COMP 400: Social Networks and Flow of Information

Supervisor: Prof. Joseph Vybihal

Sagar Nandeshwar - Student ID: 260920948

Abstract:

In this paper, I introduced a network framework to generate random graphs that simulate Social Network. The framework has many parameters which could be adjusted to align the graph with a particular subset of network and environment. I have generated graphs and their similarity with subset data from Twitter. With only a few adjustments in graph parameters I was able to achieve high similarity of Twitter data. I have also proposed a spread mechanism to track the flow of information in the social network. I populated the network with true and fake information and tracked the spread mechanism on my proposed network framework as well as the data collected from Twitter.

1. Introduction

Social network graphs are a visual representation of social relationships or connections between individuals or entities on the internet. It encompasses the arrangement of nodes (representing individuals or entities) and edges (representing relationships or interactions) that define the network's topology. They are incredibly important to understand and analyze various aspects of social interactions, user influence, information diffusion, recommendation systems, and community detection. In recent times it has become increasingly difficult to collect users' data from social media websites due to new and changing restrictions and privacy concerns. Also due to the amount of data on the internet, it is also infeasible to build real-world size graphs. Therefore, many big organizations are now opting to build random graphs to understand and analyze the dynamics of the internet.

Random graphs are mathematical structures that are used to model random or stochastic phenomena in various fields, including computer science, statistics, and network theory. They are generated by some random process, meaning that the graph's structure is determined by chance rather than by a specific, deliberate construction.

In this paper, I have proposed a network framework to generate random graphs that simulate Social Network. This framework follows inhomogeneous random graphs models to calculate the probability of edges between the nodes. I determine that this probability depends on the distance between the nodes, similarity between the node and influence of the nodes. The scale and significance of each of these three factors can be adjusted to match with the graph of particular interest.

Simultaneously, I have also proposed a spread mechanism to track the flow of information over the internet. This spread mechanism takes key features from Bootstrap percolation and First Pass percolation, which studies to track the spread of virus in human populations.

I finally tested the Network Framework and Spread Mechanism with the data collected from Twitter to evaluate its real-world significance.

2. Background

2.1 Graph Structure

A social network graph is represented as G(N, E), where N is the set of nodes that typically represent individuals, organizations, or entities, while E is the set of various types of connections or relationships between them. Nodes as well as the edges can have many attributes that define characteristics of the graph.

2.2 Previous studies

2.2.1 Erdős-Rényi (ER) model[1]

This is one of the earliest random graph models to simulate social networks. Here a graph G(n,p) model, a graph is constructed by connecting labeled nodes randomly. Each edge is included in the graph with probability p, independently from every other edge. Equivalently, the probability for generating each graph that has n nodes and M edges is

$$p^M(1-p)^{(nC2)-M}$$

The parameter p in this model can be thought of as a weighting function; as p increases from 0 to 1 the model becomes more and more likely to include graphs with more edges and less and less likely to include graphs with fewer edges.

The Erdős–Rényi model does not capture this clustering effect because it assumes that all connections are formed independently. The model assumes that each pair of nodes has an equal and independent probability of being connected. This is not reflective of social networks where the likelihood of forming connections is not uniform and is often influenced by shared attributes, geographical proximity, or existing network structures. The degree distribution in an Erdős–Rényi graph is binomial or Poisson, depending on the size of the network, which does not match the characteristics observed in real social networks.

2.2.2 Barabási-Albert (BA) model[2]

The BA model is scale-free, meaning their degree distribution follows a power law. Initially start with a small number (m_0) of connected nodes and gradually, at each time step add a new node with m (≤ m_0) edges that will be attached to different nodes already present in the system. When choosing the nodes to which the new node connects, do so with probability proportional to the number of links that the existing nodes already have.

$$pi = \frac{ki}{\Sigma k}$$

where k is degree of node i

In social networks, there are often tightly knit groups or communities, which the BA model's mechanism of preferential attachment does not naturally replicate. The BA model assumes that all nodes are identical in their ability to attract new links (apart from their degree), which is an oversimplification. In real social networks, nodes (individuals) can have varying intrinsic attributes that affect their connectivity, such as social status, geographical location, or interests.

2.2.3 The configuration model[3]

For $n \in N$, let k = (k1,...,kn): $\frac{1}{2}\Sigma k_i$ be a sequence of degrees. A graph G(V,E) is assigned ki "half edges" to vertex vi then choose two half edges uniformly at random and connect them, forming an edge. Repeat until no half edges remain unattached. The advantage of the configuration model is its tractability. For example, the clustering coefficient has a
compact expression, namely

$$\frac{(< k^2 >< k >)^2}{n < k >^3}$$

However, we can see that if we fix <k> and <k^2>and let n vary, the clustering coefficient vanishes with large n. This is a problem because the Twitter graph has a non-vanishing clustering coefficient.

2.2.4 Stochastic Block Model[4]

For n,k ∈ N, let B ∈ I^(k*k) and C:V->[k] be a surjection. Graph (V,E) is constructed n, connect vertices u and v with probability pu,v = BC(u)C(v)

It lacks the scale-freeness seen in real work networks; vertices in the same community have the same degree in expectation. Thus, it can't be an accurate Twitter analogue, either.

2.2.5 Directed Inhomogeneous Random Graph[5]

For n ∈ N, let w+ = (w1+,...,wn+), w- = (w1-,...,wn-) sequences of weights. In order to create graph G(V,E) with V=n, initialized a directed edge from vi to vu, when i not equal tp j probability with

$$p_{i,j} = \min\left\{1, \frac{w+\cdot w-}{n}\right\}$$

This model is a directed generalization of the Chung-Lu random graph model. Its directed edges help it capture the directed links on Twitter. However, it lacks geometry, and thus cannot encode the clustering and communities seen on Twitter.

2.2.6 Geometric Inhomogeneous Random Graph

For n ∈ N, let w (w1,...wn) be a sequence of positive weights. Let W = Σw the total weight. For a dimension d > 0, call the ground space the d-dimensional torus T^d. Use the inf-norm as a distance function of the ground space, such that, for x,y∈ T^d |x-y| is define as max_i |xi-yi|. In order to construct a graph G(V,E), connect vertices u and v with probability

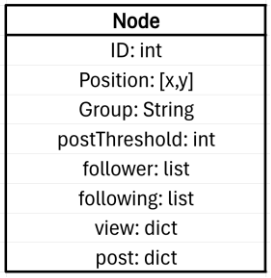
$$p_{u,v} = \Theta\left(\min\left\{1, \frac{1}{|xu - xv|^{\alpha d}} \cdot \left(\frac{wu \cdot wv}{W}\right)^{\alpha}\right\}\right)$$

3. Inhomogeneous Random Graph to consider direction, geometric location, similarity and node’s influence.

An inhomogeneous random graph is the graph where the chance of connection between two nodes depends on the probability which is not the same for all pairs if the nodes. In this paper I suggested that this probability depends on 3 main points, Distance, Similarity and Influence. The edges are directed, which represents the flow of information.

3.1 Network Structure

3.1.1 Node



Node

ID: int

Position: [x,y]

Group: String

postThreshold: int

follower: list

following: list

view: dict

post: dict

The nodes represent users of the social network. They have following attributes

ID: Unique identification of the user

position: The location of the user.

group: The community belonging to

Follower: List of users that follows them, i.e. all the users that reads this node’s post

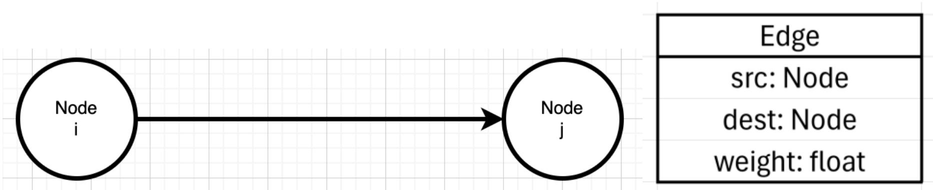
Following: List of users this node follows, i.e. all the users that who post are accessible to this node

view: List of all the messages that are posted on this node’s following users

post: List of all the messages this node posts and are available to its followers.

postThreshold: Willingness to post a message present in the view

3.1.2 Edges:



Edge

src: Node

dest: Node

weight: float

Edges represent the relation between nodes, more specifically they represent flow of information. In the proposed model we have directed edge from node i to node j.

node i’s follower is node j; node j view node i post

node j us following node i;

3.1.3 Probability of Edge

The probability of on directed edge from node u to node v depend on the following factors

1: Distance Between the nodes

People will tend to follow other users that are present in the same geometric location, as they share the same physical environments and events happening around them. Hence, the Pr(u,v) = probability of directed edge from node u and node v is inversely proportional to the distance between them |xu-xv|, such that

$$\Pr_d\big((u,v)\big) = \frac{D - d(u,v)}{D} \cdot r \cdot df$$

where D is the dimension of the entire geometry of the graph, d(u,v) is the Euclidean distance between node u and node v, and r is the radius (or distance boundary), and “df” is distance factor

2: Similarity between the nodes

Based on The Laws of Attraction, people tend to follow other people who share similar thoughts. Hence the Pr(u,v) id directly proportional to the similarity between the nodes. In this paper, we define two nodes as similar if they belong to the same group.

$$\Pr_s\big((u,v)\big) = 0.5\ +\ SimilairtyScore \cdot I_s \cdot sf$$

Is is +1 for nodes belonging to same group and -1 nodes of different group. SimilarityScore gives is probability that the two nodes with same group have edge. sf defines the significance of similarity in the edge between u and v.

3: Influence factor of the node

People tend to follow and believe the views that are said by famous and influential people. We define a node influence based on its weight.

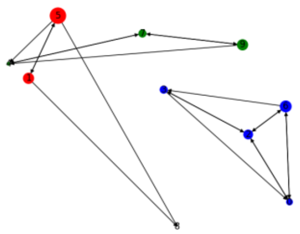
$$\Pr_i(u) = \frac{follower(u)}{N} \cdot \frac{follower(u)}{following(u)} \cdot if$$

‘if’ is influence factor that determines the significance of influence in the edge between u and v.

3.2 Building graph

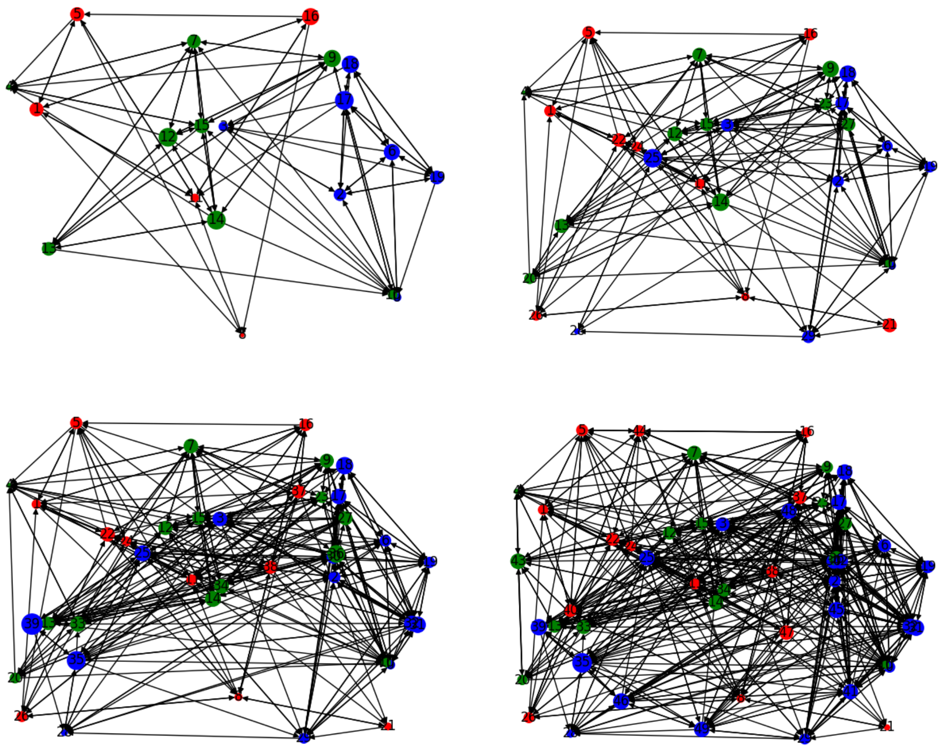
3.2.1 Initialization Phase

At the beginning I initialize the nodes with random location, random group and natural influential factor. The neutral influential factor means that node’s influence factor plays no role in the probability of edge between them, hence edge probability solely depends on the distance and similarity between them.



3.2.2 Expansion Phase

Gradually, I started adding a new set of nodes to the graph. These nodes again have random location and random groups drawn from the same probability distribution as before, however the node influence is drawn from normal distribution.



3.3 Graph Properties

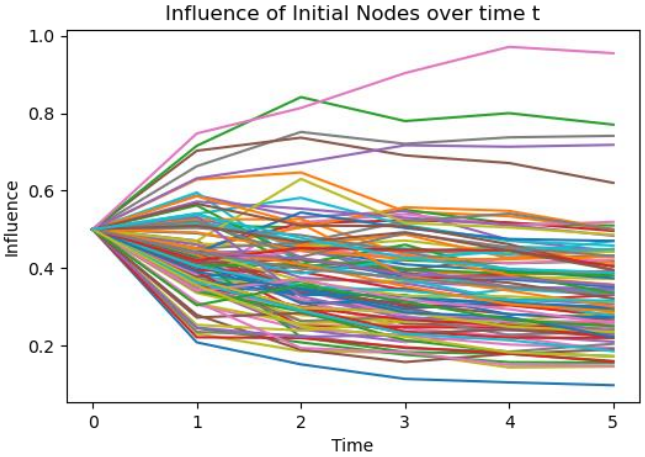
The proposed model seems to follow the following properties of the graph,

3.3.1 Power laws and scale free network[7]

When we say that a graph follows a power-law distribution, it means that there is a small number of highly connected nodes (called hubs) and a large number of nodes with relatively few connections. These hubs play a crucial role in the network and can have a significant impact on information flow, robustness, and network dynamics.

This phenomenon is also known as the “rich get richer” which is associated with preferential attachment, where new nodes entering the network are more likely to connect to existing nodes that already have a high degree. Hence the nodes with high degree tend to accumulate connections or links over time.

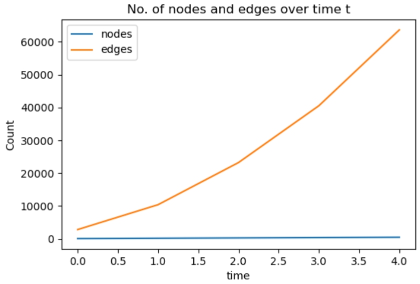
Graphs that follow a power-law distribution are often referred to as "scale-free networks” because the distribution of node degrees remains consistent regardless of the size or scale of the network.

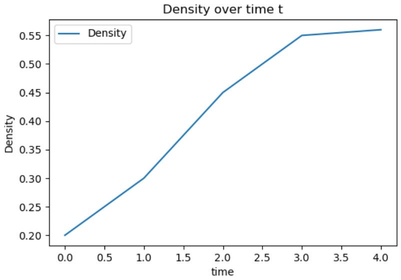


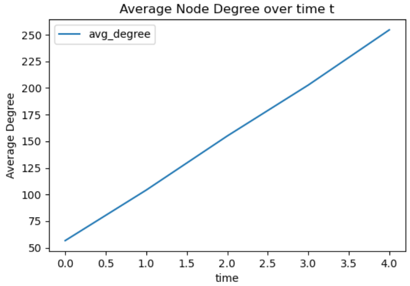
I build a graph with 100 initial nodes and observe their influence as I gradually add more nodes to the graph. I observe that only a few nodes have high influence factor while they have normal or very low influence factors. Those who have high influence tend to increase it over time, while others remain low and consistent.

3.3.2 Increase in network density over the time[8]

Network density measures the proportion of possible edges that are present in the network. Most of the social networks the network density increases over time.







As the number of nodes increases in the graph, the number of edges, graph density and the average node degree increases.

3.3.3 Small world phenomena[9]

In many real-world networks, individuals or nodes are connected to each other through relatively short chains of acquaintances, even in large and complex networks, i.e even in large networks, the average distance between nodes is relatively small. This phenomenon is often encapsulated in the famous "six degrees of separation" idea.

We observe the diameter (i.e. the maximum eccentricity) and average shortest distance of the graph is 3 and 1.73 respectively. Hence, we can say that the graph follows small world phenomena.

(I calculated Graph centrality and clustering to be compared with the data collected From Twitter)

3.3.4 Centrality:

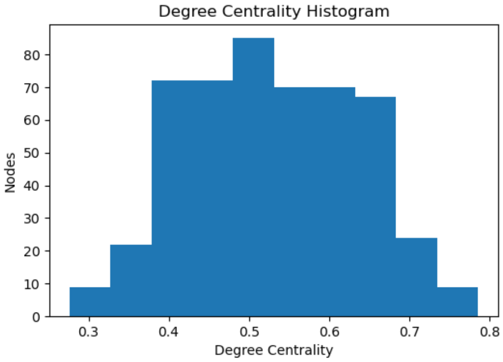
Centrality measures identify and quantify the importance or significance of nodes (individual entities) within a network. It helps to identify nodes that play important roles in the network such as influential individuals or entities who act as bridges, connectors, or opinion leaders within the social network.

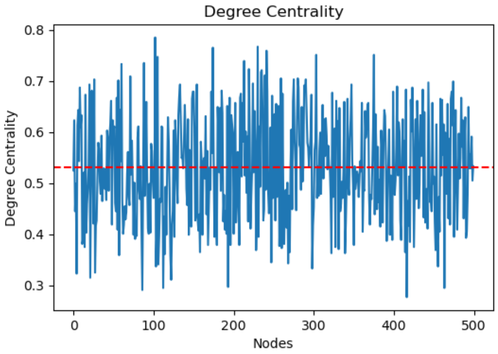
Degree Centrality:[10]

Degree centrality measures how many direct connections (edges) a node has. Nodes with higher degree centrality are often seen as more central or influential. Degree centrality of a node u is the fraction of nodes it is connected to. It is defined as

$$D(u) = \frac{\deg(u)}{N-1}$$

where N is the total number of nodes in the network



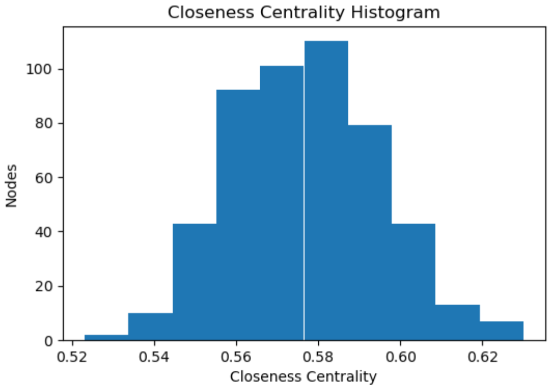


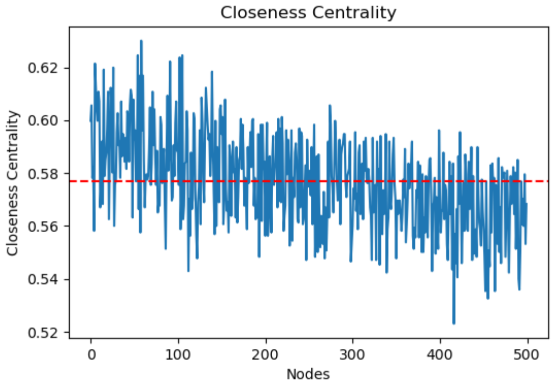
Closeness Centrality:[11]

Closeness centrality measures how quickly a node can reach all other nodes in the network. Nodes with higher closeness centrality are more central in terms of overall network accessibility. Closeness centrality of a node u is the reciprocal of the average shortest path distance to u over all n-1 reachable nodes.

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} \quad d(u,v)}$$

where d(v, u) is the shortest-path distance between v and u, and n-1 is the number of nodes reachable from u.





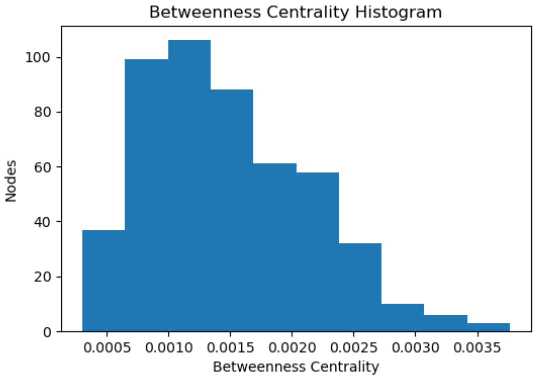
Betweenness Centrality:[12]

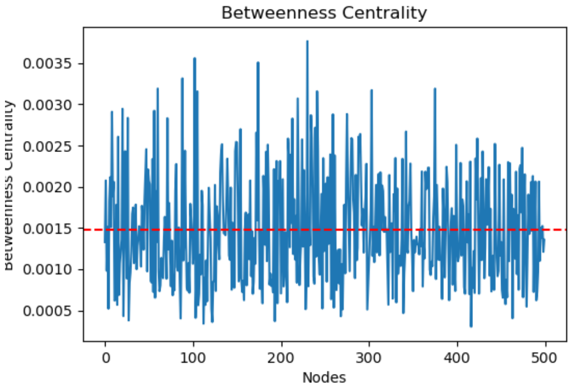
Centrality betweenness identifies nodes that act as bridges or intermediaries in the network. Nodes with high betweenness centrality lie on many of the shortest paths between other nodes. Betweenness centrality of a node v is the sum of the fraction of all-pairs shortest paths that pass-through v

$$c_b(v) = \Sigma_{s,t\in V} \frac{\sigma(s,t;\ v)}{\sigma(s,t)}$$

where V is the set of nodes, \sigma(s,t) is the number of shortest (s,t) path paths, and $\sigma(s,t|v|)$

is the number of those paths passing through some node v other than s,t





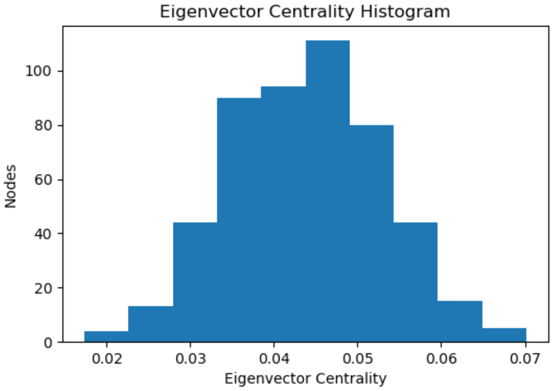
Eigenvector Centrality:[13]

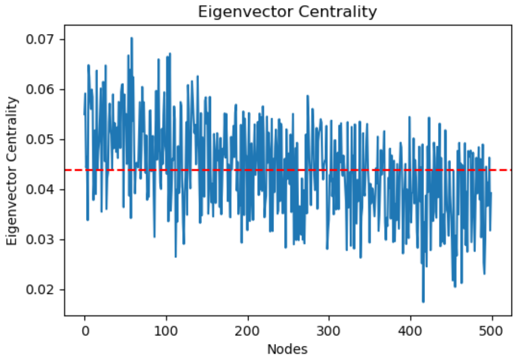
Eigenvector centrality measures a node's importance based on the connections of its neighbors. A node is considered central if it is connected to other central nodes.

Eigenvector centrality computes the centrality for a node by adding the centrality of its predecessors. The centrality for node i is the i-th element of a left eigenvector associated with the eigenvalue \lamda of maximum modulus that is positive. Such an eigenvector x is defined up to a multiplicative constant by the equation

$$\lambda x^T = x^T A$$

where A is the adjacency matrix of the graph G.





3.3.5 Clustering coefficient

The clustering coefficient measures the extent to which nodes in a network tend to cluster together. It quantifies the presence of triads (groups of three nodes and their connections) in the network. High clustering suggests that nodes tend to have connections to each other's connections, indicating a tightly knit community.

Clustering coefficient:

The clustering of a node is the fraction of possible triangles through that node that exist

$$c_u = \frac{2T(u)}{\deg(u)\,(\deg(u) - 1)}$$

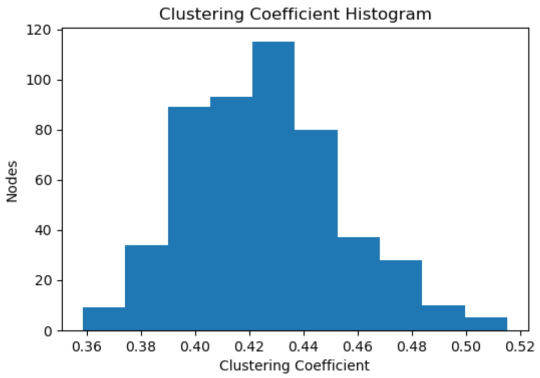
where T(u) is the number of triangles through node u and deg(u) is the degree of u.

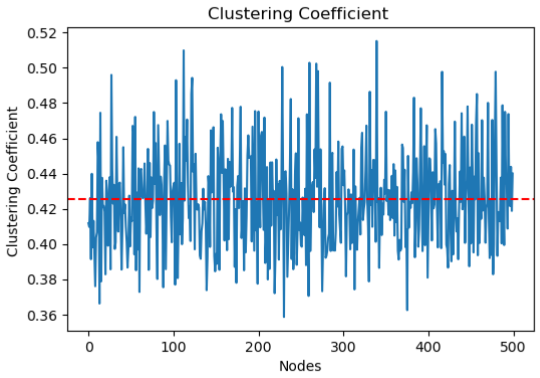
average clustering

The clustering coefficient for the graph is the average,

$$C = \frac{1}{n}\Sigma_{v \in G} c_v$$

where n is the number of nodes in G.





3.4 Comparison with Twitter data

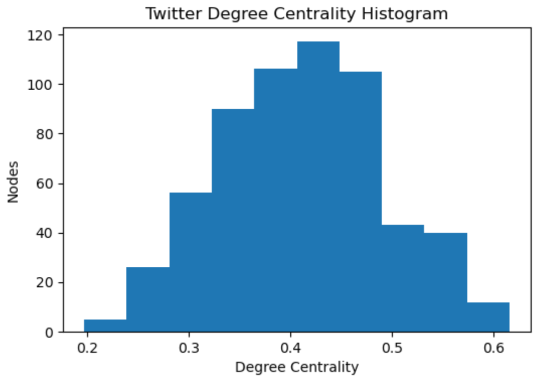
Twitter is one of the most popular and influential social networking sites with 500 million active users monthly. The site is highly dynamic and has a huge impact on the real-world.

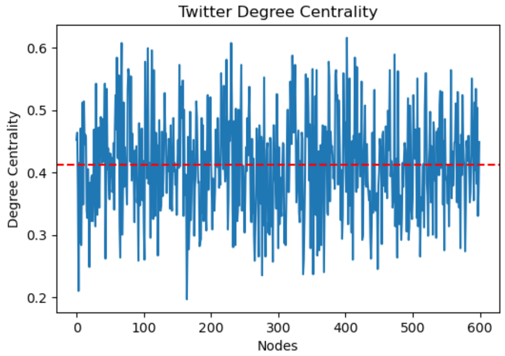
Data collection:

For the last 4 months, I have been searching for a set of nodes that are clustered together, Twitter data is very diverse and in general has a very low clustering. This is due to the fact that users usually tend to follow more influential accounts than normal users, so it was challenging to find the dataset that is clustered together. I was also able to collect such from 2018-2020 involving elections and politics. I found nearly 1000 users that are more than usually clustered together. I proposed the data to remove any spam accounts and build a graph with 800 nodes.

I build a random graph with the proposed network structure by adjusting the parameters to match with the collected as closely as possible. The following is the result.

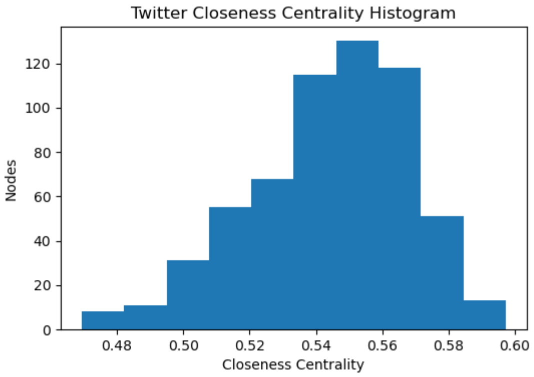
Degree Centrality:

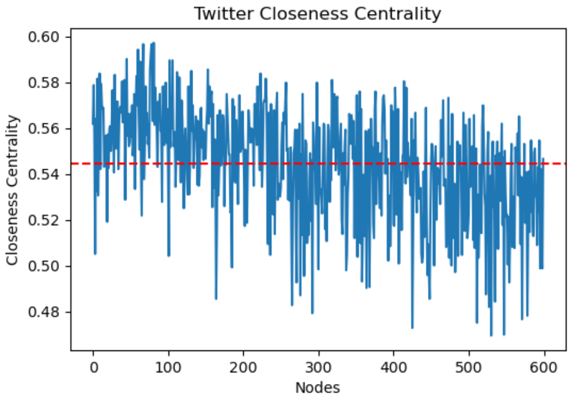




The Degree Centrality that we have obtained from our random graph is very similar to Twitter data. The average Degree Centrality that we have got is around 0.54 whereas for twitter it is 0.41. Also, both twitter data and my random graph seems to follow normal distribution for Degree Centrality for the nodes

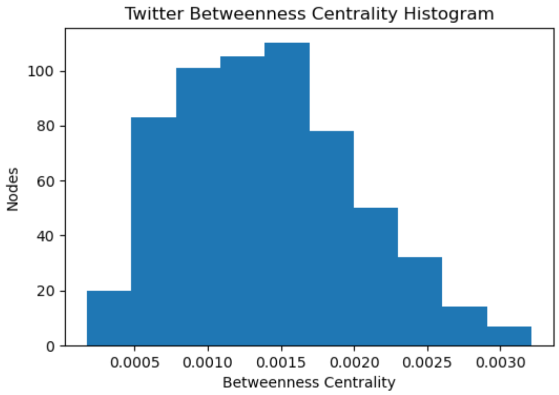
Closeness Centrality:

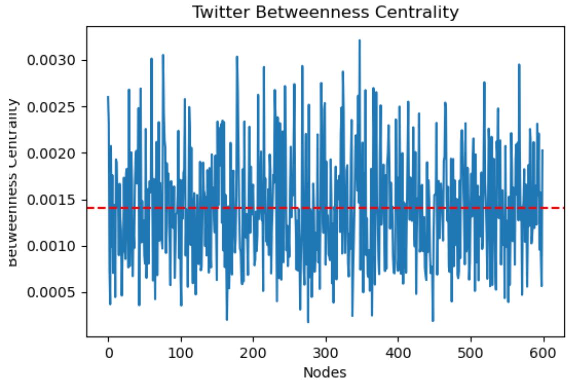




Again, my random graph and twitter are very similar for closeness centrality. While mine average closeness centrality is 0.57 and twitter closeness centrality 0.55. For both twitter and my random graph no. of nodes with closeness centrality above or below 0.56 gradually decreases as we move away from 0.57 closeness centrality.

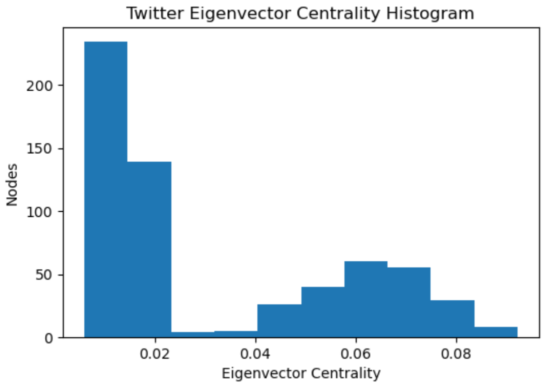
Betweenness Centrality:

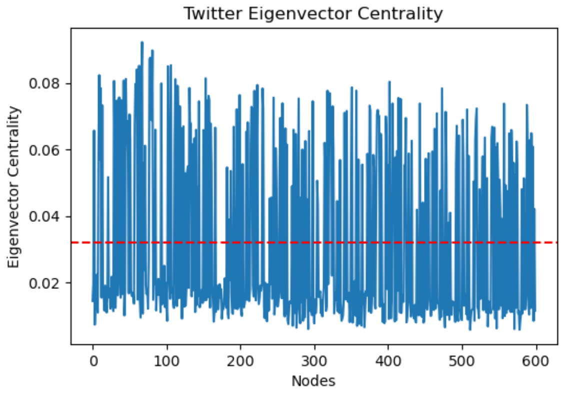




Again, my random graph and twitter have similar Betweenness centrality. While our average Betweenness centrality is 0.0015 and twitter closeness centrality 0.0014. For my graph the majority of nodes have Betweenness Centrality of 0.0010 or less, the number of nodes with higher Betweenness Centrality gradually decreases. For twitter data Betweenness Centrality for most of the nodes is 0.0015 or less and again the number of nodes with higher Betweenness Centrality gradually decreases

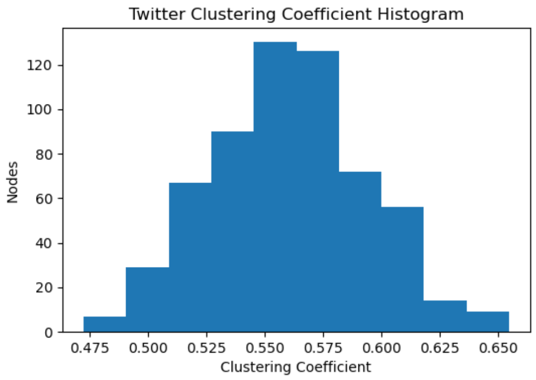
Eigenvector Centrality:

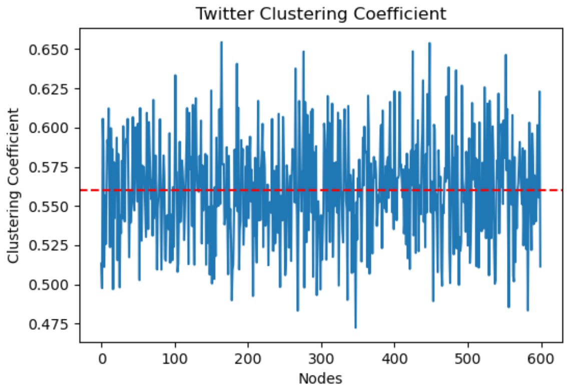




For Eigenvector Centrality our average is 0.05 whereas Twitter average is 0.03. While in our graph most of nodes have Eigenvector Centrality 0.05 or less, where else for twitter it is a bit different. Most of the nodes have Eigenvector Centrality 0.02 and good amount nodes have Eigenvector Centrality have higher than 0.05.

Clustering coefficient





Clustering coefficient is almost similar in our random graph and twitter data. For us the avg Clustering coefficient is 0.43 whereas in twitter it is 0.56. Our graph as well as twitter data seems to follow normal distribution of nodes for Clustering coefficient

Twitter Graph Diameter: 4

Average shorted path: 2.1

Lastly Twitter have Graph Diameter of 4 and Average shorted path: 2.1, we have a bit more clustered data with Graph Diameter of 3 and Average shorted path: 1.7

While making the random graph I gave more significance to the distance and similarity and node’s influence, i.e. these factors have more impact than node influence. This shows that the users from the collected twitter data are living in close geometric locations and believe in similar policies. The lack of node influence shows that the political views of the user are not influenced by famous entities.

3.6 Range of Parameters

groupList: List of all the groups that could be assigned to a node.

groupListProb: Define frequency of each group in the graph

D_dimension :Dimension of the map on which graph is created

strictness: Control the biases of edge probability. Usually in the range of 1-10. The higher the strictness is the more biased the model becomes.

similarity_score: Controls of the probability of edge between the two nodes given their respective groups. Increasing the similarity score increases the probability of edge between two same groups and decreases the probability of edge between different groups, and vice versa. In the range of 0-0.5.

similarityFactor: It multiplies the similarity scores. Preferred in the range of 0-2

distanceBoundary: Radius from a node within which the probability of edge increases and decreases as we move out of the radius. Usually in the range of 0-1

distanceFactor: It multiplies the distanceBoundary. Preferred in the range of 0-2

influenceFactor : Control the significance of sender node influence on the probability of edge. Preferred in the range of 0-2

activeStateProb: Controls the probability of nodes being active or not. Preferred in the range of 0-1

4. Spread Mechanism

Information travels across the network through edges and nodes. This spread of information is very similar to the spread of viruses and disease in the human population. I analyzed two common Percolation methods - Bootstrap Percolation and First Pass Percolation - that were used to study the spread of viruses in organisms and proposed a new method that takes into consideration the important features of the two methods. Finally, I analyze the proposed spread of information on my proposed network structure as well as Twitter data.

5. Background Studies

5.1 Bootstrap percolation [15]

In Bootstrap percolation, we start with a lattice or network where each node or cell can be in one of two states: active or inactive. A node becomes active if a certain number of its neighbors are active. This threshold is predefined and is a key parameter of the model. The activation process is iterated, inactive nodes check their neighbors and may switch to the active state if the rule is satisfying. With this more nodes become active over time, depending on the state of their neighbors.

Let G(V,E). Let A0 be the set of initial active states, then At the set of active of nodes at time t, is define as

At = At-1 \union {u\belongs V | N(u) \intersection At-1 > 2}

where N(v) is the set of vertices adjacent to vertex u in G.

The model is intuitively rational that people tend to believe in something if more and more people around them start talking about it. However, the model is highly deterministic, as it can predict the active state of the cell given information about the grid and infected sites. This not exactly synonymous with the real world, as the real world tend to be much more complex

5.2 First-passage percolation [16]

Each edge of the lattice is assigned a random weight. These weights are usually positive numbers and can be thought of as representing the time or cost it takes for a condition to pass through that edge. The distribution of these weights is a critical aspect of the model and is typically determined beforehand, often using a probability distribution like a uniform or exponential distribution. The process begins at a specific site or a set of sites on the lattice. This is the origin from which the condition starts spreading. The condition spreads across the edges from the starting point. The key aspect here is that the spread through each edge takes time/or cost, as determined by the weight of that edge. The primary quantity of interest in this model is the "first-passage time" from the starting point to another site or set of sites. This is the minimum time it takes for the condition to reach these sites, considering the weights on the edges.

Let G(V,E) be the graph and weight of the edge e between node u and v, We which is i.i.d. copy of the random variable W. Let path pi = (pi0,...,pik) is set of vertices connected by edges, such that the cost of the path is sum of the edges used in the path, that is

$$c(\pi) = \sum_{i=0}^{k-1} L(\pi_i, \pi_{i+1})$$

Let be the set of paths starting in the vertex set A and ending in the vertex set B. The distance

between the two sets of vertices as the cost of the cheapest path between them:

$$d_G(A, B) = \min_{\pi} \quad c(\pi)$$

Finally, the budget of a piece of information as a positive random variable B with some probability distribution, and the spread of that piece of information, conditioned on it originating from vertex v0, as the random variable Uv0 = {v\belongsV:d(v0,v)<B}

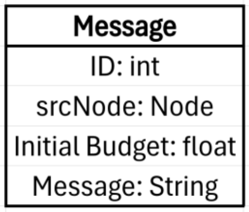
However, the spread of information only depends on the initial budget of the message and edge cost. The transverse of information is independent on Neighbours, which is not the generally the case in the real world.

6. Spread Mechanism Model

I proposed a new spread mechanism that takes into consideration the cost of travel from source to destination node and as well as the impact of neighbours on that cost of travel.

6.1 Structure

6.1.1 Message:



Message
ID: int
srcNode: Node
Initial Budget: float
Message: String

6.2.2 Edge weight

The weight of edge represents the cost it takes for a message to travel from node i to node j. This cost depends on two factors.

Similarity between the node i and node j: If the nodes are similar, the weight of the edge will be less and vice versa. This is based on the intuition that people tend to believe information that is shared by alike users.

Influence of node i: The more influential the node i (sender) is, the lesser the weight of edge will be. This is based on the intuition that things said by famous and influential people are more trusted by the user than the normal users.

$$c(e) = w_e \cdot \inf luence(i)$$

6.2.3 post thresholds:

Represent the tendency of nodes to post a message that they view on their profiles.

6.3 Simplified Pseudo Algorithm:

```
Graph
Seed(root)
for message in view():
if root.budget(message) > root.postThreshold
root.post(message, budget)
root.inform_followers()

Nodes:
-> post information receives from node A
if message is in this.view:
this.budget(message) = this.budget(message) + A.budget(message)
else:
        this.view.add(message)
        this.node.budget(message) = A.budget(message)
if this.budget(message) > postThreshold
this.post(message)
this.inform_followers()
```

6.4 Flow of information

1: A source node (seed) will create a message and give an initial budget.

2: This message is passed on to the source node Neighbours.

While passing through each edges the budget of the message is decreases by the cost of the edge

3: Nodes accumulated the message received by the Neighbour.

The budget of the same message keeps on adding as more and more messages are getting accumulated from the Neighbour

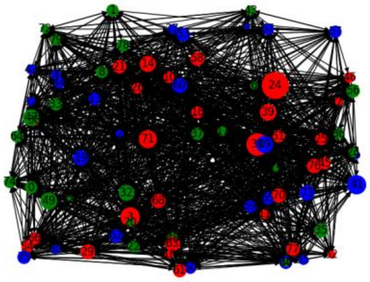
4: If budget of the message in node n crosses its post thresholds, node n will post the message in its profile which could be view by its followers

5: This cycle will continue till all the nodes have published the message in their view or have budget of zero or less than their post threshold for that message.

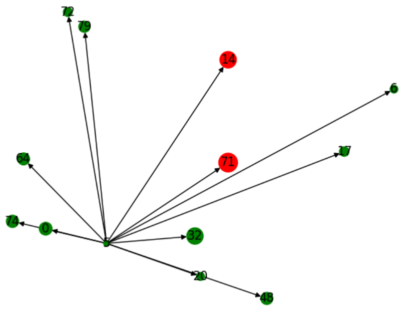
Lastly, with a very small probability some nodes could be inactive and do not participate in the information flow.

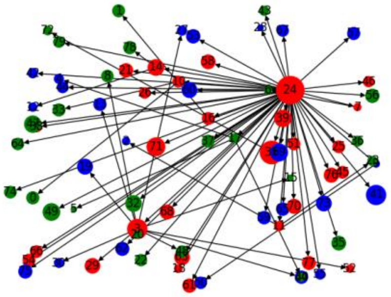
6.5 Flow of information from most and least influential node:

I created a graph with 80 nodes based on my proposed framework. I then selected the most influential and least influential nodes from the graph as my seed. I ran the algorithm and observe the following results



Graph





Information Flow with

Low Influence

Information Flow with

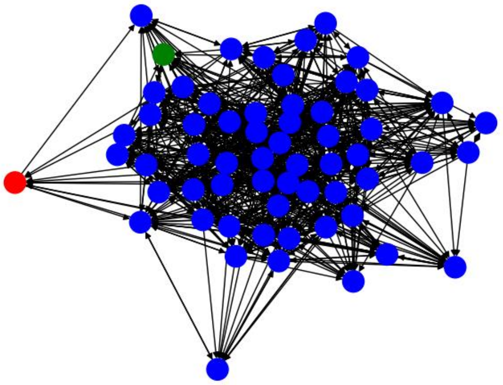
High Influence

The message spread by the most influential node is received by nearly all of the nodes on the graph. There were only 14 bottleneck nodes that stopped the flow of information and only 7 nodes that did not receive the information. On the other hand, message spread by the least influential node was only able to reach 12 nodes. There were only 12 bottleneck nodes, and 67 nodes did not receive the message. This shows how a user's influence impacts the spread of information on the internet.

(Bottlenecks are the nodes that have viewed the message, but the message budget is lower than their post threshold in order to post that message.)

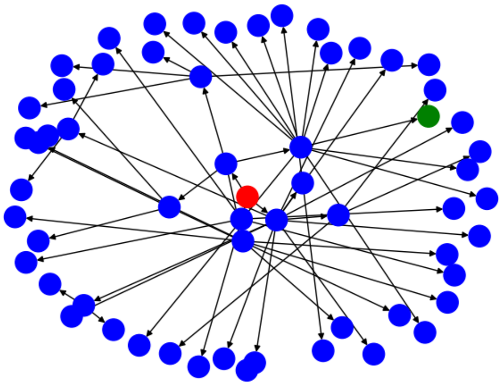
6.6 Flow of Information on Twitter:

To test the spread mechanism, I searched for a celebrity, say node A (green), active on twitter that retweeted a tweet from a lesser-known user, node B (red). Node A does not follow node B, but still received the tweet and decided to retweet on their profile. This shows that there is a connection as well as information flow from node B to Node A. I search from Node B followers and following and build a graph with node A and Node B collected. Finally, populate the graph with a message from node B as seed and get the information flow sequences. The sequence is likely the path of information flow based on my algorithm.



Connect Twitter Graph

with Node A and B

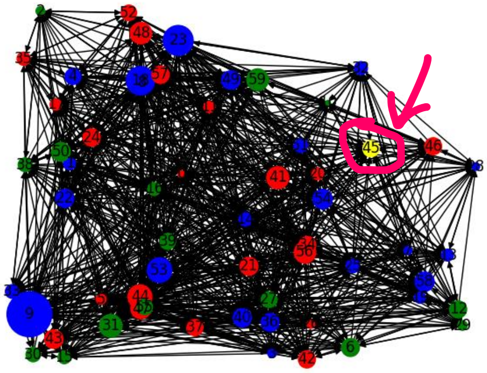


Graph Flow of

Information

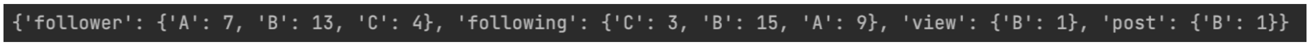
6.7 Unknown Group Detection:

To test the application of network and algorithm, I added a node with unknown group and random location and influence on a graph.



In order to get the information of the node's group, I collected information about its followers and following. I also populated the network with two messages from different sources and observed the target node's view and posts.

Output:



```
{'follower': {'A': 7, 'B': 13, 'C': 4}, 'following': {'C': 3, 'B': 15, 'A': 9}, 'view': {'B': 1}, 'post': {'B': 1}}
```

'follower': {'A': 7, 'B': 13, 'C': 4} 'following': {'C': 3, 'B': 15, 'A': 9} 'view': {'B': 1} 'post': {'B': 1}

With the above information we can say the node belongs to the group "Blue" with some high confidence.

7. Conclusion

The proposed network frameworks build a random graph based on distance between the nodes, similarity between the nodes, and node's influence. All of these can be adjusted to align the model with a specific subset. Graph has followed the general social network behaviors such as scale free growth, power Law, increase in network density over the time and small world phenomena. The graph has also shown very similar properties when compared to real Twitter data. We have also seen how the network framework can be used to predict the user characters, which could be useful for recommendation systems. We also collect the count of messages that could be used to detect spamming of messages. The network framework can be used to use the dynamics of the users based on the parameters of graph. These parameters can also be studied more to build an ML model for graph generation.

The proposed spread mechanism has incorporated important features First-passage percolation and Bootstrap percolation. The data traveling over the network decays over the time and distance, however for a particular node it could regain its budget based on accumulation from Neighbours. These features show similar behaviors spread information over the internet, as a message may not directly travel around the globe but gradually reach from place to place. With the proposed spread mechanism, we are able to study the importance of nodes' influence on the information flow. We have also seen how the spread mechanism can be used to track the information flow.

8. Deliverables Table

S.No.	Task	Developer	Data
1	Resource Collection	Sagar Nandeshwar	Sept 23rd, 2023
2	Network Framework	Sagar Nandeshwar	Oct 26th, 2023
3	Data Collection and Processing	Sagar Nandeshwar	Nov 15th, 2023
4	Flow of Information	Sagar Nandeshwar	Dec 5th, 2023

Information for Next Development Team

The network framework and spread mechanism is built in python. All of the source code can be found on src folder. I have also collected two sets of Twitter data as mentioned on the paper and stored it in the data folder.

9. Reflection

What would you do again (what have you learned positive) Future Work?

I would study the flow of information in context of fluids and energy potential to include a more dynamic version of spread mechanism. I have really enjoyed the time mining social media networks and learned a lot about our society social psychology. I have learned how significant graph theories and graph structures are in our day-to-day lives on the internet.

What would you do differently (what have you learned negative)?

I would narrow down my research to work on a particular topic and build my graph and spread algorithm accordingly. The data in internet is very diverse but some topics have structure and particular flow of data. This would have helped me to better understand the structure of the graph and the parameters involved. I would also avoid using Twitter as not only is it very difficult to retrieve useful information, but also the platform is not as active as it was a few years ago. I would have also explored ML based model for graph generation and Reinforcement Learning based approach for spread Mechanism.

10. Things to Fix Features to Add Suggestions

My model assumes that the parameters of the Random Graph are independent of another. However, if we were able to define a concreate relation between them, we will be able to reduce the no. of parameters, which will reduce the complexity of the model

The spread of information is not done distributed system. This was important to run the algorithm in feasible amount of time. However, alternative a node can trigger its own spread of information creating chain of information flow. This may be a very expensive computation, but it is closer to the real world spread of information.

Reference:

[1] Erdős, P.; Rényi, A. (1959). "On Random Graphs. I"

[2] Albert, Réka; Barabási, Albert-László (2002). "Statistical mechanics of complex networks"

[3] Network Science by Albert-László Barabás

[4] Holland, Paul W; Laskey, Kathryn Blackmond; Leinhardt, Samuel (1983). "Stochastic blockmodels: First steps"

[5] Bootstrap Percolation in Directed Inhomogeneous Random Graphs- Thilo Brandis et al.

[6] Geometric inhomogeneous random graphs - Karl Bringmann et al.

[7] Onnela, J.-P.; Saramaki, J.; Hyvonen, J.; Szabo, G.; Lazer, D.; Kaski, K.; Kertesz, J.; Barabasi, A. -L. (2007). "Structure and tie strengths in mobile communication networks"

[8] Evaluating the Evolution of Social Networks - Melissa B. Scribani et al.

[9] John F. Buford “Networking and Applications” 2009

[10] Peter V. Marsden, in Encyclopedia of Social Measurement, 2005

[11] Linton C. Freeman: Centrality in networks: I. Conceptual clarification. Social Networks

[12] Ulrik Brandes: A Faster Algorithm for Betweenness Centrality. Journal of Mathematical Sociology 2001

[13] Abraham Berman and Robert J. Plemmons. “Nonnegative Matrices in the Mathematical Sciences.” Classics in Applied Mathematics. SIAM, 1994.

[14] Generalizations of the clustering coefficient to weighted complex networks by J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, and J. Kertész, Physical Review E, 75 027105 (2007)

[15] Chalupa, J.; Leath, P. L.; Reich, G. R. (1979), "Bootstrap percolation on a Bethe lattice",

[16] Hammersley, J. M.; Welsh, D. J. A. (1965). "First-Passage Percolation, Subadditive Processes, Stochastic Networks, and Generalized Renewal Theory".