Priyanka Rathnam (pcr43)
Minzhi Wang (mw787)
Kanchan Yawalkar (ksy9)
CS 6850 Reaction Paper

Introduction

Cascading behavior in networks can be heavily influenced by the initial nodes in the network that are infected. Whether some piece of information or "an infection" spreads throughout the network or dies out is partially a product of how successful the initial spreaders are. Not only is it important for the initial spreaders to be successful in infecting or spreading information to neighboring nodes, in order for information or an infection to propagate through the entire network, it is also important for spreaders to be selected in a way that maximizes the potential number of nodes that will receive the information or infection. Choosing an optimal set of influential initial spreaders is critical to the scale of propagation in a cascade network. This paper explores and builds on papers that propose algorithms of choosing an optimal set of initial spreaders and factor in path diversity when selecting this set. This area of research is interesting for studying information spreading, epidemic control as well as viral marketing. The ideas discussed heavily apply to strategies of social influencers and how the success of viral marketing can be affected based on the choice of initial influencers.

Identifying a set of influential spreaders in complex networks

When making the decision to choose the first set of initial spreaders, traditional algorithms that rank node influence such as PageRank, ClusterRank, and k-shell decomposition face the challenge that some spreaders are very close to each other in the network. As a result, many of the neighbors that they infect or will infect in the future overlap, and the cascade can get stuck in local clusters, undermining the efficiency of the spreading. To overcome this limitation, Kempe et al. proposed a hill-climbing based greedy algorithm, which finds a set of initial nodes that would lead to the widest range of cascading. However, the greedy algorithm does not scale well to large network structures.

The paper "Identifying a set of influential spreaders in complex networks" discussed the limitations of pure influence-ranked selection of initial spreaders, and proposed a new algorithm to find a set of initial spreaders that would yield the most desired cascading effects. In the model, each node has two properties: the first is its voting power, namely the weight of its votes to each of its neighbors; the second property is the number of votes received by the node, which is calculated by summing up the voting power of all of the node's neighbors. In each iteration, the proposed VoteRank algorithm finds the top most elected spreader in the network and updates the voting weights of that node and its direct neighbors. When a node is elected as an initial spreader, the voting abilities of the nodes' neighbors are decreased by $\frac{1}{average\ degree\ in\ G}$ in order to ensure the diversity of the infection paths and increase the propagation range. As a result, the numbers of votes received by the nodes that are of distance 2 away from the elected node are also updated, since the voting abilities of their neighbors (nodes that are of distance 1 away from the elected node) are decreased. The algorithm stops when *r,* the targeted number of initial spreaders are found [3].

The paper argues that the resulting set of nodes generated from this VoteRank algorithm are far apart globally, but each is influential in its own local network. Once the initial r spreaders (infected nodes) are selected, the infection process is straightforward: at each time step, each infected node has probability μ of infecting one of its neighbors. Meanwhile, each infected node has probability β of being recovered. To measure the effectiveness of this algorithm, the paper proposed a performance metric called “infected scale”, $F(t) = \frac{nI(t) + nR(t)}{n}$, where nI(t) denotes the number of nodes infected at time t and nR(t) denotes the number of nodes recovered at time t. The final infected scaled is denoted by $F(tc) = \frac{nR(tc)}{n}$, where nR(tc) is the number of recovered nodes when cascading achieves a steady state. The infected rate is defined as $\lambda = \frac{\mu}{\beta}$. Another performing metric they used for model evaluation is the average shortest path length L_s between each pair of source spreaders S, where $Ls = \frac{1}{|S|(|S|-1)} \sum_{u,v\varepsilon S} l_{u,v}$, with $l_{u,\, v}$ denoting the length of the shortest path from node u to node v. This VoteRank model is tested on four datasets: Youtube, Cond-Mat, Berkstan, Notre Dame, and outperformed other models such as ClusterRank, IndegreeRank, and K-shellRank in the performance metrics described above [3].

This paper touched upon some network topics we talked about in class, such as in-degree, out-degree and closeness, and showed applications of these network structures in terms of cascading. Moreover, it discussed and explored other important properties of network graphs such as betweenness, centrality, and path diversity. Instead of a global approach, its local approach allows it to generate a set of source spreaders that are decentralized in the network. By intentionally making L_s larger, the spreading on the global scale becomes more effective due to the lack of overlap in the local structures. Additionally, because of the nature of a local-based approach, this method does not suffer much when scaled up to a large network. The algorithm must store all nodes, all the nodes’ properties, and all the edges in memory. The space complexity is therefore O(nm). The time complexity, however, is more advantageous compared to that of other algorithms. To initialize the voting ability of each node, the computation complexity is O(m) where m is number of edges in the graph. The complexity of picking the top elected node in the graph is O(n), where n is the number of nodes in the graph. This election step can be further optimized to O(log(n)). Updating voting ability only affects distances of up to 2 nodes away from the elected node, instead of the entire network, so it has a complexity of O (rm^2/n^2). To conclude, the total time complexity of this algorithm is O($m + rlog(n) + rm^2/n^2$) [3].

Path diversity improves the identification of influential spreaders

In the paper “Path diversity improves the identification of influential spreaders”, Chen et. al. explored the importance of path diversity in identifying influential spreaders in cascading networks. They posited that many alternative methods to identify such spreaders rely heavily on the number of paths for propagation and overlook the significance of path diversity. This claim was based in the observation that the spreading ability of a node is significantly reduced if its propagation paths overlap i.e. its path diversity is low. Chen et. al. therefore introduced an information entropy metric to reflect path diversity, which they incorporated into a spreader ranking algorithm, aiming to increase ranking performance [1].

To ensure that their algorithm was scalable, Chen et. al. limited their path diversity metric to local paths. They noticed that the diversity of a target node is related to the "evenness" in degree of its neighbors. To formalize this interpretation of diversity, Chen et. al. proposed the following measure

$$H_i = \frac{\sum_{j \in \Gamma_i} -p_j log(p_j)}{log(k_i)}$$

$$where\ p_j = \frac{k_j}{\sum_{l \in \Gamma_i} k_l},\ \Gamma_i = set\ of\ neighbors\ of\ node\ i,\ k_j\ =\ degree\ of\ node\ j$$

which describes the normalized diversity of the target node's paths and falls between 0 and 1. The numerator of the H_i metric corresponds to the information entropy of target node i and the denominator of H_i normalizes the entropy by the target node's degree. The higher the value of H_i, the more even the degree of target node i's neighbors, and therefore, the more diverse its paths [1].

Chen et. al. proposed a local spreader ranking algorithm, KED, which combined their formulation of path diversity with the number of propagation paths. Using the SIR model to simulate the infection process, they tested their algorithm on four social networks, Youtube, Orkut, EmailEU, and Digg, and compared its performance to degree centrality, PageRank, LeaderRank, and k-shell. To compare performances, they found each algorithm's top-50 spreaders, and after simulation they measured the ratio of KED's average final infection coverage to that of the other algorithms. Their results showed that KED performed better than degree centrality, PageRank, and LeaderRank on these datasets, and that in most cases it performed better than k-shell. These findings suggest that the incorporation of path diversity information into the ranking algorithm helped identify nodes with strong spreading ability [1]. This conclusion is quite interesting in relation to the concepts we discussed in this course regarding cascades. Chen et. al. have proposed a method of improving the selection of spreaders which directly impacts areas such as information spreading, epidemic control, and viral marketing [1]. We have seen examples of these areas in class, such as the success of Hotmail via viral marketing. These findings could increase understanding of cascades and how to select spreaders for our benefit.

Critiques

While both papers provide strong proposals for algorithms that can speed up the spreading in a cascade network, the justification for the effectiveness and efficiency of these algorithms is somewhat lacking. In "Identifying a set of influential spreaders in complex networks," three different performance metrics are discussed [3]. One is the infected scale, which is a function of the numbers of infected, recovered, and total nodes. Another is the final affected scale, which is a function of the numbers of recovered and total nodes. The third is based on the average shortest path length between spreaders. While the performance metrics seem rigorous on the surface because three separate metrics are used, it should be noted that none of the metrics is solely based on the number of infected nodes (without regard to the number of recovered nodes). This indicates that it could be likely that the performance metrics used in the papers were selected because they were skewed to their data. In addition to the metrics mentioned above, another desired metric may be percentage of infected nodes at the time of termination, a function of the numbers of infected nodes and total nodes (similar to the final affected scale).

Moreover, the justifications for the algorithms' performance or effectiveness in both papers are primarily empirical, with little to no mathematical analysis or modeling to back up the results that were obtained. While the empirical data from multiple datasets show the algorithms' effectiveness in comparison to others to an extent, the performance would have been more strongly justified had there been supporting mathematical analysis in addition to the empirical data.

Further Research Questions

The topic of information cascading is very broad and there are numerous directions we could explore from the ideas in these papers. Based on the ideas in "Identifying a set of influential spreaders in complex networks," we could create a modification of the VoteRank algorithm to extend votes to neighbors that are two steps away instead of just one. We could then see if this modification leads to a better selection of initial spreaders that results in a larger cascade (fewer time steps needed to reach infection bound/termination or more infected nodes after a given fixed number of time steps).

Another direction would be to synthesize the ideas from both papers to create a more effective algorithm. We could implement a version of the VoteRank algorithm from the first paper that incorporates the concept of information entropy from the second paper. We can achieve this by weighting the number of votes received by each node according to its information entropy.

The two papers primarily focus on ways to select initial spreaders in such a way as to increase the number of nodes that eventually get infected and the speed and the range of the cascade. In order to explore how to increase the spread of the cascade or to achieve a certain level of propagation within a specified number of time steps, it would be useful to determine the optimal number of initial spreaders instead of choosing an arbitrary number. By testing different configurations with different parameters, we can establish the optimal number of initial spreaders based on the network structure of the data we are looking at.

The performance metrics used in Zhang et al.'s paper are mentioned as a potential weakness above. To mitigate these concerns, we can introduce additional performance metrics that can be used to test our algorithm more objectively. Metrics that relate to the number of infected nodes and metrics that measure how fast or widespread a cascade is given r initial spreaders should be used. We will discuss these metrics further at the end of this paper.

Algorithm Implementation and Model Testing

We implemented the VoteRank algorithm that was described in the paper "Identifying a set of influential spreaders in complex networks", and downloaded the Condensed Matter dataset from SNAP to test our algorithm [2]. The Condensed Matter dataset is a representation of a collaboration network from the e-print arXiv. Each node in the graph is an author of a paper and each edge indicates a co-authorship. This dataset contains 23133 nodes and 93497 edges. The average degree in the graph is about 8.081, a parameter we used in our algorithm to decrease the voting abilities of nodes whose neighbor is elected as one of the initial spreaders.

The original model in Zhang et al.'s paper incorporated the factor of recovery into their infection model, where an infected node at any time step can recover from the infection with some probability [3]. For our implementation, we decided to take the probability of recovery out, and assumed that once a node is infected, it stays infected and has the ability to infect other neighboring nodes. Additionally, the original paper did not specify several parameters concerning the model, including the number of initial elected spreaders (r) and the termination point/infected bound of the algorithm (b). In our implementation, we set these parameters, along with infection rate (μ), to be user-specific, so we could test the model with different configurations and find the optimal setting.

Below is a table of results for different configurations of our implementation.

	# of initial spreaders (r)	Infection rate (μ)	Infected bound (percentage of nodes infected that determines termination)	Average # of time steps until termination (in 100 trials)
Config 1	20	0.5	50%	29.54
Config 2	20	0.5	80%	45.66
Config 3	20	0.3	50%	46.93
Config 4	20	0.3	80%	73.76
Config 5	50	0.5	50%	26.02
Config 6	50	0.5	80%	41.95
Config 7	50	0.3	50%	41.48
Config 8	50	0.3	80%	68.38
Config 9	100	0.5	50%	23.33

Config 10	100	0.5	80%	39.29
Config 11	100	0.3	50%	37.29
Config 12	100	0.3	80%	64.01

As we can see in the table above, if either the number of initial spreaders or the infection rate increases as the infection bound remains fixed, the average number of time steps necessary to reach termination decreases. Moreover, the average number of time steps needed to reach termination increases as the infected bound increases, meaning that more time steps are needed to infect more nodes given a fixed number of initial spreaders and a fixed infection rate. It is interesting to observe that the effect of increasing the number of initial spreaders seemed to diminish as the number of initial spreaders increased. We intend to look more into this phenomenon for our final project in order to find an optimal number of initial spreaders.

Model Proposal

We have already mentioned some of our ideas for extending the VoteRank algorithm designed by Zhang et. al. to improve its performance in selecting influential spreaders within the network. We will now describe how we will implement one of these extensions. We wish to leverage the utility of information entropy in measuring path diversity to add a new dimension to the VoteRank algorithm, and we hypothesize that it will improve the algorithm's performance. In order to test such an addition, we will modify our implementation of the VoteRank algorithm's voting system. Currently, the number of votes received by each node corresponds to the sum over the voting power of each of that node's neighbors. We would like to weight these votes according to the node's information entropy, described by Chen et. al. as the node's path diversity. Chen et. al. pointed out that the incorporation of path diversity avoids over-emphasizing nodes with overlapping propagation paths when determining their spreadability [1]. To weight these votes, we propose to multiply the current number of votes received by node i by $H_i = \frac{\sum_{j \in \Gamma_i} -p_j log(p_j)}{log(k_i)}$ since H_i is normalized to fall into a range from 0 to 1. We can also explore the effect of adding votes scaled by information entropy rather than scaling votes as a whole. In this setting, we would calculate the number of votes received by node i as the following

$$votes\ received\ by\ node\ i = \ current\ number\ of\ votes\ +\ \alpha H_i * current\ number\ of\ votes$$

We would tune the parameter α during testing to optimize the algorithm. We plan to test the various configurations of our algorithm on the Condensed Matter dataset as well as a simulated dataset. The purpose of the simulated dataset is to observe the effect that our addition of information entropy to the VoteRank algorithm has on networks with heavily overlapping propagation paths.

We manually ran the VoteRank algorithm with and without information entropy on a network structure that we created. In the unmodified VoteRank algorithm (steps are described in previous section), node B and C would be elected as the first two initial spreaders. In our modified

VoteRank algorithm, with information entropy incorporated, here are the steps we take at each time step.

At time step 0 (t = 0):

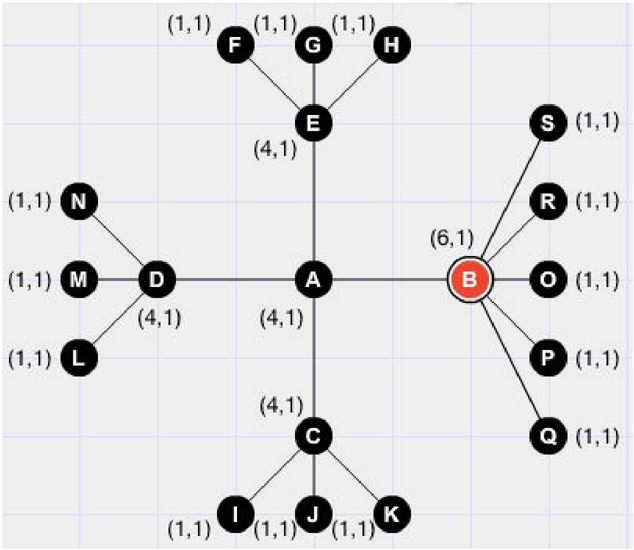
1. Node B gets elected as the first initial spreader, since it has the max number of votes received, calculated by multiplying the information entropy with the number of neighbors.
2. We decrease node B's number of votes received and voting power to 0, and decrease all of node B's neighbors' voting powers by $\frac{1}{1.8947}$, where 1.8947 is the average degree of the network.

At time step 1 (t = 1):

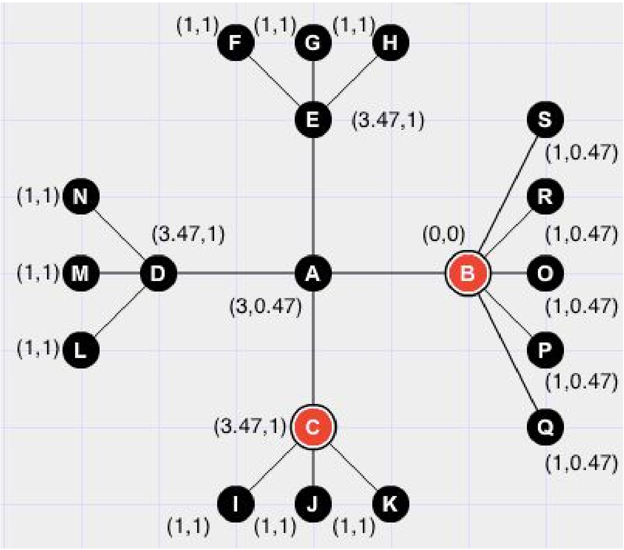
1. Node A gets elected as the second initial spreader, since it has the max number of votes received in this new system.

The modified VoteRank algorithm that incorporates information entropy is superior to VoteRank in this example. Node A has higher information entropy than node C. It is clear that node C must infect node A in order to reach any nodes other than nodes I, J, and K. While node C can only reach 3 nodes without infecting A, node A has a greater reach and can spread its infection to up to 8 nodes without having to infect node C.

VoteRank without information entropy:

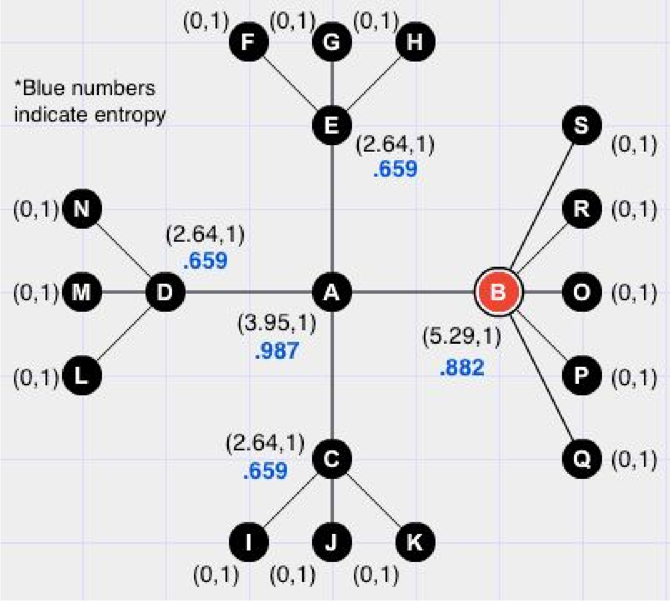


t = 0

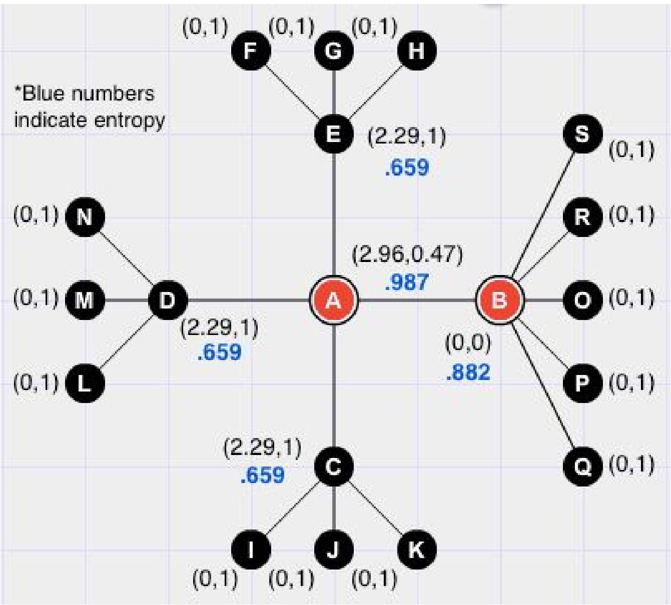


t = 1

VoteRank with information entropy:



t = 0



t = 1

Performance metrics

In order to measure the performance of our extended model, we will use metrics that are similar to those discussed in “Identifying a set of influential spreaders in complex networks,” but they will be modified to suit our version of the algorithm. Since our model will not include recovered nodes, we will exclude the number of recovered nodes from our metrics and instead focus on the number of infected nodes. We will measure a variation of the infected scale; we will calculate the proportion of infected nodes to total nodes. We will also do additional testing to analyze the time steps needed until termination given an infection bound based on variations of different parameter such as number of initial spreaders.

Link to our implementation (The code is also included in the submitted zip file):
https://github.com/vickywang8/CS6850_Reaction

Citations

[1] Chen, Duan-Bing, et al. "Path diversity improves the identification of influential spreaders." *EPL (Europhysics Letters)* 104.6 (2014): 68006.

[2] Leskovec, Jure, Jon Kleinberg, and Christos Faloutsos. "Graph evolution: Densification and shrinking diameters." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007): 2.

[3] Zhang, Jian-Xiong, et al. "Identifying a set of influential spreaders in complex networks." *Scientific reports* 6 (2016): 27823.