IDENTIFYING INFLUENTIAL INDIVIDUALS IN MICROBLOGGING NETWORK

TEAM MEMBERS:

17BCE2001 - TANISHQ BAFNA

17BCE0371 - SHUBHAM GOYAL

REPORT SUBMITTED FOR THE FINAL PROJECT REVIEW OF

COURSE CODE: CSE3021

COURSE NAME: SOCIAL AND INFORMATION NETWORKS

SLOT: C2 + TC2

PROFESSOR: DR W.B.VASANTHA

INDEX

ABSTRACT

Identifying influential people who lead to quicker and more extensive spreading of impact in networks is of theoretical significance and practical value to either accelerating the speed of propagation in the case of product promotion or hindering the pace of diffusion involved in rumours. Conventional strategies, extending from centrality files to dispersion based procedures, as of now consider the number and impacts of supporters, and yet neglect to utilize the attributes of web-based life. An epic methodology called Partition Rank for finding a pre-fixed number of influential people in microblogging situations is proposed in this investigation to augment the effect; it joins premium likeness with social association between clients by means of chart parcelling. Test results on fake and genuine world microblogging networks delineate that our plan beats the other cutting edge strategies inadequacy and effectiveness.

KEYWORDS: Extensive spreading, Partition Rank, Centrality, Microblogging networks.

INTRODUCTION

In real-world, many complex systems can be represented as complex networks, in which, many activities such as advertising over media and word-of-mouth on social networks can be described by information spreading on complex networks maximizing the scale of spreading is a common target. If a market manager wants to advertise a new product on Twitter.com, she/he tries to choose a small number of users to provide them with free products in exchange for posting tweets about the product to influence their friends to buy the products. So, the task of market manager is to choose a few users such that the Product information can be transmitted to more users and, more products can be sold finally. With the topology unchanged or changed slightly, the location of source spreaders determines the final scale of spreading on large degree.

The problem of choosing initial nodes as source Spreaders to achieve a maximum scale of spreading is defined as influence maximization problem. Our research focuses on the strategy of choosing a set of critical nodes as source spreaders in this report. As influential nodes have strong ability to affect other nodes, selecting top-ranked influential nodes as source spreaders is a common and classical strategy. Up to now, many ranking methods have been proposed, such as degree, closeness betweenness Centralities, and other heuristic algorithms. Random-walk based methods such as well-known PageRank and Leader Rank have been receiving great attention and shown significant value in last few years. Addressed a direct method to search for influential spreaders by following the real spreading dynamics in a wide range of networks. Some other methods such as Twitter Rank are also useful and effective. Recently a local based method Cluster Rank has also good performance in some cases. Shows that the crucial factor of node's influence is its location in network measured by k-shell value. Under this measuring strategy, nodes with larger k-shell values usually have more ability to spread.

LITERATURE REVIEW

IDENTIFYING INFLUENTIAL NODES IN BIPARTITE NETWORKS USING THE CLUSTERING COEFFICIENT.

Locating important nodes in a network is often crucial as this could aid in terminating the spread of diseases or alternatively assist the spread of knowledge and information. A number of centrality measures are currently used to identify important nodes, but as Kitsak et al. (2010) point out, these measures may not reveal the truly important nodes. This is especially the case if the network has a community structure, where centrality measures may only reveal important nodes from one of the communities. This paper uses a very different approach, defining new clustering coefficients and using these to find influential nodes across communities.

[J. Liebig, A. Rao School of Mathematical and Geospatial Sciences, RMIT University, Melbourne 3001, Australia]

IDENTIFYING INFLUENTIAL NODES IN ONLINE SOCIAL NETWORKS USING PRINCIPAL COMPONENT CENTRALITY

Centrality is a measure to assess the criticality of a node's position. Node centrality as a measure of a node's importance by virtue of its central location has been in common use by social scientists in the study of social networks for decades. Over the years several different meanings of centrality have emerged. Among many centrality measures, eigenvalue centrality (EVC) is arguably the most successful tool for detecting the most influential node(s) within a social graph. Thus, EVC is a widely used centrality measure in the social sciences. As we demonstrated earlier in [11], one key shortcoming of EVC is its focus on (virtually) a single influential set of nodes that tends to cluster within a single neighbourhood.

[Hayder Radha Department of Electrical & Computer Engineering Michigan State University East Lansing, MI 48824, USA {ilyasmuh, Radha}@egr.msu.edu]

IDENTIFYING INFLUENTIAL SPREADERS IN ARTIFICIAL COMPLEX NETWORKS

With the development of complex network science, lots of real-world complex systems could be described by complex networks. And the dynamical disease models and spreading behaviours, such as susceptible-infectious-susceptible (SIS), susceptible-infectious-recovered (SIR) models and so on, have been extensively investigated in various complex networks. Just as an old Chinese saying “One who mixes with vermilion will turn red, one who touches pitch shall be defiled therewith”. Obviously, whether a person will turn “red” or not as well as how fast of this transition depends on the properties of the original spreader in the networks. Therefore, identifying influential spreaders in complex networks have fundamental importance.

[WANG Pei · TIAN Chengeng · LU Jun-an The Editorial Office of JSSC & Springer-Verlag Berlin Heidelberg 2014]

A GRAPH EXPLORATION METHOD FOR IDENTIFYING INFLUENTIAL SPREADERS IN COMPLEX NETWORKS

Understanding spreading process in real-world complex networks is a central subject in network analysis, due to the variety of applications which occur - such as the control of the spread of a disease, the viral marketing, as well as the network vulnerability to external attacks. Key role in these processes play the high spreading efficient nodes which are often called *influential spreaders*, representing the nodes that are more likely to spread information or a virus in a large part of the network. Thorough research has been realized in order to connect the topological properties of network nodes with their spreading efficiency. Kitsak et al. (2010) proposed the k-core decomposition method (Seidman 1983) as an *influential spreaders identifier*, showing that the k-core values constitute a more reliable measure than *degree centrality* and *betweenness centrality*. One of the core results is that the placement of a node (node global property) is more important than its degree (node-local property). That is, two nodes with the same degree but different placement in the network, where the one is connected with the periphery of the network and the other one with the innermost core may not have equal spreading efficiency. Thus, highly connected nodes are not always the best spreaders, while less connected nodes but, at the same time, well connected with the core of the network may strongly affect the spreading process.

[Nikos Salamanos, Elli Voudigari and Emmanuel J. Yannakoudakis. Salamanos et al. Applied Network Science (2017)]

OBJECTIVE OF THE PROJECT

Identifying influential people who lead to quicker and more extensive spreading of impact in networks is of theoretical significance and practical value to either accelerating the speed of propagation in the case of product promotion or hindering the pace of diffusion involved in rumours.

In this Project are main objective is to find influential people (vertex) in a social media environment such are microblogging sites (twitter, reddit) and photo blogging sites (Instagram). The social media of a person (vertex) is like a complex directed graph were the outward pointing edge would infer that a person follows another person (vertex) and the inward pointing edge means that he is being followed by another person (vertex). Using properties of a network (Centrality, Prestige, Degree, Hubs and Authorities, Page Rank, Community Detection) we would find the influential people on the internet.

INNOVATION COMPONENT IN THE PROJECT

With the help of R software we will generate a graph based on the number of people each person follows. The graph would contain around 40 to 50 people (vertices) which would be interconnected or closely knit. With the help of properties of a network (Centrality, Prestige, Degree, Hubs and Authorities, Page Rank, Community Detection) we would find the influential people on the internet. Rather than focusing on one network property we would examine the effect of some of the important properties on a social media network to collectively decide who the influential people on the internet are.

We would also analyse the effect of the community on the person (vertex) on choosing the people (vertex) the person (vertex) he decide to follow.

WORK DONE AND IMPLEMENTATION

PROPOSED WORK AND IMPLEMENTATION

Methodology adapted: We will collect through various sources over the internet and microblogging and photo blogging sites to get the necessary data set.

This data set will be given to the algorithm which will help us produce a graph and will give us the properties related to it such as centrality, betweenness, clustering coefficient, degree of the graph. The Data set will produce a very complex graph. With the help of R programming we will be able to produce all the necessary statistics.

CENTRALITY

Degree Centrality:- Let the total number of actors in the network be n. Undirected Graph: In an undirected graph, the degree centrality of an actor i (denoted by CD(i)) is simply the node degree (the number of edges) of the actor node, denoted by d(i), normalized with the maximum degree, n-1.

$$C_D(i) = \frac{d(i)}{n-1}$$

The value of this measure ranges between 0 and 1 as n-1 is the maximum value of d (i).

Directed Graph: In this case, we need to distinguish in-links of actor I (links pointing to i), and out-links (links pointing out from i). The degree centrality is defined based on only the out-degree (the number of out links or edges), d_o (i).

$$C'_D(i) = \frac{d(i)}{n-1}$$

Closeness Centrality: - This view of centrality is based on the closeness or distance. The basic idea is that an actor xi is central if it can easily interact with all other actors. That is, its distance to all other actors is short. Thus, we can use the shortest distance to compute this measure. Let the shortest distance from actor i to actor j be d (i, j) (measured as the number of links in a shortest path). Undirected graph: The closeness centrality CC (i) of actor i is defined as: The value of this measure also ranges between 0 and 1 as n-1 is the minimum value of the denominator, which is the sum of the shortest distances from i to all other actors.

$$C_c(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}$$

Directed graph: The same equation can be used for a directed graph. The distance computation needs to consider directions of links or edges.

$$C_c(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}$$

Betweenness Centrality: - If two non-adjacent actors j and k want to interact and actor i is on the path between j and k, then i may have some control over their interactions. Betweenness measures this control of i over other pairs of actors. Thus, if I is on the paths of many such interactions, then i is an important actor.

Undirected graph: Let p_{jk} be the number of shortest paths between actor j and k. The betweenness of an actor i is defined as the number of shortest paths that pass i (denoted by p_{jk} (i), j i and k i) normalized by the total number of shortest paths of all pairs of actors not including i:

$$C_B(i) = \sum_{j<k} \frac{P_{jk}(i)}{P_{jk}}$$

PRESTIGE

Prestige is a more refined measure of prominence of an actor than Centrality. A prestigious actor is defined as one who is object of extensive ties as a recipient. In other words, to compute the prestige of an actor, we only look at the ties (links) directed or pointed to the actor (in-links). Hence, the prestige cannot be computed unless the relation is directional or the graph is directed.

The main difference between the concepts of centrality and prestige is that centrality focuses on out-links while prestige focuses on in-links. The third prestige measure (i.e., rank prestige) forms the basis of most Web page link analysis algorithms, including PageRank and HITS.

Degree Prestige: Based on the definition of the prestige, it is clear that an actor is prestigious if it receives many in-links or nominations. Thus, the simplest measure of prestige of an actor i (denoted by $P_D(i)$) is it's in degree. Where d_I (i) is the in-degree of i (the number of in-links of i) and n is the total number of actors in the network. As in the degree centrality, dividing by n 1 standardizes the prestige value to the range from 0 and 1. The maximum prestige value is 1 when every other actor links to or chooses actor i.

$$P_D(i) = \frac{d_I(i)}{n-1}$$

COMMUNITY DETECTION

What is community detection?

The process of finding clusters of nodes (''communities'') – With Strong internal connections and – Weak connections between different communities

• Ideal decomposition of a large graph – Completely disjoint communities – There are no interactions between different communities.

• In practice, – find community partitions that are maximally decoupled.

Why Detecting Communities is Important?

The club members split into two groups (gray and white)

• Disagreement between the administrator of the club (node 34) and the club's instructor (node 1),

• The members of one group left to start their own club

Why Community Detection?

Network Summarization

– A community can be considered as a summary of the whole network

– Easier to visualize and understand

Preserve Privacy

– [Sometimes] a community can reveal some properties without releasing the individuals' privacy information

PAGE RANK

PageRank (PR) is an algorithm used by Google Search to rank websites in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages. According to Google:

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

ALGORITHM

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

If the only links in the system were from pages B, C, and D to A, each link would transfer 0.25 PageRank to A upon the next iteration, for a total of 0.75.

$$PR(A) = PR(B) + PR(C) + PR(D).$$

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

i.e. the PageRank value for a page u is dependent on the PageRank values for each page v contained in the set Bu (the set containing all pages linking to page u), divided by the number L(v) of links from page v. The algorithm involves a damping factor for the calculation of the PageRank. It is like the income tax which the govt. extracts from one despite paying him itself.

HITS ALGORITHM

HITS; also known as hubs and authorities) is a link analysis algorithm that rates Web pages, developed by Jon Kleinberg. The idea behind Hubs and Authorities stemmed from a particular insight into the creation of web pages when the Internet was originally forming; that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that they held, but were used as compilations of a broad catalog of information that led users direct to other authoritative pages. In other words, a good hub represented a page that pointed to many other pages, and a good authority represented a page that was linked by many different hubs

In the HITS algorithm, the first step is to retrieve the most relevant pages to the search query. This set is called the root set and can be obtained by taking the top pages returned by a text-based search algorithm. A base set is generated by augmenting the root set with all the web pages that are linked from it and some of the pages that link to it. The web pages in the base set and all hyperlinks among those pages form a focused subgraph. The HITS computation is performed only on this focused subgraph. According to Kleinberg the reason for constructing a base set is to ensure that most (or many) of the strongest authorities are included.

Authority update: Update each node's authority score to be equal to the sum of the hub scores of each node that points to it. That is, a node is given a high authority score by being linked from pages that are recognized as Hubs for information.

- Hub update: Update each node's hub score to be equal to the sum of the authority scores of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node is calculated with the following algorithm:

- Start with each node having a hub score and authority score of 1.
- Run the authority update rule
- Run the hub update rule
- Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.
- Repeat from the second step as necessary.

DATASET USED:

Our data set is generated from live data such as information from Instagram through Instagram API and Twitter data. The data has been generated by creating a google form where we asked people to give a detailed analysis of people the followed and no of followers.

We have generated our own data set.

Is your project based on any other reference project (Stanford Univ. or MIT)?

This project is not based on any of the projects from (MIT or Stanford Univ.).

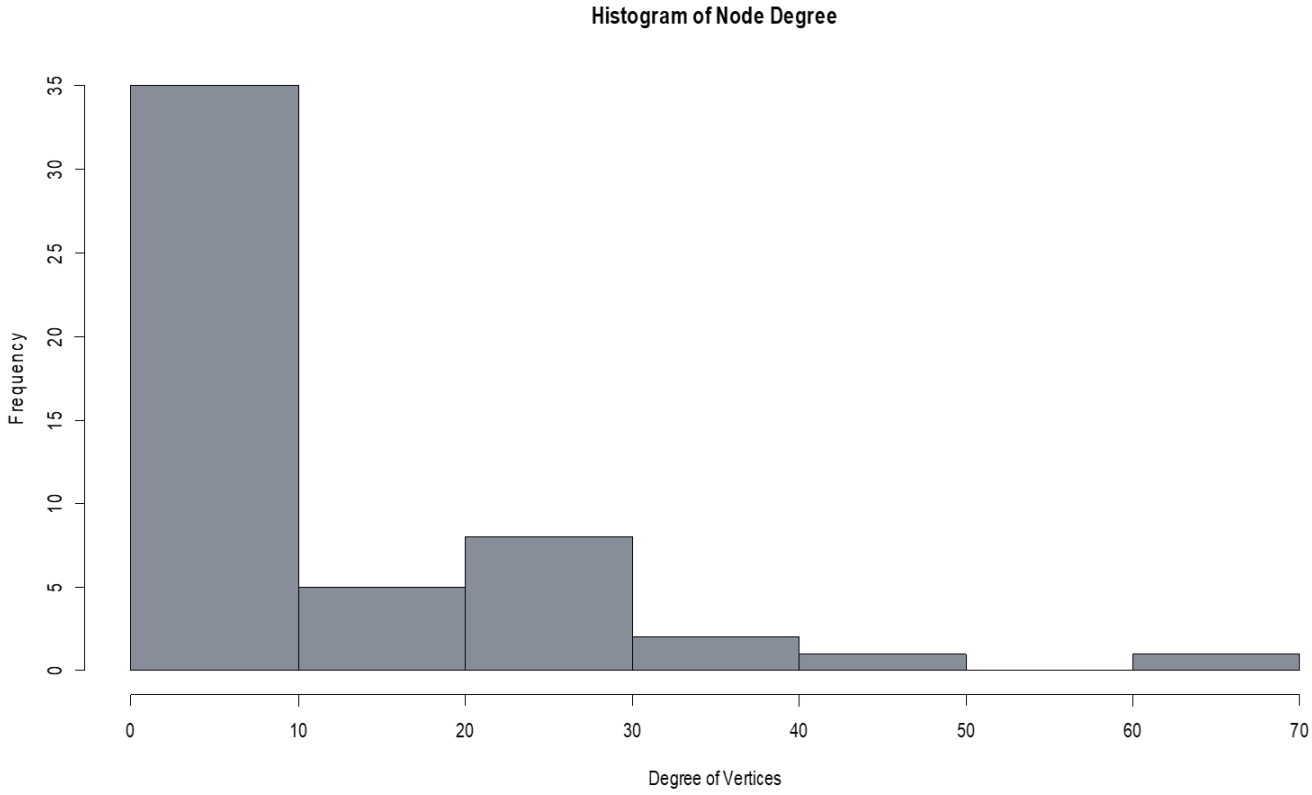
Tools used

Hardware and software requirements: R software for designing graph and finding necessary details from the graph.

With the help of the properties of a graph, we will find the most popular or influential person (node) of a microblogging site (graph).

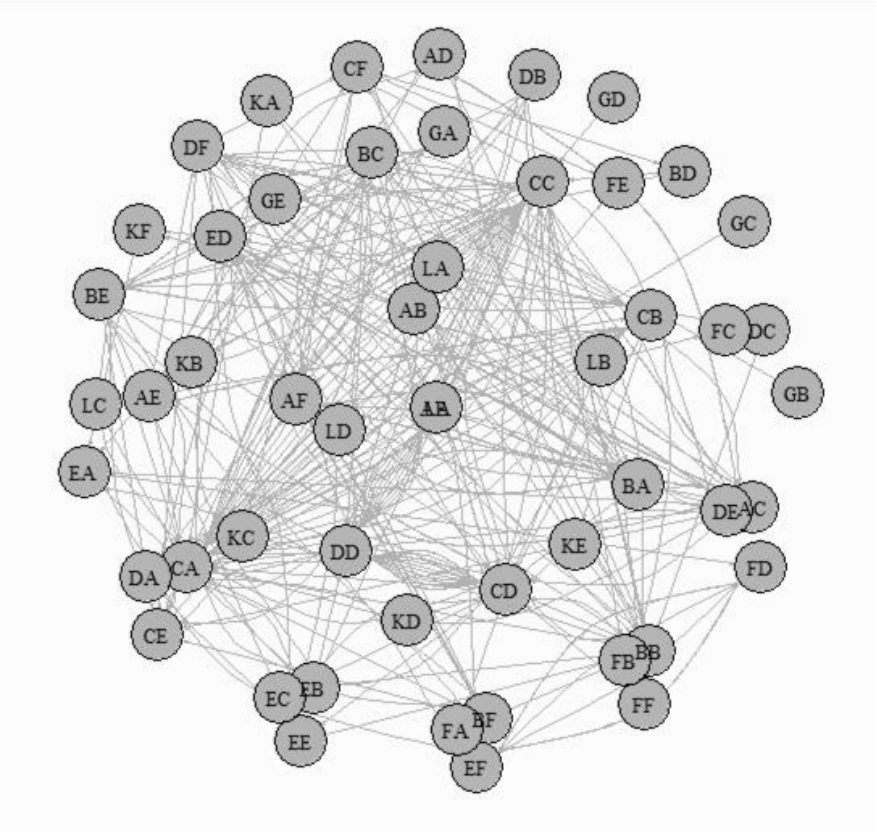
Library (igraph) and library (sna) which are preinstalled with the package.

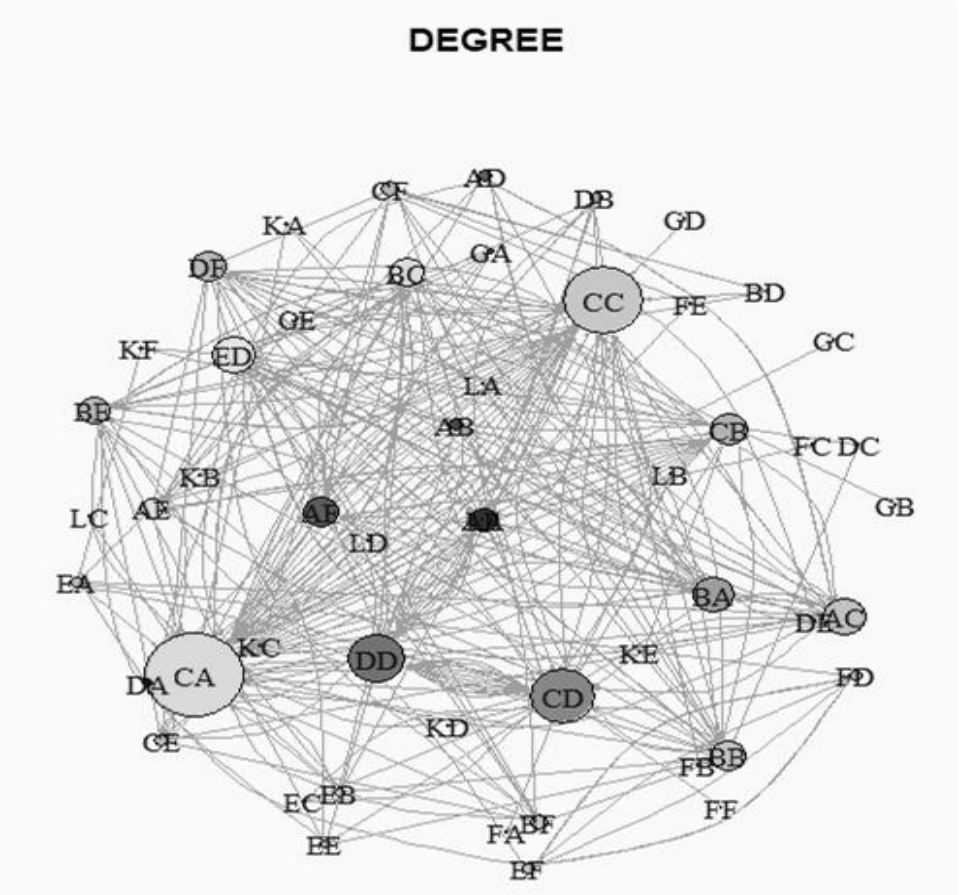
SCREENSHOT AND DEMO



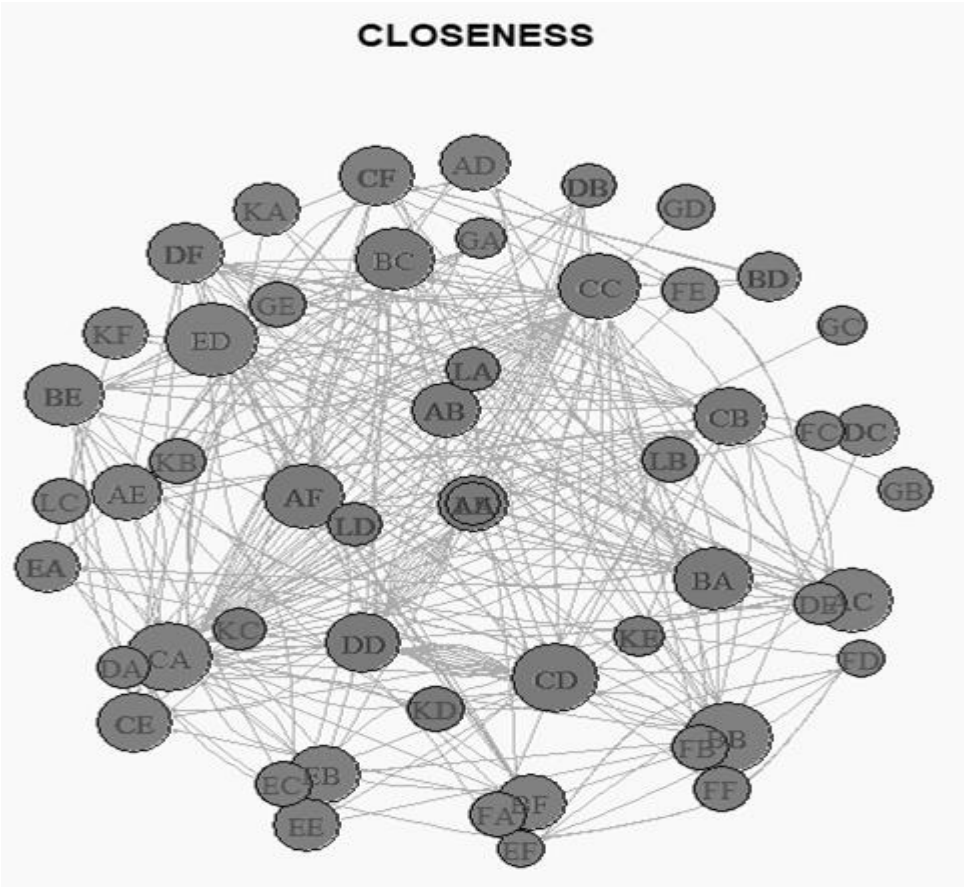
This graph shows the frequency of the nodes vs the degree of nodes

THE NETWORK GRAPH

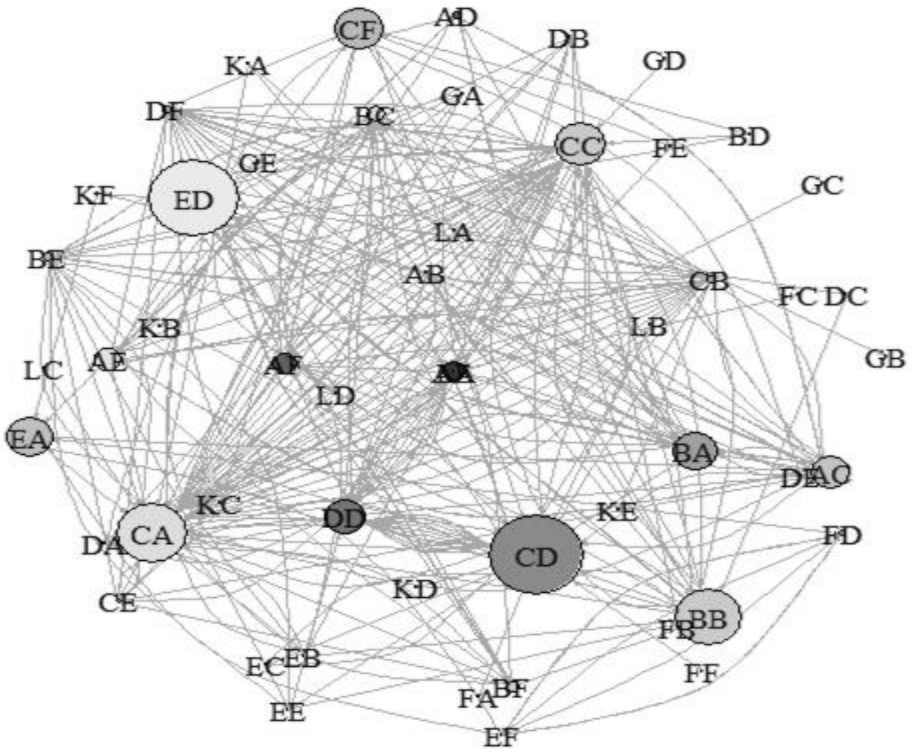




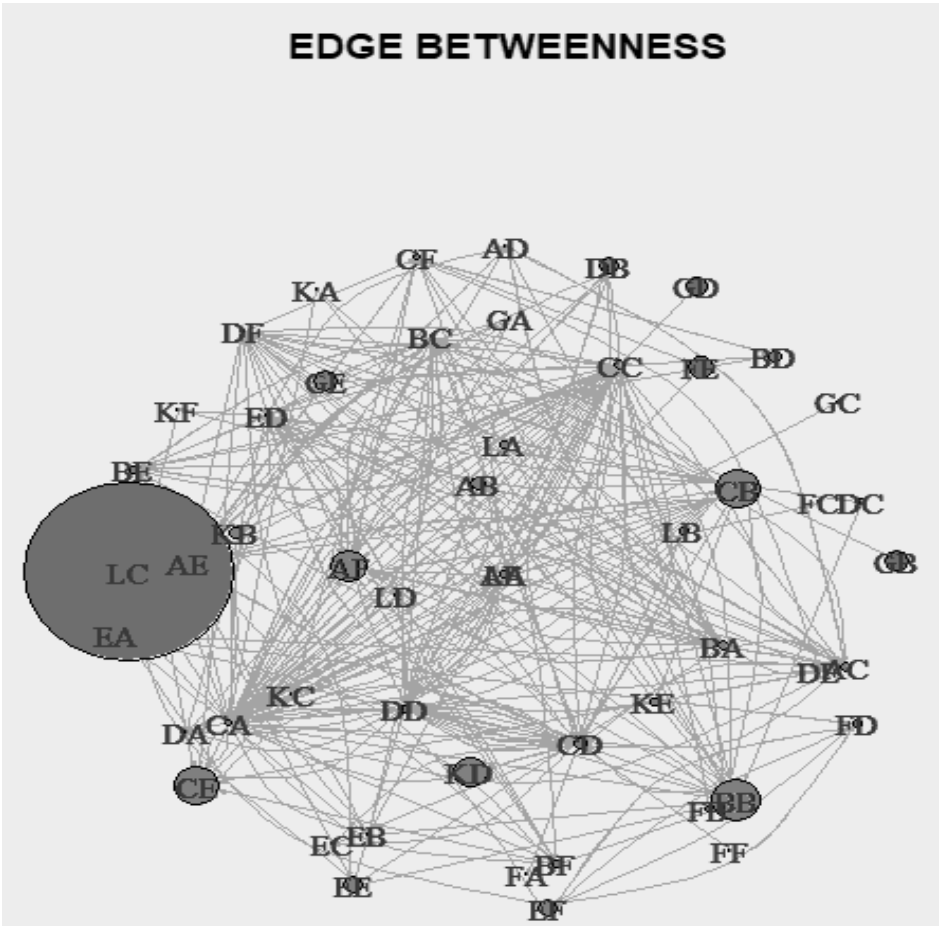
CLOSENESS



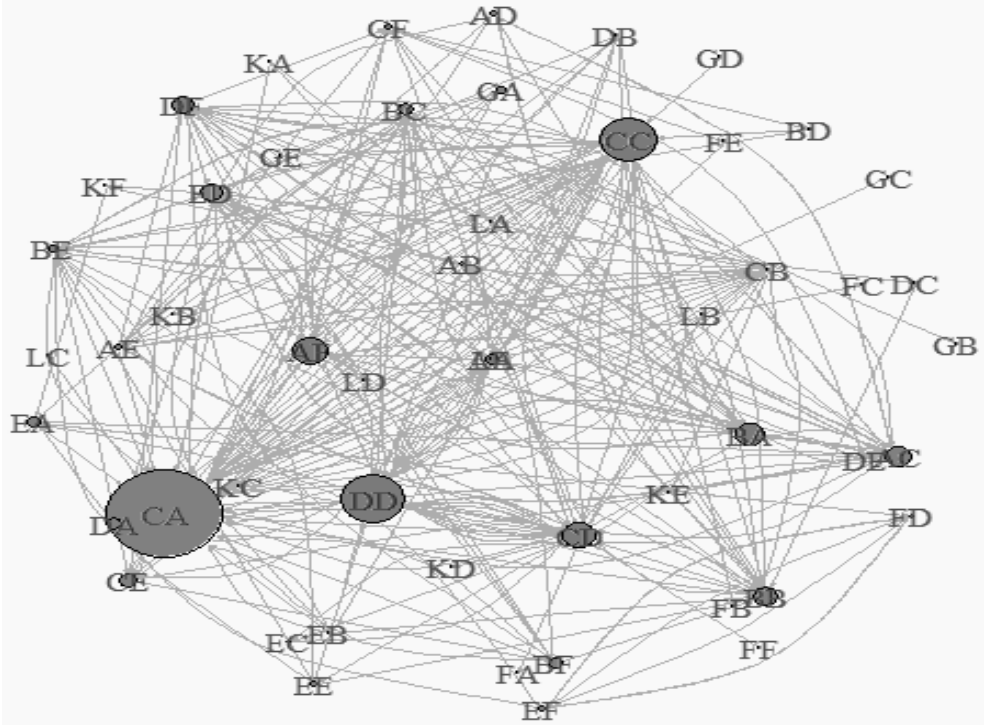
BETWEENNESS



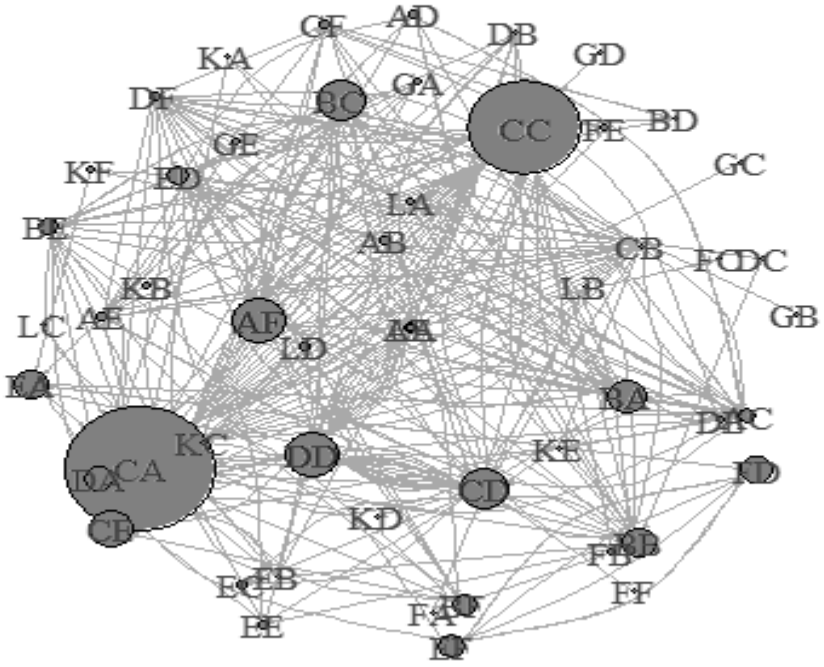
EDGE BETWEENNESS



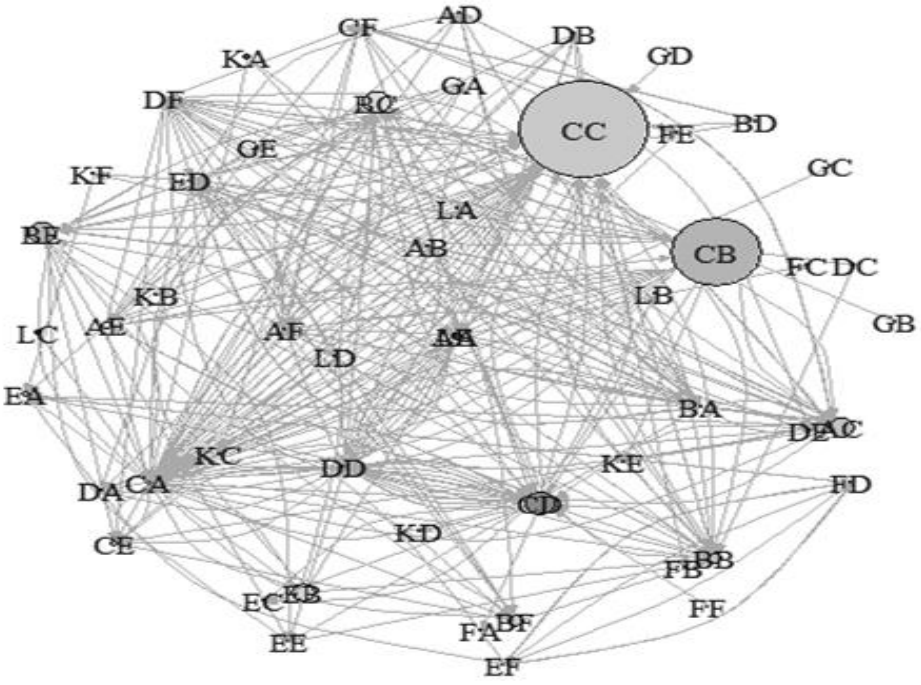
PRESTIGE



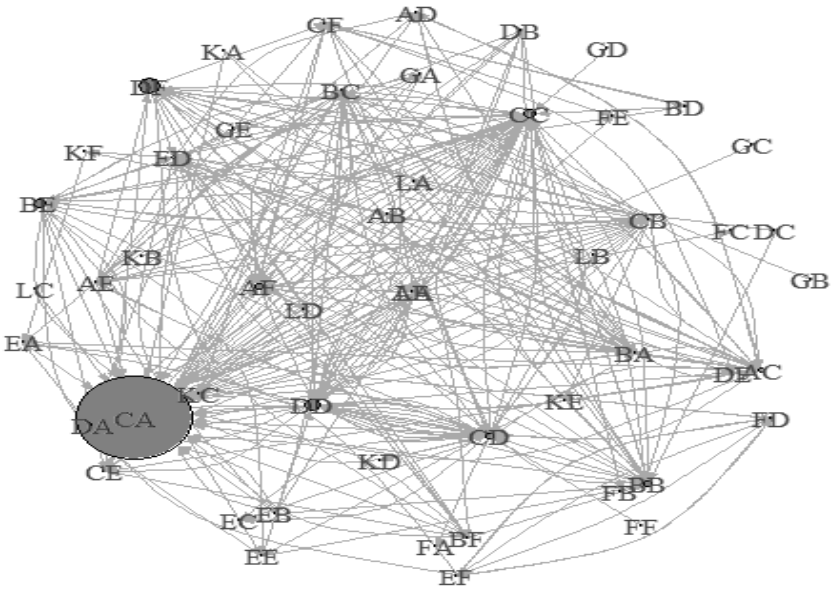
PAGE RANK



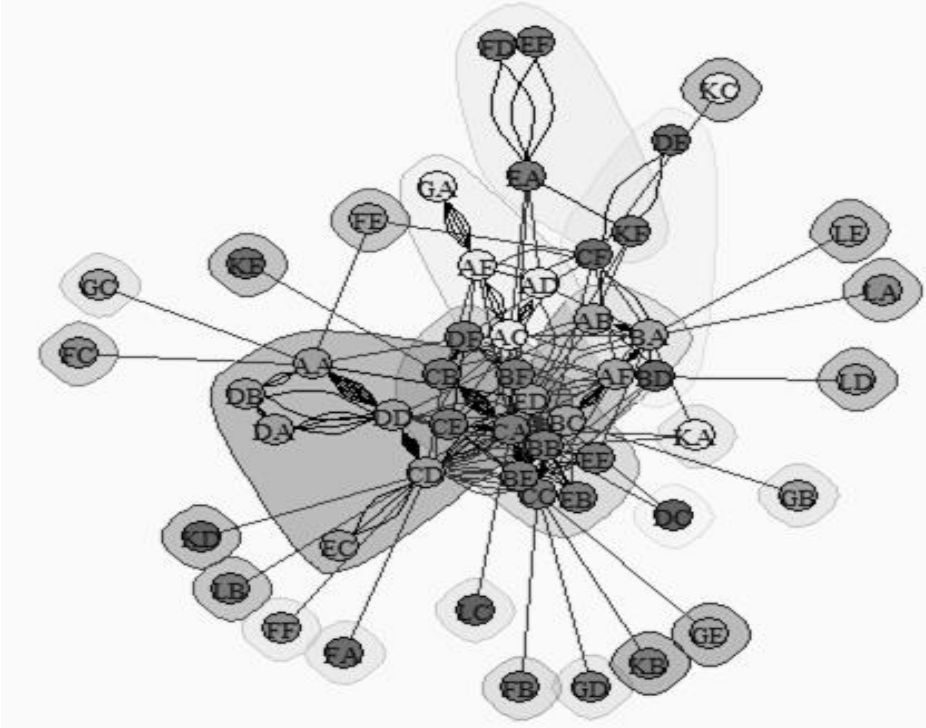
HUBS



AUTHORITIES



COMMUNITY DETECTION



RESULTS AND DISCUSSION

With the different graphs generated we can find the influential people of the network.

1] With the degree graph we can infer that node CA, CC, CD, DD could be considered as influential node as maximum number of node have an outward edge pointing towards them.

2] With the closeness graph we cannot infer anything substantial because we cannot find any node with any extraordinary characteristics.

3] With the betweenness graph we can infer that CA, CD, ED, BB, CC can be considered as influential node because these nodes are the nodes which come in the between the shortest path of two nodes the maximum time.

4] With the prestige graph we can infer that node CA, CC, DD, AF could be considered as influential nodes as they have the maximum prestige i.e. these nodes have the maximum ties with other nodes.

5] With the Page Rank graph CA, CC, DD, BC, AF, CD would be considered as influential nodes as they have a higher page rank.

6] With authority graph we can infer that node CA has a higher authority over other nodes.

7] With the community graph we can see that any majority of nodes from different community follow node CA, CC, BF, CD, ED, and BB.

FINAL RESULT: - NODES CA, CC, DD, AF ARE THE INFLUENTIAL NODE (PEOPLE) OF THE NETWORK.

WE COULD CONSIDERED NODE CA AS THE MOST INFLUENTIAL NODE OF THE NETWORK.

REFERENCES

1. Asian, A. S., Denley, T. M., & H•activist, R. (1998). Bipartite graphs and their applications. (1st ed.). Cambridge: Cambridge University Press.
2. BBC (2014). Obituary: Noordin Mohamed Top. online. URL: http://news.bbc.co.uk/2/hi/ Asia-pacific/4302368.stm.
3. Bonacich, P. (1978). Using Boolean algebra to analyze overlapping memberships. Sociological Methodology, (pp. 101{115).
4. Borgatti, S. P. (2012). Social network analysis, two-mode concepts in. In Computational Complexity (pp. 2912{2924). Springer.
5. Borgatti, S. P., & Halgin, D. S. (2011). Analyzing a_liation networks. In The Sage handbook of social network analysis (pp. 417{433). Thousand Oaks: Sage Publications.
6. R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley New York, 2001.

R CODE

```
library(igraph)
library(sna)
library(networkR)
library(DirectedClustering)


# Read data file
data <- read.csv(file.choose(), header=T)
y <- data.frame(data$first, data$second)

# Create network
net <- graph.data.frame(y, directed=T)
```

```
c <-get.adjacency(net, type=c("both", "upper", "lower"),
      attr=NULL, names=TRUE,sparse=FALSE)

V(net)
E(net)
V(net)$label <- V(net)$name
V(net)$degree <- degree(c, g=1, nodes=NULL, gmode="digraph", diag=FALSE,
tmaxdev=FALSE, cmode="freeman", rescale=FALSE, ignore.eval=FALSE)
set.seed(123)

# Histogram of node degree
hist(V(net)$degree,
    col = 'green',
    main = 'Histogram of Node Degree',
    ylab = 'Frequency',
    xlab = 'Degree of Vertices')

# Network diagram
set.seed(123)
plot(net,
    vertex.color = 'green',
    vertext.size = 2,
    edge.arrow.size = 0.1,
    vertex.label.cex = 0.8)

diameter(net, directed=F, weights = NA)
edge_density(net, loops = F)
ecount(net)/(vcount(net)*(vcount(net)-1))
reciprocity(net)
```

```
# Degree
V(net)$degree <- degree(c, g=1, nodes=NULL, gmode="digraph", diag=FALSE,
tmaxdev=FALSE, cmode="freeman", rescale=FALSE, ignore.eval=FALSE)
set.seed(123)
plot(net,
    vertex.color = rainbow(52),
    vertex.size = V(net)$degree*0.4,main='DEGREE',
    edge.arrow.size = 0.1,
    layout=layout.kamada.kawai)

# Closeness
V(net)$closeness<-closeness(c, g=1, nodes=NULL, gmode="digraph", diag=FALSE,
tmaxdev=FALSE, cmode="undirected", geodist.precomp=NULL,
rescale=FALSE, ignore.eval=TRUE)
set.seed(123)
plot(net,
    vertex.color = rainbow(52),
    vertex.size =  V(net)$closeness*40,main='CLOSENESS',
    edge.arrow.size = 0.1,
    layout=layout.kamada.kawai)

# Betweenness
betweenness.igraph(net, directed=T, weights=NA)
V(net)$betweenness <-betweenness(c, g=1, nodes=NULL, gmode="digraph", diag=FALSE,
tmaxdev=FALSE, cmode="directed", geodist.precomp=NULL,
rescale=FALSE, ignore.eval=TRUE)
set.seed(123)
plot(net,
    vertex.color = rainbow(52),
    vertex.size =  V(net)$betweenness*0.06,main='BETWEENNESS',
```

```
    edge.arrow.size = 0.1,
    layout=layout.kamada.kawai)


edge_betweenness (net, directed=T, weights=NA)
set. Seed(123)
plot(net,
    vertex.color = rainbow (52),
    vertex.size = edge_betweenness(net, directed=T, weights=NA)*0.34,main='EDGE BETWEENNESS',
    edge.arrow.size = 0.1,
    Layout=layout.kamada.kawai)
# Prestige
V(net)$prestige<- prestige(c, g=1, nodes=NULL, gmode="digraph", diag=FALSE,
cmode="indegree", tmaxdev=FALSE, rescale=TRUE, tol=1e-07)
set.seed(123)
plot(net,
    vertex.color = rainbow(52),
    vertex.size =  V(net)$prestige*150,main='PRESTIGE',
    edge.arrow.size = 0.1,
    layout=layout.kamada.kawai)


# PageRank
V(net)$pg<- page_rank(net, algo = c("prpack", "arpack", "power"),  vids = V(net),
                directed = TRUE, damping = 0.85, personalized = NULL, weights = NULL,
                options = NULL)$vector
plot(net,
    vertex.size=V(net)$pg*300,
    main = 'PAGE RANK',
    vertex.color = rainbow(52),
    edge.arrow.size=0.1,
```

```
    layout = layout.kamada.kawai)

#Hubs and Authorities
hits(c, maxiter = 100L, tol = 1e-05)
net <- graph.data.frame(y, directed=T)
hs <- hub_score(net)$vector
as <- authority_score(net,scale = TRUE, weights = NULL,)$vector
par(mfrow=c(1,2))
set.seed(123)
plot(net,
    vertex.size=hs*30,
    main = 'HUBS',
    vertex.color = rainbow(52),
    edge.arrow.size=0.1,
    layout = layout.kamada.kawai)
plot(net,
    vertex.size=as*30,
    main = 'AUTHORITIES',
    vertex.color = rainbow(52),
    edge.arrow.size=0.1,
    layout = layout.kamada.kawai)
par(mfrow=c(1,1))

# Community detection
net <- graph.data.frame(y, directed = F)
cnet <- cluster_edge_betweenness(net)
plot(cnet,main="COMMUNITY DETECTION",
    net,
    vertex.size = 10,
    vertex.label.cex = 0.8)
```