*Analysis of Diffusion in Social Network*

*WASHINGTON STATE UNIVERSITY*

*Under the guidance of: Assefaw Hadish Gebremedhin*

**Yuhui Wang, Shashvath PV, Siddhant SrivastavA**

**Abstract**

*This paper is mainly about the new model for diffusion which we proposed basically on analysis on the distribution model, and we use pagerank as a support for our model construction, also we want to try to solve the diffusion based on the community detection which can help us more understanding about the diffusion process between two community which not only two induvial nodes.*

**Introduction**

The problem we focus on is mainly about the diffusion in the social network. The reason why we focus on this, since we want to know more about how people react to other’s in the social network, the diffusion takes a really important role in this particular way to understand people’s reaction in the social network. We first think about can we directly find a way to tell what probability the diffusion is will take place from a specific node to another. However, we fail in the directly study, since for any diffusion model being introduced, they are hardly to defined the probability out. This situation even worse in the social network. Therefore, we consider another way to think about this problem, we consider the whole network together and try use the statistic model to find how many nodes can be active in the fixed time, and coming back to find a way solve this problem. Our basic approach is mainly construct a new diffusion model based on the statistic distribution analysis for independent Cascade Model and PageRank analysis. Also we want to know what happen with it combined with the community detection. We basically use Poisson Distribution and Binomial Distribution to check about the model, and we use PageRank priority to defined the probability which diffusion process take place. Then we simulate the result use this diffusion model. After that, we implement the community detection algorithm for checking about the diffusion between not only two nodes but also two communities.

**Problem Definition**

The problem we focus on is about how can we get the probability from one node to diffuses to another node, which directly connected. We are interesting in this problem, since we figure out if we can have a way to calculate the probability, we can give the prediction for the how can we maximize the influence when we want to show something, and make it spreads as far as we can. The important of this thing is about how the social network share their information, and how the influence will be taken place in the social network.

**The Model, Algorithm and Implementation**

Before introduce about the Model, we need expressed the idea how this model came out. We use statistic distribution model to analysis the Independent Cascade Model. Independent Cascade Model is Given a graph G=(V,E), start with an initial set of active nodes S, it is processing in the discrete time steps. When node V being active in step T, it will have a single chance to active all his inactive neighbors, which for each neighbor W being active with the probability Pvw. And for a node which has multiple new active neighbors, their attempts are in random sequence. Once V succeed, which means W have been active in the steps T+1. Process runs until no more nodes can be active in the G. This Model mainly have 2 important things we noticed. One thing is that from specific node V to active W, it only try once, after it done, it won’t try again. Another thing is that the steps form T to T+1 it means all the nodes active in step T have finished the activation process in the T+1. That is with definition of the steps, for any T in the diffusion process, we can regard as from T to T+1 it will have a fixed average time in one graph G which being given. That is we can use Poisson Distribution to simulate result. “Poisson Distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event.” That is, in our case, for given K nodes being active in the T to T+1 time, what is the probability of this K nodes being active. The formula as follow:

For this formula x means the how many nodes we suppose to being active in T to T+1. means the events rates, which represent average the nodes being active in time T to T+1. With this Poisson Distribution, we think about the case in the social network, we can give the prediction how many nodes being active in a specific time interval. Then we think deep in it, we figure out that binomial Distribution have really close connection to the Poisson Distribution. “Binomial Distribution with parameters n and p is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments, each of which yields success with probability *p.”* The formula as follow:

In this formula n means total number, k means how many events happen, p shows the probability for each one happened. In statistic, there is in some case Binomial Distribution will similar to the Poisson Distribution, which it is when n is large enough and p is small enough. That is in our case, we consider n as the degrees of each nodes, which in a large graph the n is big enough, that is in this situation, why can’t we use the Binomial Distribution to represent the Distribution we are looking for. That is means for any nodes inside the network, they share the same probability to all his neighbor. After this conclusion, we think if it is possible that all nodes share the same probability in the social network, in reality, it is impossible. We found some examples in the network. Which Donald J. Trump share a video with 468K views and 5529 of them share the video, but another guys called Billy Clark shared the same video no one share the video from him. That is different nodes have different probability for diffusion process in the social network. Therefore, we used the PageRank to give us some idea about the what nodes will have the biggest influence in the network.

**Page Rank in social networks:**

PageRank serves three purposes in a social network, where the nodes are people and the edges are some type of social relationship. First, it can help solve link prediction problems to find individuals who will become friends soon. Second, it serves a classic role in evaluating the centrality of the people involved to estimate their social status and power. Third, it helps evaluate the potential influence of a node on the opinions of the network. PageRank has been used to rank individuals in the Twitter network by their importance (Java, 2007) and to help characterize properties of the Twitter social network by the PageRank values of their users.

**Page rank with Diffusion:**

We decided to perform a diffusion analysis and build a new model for the user behavior in a particular social network. In order to perform diffusion, we required page rank for the prediction model of diffusion.

The network we took into consideration is the Facebook network and obtained the dataset from snap datasets. The URL for the data set is as shown below:

<https://snap.stanford.edu/data/egonets-Facebook.html>

The network data set we took into consideration consist of 4039 nodes and 88234 edges. This is an unsigned network. In order to compute the page rank for the network and also the properties of the network we used R studio.

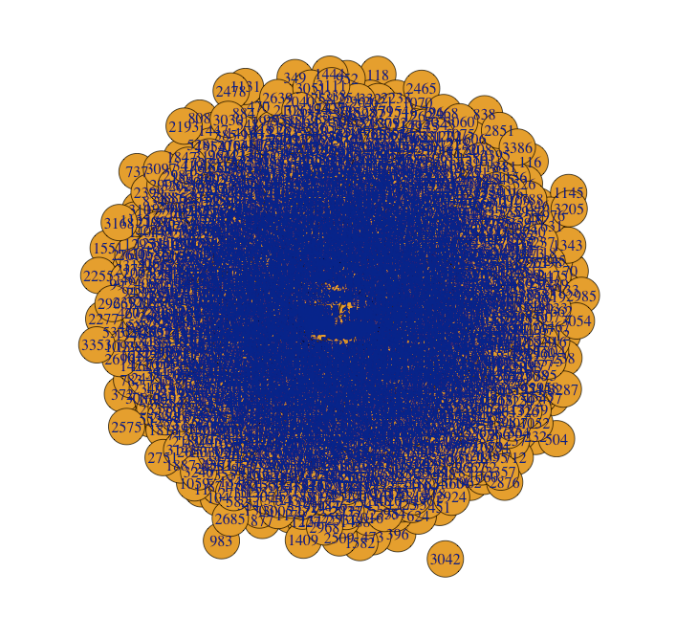


Fig: Graph of the network selected

Considering the above mentioned network, we intend to know the following properties in the network:

1. PageRank
2. Betweenness
3. Centrality
4. Max degree

Page rank:

Max(page\_rank(a1)$vector) 🡪 0.005067841

Which(Max(page\_rank(a1)$vector)) 🡪 3831

Max Degree:

Max(degree(a1)) 🡪 214

max(degree(a1)) 🡪 3831

Betweenness:

max(betweenness(a1)) 🡪 60510.5

which.max(betweenness(a1)) 🡪 3831

From the above results, we can infer that page rank for this network is mainly concentrated on the node 3831 which has the highest value for degree, page rank and betweenness. We decided to plot the graph of page rank value with the index and the graph we obtained as follows:

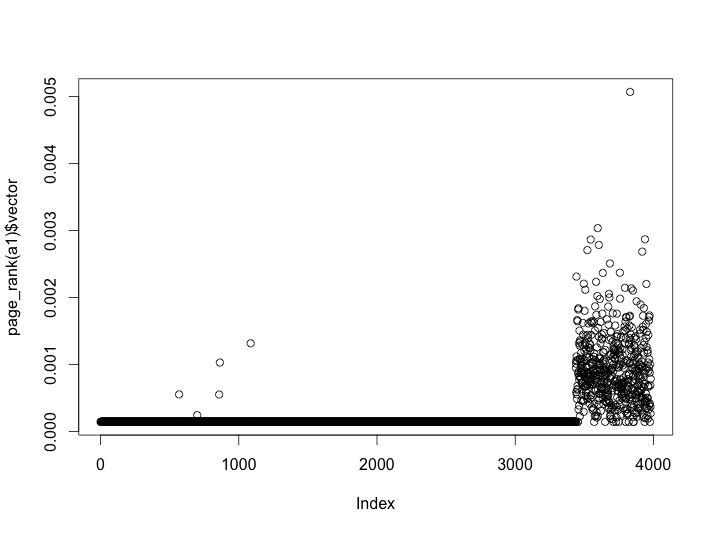


Fig: Plot of Page Rank vs Index

From the above plot we can see that the nodes which have the maximum page rank are more concentrated towards the right hand side suggesting that the nodes with high index number have high page rank. There is a sudden spike in the page rank value once it reaches the index value of 3500.Most of the nodes have the page rank value which is more dispersed unlike something around 0 for the nodes below 3500. We can see that the node which has the maximum page rank has the index of 3831 with a value of 0.005067.

Therefore, we construct the Probability model based on the highest priority in the network with the standard probability Ps. For ith node in the network, we mark it as Ni, the probability will be the ith node’s priority divide the highest priority times the standard probability. The formula as follow:

For this formula P means the probability, and p stands for the priority.

For simulation, we use the algorithm as follow:

1. Input the graph G and node number n
2. Run the PageRank function on the graph to get the priority for each nodes, find the highest node’s priority and V’s priority.
3. We search for the node V in the graph g with the node number n. And mark this node active=1.
4. We find all the neighbors for the V.
5. We calculate the probability base on the formula we just showed, and assigned the =0.4. For calculation the P(V)
6. Run a random number generator for each neighbor which is not being active.
7. Check about the random number if it is higher than the P(V), then assigned not being active, if it is lower than or equal to P(V), we active it and assign active =1 for this nodes , and put this node number to a list L.
8. We then pick a node number from list L, and repeat the process from step 3. Until list is empty.
9. We get the nodes from the graph G which all active=1.And construct new graph with this nodes and its edge between the nodes inside the active=1 sets.

**Community Detection:**

Social, technological and information systems can often be described in terms of complex networks that have a topology of interconnected nodes combining organization and randomness. The typical size of large networks such as social network services, mobile phone networks or the web now counts in millions when not billions of nodes and these scales demand new methods to retrieve comprehensive information from their structure. A promising approach consists in decomposing the networks into sub-units or communities, which are sets of highly inter-connected nodes. The identification of these communities is of crucial importance as they may help to uncover a-priori unknown functional modules such as topics in information networks or cyber-communities in social networks. Moreover, the resulting meta-network, whose nodes are the communities, may then be used to visualize the original network structure.

The problem of community detection requires the partition of a network into communities of densely connected nodes, with the nodes belonging to different communities being only sparsely connected. Precise formulations of this optimization problem are known to be computationally intractable. Several algorithms have therefore been proposed to find reasonably good partitions in a reasonably fast way. This search for fast algorithms has attracted much interest in recent years due to the increasing availability of large network data sets and the impact of networks on every day life.

A new approach to the detection of communities implemented in this paper is that instead of trying to construct a measure which tells us which edges are most central to communities, the focus is instead on those edges which are least central, the edges which are most ‘between’ communities. Rather than constructing communities by adding the strongest edges to an initially empty vertex set, they are constructed by progressively removing edges from the original graph. Vertex ‘betweenness’ has been studied in the past as a measure of the centrality and influence of nodes in networks. The betweenness centrality of a vertex *i* is defined as the number of shortest paths between pairs of other vertices which run through *i*. It is a measure of the influence of a node over the flow of information between other nodes, especially in cases where information flow over a network primarily follows the shortest available path. In order to find which edges in a network are most ‘between’ other pairs of vertices, the algorithm generalizes Freeman’s betweenness centrality to edges and defines the edge betweenness of an edge as the number of shortest paths between pairs of vertices that run along it. If there is more than one shortest path between a pair of vertices, each path is given equal weight such that the total weight of all the paths is unity. If a network contains communities or groups that are only loosely connected by a few inter-group edges, then all shortest paths between different communities must go along one of these few edges. Thus, the edges connecting communities will have high edge betweenness. By removing these edges, the algorithm separates groups from one another and so reveal the underlying community structure of the graph.

The algorithm proposed for identifying communities in a social network is simply stated as follows:

1. Calculate the betweenness for all edges in the network.

2. Remove the edge with the highest betweenness.

3. Recalculate betweennesses for all edges affected by the removal.

4. Repeat from step 2 until no edges remain.

As a practical matter, the algorithm calculates the betweennesses using the fast algorithm of Newman [3], which calculates betweenness for all m edges in a graph of n vertices in time O(mn). Since this calculation has to be repeated once for the removal of each edge, the entire algorithm runs in worst-case time O(m2n). However, following the removal of each edge, we only have to recalculate the betweennesses of those edges that were affected by the removal, which is at most only those in the same component as the removed edge. This means that running time may be better than worst-case for networks with strong community structure (ones which rapidly break up into separate components after the first few iterations of the algorithm).

Summing up, Girvan-Newman algorithm is a hierarchal decomposition process where edges are removed in the decreasing order of their edge betweenness scores. This is motivated by the fact that edges connecting different groups are more likely to be contained in multiple shortest paths simply because in many cases they are the only option to go from one group to another. We plot the dataset on Gephi using Yiphan Hu Proportional and further we use R to identify the communities based on Girvan-Newman algorithm.

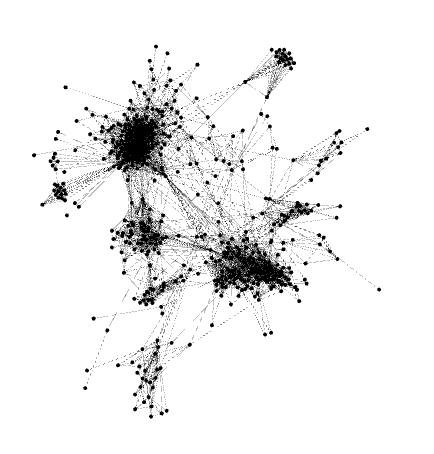


Fig: Graphical representation of Nodes

**Algorithm -**

# First we load the ipgrah package

library(igraph)

g <- read.graph("~/Desktop/Stuff/Masters Subjects/R Language/3437.edges", format= c("edgelist"))

# Checking for communities using the

# Grivan-Newman algorithm

# 1st we calculate the edge betweenness, merges, etc.

ebc <- edge.betweenness.community(g, directed=F)

vcount(g)

# Now we have the merges/splits and we need to calculate the modularity

# for each merge for this we'll use a function that for each edge

# removed will create a second graph, check for its membership and use

# that membership to calculate the modularity

mods <- sapply(0:ecount(g), function(i){

g2 <- delete.edges(g, ebc$removed.edges[seq(length=i)])

cl <- clusters(g2)$membership

# compute modularity on the original graph g

# and not on the induced one g2.

modularity(g,cl)

})

# We can now plot all modularities

plot(mods, pch=20)

# Now, let's color the nodes according to their membership

g2<-delete.edges(g, ebc$removed.edges[seq(length=which.max(mods)-1)])

V(g)$color=clusters(g2)$membership

# Let's choose a layout for the graph

g$layout <- layout.fruchterman.reingold

# Plot it

plot(g, vertex.label=NA)

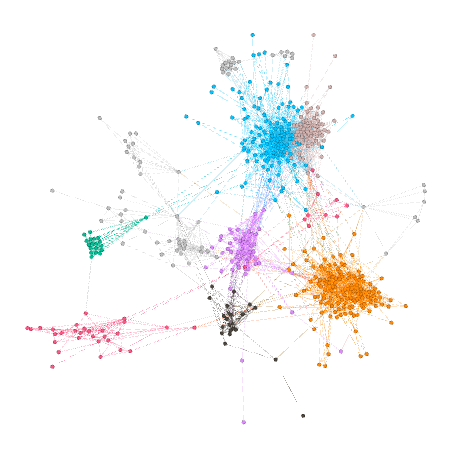


Fig: Colored member nodes of communities

We identify 11 communities each represented by a different colour on the graph.

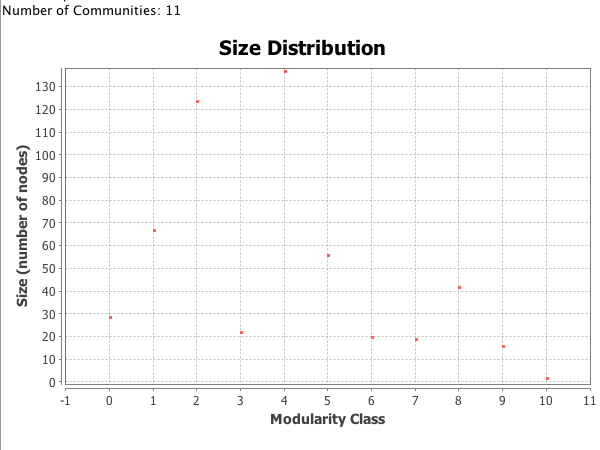


Fig: Size Distribution of the Modularity Class

communities(ebc)$'3447'

communities(ebc)$'3448'

neighbors(g,3503,mode = 'out')

are.connected(g, 3503, 3899)

We then find that Nodes 3503 and 3899 are connected but belong to two different communities 3477 and 3448.

**Results and Discussion**

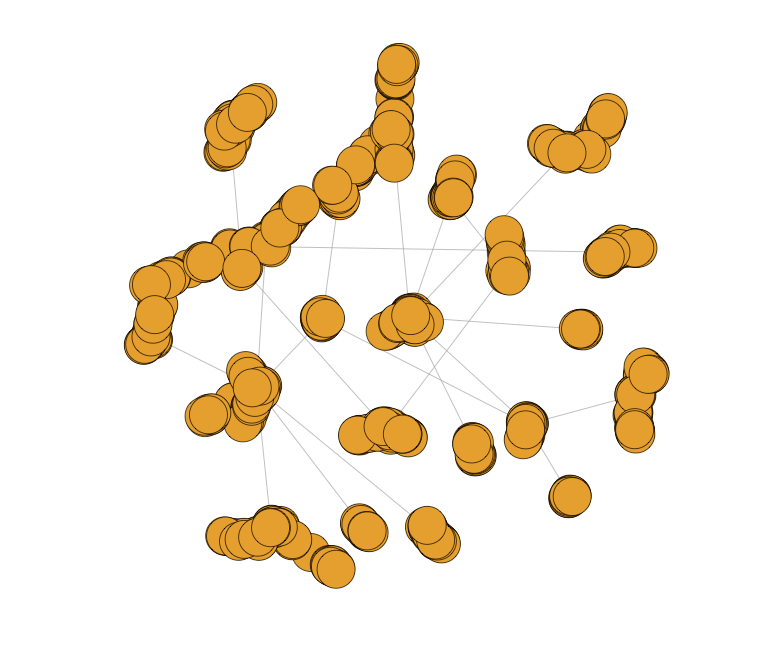


Fig: Diffusion simulation (from the highest probability node)

Diffusion probability of the two nodes identified in the social network through the Girvan-Newman is different and its diffusion would be different with its neighboring node.

**Related Work**

Google’s PageRank method was developed to evaluate the importance of web-pages via their link structure. The mathematics of PageRank, however, are entirely general and apply to any graph or network in any domain. Thus, PageRank is now regularly used in bibliometrics, social and information network analysis, and for link prediction and recommendation. To do this, Google designed a system of scores called PageRank that used the link structure of the web to determine which pages are important. More generally, we can consider random surfer models on a graph with an arbitrary set of nodes instead of pages, and transition probabilities instead of randomly clicked links. Teleporting is the essential distinguishing feature of the PageRank random walk that had not appeared before in the literature. It ensures that the resulting importance scores always exist and are unique. It also makes the PageRank importance scores easy to compute. Two uses underlie the majority of PageRank applications. In the first, PageRank is used as a network centrality measure. A network centrality score yields the importance of each node in light of the entire graph structure; the goal is to use PageRank to help understand the graph better by focusing on what PageRank reveals as important. It is often compared to or contrasted with a host of other centrality or graph theoretic measures. In the second type of use, PageRank is used to illuminate a region of a large graph around a target set of interest; for this reason, we call this second use a localized measure. It is also commonly called personalized PageRank based on the discussion of personalized teleportation behaviors in the original PageRank manuscript where the random surfer teleports only to pages that are interesting to the user.

The traditional method for detecting community structure in networks is hierarchical clustering. One first calculates a weight Wij for every pair i, j of vertices in the network, which represents in some sense how closely connected the vertices are. Then one takes the n vertices in the network, with no edges between them and adds edges between pairs one by one in order of their weights, starting with the pair with the strongest weight and progressing to the weakest. As edges are added, the resulting graph shows a nested set of increasingly large components (connected subsets of vertices), which are taken to be the communities. Since the components are properly nested, they can all be represented using a tree, in which the lowest level at which two vertices are connected represents the strength of the edge which resulted in their first becoming members of the same community. A ‘slice’ through this tree at any level gives the communities which existed just before an edge of the corresponding weight was added.

Many different weights have been proposed for use with hierarchical clustering algorithms. One possible definition of the weight is the number of node-independent paths between vertices. Two paths which connect the same pair of vertices are said to be node-independent if they share none of the same vertices other than their initial and final vertices. It is known that the number of node-independent paths between vertices *i* and *j* in a graph is equal to the minimum number of vertices that need be removed from the graph in order to disconnect *i* and *j* from one another. Thus this number is in a sense a measure of the robustness of the network to deletion of nodes. This and other pathologies, along with poor results from these methods in some networks where the community structure is well known from other studies, make the hierarchical clustering method, although useful, far from perfect.

**Conclusion**

As we have done for the new model, we try combine the distribution model to check about the diffusion process, and construct the new model for the diffusion process. We believe, with highest betweenness and degree, we have highest probability to diffuse to the other nodes connect by it. Also, the community detection for the diffusion process will take place for analysis the diffusion between two communities.

**Future Enhancements:**

1. More real data to analyze and compare with the diffusion results that are simulated on a specific network.
2. To check how this model functions and where it can be used in the real world. In order to know the aspects and applications.
3. Use community detection when 2 different nodes in different community which has connection between them and the communities have no intersection, check whether the diffusion model still works the same.
4. An other assumption about community detection is by considering each individual community (such that there is no intersection between the communities) as a node and construct a new graph and then check diffusion from one community to another.

**Bibliography**

1. Gleich, David. "Expanders, tropical semi-rings, and nuclear norms: oh my!."*XRDS: Crossroads, The ACM Magazine for Students* 19, no. 3 (2013): 32-36.
2. Granovetter, Mark S. "The strength of weak ties." *American journal of sociology* (1973): 1360-1380.
3. Girvan, Michelle, and Mark EJ Newman. "Community structure in social and biological networks." *Proceedings of the national academy of sciences* 99, no. 12 (2002): 7821-7826.
4. Romero, Daniel M., Brendan Meeder, and Jon Kleinberg. "Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter." In *Proceedings of the 20th international conference on World wide web*, pp. 695-704. ACM, 2011.
5. Kempe, David, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network." In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137-146. ACM, 2003.