SEED SELECTION FOR INFLUENCE MAXIMIZATION IN SOCIAL NETWORKS

BS Project Report by Rishabh Bhonsle
Roll Number: 17217
IISER Bhopal

This project was created under the guidance of
Dr. Kundan Kandhway

ABSTRACT

From rumors to marketing strategies, we observe ideas and influence propagating in a population through promotional events or by "word of mouth". But how does this information actually spread in a given social network? If one had to influence a given population (social network) or spread information through a given population, with limited resources, which individuals should he/she select as starting points (seeds)? These are some fundamental questions that one comes across during a social network analysis. All these questions would be answered through the model we build in this project based on the paper "Maximizing the Spread of Influence through a Social Network" by David Kempe, Jon Kleinberg and Eva Tardos.

The optimization problem of selecting the most influential nodes is NP-hard here. We thus approach the problem by analyzing the solution of a NP-complete problem - The Set Cover Problem based on a natural greedy strategy and then reduce the problem to our Influence Maximization Problem. As the problem at hand is NP-hard, we provide provable approximation guarantees for the best possible solution in polynomial time. We find that this solution is provably within 63% of optimal.

To conclude, we then provide computational experiments on a population network showing that in addition to their provable guarantees, our approximation algorithm significantly out-perform node-selection heuristics based on the well-studied notions of degree centrality, betweenness centrality and pagerank from the field of social networks.

INTRODUCTION

PROBLEM STATEMENT

Given a social network – the graph of relationships and interactions within a group of individuals, a diffusion model – the dynamics of spread of information through that network and a seeding budget – number of few “influential members” that can be chosen in the network to trigger a cascade of influence to spread the information we desire; the problem lies in choosing the most optimal “influential members” such that the spread of information through the triggered cascade is maximum in the network.

MOTIVATION AND SCOPE

A social network plays a fundamental role as a medium for the spread of information, ideas, and influence among its members. An example provided in the paper "Maximizing the Spread of Influence through a Social Network" by David Kempe, Jon Kleinberg and Eva Tardos are the use of cell phones among college students, the adoption of a new drug within the medical profession, or the rise of a political movement in an unstable society. If we want to understand the extent to which such ideas are adopted, it can be important to understand how the dynamics of adoption are likely to unfold within the underlying social network. Such network diffusion processes have a long history of study in the social sciences. We shall study a few of them here. But in real world networks, populations form huge social networks. Thus if we desire for an entire population to adopt a certain idea or an innovation or stop the spread of an uprising or a political movement at its roots, the

main question then arises is what part of the population do we target as the "early adopters" of the innovation or the "early influencers" of a movement such that they eventually influence a maximum portion of the population as it becomes almost impossible to target each and every node of the population. Thus, lies the motivation to our problem.

REPORT OUTLINE

Before we get to the problem, we need to familiarize ourselves with some required pre-requisites. We shall follow a brief outline as follows.

1. Introduction to Networks
   In this section we shall quickly familiarize ourselves with the basic aspects of a network and study about few common degree centralities.
2. Defining a Network for our Problem
   In this section we shall choose a small population network based on which we shall model our problem.
3. Choosing a model for the Spread of Information in our Network
   In this section we shall discuss two popular methods in sociology by which information/influence would propagate through population networks and shall model them on ours.
4. The Influence Maximization Problem
   Finally, in this section we start off by defining the NP-complete "Set Cover Problem" and analyze its natural greedy approach to the solution. We then reduce this problem to out NP-hard "Influence Maximization Problem" for seed selection and measure its performance with seed selection based on the centrality measures discussed in section 1.

1. INTRODUCTION TO NETWORKS

In considering models for the spread of an idea or innovation through a social network, we represent the network by a graph G. In mathematics, and more specifically in graph theory, a graph is a structure amounting to a set of objects in which some pairs of the objects are in some sense "related". The objects correspond to mathematical abstractions called nodes (also called vertices or points) and each of the related pairs of vertices is called an edge (also called link or line). We use NetworkX library for our computational experiments. NetworkX is a Python library for studying graphs and networks. It provides data structures for graphs (or networks) along with graph algorithms, generators, and drawing tools.

In graph theory and network analysis, indicators of centrality identify the most important vertices within a graph. We shall use the following three centrality measures:

1. **Degree Centrality**

Degree centrality assigns an importance score based simply on the number of links held by each node. In the scenario of social networks, it is used for finding very connected individuals, popular individuals, individuals who are likely to hold most information or individuals who can quickly connect with the wider network.

The degree centrality of a node j is given by

$$C_D(j) = \sum_{j=1}^{n} A_{ij}$$

where, $A_{ij} = 1$ iff there exists an edge between nodes i and j

2. **Betweenness Centrality**

Betweenness centrality measures the number of times a node lies on the shortest path between other nodes. In the scenario of social networks, it is used for finding the individuals who influence the flow around a system.

The betweenness centrality for a node v is given by

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest $(s,t) - paths$ and $\sigma(s,t|v)$ is the number of those paths passing through some node v other than s,t. If $s = t$, $\sigma(s,t) = 1$ and if $v \in s,t$, $\sigma(s,t|v) = 0$

3. **Closeness Centrality**

Closeness centrality scores each node based on their 'closeness' to all other nodes in the network. In the scenario of social networks, it is used for finding the individuals who are best placed to influence the entire network most quickly.

The closeness centrality for a node u is given by

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

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

4. **Eigenvector Centrality**

Like degree centrality, EigenCentrality measures a node's influence based on the number of links it has to other nodes in the network. EigenCentrality then goes a step further by also considering how well connected a node is, and how many links their connections have, and so on through the network. In the scenario of social networks, it is used as a good 'all-round' SNA score, handy for understanding human social networks.

The eigenvector centrality for a node i is given by

$$Ax = \lambda x$$

where A is the adjacency matrix of the graph G with eigenvalue λ. By virtue of the Perron–Frobenius theorem, there is a unique and positive solution if λ is the largest eigenvalue associated with the eigenvector of the adjacency matrix A.

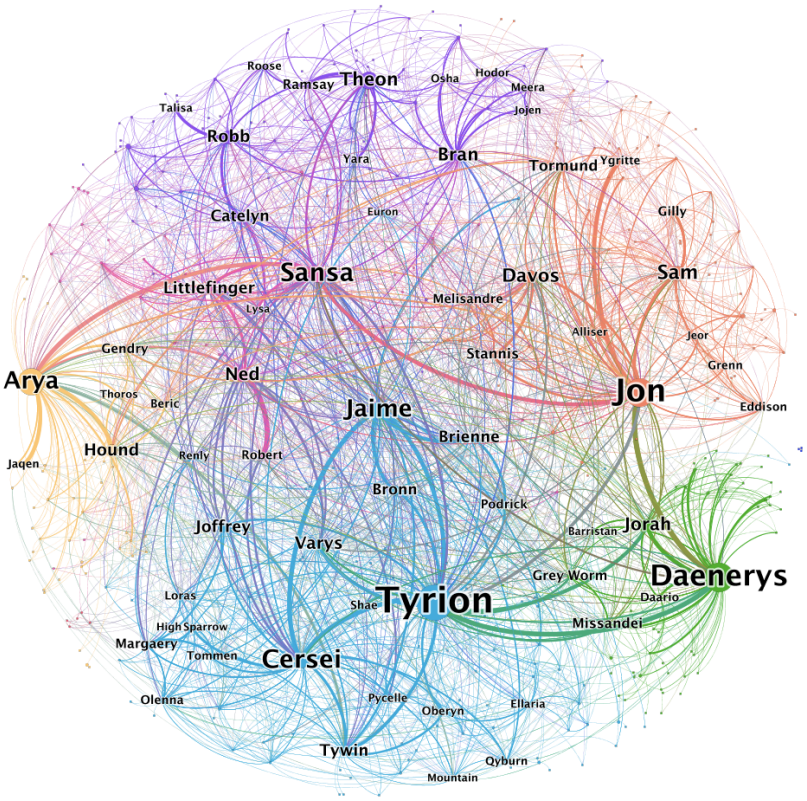
5. **PageRank Centrality**

PageRank is a variant of EigenCentrality, also assigning nodes a score based on their connections, and their connections' connections. The difference is that PageRank also takes link direction and weight into account – so links can only pass influence in one direction, and pass different amounts of influence. Because it considers direction and connection weight, PageRank can be helpful for understanding citations and authority.

2. DEFINING A NETWROK

Now that we have introduced a few basic terminologies of a network, let us define the network model that we wish to experiment on. As real-world social networks are huge and time-consuming to perform computational experiments on, we shall simulate on a smaller model that mimics real world population dynamics. The model that satisfies these parameters, that we consider here is the "Network of Thrones" model based on the books of famous TV Show Game of Thrones.

The dataset can be found via the link - https://networkofthrones.wordpress.com



The Dataset - "Network of Thrones"

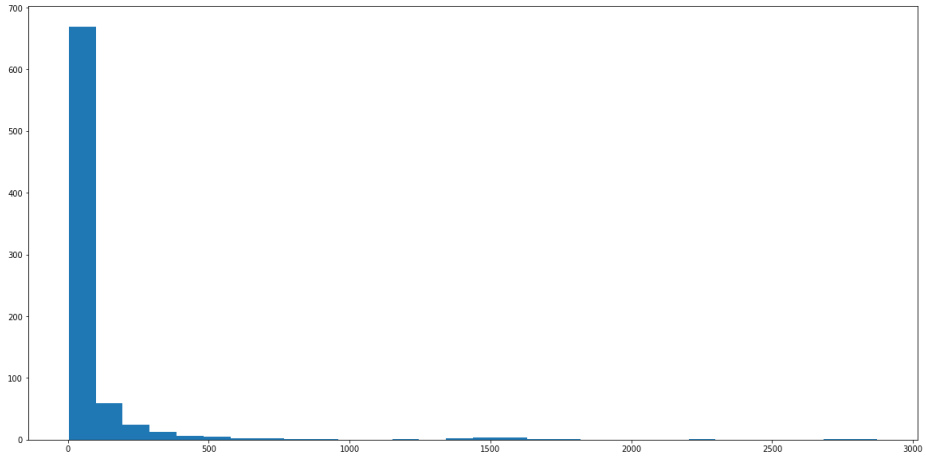
Two names (or nicknames) of characters that appear within 15 words of one another each time are linked together. Each link corresponds to an interaction between the two characters. Note that this interaction could be direct or indirect. Here are some of the types of interactions that our method picks up.

- Two characters appearing together in the same location
- Two characters in conversation
- One character talking about another character
- One character listening to a third character talk about a second character
- A third character talking about two other characters
- And so on…

For simplicity, these links are marked “undirected” meaning that the links are mutual, even when one character references another character (which could instead be seen as a one-way link). Further work on how the dataset was cleaned and details can be found using the link above.

Does this work as a Population Network?

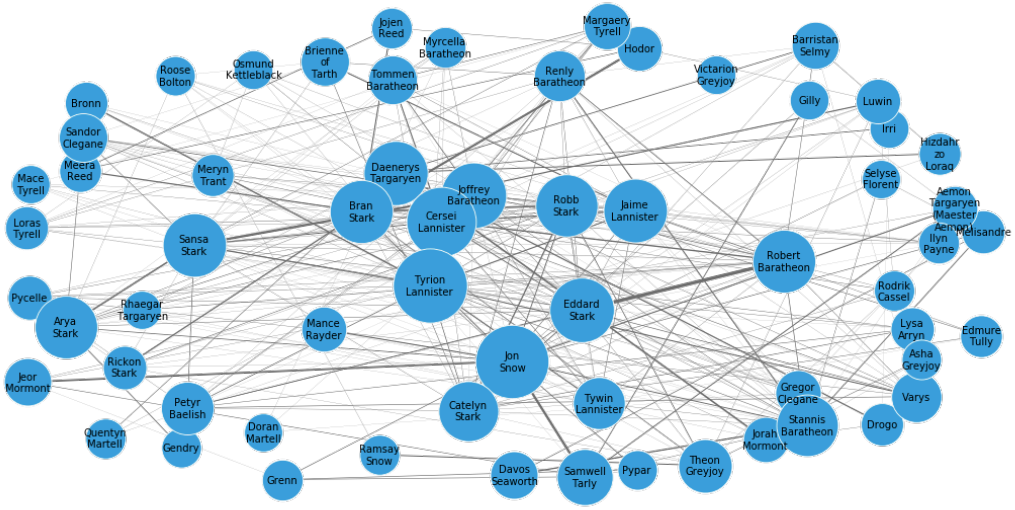
On loading the dataset and plotting its degree distribution (plot of number of nodes having a particular degree on y-axis vs. the particular degrees on x-axis), we get a plot as follows:



We thus observe that the nodes in our graph data indeed follow a power law distribution similar to population networks and hence can be considered for our project.

Finalizing our Dataset

We further shorten our dataset by removing redundancies and special cases by excluding nodes that have a degree less than 200. The resulting dataset that we thus consider for our project has 63 nodes and 496 edges with an average degree of 15.75



3. CHOOSING A SPREADING MODEL

Now that we have our social network model at hand, we must define the process by which influence spreads in the network. In considering operational models for the spread of an idea or innovation through a social network G, we shall consider each individual node as being either active (an adopter of the innovation) or inactive. How this state of "activeness" or "inactiveness" influences a linked node not yet exposed to the innovation, is what we shall define here via a spreading model. The models that we will be defining are famous models in the field of sociology and have been studied to quite an extent. Before we define them, we shall consider a few basic axioms to better understand our model –

- Each node's tendency to become active increases monotonically as more of its neighbors become active.
- Nodes can switch from being inactive to being active, but do not switch in the other direction.

Thus, the process will look roughly as follows from the perspective of an initially inactive node v: as time unfolds, more and more of v's neighbors become active; at some point, this may cause v to become active, and v's decision may in turn trigger further decisions by nodes to which v is connected. We thus define the two models below as described in the paper "Maximizing the Spread of Influence through a Social Network" by David Kempe, Jon Kleinberg and Eva Tardos.

1. The Linear Threshold Model

Granovetter and Schelling were among the first to propose models that capture such a process; their approach was based on the use of node-specific thresholds. Many models of this flavor have since been investigated but the following Linear Threshold Model lies at the core of most subsequent generalizations.

In this model, a node v is influenced by each neighbor w according to a weight $b_{v,w}$ such that $\sum_{w\ neighbour\ of\ v} b_{v,w} \leq 1$. The dynamics of the process then proceed as follows. Each node v chooses a threshold θ_v uniformly at random from the interval [0, 1]; this represents the weighted fraction of v's neighbors that must become active in order for v to become active. Given a random choice of thresholds, and an initial set of active nodes A_0 (with all other nodes inactive), the diffusion process unfolds deterministically in discrete steps: in step t, all nodes that were active in step t − 1 remain active, and we activate any node v for which the total weight of its active neighbors is at least θ_v:

$$\sum_{w\ active\ neighbour\ of\ v} b_{v,w} \leq \theta_v$$

Thus, the threshold θ_v intuitively represent the different latent tendencies of nodes to adopt the innovation when their neighbors do; the fact that these are randomly selected is intended to model our lack of knowledge of their values — we are in effect averaging over possible threshold values for all the nodes.

2. The Independent Cascade Model

The Independent Cascade Model is conceptually simplest model of the types of Dynamic Cascade Models investigated in the context of marketing by Goldenberg, Libai, and Muller. We again start with an initial set of active nodes A_0, and the process unfolds in discrete steps according to the following randomized rule. When node v first becomes active in step t, it is given a single chance to activate each currently inactive neighbor w; it succeeds with a probability $p_{v,w}$ —a parameter of the system — independently of the history thus far. (If w has multiple newly activated neighbors, their attempts are sequenced in an arbitrary order) If v succeeds, then w will become active in step $t + 1$; but whether or not v succeeds, it cannot make any further attempts to activate $w$in subsequent rounds. Again, the process runs until no more activations are possible.

The Linear Threshold and Independent Cascade Models are two of the most basic and widely-studied diffusion models, but of course many extensions can be considered.

For the purpose of our experiment, we shall choose the Independent Cascade Model and implement it in python as follows:

```
def independent_cascade(G,t,infection_times):
    #doing a t->t+1 step of independent_cascade simulation
    #each infectious node infects neigbors with probabilty proportional to the weight
    max_weight = max([e[2]['weight'] for e in G.edges(data=True)])
    current_infectious = [n for n in infection_times if infection_times[n]==t]
    for n in current_infectious:
        for v in G.neighbors(n):
            if v not in infection_times:
                if  G.get_edge_data(n,v)['weight'] >= np.random.random()*max_weight:
                    infection_times[v] = t+1
    return infection_times
```

4. THE INFLUENCE MAXIMIZATION PROBLEM

We have now defined a social network model that mimics population dynamics and have also defined the model by which influence spreads in a network. We now come to the problem that we aim to solve through this project – How does one maximize this spread of influence in a social network given the structure (graph) of the social network, influence model and a budget k.

Let us define the new term that we have just encountered – the budget k. Consider an example. Let's say that we wish to release a new product/innovation in the market. One approach to do that is by distributing free samples of the product. The consumer uses the free sample and if found appealing, then recommends it to his/her friends. The friends then purchase the product and recommend it to their friends and the influence cascade unfolds. However, in case of huge populations, we cannot distribute the free samples to all the individuals. The number of free samples is thus our budget.

To generalize the concept, the budget k, is the number of nodes that we would want to "activate", so that the spread of influence through these nodes in the entire network is maximum. We call these k chosen nodes – The Seed Nodes.

The Influence Maximization Problem thus demands us to find this k-set of seed nodes in a social network given the structure (graph) of the social network, the influence model and a budget k.

Input

A	A set of n nodes or entities of a social network connected via edges with varying weights corresponding to the relation between the nodes
k	Number of active (seed) nodes in set A
ISM	A Diffusion Model (here, the Independent Cascade Model)

Output

A_0	The bounded set of k seed nodes from the n population nodes such that the influence $\sigma(A_0)$ of the seed nodes on the population network A via the given diffusion model is maximum.

Problem Analysis

Consider a social network of n nodes. To solve this problem efficiently, we will need to find the spread of influence $\sigma(A_0)$ of every possible k-node set A_0 in the network and then comparing every σ to find the biggest σ. The k-node set A_0 corresponding to that σ would thus be our answer.

The number of ways of choosing the k-node set from n nodes would be

$$\binom{n}{k}$$

The spread of influence would then be calculated by the effect of each of those k nodes on the n-node population, thus contributing to a factor of

$$n * k$$

Thus, the runtime complexity of the problem would be

$$O\left(\binom{n}{k} * n * k\right)$$

We see that we have encountered a factor of a factorial. Thus, this is a NP (non-deterministic polynomial-time) Hard problem. To solve it we must compromise on either correctness or speed. As per the usual approach to every new NP hard problem, we shall try and solve an equivalent NP complete problem which we shall reduce to our NP hard problem.

NP-Hard problems(say X) can be solved if and only if there is a NP-Complete problem(say Y) that can be reducible into X in polynomial time whereas NP-Complete problems can be solved by a non-deterministic Algorithm in polynomial time.

Thus, the NP-complete problem we shall consider is the famous Set Cover Problem. We solve this problem in polynomial time using the non-deterministic Natural Greedy Algorithm and then reduce this problem to our Influence Maximization Problem.

The Set Cover Problem

The set cover problem is a classical question in combinatorics, computer science, operations research, and complexity theory. It is one of Karp's 21 NP-complete problems shown to be NP-complete in 1972. The problem definition can be expressed as follows:

Given a ground set U consisting of n entities, its subsets $T_i \subseteq U$ and a budget k corresponding to the number of subsets that can be considered, the problem asks to find the subsets to be chosen such that their coverage on U i.e.

$$f_{cov}(k) = |\bigcup_{i \in K} T_i|$$

is maximum.

Input

U Ground set consisting of n entities

T_i m subsets of the ground set

k Budget

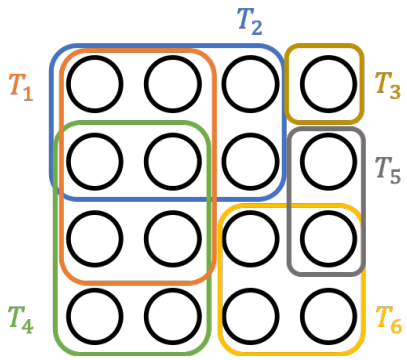
Output

A_0 The set of $K \subseteq \{1,2, \dots m\}$ of k subsets that maximize the coverage on U

$$f_{cov}(k) = |\bigcup_{i \in K} T_i|$$

Example

Given the ground set of 16 black circles below and subsets color coded respectively, what subsets must be chosen such that the coverage is maximum for a seeding budget = 4?



By observation, the answer in the above case would be either

$$T_2 + T_4 + T_6 + T_5 \ or \ T_2 + T_4 + T_6 + T_3$$

thus, giving a coverage of 15 out of the 16 elements in the ground set.

Natural Greedy Algorithm Intuition

Consider the trivial case of k =1.

In this case, the subset to be chosen would be the largest subset of all T_i

Consider the case where k=2.

Here, the task of the algorithm would be to choose a subset such that the number of elements in it apart from those in the previously chosen subsets is maximum i.e.

$$[\, f_{cov}(k \cup \{i\}) - f_{cov}(k)] \ is \ maximum$$

where i corresponds to the index of the next biggest new element subset

$$f_{cov}(k) = |\bigcup_{i \in K} T_i\,|$$

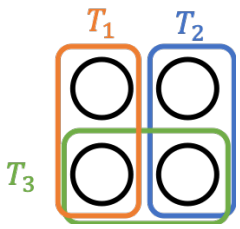
We repeat this process for greater values of k and thus we greedily keep increasing coverage using this natural greedy algorithm.

Here, we thus repeat our process for k steps, each time iterating over m total subsets and comparing, each time, the amounts of new element subsets of sizes of the order say s. Thus, the runtime complexity of the current algorithm becomes:

$$O(k * m * s)$$

Drawback of the Natural Greedy Algorithm

As stated earlier, the Set Cover Problem is a NP-complete problem. We thus found out a non-deterministic solution that runs in polynomial time by the algorithmic intuition as stated above. We shall see this non-determinism using the following example.



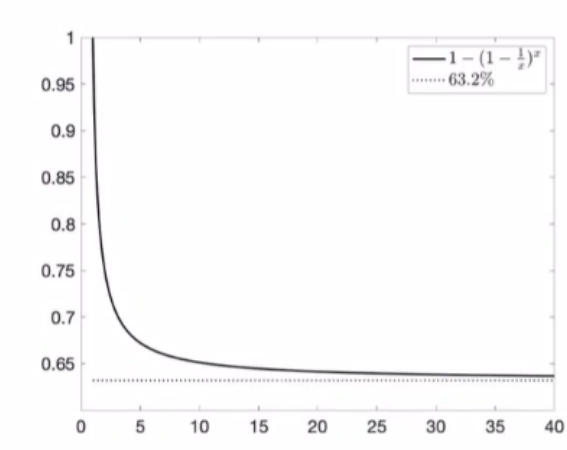
Given the ground set of 4 black circles below and subsets color coded respectively, what subsets must be chosen such that the coverage is maximum for a seeding budget = 2?

Say T_3 gets chosen at $k = 1$. In the next step, at $k = 2$ the greedy algorithm now has to choose between one of T_1 or T_2 thus making the net solution sub-optimal as compared to $T_1 + T_2$. Here lies the non-determinism.

But, the algorithm provides a good approximation of the optimal solution in polynomial time as compared to the optimal solution in non-polynomial time. This approximation is guaranteed to be equivalent to

$$1 - \left(1 - \frac{1}{k}\right)^k \text{ of optimal}$$

which amounts to roughly 0.63% of optimal in case of big k and n values. Thus, it is guaranteed that the algorithm would always do better than 0.63% of its optimal case solution. This can be observed by graphing these approximation values of percentage of optimal solution (y-axis) corresponding to the k-value (x-axis) to get the following curve.



Proof of Approximation Guarantee

As this is an implementation project, we shall not discuss theorems in detail but we shall get us familiarized with the concepts behind them by proving them intuitively.

Statement – For every possible input of maximum coverage problem, if budget = k, then this algorithm guarantees

$$coverage\ of\ greedy\ algo \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) * (max.\ \ possible\ coverage)$$

Proof – We shall begin our proof by first stating a lemma

Lemma 1 – Let C^* = max. possible coverage. Then each subset chosen by greedy coverage covers at least

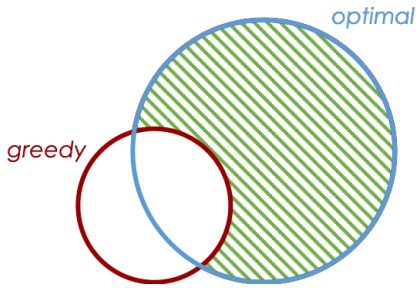
$$\frac{1}{k}(C^* - \#\ elements\ already\ covered)$$

Proof of Lemma – Let $k' = subset\ of\ k\ indices$ (eg. Consider k' to be the subsets of an optimal solution). Let the current iteration be j, and thus $j-1$ subsets would be chosen by now by the greedy coverage algorithm

Claim –

$$\sum_{i \in k'} (coverage\ increase\ from\ T_i) \geq C' - \ (\#\ elements\ already\ covered)$$

As we chose k' to be the subsets of an optimal solution C', C' thus becomes the coverage of k'. We represent this using the blue circle below. The coverage thus far by the greedy algorithm is represented by the dark red circle whereas the subsets in the optimal solution that are not yet covered by the greedy algorithm is represented by the green shaded region below.



Now we can say that the maximum possible extent of the green region i.e. subsets in the optimal solution that are not yet covered by the greedy algorithm at the j^{th} iteration, will include all the k' subsets that eventually form the optimal solution. Thus,

max. possible green region = $\sum_{i \in k'}(coverage\ increase\ from\ T_i)$ …(LHS)

Whereas the minimum possible extent of the green region i.e. subsets in the optimal solution that are not yet covered by the greedy algorithm at the j^{th} iteration, will include all the elements are not yet covered by the greedy algorithm but are part of the optimal solution and. Thus,

min. possible green region = $C' - (\#\ elements\ already\ covered)$ …(RHS)

The green region forms the upper and lower bounds of the LHS and RHS terms of our claim respectively. Thus, our claim holds.

Now, the LHS, $\sum_{i \in k\prime}(coverage\ increase\ from\ T_i)$ is sum of k' terms. Thus, the maximum element out of the k' terms is obviously greater than the average of their sums i.e.

$$\max_{i \in k\prime}(coverage\ increase\ from\ T_i) \geq \frac{1}{k}\sum_{i \in k\prime}(coverage\ increase\ from\ T_i)$$

$$\therefore \max_{i \in k\prime}(coverage\ increase\ from\ T_i) \geq \frac{1}{k}(C' - \#\ (elements\ already\ covered))$$

As greedy coverage selects at least as good as the next best subset, Lemma 1 is proved.

Now, coming back to our main proof, let us define some terms.

$$C^* = max.\ possible\ coverage$$

$$C_j = greedy\ coverage\ for\ first\ j\ subsets$$

By Lemma 1, $\forall j$,

$$C_j - C_{j-1} \geq \frac{1}{k}(C^* - C_{j-1})$$

Applying Lemma 1 to the final iteration of the greedy algorithm i.e. $j = k$,

$$C_k - C_{k-1} \geq \frac{1}{k}(C^* - C_{k-1})$$

$$\therefore C_k = \frac{C^*}{k} - \left(1 - \frac{1}{k}\right)C_{k-1}$$

Similarly, for $j = k - 1$

$$C_{k-1} = \frac{C^*}{k} - \left(1 - \frac{1}{k}\right)C_{k-2}$$

Thus, we get a recursive function for $j = k - 2, k - 3, k - 4, \ldots$ Considering $C_0 = 0$ (Zero coverage for the 0th iteration), we get

$$C_k \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) * C^*$$

This completes our proof.

Reducing the Set Cover Problem to our Influence Maximization Problem

Consider the subsets in the set cover problem to be analogous to the set of influenced nodes generated by a single node in the considered social network. Each node when seeded influences its neighbours who then influence their neighbours in accordance to the spreading model (in our case, the Independent Cascade Model) defined for the social network finally creating a corresponding set of influenced nodes.

We face an issue here. As the spreading models are based on probabilities, a node can make its neighbour active a total of p times of the total trials. Thus this net corresponding

set of influenced nodes for the same single node is bound to vary over various trails. Let us assume that the ability for a node to make its neighbour active is possible only if the edge joining both the nodes becomes active. Thus we have now associated the probabilities of the cascade model to the network edges.

Say that our network has m edges. We shall "flip" all the edges of our social network with a probability p of making an edge active and a corresponding probability of $p - 1$ of making it inactive. The spread of influence of a node then becomes the reachability of this node to other nodes via the "activate" edges that are now present in our social network graph. Let's say we have a seeding budget of k nodes, we thus apply the natural greedy algorithm similar to the set cover problem as follows

Consider the trivial case of $k = 1$. In this case, the node to be chosen would be the largest subset of influence generated by that node out of all the n nodes.

Consider the case where $k = 2$ or more. Here, the task of the algorithm would be to choose a node such that the number of nodes influenced by it apart from those which were influenced by the previously chosen node(s) is maximum i.e. we repeat this process for greater values of k and thus we greedily keep increasing influence in a network using this natural greedy algorithm.

This was the scenario for a single "flip-outcome" of all the m edges of our social network. In the general case, we thus compute the weighted average of this seed selection by repeating the above algorithm for all possible flips i.e. 2^m (each edge can either be "active" or "inactive"). We thus apply our greedy algorithm for k iterations, comparing the spread of influences of n nodes per iteration with those of the previous iterations for all the 2^m network structure possibilities and averaging the spread of influence at the end thereby creating a runtime complexity of

$$O(k * m * 2^m)$$

We have thus reduced our factorial time complexity to an exponential time complexity. However, note that the 2^m factor is just used for averaging all possible outcomes of the cascade model. As probabilities are involved to generate each "flip-outcome" of all the 2^m possibilities, by the power of uniform probability distributions, we may instead average the spread of influence over a large number of times (say s) instead of 2^m by generating these "flip-outcomes" of all the m edges of our social network randomly. Our time complexity thus reduces to

$$O(k * m * s)$$

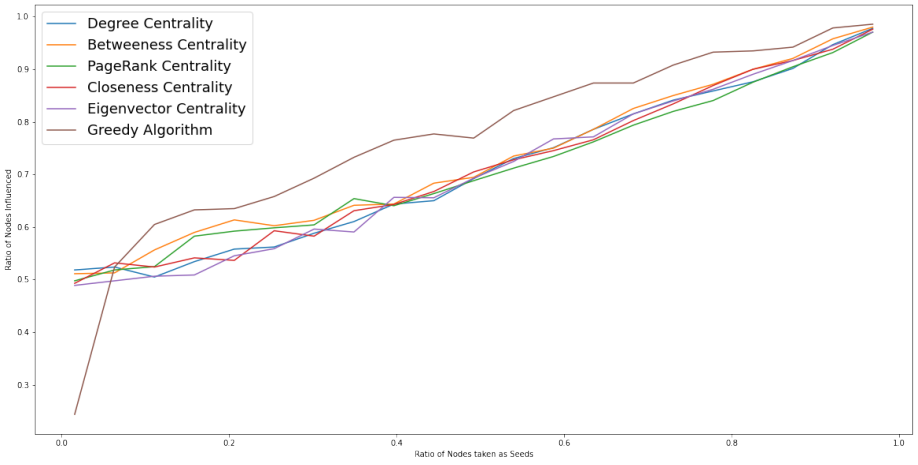
where s is just a large constant number.

As the method used here is similar to the set cover problem averaged over many trails, the approximation guarantee for the influence maximization problem too would be around 63% of the optimal. A non-deterministic price to pay for polynomial time complexity!

5. RESULTS AND CONCLUSION

To prove that the set of nodes “influenced” by nodes selected by the natural greedy algorithm for a given social network (here, the “Network of Thrones” dataset) and a given spreading model (here, the Independent Cascade Model) over various budget values k (number of nodes to be chosen as seed nodes) provides a better seed selection to maximize the spread of influence in a social network, we shall compare the results of the spread of influence by the seeds generated by the greedy algorithm to those of the seeds selected via node-centrality heuristics discussed in section 1 namely - 'Degree Centrality', 'Betweenness Centrality', 'PageRank Centrality', 'Closeness Centrality' and 'Eigenvector Centrality'. Plotting the graphs of the ratio of k-values to the total nodes of the social network on the x-axis and the ratio of the spread of influence generated by the seeds of each of the node-centrality heuristics and the greedy algorithm to the total nodes of the social network on the y-axis, we see that our greedy algorithm performs significantly better than the node-centrality heuristics for seed selection thereby successfully maximizing the spread of influence in social networks.

Note that this experiment was run for 20 trials only with the greedy algorithm averaging over mere 400 runs for the lack of a better processing machine. If done over greater trials and greater value of runs for the greedy algorithm, the difference in optimality would be much more pronounced.



6. FUTURE WORK

Seed Selection Algorithms

Despite the work on maximizing the spread of influence through a social network has been recently undertaken (since the turn of the 21st century), a lot of progress in research has been made in this field thus creating more opportunities to explore more variants and better variants of natural greedy algorithms or even better algorithms than the natural greedy selection.

The Social Network Model

In our project, we considered a very basic social network based on simple population dynamics following the power law distribution. But apart from this, there are a lot of variations of the social network which can simulate closer-to-real world population dynamics. We have discarded some important properties of viral marketing in order to simplify the problem. Apart from time, budget and memory networks, few consider the memory and social reinforcement effects, which are two common and significant characteristics of social networks. Memory effect means that previous contacts in a social network could affect the information spread in real time. Social reinforcement effect means that if more than one neighbor approves the information and transfers it to you, there is a high probability that you will approve it.

Diffusion Models

Last but not the least, we considered only the standard diffusion model for modeling the spread of influence in our social network – The Independent Cascade Model. Future work would include simulating this problem on even more variants of the diffusion models (like the Linear Threshold Model) and creating more diffusion models that mimic close-to-real population dynamics to then simulate on.

7. REFERENCES

"Maximizing the Spread of Influence through a Social Network" by David Kempe, Jon Kleinberg, Eva Tardos

“Influence analysis in social networks: A survey” by Sancheng Peng, Yongmei Zhou, Lihong Cao, Shui Yu, Jianwei Niu, Weijia Jia

“A survey on influence maximization in a social network” by Suman Banerjee, Mamata Jenamani, Dilip Kumar Pratihar

https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)

https://en.wikipedia.org/wiki/Centrality

https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/

https://networkx.org/documentation/stable/index.html

https://networkofthrones.wordpress.com

https://www.geeksforgeeks.org/

YouTube lectures based on the book Algorithms Illuminated, Part 4: Algorithms for NP-Hard