Big Data Analytics Project

Option A

Analyzing ACM Citation Network

Manish Patil

Rohit Tatar

**INDEX**

1. **Introduction**
2. **Methodology** 
   1. **Building the Citation network**
   2. **Visualizing In-Degree Distribution**
   3. **Implementing Weighted Page Rank**
   4. **Calculate clustering coefficient and average path length.**
3. **Results**
4. **Conclusion**
5. **References**

**INTRODUCTION**

The project processes and analyses the ACM Citation Network dataset. As the name suggests, our record contains index and references for the papers shown in the ACM Library. Initially, we reduce the ACM citation file into two columns which represent the main paper in the first column and cited paper in the second column. Second task allows visualization of graph with in-degree distribution of the ACM Citation network. Third task is to implement Weighted Page-Rank Algorithm, on the data produced in our previous step. The goal is to return the top 10 papers which have been cited the most in the dataset. Last step of our project is to calculate average clustering coefficient and average path length and finally compare them with random graphs.

**Methodology**

The source data for our project was the ACM Citation Network file called ACM-Citation-Network-v8.txt found. (<https://aminer.org/billboard/citation>)

**Step 1:** **Building the Citation network using MapReduce**

* Configure MapReduce Driver class to a customizable record delimiter.

Configuration conf = getConf();

conf.set(“textinputformat.record.delimeter”,”#\*”);

Job job = new Job(getConf());

* It enables us to create input split between “#\*”s.
* We create two regular expressions to filter indexes and citations.
* After successful match emit index as key and citations as value.
* Filtering and shuffling will be done in MapReduce execution stage between map function and reduce function.
* Only returns those papers which have citation.
* The key and value pair sent from mapper to the reducer will finally be combined for all values in dataset and emitted in tab separated format.

**Step 2: Visualizing In-Degree Distribution**

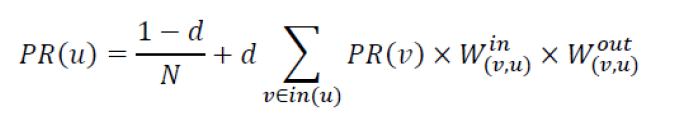
In our second step, our main objective is to produce an in-degree distribution graph from the data produced in the first step. The number of other papers which have cited our paper represented by in-degree for our paper.

* For our data, the in-degree for a particular paper would be the number of papers that have cited it .
* The in-degree distribution is number of papers with same in-degree divided by total number of papers.
* We use the file generated in the previous step to generate the graph using GraphX in Spark.
* The output links generated are in a String format and GraphX does not support Strings to be used as Nodes. Links are then converted into Long format, which is compatible with the GraphX package [2] using zipwithindex.
* Calculate the in-degree of each paper. First, we initialize each indegree with 1 and sum all the same indegree to calculate indegree distribution.

**Step 3: Implementing Weighted Page Rank Algorithm**

Implementation of Weighted Page-Rank Algorithm returns the top 10 papers with the highest page rank in our dataset. Weight is assigned to each node to imply which node is important according to their weight.

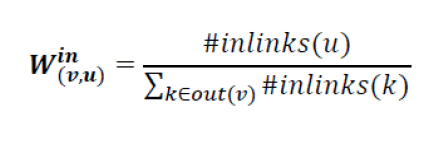
The Weighted page rank:



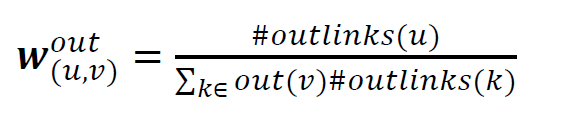
N=Total number of nodes

In(u) is indegree calculated for node u.

Win is an in-weight of link(v, u). It is calculated as the ratio of the number of in-links (incoming links) to node u over the number of in-links to all references of node v [4].



Wout is out-weight of link(v, u) . It is calculated as the ratio of the number of out-links (outgoing links) from node u and the number of out-links from all references of node v [4]:



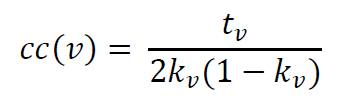
**Step 4:** **Finding Average Clustering Coefficient and Average Path Length of the network and compare it to a random graph**

Many real-world networks including social networks of a type of graph called Small-World networks. Small-world networks have two main properties:

1. Most of the nodes in the network have a small degree which means most of the nodes are not neighbors of one another, but for every given node its neighbors are also neighbors of each other. This property is typically measured using Average Clustering Coefficient of the network [5].

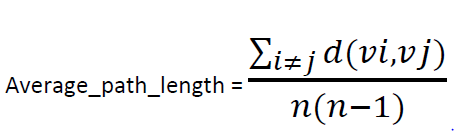
2. Despite the small degree and dense clustering of most nodes in the graph, it is possible to reach any node in the network from any other network relatively quickly by traversing a small number of edges which measured by Average Shortest Path Length of the network [5].

The local clustering coefficient of a node v in a graph is defined as follows:



* Used GraphX API to calculate triangle count and degree for each vertex in graph.
* Further input the calculated values in the formula to calculate coefficients for each vertex.
* After, we calculate the average of the coefficients.

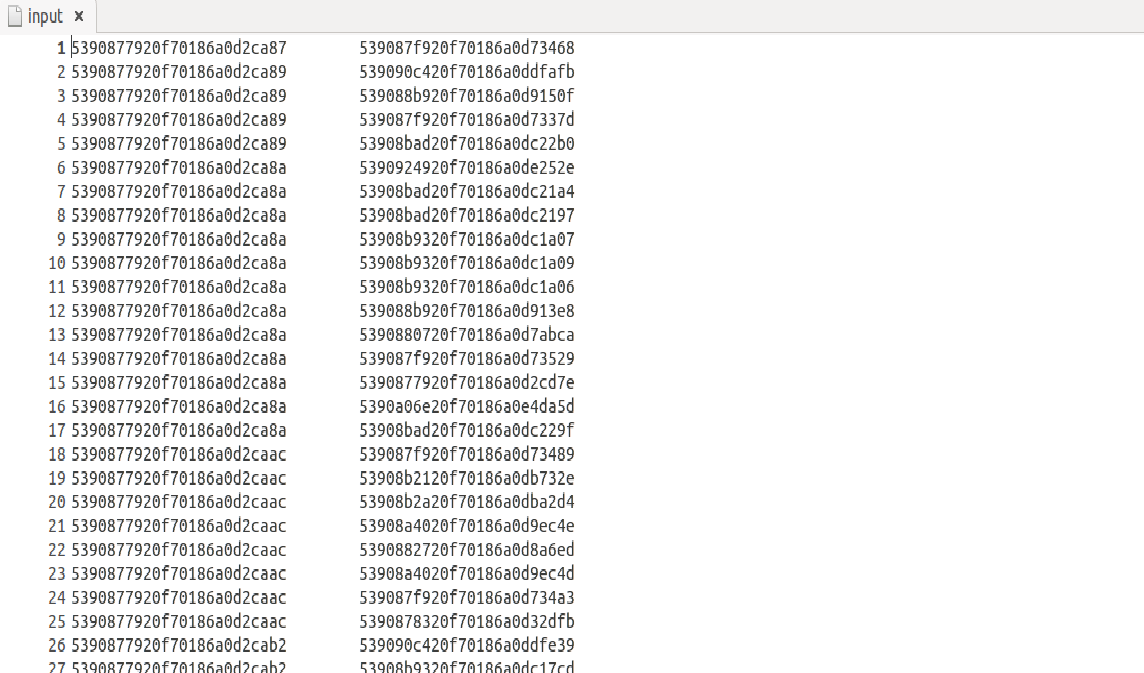
The average path length of a graph is the average length of the shortest path between every two nodes in the graph and is computed as follows

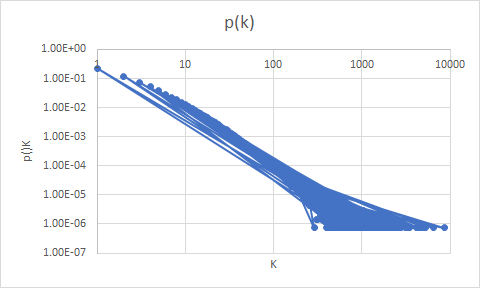


* Using the ShortestPath function from the GraphX library, we get the distance between the two vertices.
* Further we take the summation of the distances and divide it with n(n-1) where n is num of vertices.

**Results**

**Step 1: Index and paper cited are separated by tab**

****

**Step2: Visualizing the in-degree distribution of graph.**

**Step 4:**

Comparing coefficients and shortest path for random graphs and graph generated in the previous steps

1. Graph generated in the previous step:

Avg Clustering coefficient: -0.138

Avg Path Length: 1.33

1. Random Graph with n = 20

Avg Clustering coefficient: -0.057

Avg Path Length: 0.02

1. Random Graph with n = 40

Avg Clustering coefficient: -0.067

Avg Path Length: 0.05

Hence, the graph generated from ACM citation networks exhibits Small-world network.

**Conclusion**

1. MapReduce used to filter index and citations.
2. Using GraphX API, plotted the graph for indegree distribution.
3. The Weighted Page Rank algorithm provides with the top ten most influential papers.
4. Clustering coefficient and shortest path of in degree distribution graph and random graph enables to then the ACM citation graph exhibits small world network properties.

**References:**

[1] Scala Cookbook

[2] [GraphX programming Guide](http://spark.apache.org/docs/latest/graphx-programming-guide.html)

[4] W. Xing and A. Ghorbani, "Weighted PageRank algorithm," Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on, 2004, pp. 305-314.

[5] D. J. Watts and Steven Strogatz, "Collective dynamics of 'small-world' networks". Nature. 393(6684), 1998, PP.440–442.