Identification and ranking of key persons in a Social Networking Website using Hadoop & Big Data Analytics

Prerna Agarwal
Computer Engineering
Jamia Millia Islamia
+918285003430
prerna.jmi@gmail.com

Rafeeq Ahmed
Computer Engineering
Jamia Millia Islamia
+918750370087
rafeeq.amu@gmail.com

Tanvir Ahmad
Computer Engineering
Jamia Millia Islamia
+919811239735
tahmad2@jmi.ac.in

ABSTRACT

Big Data is a term which defines a vast amount of structured and unstructured data which is challenging to process because of its large size, using traditional algorithms and lack of high speed processing techniques. Now a days, vast amount of digital data is being gathered from many important areas, including social networking websites like Facebook and Twitter. It is important for us to mine this big data for analysis purpose. One important analysis in this domain is to find key nodes in a social graph which can be the major information spreader. Node centrality measures can be used in many graph applications such as searching and ranking of nodes. Traditional centrality algorithms fail on such huge graphs therefore it is difficult to use these algorithms on big graphs. Traditional centrality algorithms such as degree centrality, betweenness centrality and closeness centrality were not designed for such large data. In this paper, we calculate centrality measures for big graphs having huge number of edges and nodes by parallelizing traditional centrality algorithms so that they can be used in an efficient way when the size of graph grows. We use MapReduce and Hadoop to implement these algorithms for parallel and distributed data processing. We present results and anomalies of these algorithms and also show the comparative processing time taken on normal systems and on Hadoop systems.

General Terms

Graph Theory

Keywords

Betweenness Centrality, Closeness Centrality, Degree Centrality Big Data, MapReduce, key persons, ranking.

1. INTRODUCTION

There have been a lot of changes in the way internet has been used in these days. Due to rise in popularity of social networking websites and availability of huge amount of generated data, there is a great opportunity to mine and extract information from social networks.

A social network consists of nodes and their interconnections. It involves nodes that exist together as a linked community or a social networking platform in which every day people can distribute their opinions. A significant feature of a social network is its capability to connect people through direct links or indirect links. A social network is very dynamic, on a daily basis thousands of novel links are created and destroyed. Thus the relationship between people are temporary. The vast space of social network is the basis to extract facts and trends about the nature of relationships and connections.

Data mining targets to find these kind of hidden, previously unidentified, and potentially useful information from data [2]. Maximum of the key person extraction procedures relay on many centrality measures, i.e. local (inside the social groups) or global (for the complete network) structural features [12].

Big Data is further called as Data Intensive Technologies, a fresh technology in science, industry and business [24, 25, 26].

Big Data is not just a Hadoop problem, it is the mechanism to store, process, visualize and bring outcomes to required applications. Big Data is a fuel for data processing, source, target, and conclusion [27].

When we move from traditional database to distributed databases the following issues occur:

- Maintaining meaning and integrity for data provenance
- Data security from unauthorized access
- Data ownership and privacy

Measuring centrality of nodes requires computation which is possible in small graphs but it becomes expensive as the graphs becomes large. For example, betweenness centrality requires computation of shortest paths between all pairs of vertices. It is possible in small graphs but it becomes time consuming as the graph grows.

There is no easy way to decrease the computational complexity of centrality algorithms. Traditional methods are very expensive for graphs with millions of nodes, making it impossible to use in practical application scenarios.

The situation gets even worse when we consider real network graphs, such as social networks which are dynamic in nature and change rapidly with new nodes arriving, and old nodes being removed.

2. PREVIOUS WORK

Hüseyin Oktay and A. Soner Balkir have worked on distance calculation in big network [1]. In their paper they have used Distance calculation as key to discover several network mining applications for example, centrality and clustering using the MapReduce parallel processing framework to powerfully and

AICTC'16, August 12–13, 2016, Bikaner, India.
© 2016 ACM. ISBN 978-1-4503-4213-1/16/08$15.00.
DOI: http://dx.doi.org/10.1145/2979779.2979844

precisely evaluation of the distance for very huge network by suggesting a network structure index (NSI) on MapReduce framework supposing that networks are undirected and are unweighted.

U Kang and Spiros Papadimitriou have done observations of centrality algorithms in their paper [9] calculated node centrality for very huge graphs, for millions of nodes and edges using centrality measures, as well as reachable means to efficiently calculate them by suggesting effective closeness and LINERANK which are projected for billion-scale graphs.

Behnam Hajian [3] in his work has analyzed a network based on the connections between nodes called as behavioral analysis. The hypothesis discovered in his work is that the rank based on behavior can be measured. He has defined Influence Rank for a node as "the average Influence Rank of its neighborhoods combined with another index called Magnitude of Influence"[3]. The correlation between the indices is analyzed. This collective measure is calculated by a formula whose complexity yields non-polynomial time.

After going through the above works done in previous researches, We identified that the past researches conducted have either proposed a new method to identify the importance of a node using map reduce framework. Moreover the focus was more on proposing a new method to identify centrality measure or node ranking. Less effort have been put on using the existing algorithms for a very large network in an efficient manner to find out centrality measure accurately.

In our work, we have scaled up the usage of the existing traditional centrality algorithms on a very large network using MapReduce and Hadoop. The map reduce will scale up computation in less time on a distributed data platform.

3. EXISTING ALGORITHMS

A total of three basic centrality algorithms were taken into consideration.

3.1 Degree Centrality

Degree centrality can be understood as the instant risk for a node to get infected with whatever is flowing across the network (for example a spreading of a virus, or some information) [4].

The degree centrality of a vertex **v**, for a graph **G= (V, E)** where **|V| is the number of** vertices and **|E| is the number of** edges, is defined as:

CD (v) =degree (v) (1)

3.2 Betweenness Centrality

It depicts how many pairs of individuals would have to go through you in order to reach one another in the minimum number of hops [5].The betweenness can be represented as:

$$C_b\ (v) = \sum_{s \neq v \neq t \in v} \frac{\sigma st(v)}{\sigma st} \quad (2)$$

Where:

σst: The total number of shortest paths from node **s** to **t**

σst (v): Number of those paths which pass through **v**

3.3 Closeness Centrality

The farness of a node is defined as the sum of the distance of a node d to every other node. Closeness is defined as the reciprocal of the farness. The more central a node is the lower is its total distance to every other node. In the standard definition of the closeness centrality, the spread of information is modeled by the use of shortest paths. This model might not be convincing for all types of communication circumstances. [4].

$$C(x) = \frac{1}{\sum_y d(x,y)} \quad (3)$$

4. MAPREDUCE& HADOOP

MapReduce is a framework for writing applications which process large data (in terabytes) in parallel on many clusters of commodity hardware in a consistent, fault-tolerant manner [8].

A MapReduce *job* divides the data into autonomous sets that are processed by the *map tasks* in a parallel manner. The framework arranges the output of the maps, which are further given to the *reduce tasks*. The MapReduce framework has a master Job Tracker and one Task Tracker per cluster-node. Master is responsible for scheduling tasks on slaves, monitoring them and re-executing the unsuccessful tasks. Slaves execute the tasks directed by the master [8].As the size of networks are increasing, it is becoming difficult to mine them to find our required result. Hence, constrained pattern mining [6], is required, which aims to find frequent patterns. This results in need for research and practices in Big data analytics [11] [7] and Big data mining [21] [22].

Therefore, distributed frameworks for graph mining are becoming popular. To handle big data, researchers proposed the use of a high-level programming model—called MapReduce—to process high volumes of data by using parallel and distributed computing [23] on large clusters or on nodes (machines), which consist of master slave nodes architecture.

Advantage of using the MapReduce model is that users need to focus on "map" and "reduce" functions without going into implementation details for distributing the input data, their scheduling and program execution across multiple nodes, handling node failure by providing secondary master, or managing communication between nodes[2].

Kang et al. [9, 10] developed graph mining algorithms for the Hadoop framework, such as finding diameter of network, degree distribution within the graph and finding connected components of large graphs by generating matrix vector multiplication performing basic graph mining tasks on very large graphs. Kang et al. [9] also developed a method for different centrality definitions which is a more computationally intensive task and popular graph mining task.

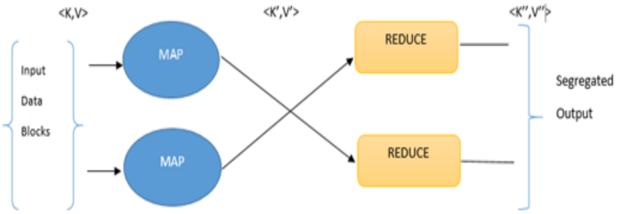

Figure 1. Computation of MapReduce

5. OUR APPROACH

5.1 Data Set

Data set is taken from Stanford website which consists of 10 anonymized networks [13]. Each network consists of edges and set of circles. Circles consists of nodes and edges are undirected. Redundant entries were removed from the data set and all networks were integrated to form a single large network.

Data set was normalized so that it could be processed by the algorithms using MapReduce. Data set was converted into CSV (comma separated values) so that it could easily be tokenized by MapReduce framework. After integration the data set consists of 88,234 edges.

Node id	Node id
1981	2390
1981	2399
3128	3336
1982	2222
1982	2142
1982	2092
1982	2603

Table 1. Input Data

5.2 Modified Centrality Algorithms

The MapReduce framework operates completely on <key, value> pairs, i.e., the framework interprets the input to the job as a set of <key, value> pairs and produce a set of <key, value> pairs as the output of the job, possibly of different forms[8].

Node ID is taken as the key and value is taken as the distance measure for each entry.

5.2.1 *Degree Centrality*

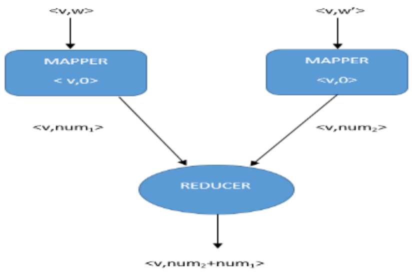

Figure 2. Execution of degree centrality algorithm on MapReduce framework

Algorithm 1 Degree Centrality algorithm

```
Map Phase:
Input: Graph G (V, E)
Output: Key value pairs <$N_i$, count>
For each edge (v, w)
Do
1: k=first (V)
2: while (k$\neq$eof)
Do
        map ($\lambda$, V, 0)
Done
3: k=next (V, k)
4: k=first (E)
5: while (k$\neq$eof)
 Do
        n=evaluate ($\lambda$, e.v) +1
        map ($\lambda$, e.v, n)
        n=evaluate ($\lambda$, e.w) +1
        map ($\lambda$, e.w, n)
        k=next (E, k)
   Done
Done
Reduce Phase:
Input: Key value pairs <V, count>
Output: Degree centrality measure according to their rank
1: For each pair <V, count>
Do
<V, count>=<V, count>+$\sum$<V, $count_i$>
2: Rank them according to the values obtained
```

5.2.2 *Betweenness Centrality*

Betweenness centrality depends on entire graph and when the dataset is divided, the interdependencies are lost which can be actually important to measure. This is one of the major limitation of hadoop.

So here, we tried to resolve such dependencies by keeping a minimum spanning tree which would help in resolving dependencies at reduce phase by joining it at the common nodes and thus identifying the interdependent shortest paths.

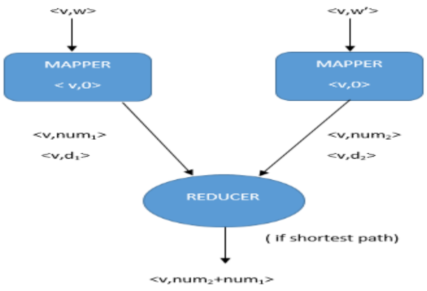

Figure 3. Execution of betweeness centrality algorithm on MapReduce framework

```
Algorithm 2        Betweenness Centrality algorithm

Map Phase:
Input: Graph G (V, E)
Output: Key value pairs <$N_i$, count>, <$N_i$, d> where d is the shortest path distance.
For each vertex v
Do
  1: Calculate the number of shortest paths that passes through v.
  2: Compute the minimum spanning tree and store the distance in d.
  Done

Reduce phase:
Input: Key value pairs <V, count>, <$N_i$, d>
Output: Betweeness centrality measure according to their rank
1: For each pair <V, count>
Do if it is shortest path
<V, count>=<V, count>+$\sum$<V, $count_i$>
Done
2: Calculate total number of paths through each pair of vertices (u, v) =t
3: Calculate betweenness centrality fraction for each vertex v:
BC (v)=$count_v$/t
4: Normalise them if required
5: Rank them according to the values obtained.
```

5.2.3 *Closeness Centrality*

```
Algorithm 3        Closeness Centrality algorithm

Map Phase:
Input: Graph G (V, E)
Output: Key value pairs <$N_i$, count>
For vertex v
Do
    Calculate sum of its distance to all other nodes.
Done
Reduce phase:
Input: Key value pairs <V, count>
Output: Closeness centrality measure according to their rank
1: For each pair <V, count>
Do
<V,count>=$\sum$(<V,$count_i$>+<V,count>)
Done
2: calculate reciprocal of the count values of each vertex.
3: Normalize it.
4: Rank them according to the values obtained.
```

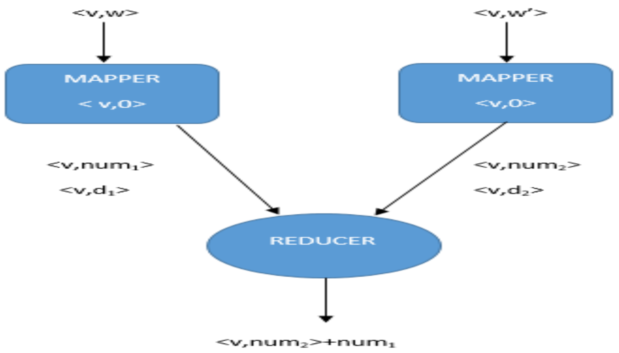

Figure 4. Execution of closeness centrality algorithm on MapReduce framework

Since the graph is assumed to be connected therefore the degree, distance, and the number of paths cannot be 0.The reciprocal of distance for each node is taken so as to find the closeness centrality.

6. RESULTS

The result obtained is the comparative identification of key nodes by each algorithm based on their centrality measure. The key nodes identified by respective algorithms are:

Node id	Betweeness Centrality	Node id	Degree centrality	Node id	Closeness Centrality
107	7833120.28	107	1045	107	3917082.644
1684	5506573.373	1684	792	1684	2753289.189
3437	3849012.303	1912	755	3437	19241519.65
1912	3737836.424	3437	547	1912	1868929.712
1085	2429155.516	0	347	1085	1214579.25
0	2384992.226	2543	294	0	1192669.613
698	1880048.492	2347	291	698	940044.2464
567	1569993.811	1888	254	567	785004.4
58	375189.9667	1800	245	58	687600.98
428	1048328.135	1663	235	428	524174.5677
563	9023405.993	1352	234	563	511705.4965

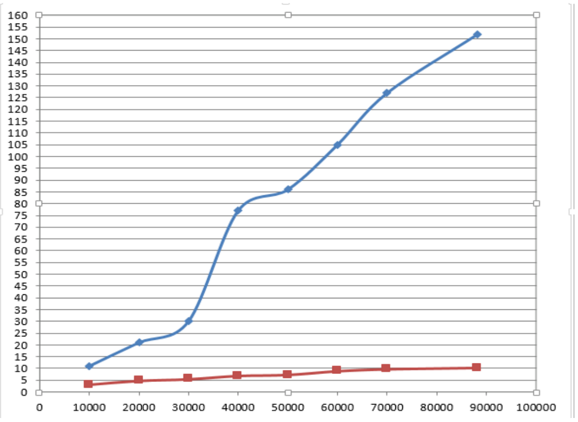

Table 2: Comparison of the top key nodes identified by each algorithm.

Figure 5. Comparison of time taken on hadoop system and standalone system

The results obtained after running the algorithms on standalone system and hadoop cluster were analyzed and compared.

Red line indicates the time taken on cluster using hadoop on different number of tuples. Blue line indicates the time taken on standalone system on different number of tuples.
The time is taken in seconds and is represented on y axis. X-axis denotes the number of tuples in input. The area between the two lines indicates the speedup of hadoop system over standalone system. This graph clearly depicts that hadoop system is scalable and robust for performing graph mining on very large networks containing millions of nodes where standalone system fails.
When this data increases to MB's and GB's the time taken by hadoop system increases linearly as compared to standalone system where the time taken increases in polynomial time.

7. FUTURE WORK

This work was aimed at the task of identifying and ranking the important or key persons using basic centrality algorithms. It can be further move ahead to write and implement other centrality algorithms such as Eigen value centrality using hadoop and MapReduce which uses more parameters to define the centrality of a node in a graph to achieve more accuracy and precision in identifying and ranking the nodes.

Moreover these results also suggests that it can be further moved to study the features of the nodes using text mining to see the behavior of the person on the social networking website to judge its importance.

It strongly supports the idea of inputting different types of data set with different attributes to conclude the importance of a node in a different way. Those attributes which majorly decides the importance of the node should be given more preference than other attributes.

Network analysis has found applications in many domains beyond social sciences; for example, the study of food chains in different ecosystems [17], identifying criminal and terrorist networks from traces of collected communications [15], [16],and understanding the interaction of proteins in metabolic pathways [18]. How contacts among humans could affect the spread of diseases and rumor in social networks as well as avoiding the spread of computer worms in a network [19][20]. Influence maximization problem of finding a small set of most influential nodes in a social network can be find out so that their accumulated impact in the network is maximized [14].

Another upgrade to the work can be in the form of dividing the dataset into clusters and carrying out the analysis on those clusters so as to identify the clusters having more number of key persons.

8. REFERENCES

[1] H. Oktay, A. S. Balkir, I. Foster, and D. Jensen "Distance estimation with MapReduce for large networks", in *Proceedings of the Workshop on Information Networks*, WIN, pp. 1-6, 2011

[2] Carson Kai-Sang Leung, R.K. MacKinnon and Fan Jiang "Reducing the Search Space for Big Data mining for Interesting Patterns from Uncertain Data"**,** Carson Kai-Sang Leung et al, 2014 *IEEE International Congress on Big Data*, 27-06-14 to 02-07-14, Page(s):315 – 322, Print ISBN: 978-1-4799-5056-0, DOI:10.1109/BigData.Congress.2014.53

[3] B. Hajian, T. White "Modelling Influence in a Social Network: Metrics and Evaluation," Privacy, Security, Risk and Trust (PASSAT) and *2011 IEEE Third International Conference on Social Computing (SocialCom)*, pp.497-500, 9-11 Oct. 2011,doi: 10.1109/PASSAT/SocialCom.2011.118.

[4] Definitions of centrality algorithms: https://en.wikipedia.org/wiki/Centrality

[5] Betweeness centrality: www.cse.unr.edu/~mgunes/cs765/cs790f10/Lect6_Centrality.ppt

[6] R.T., Ng, L.V.S. Lakshmanan, J. Han, & A. Pang "Exploratory mining and pruning optimizations of constrained associations rules*" SIGMOD '98 Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, Pages 13-24, Volume 27, Issue 2, June 1998. Pages 13-24, doi:10.1145/276305.276307

[7] H. Yang & S. Fong, "Countering the concept-drift problem in big data using iOVFDT", in *Proceedings of the 2013 IEEE International Congress on Big Data*, pp. 126–132, 2013, doi:10.1109/BigData.Congress.2013.25.

[8] Hadoop information: http://hadoop.apache.org/

[9] U. Kang, S. Papadimitriou, J. Sun, and H. Tong "Centralities in large networks: Algorithms and observations", in *SIAM International Conference on Data Mining* (SDM), 2011, doi:10.1137/1.9781611972818.11.

[10] U. Kang, C. Tsourakakis, and C. Faloutsos "Pegasus: A peta-scale graph mining system – implementation and observations", *ICDM '09 Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, Pages 229-238, ISBN: 978-0-7695-3895-2 doi:10.1109/ICDM.2009.14

[11] P. Agarwal, G. Shroff, & P. Malhotra, "Approximate incremental bigdata harmonization", *in IEEE Big Data Congress*, pp. 118–125,2013,doi: 10.1109/BigData.Congress.2013.24.

[12] P. Bródka, K. Musiał, P. Kazienko "A Performance of Centrality Calculation in Social Networks*". IEEE Computer Society*, 24-31, 2009, doi: 10.1109/CASoN.2009.20.

[13] Database: https://snap.stanford.edu/data/egonets-Facebook.html

[14] W. Chen, Y. Yuan, and L. Zhang "Scalable influence maximization in social networks under the linear threshold model", in *IEEE International Conference on Data Mining*, pp. 88–97, 2010,doi:10.1109/ICDM.2010.118.

[15] L. Freeman "The development of social network analysis", Empirical Press, 2004.

[16] S. Borgatti, A. Mehra, D. Brass, and G. Labianca, "Network analysis in the social sciences", *Science 13 February 2009*, vol. 323, no. 5916, p.892, 2009,DOI: 10.1126/science.1165821. http://www.sciencemag.org/content/323/5916/892

[17] F. Schweitzer, G. Fagiolo, D. Sornette, F. Vega-Redondo, A. Vespignani, and D. White, "Economic networks: The new challenges", vol.325, no. 5939, p. 422, 2009,DOI: 10.1126/science.1173644.

[18] B. Balasundaram, S. Butenko, I. Hicks, and S. Sachdeva, "Clique relaxations in social network analysis: The maximum k-plex problem", *Operations Research*, p. 26, 2009, doi:10.1287/opre.1100.0851.

[19] S. Eubank, V. Kumar, M. Marathe, A. Srinivasan, and N. Wang "Structure of social contact networks and their impact on epidemics", *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 70, p. 181, 2006,Doi: 10.1109/TETC.2015.2398353.

[20] D. Shah and T. Zaman "Detecting sources of computer viruses in networks: theory and experiment", *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 1, pp. 203–214, 2010, doi:10.1145/1811099.1811063.

[21] A. Azzini & P. Ceravolo, "Consistent process mining over Big data triple stores", *in IEEE Big Data Congress*, pp. 54–61,2013, doi: 10.1109/BigData.Congress.2013.17.

[22] E. ¨O lmezo˘gullari & I. Ari, "Online association rule mining over fast data", in *IEEE Big Data Congress*, pp. 110–117,2013,doi: 10.1109/BigData.Congress.2013.7

[23] M.J. Zaki "Parallel and distributed association mining: a survey", *IEEE Concurrency*, Volume 7 Issue 4, October 1999 ,Pages 14-25, doi:10.1109/4434.806975

[24] "Global Research Data Infrastructures: Towards a 10-year vision for global research data infrastructures", Final Roadmap, March 2012. *Seventh Framework Programme (FP7) - Infrastructures,"Capacities – Research Infrastructures" - Project Number: 246682*, www.grdi2020.eu

[25] "Riding the wave: How Europe can gain from the rising tide of scientific data", *Final report of the High Level Expert Group on Scientific Data*. October 2010[Online]. Available at http://cordis.europa .eu/fp7/ict/e-infrastructure/docs/hlg-sdi-report.pdf

[26] Y.Demchenko, P.Membrey, P.Grosso, C. de Laat "Addressing Big Data Issues in Scientific Data Infrastructure", in *First International Symposium on Big Data and Data Analytics in Collaboration* (BDDAC 2013). Part of The 2013 Int. Conf. on Collaboration Technologies and Systems (CTS 2013), May 20-24, 2013, San Diego, California, USA.

[27] Y. Demchenko, C. de Laat, P. Membrey "Defining architecture components of the Big Data Ecosystem", *in Collaboration Technologies and Systems (CTS),International Conference*, vol.,no.,pp.104-112,2014,doi:10.1109/CTS.2014.6867550.