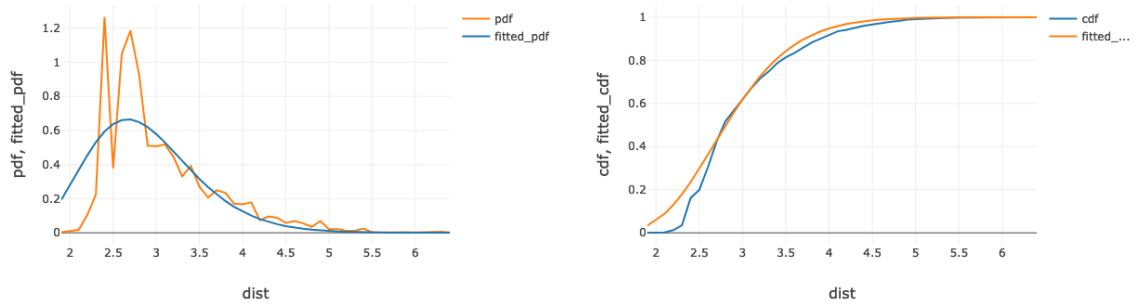
	Parallel BFS on landmarks (15 steps)	Parallel BFS on all nodes (15 steps)

Time	1.7 min (BFS) + 5.7 min (conversion)	5 min
Error	mean: +0.215 sd: 0.42	--

The network is found to be a large connected component with network diameter 12, so both approaches are able to cover the distance of all node-pairs. Based on the fact that error of the landmark estimation is quite small, we would conclude the choice of landmarks is effective.

Analysing the distribution of mean distance of one node to any other nodes yields us a right-skewed distribution. The fitting in Figure3) is a gamma-distribution

$$D \sim Gamma(\alpha = 8.9, \beta = 4.74, min = 1)$$



3.4 Analysis of Node PageRank Centrality

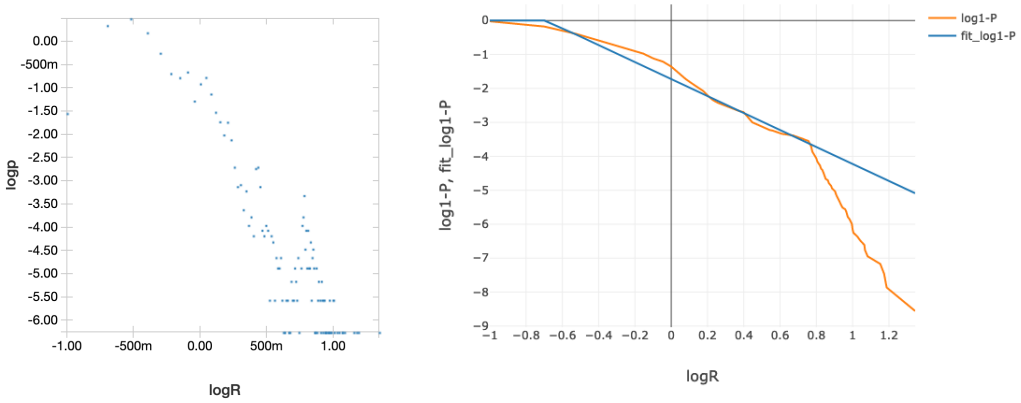
PageRank is a link-analysis algorithm and it has the ability to capture important nodes of the network. To determine PageRank centrality of each node, we implement a parallel PageRank algorithm in ***network_analysis_PageRank.ipynb***. It is efficient in a sense that it provide more dynamic information than degree or closeness centrality and quite quick in Spark (~ 1.85 minutes for 100 iterations)

Analysing the distribution of PageRank weighting distribution gives us a typical power law distribution Likewise in 3.2, we perform linear regression to obtain, $\alpha' = 2.50$, $\alpha = 3.50$, $x > 0.205$.

Hence, we fit the distribution of node degree as shown in Figure2 by

$$p(x) \propto x^{-3.50}$$

where $x > 0.2$ is the total pageRank of a node.



3.5 Analysis of Clustering Coefficient

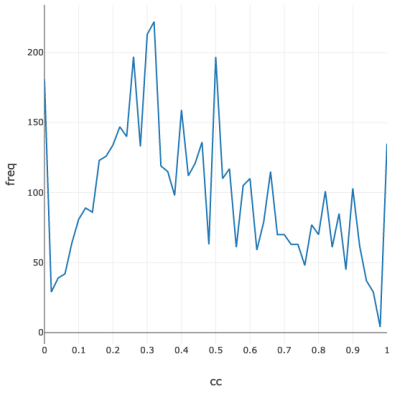
clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. It is defined as the proportion of number of links between neighbors of a node divided by the maximum possible number of links between them.

$$C_i = \frac{2e_i}{k_i(k_i - 1)}$$

where k_i is the degree of node i, ei is the number of edges between neighbors of i.

In this project, we implement own own clustering coefficient algorithm can be seen in **network_analysis_clustering_coefficient.ipynb**

Investigating the distribution of clustering coefficient, except many nodes have 0 or 1 clustering coefficient, there are roughly more nodes with low clustering efficient.



4. Conclusion

In this project, we have examined the different nodes metadata. We utilize Spark and python statistics to provide fitting to the metadata - network traffic is a log-normal distributed, node degree and PageRank centrality are power-law distributed while for node-distance, we map it by a gamma distribution.

In order to find the power-law distribution parameters, we utilize the linear regression module by SparkML on a log-scale or a semi-log scale graph and utilize satisfactory results. Computationally, setting partition number to be 1x~3x of the total number of threads is optimal for high CPU utilization and low partition creation overhead.

We have examined the landmark approach of node distance estimation and reached an effective result - error is acceptable while graph searching time is 3x faster. Through implementing and running a clustering coefficient algorithm, we also discover this Wikipedia 'squirrel' topic network being quite sparse.

[1] *Wikipedia Article Networks*. SNAP. (n.d.). https://snap.stanford.edu/data/wikipedia-article-networks.html.

[2] B. Rozemberczki, C. Allen and R. Sarkar. *Multi-scale Attributed Node Embedding.* 2019.