AIA: Maximizing the Spread of Influence through a Social Network

James Healey
BSc Computer Science,
Teesside University,
m2019524@tees.ac.uk

George Milner
BSc Computer Science,
Teesside University,
m2049480@tees.ac.uk

Chris Percival
BSc Computer Science,
Teesside University,
e5116399@tees.ac.uk

Keywords: NetLogo, diffusion models, approximation algorithms, influence spread, social networks

Hypothesis

Is targeting highly-connected individuals the most effective way to spread influence throughout a social network?

Abstract

The ability to identify the most influential users in a social network could have applications in online marketing or advertising. If marketers can target these influential users, it can allow them to increase the effect of their advertising campaign (Domingos, 2005).

In this paper, we first model a network, then run experiments on this model and compare the results to the findings of Kempe, Kleinberg, and Tardos (Kempe, Kleinberg, & Tardos, 2003) to identify reasonable ratios and values to be used. We use these values and explore the speed of spread using different algorithms, then extend the model to include interest depreciation, and rerun the experiments of the original model and explore how the results relate to the hypothesis.

We found that in both the original and extended model, the greedy algorithm performed highest, followed by the high degree centrality algorithm, with the distance centrality and random algorithms performing equally poorly.

1. Introduction

Kempe, et al (Kempe, Kleinberg, & Tardos, 2003) considered two different diffusion models and four approximation algorithms.

Diffusion models:

- Linear Threshold Model
- Independent Cascade Model

Approximation algorithms:

- Random
- Distance Centrality
- Degree Centrality
- Greedy Hill Climbing

Using data on the authors of academic papers as a network, they ran the algorithms on the two different models to find the optimum strategy to target individuals to achieve maximum influence. They found that although the greedy algorithm performed the best, the highest performing algorithm that is feasible for use in a real life scenario is the high degree centrality algorithm.

Building on this work, we first create another model of a social network. A variant of the Independent Cascade Model, each node is given an interest level that is added to every time another user attempts to influence them. We believe this model is closer to real life, as unlike a standard Independent Cascade Model, users have memory of previous influence attempts. In real life, this may equate to a

Twitter user who has previously seen and ignored a link then seeing the link for a second time and feeling more inclined to engage.

Given this model, we then run the same four approximation algorithms as Kempe, and adjust the values used until the results of experimentations are similar. This gives us values that have been backed by a well-established scientific paper.

We investigate the speed of influence spread for the different algorithms, and the effect of using the sorting algorithms when agents choose which of their followers to attempt to influence.

We then extend the model to include interest depreciation, which again brings the model closer to real life, as user interest generally decreases over time. Using the extended model and the values and ratios from before, we rerun the experiments, and explore the results against the hypothesis, the previous results, and the results from Kempe.

NetLogo was chosen as the modelling environment as it provides a simple but powerful graphical representation of problems, built upon simple agents.

2. Model

The model simulates the spread of influence throughout a network when certain nodes (users) are initially set to active (i.e. targeted). There are four algorithms that we use to determine how the influence spreads. The size (total number of nodes) and connectedness (number of edges/links/followers) of the network can be specified.

The maximum number of nodes are validated: Given n nodes, the maximum number of links can be no more than 2n - 1.

The model terminates when each active node has attempted to influence all of their inactive followers.

2.1. Algorithms

There are four algorithms that we use with the model to determine which nodes are targeted. These algorithms can be used to select the initial active nodes, and, optionally, when an active node attempts to influence a follower (algorithm-every-tick). If this flag is disabled, the random algorithm is used for all preceding node targeting selection (which is arguably closer to real life).

When determining the initially active nodes during setup, the potential target list is the entire network. Each tick, every active node will attempt to activate one of their followers. Each node can only attempt to activate a follower once, and therefore an active node can deplete its potential target list.

1 Random - The potential targets are randomised.

2 Degree centrality - The potential targets are sorted on the number of outward links (followers).

3 Distance centrality (closeness) - The potential targets are sorted on the average of their influence on their followers.

4 Greedy Hill Climbing - The greedy algorithm selects the node that gives the largest short term/marginal gain. The potential targets are sorted on the number of their followers which will be activated by their influence.

```
to-report get-potential-targets [ agentset ]
  if algorithm = "random" [
    report shuffle sort agentset
  ]
  if algorithm = "distance centrality" [
    report sentence (sort-on [(- mean [influence] of my-out-links)] agentset with [count my-out-links > 0]) sort agentset with [count my-out-links = 0]
  ]
  if algorithm = "degree centrality" [
    report sort-on [(- count out-link-neighbors)] agentset
  ]
  if algorithm = "greedy hill climb" [
    report sort-on [(- count my-out-links with [influence > [interest-
threshold] of end2])] agentset
  ]
end
```

2. Influence

We model influence as directed links between two nodes, each link has a value for the influence of the origin node on the destination node, a random number from zero to a definable max influence. Every node in the network has an interest threshold and an interest level. A node is defined as active if the interest level is above the interest threshold. Every tick, each active node attempts to influence one of their inactive followers. The influence from the link of the active node is added to the interest level of the target node. Each node keeps track of which followers it has attempted to influence, and does not attempt to influence them again.

- influence-max - the maximum influence value of a link. Each link has an influence value randomly assigned from zero to influence-max.
- interest-max - the maximum interest threshold value of an agent. Each agent has an interest threshold randomly assigned from zero to interest-max.

3. Results

3.1. Results from Kempe, Kleinberg, and Tardos

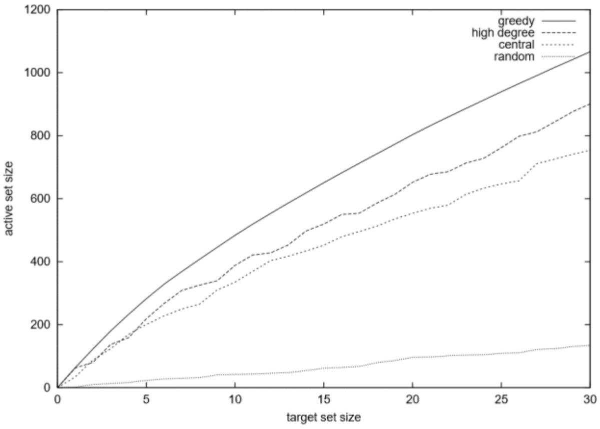


Figure 1: Results for the linear threshold model

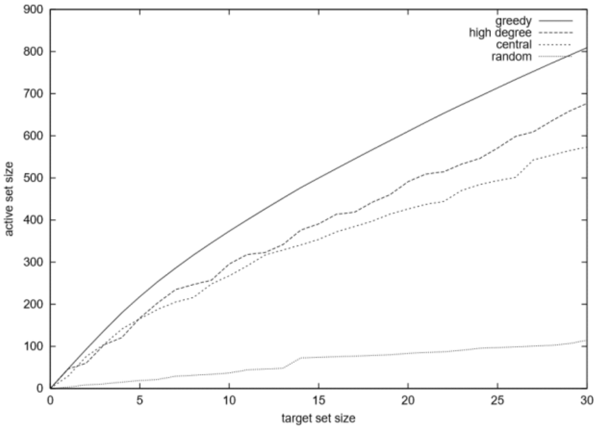
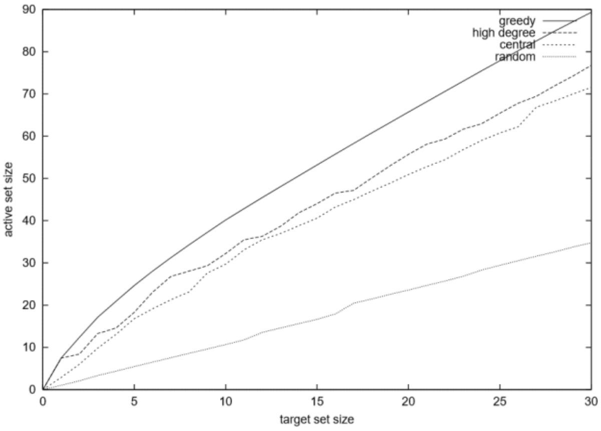


Figure 2: Results for the weighted cascade model



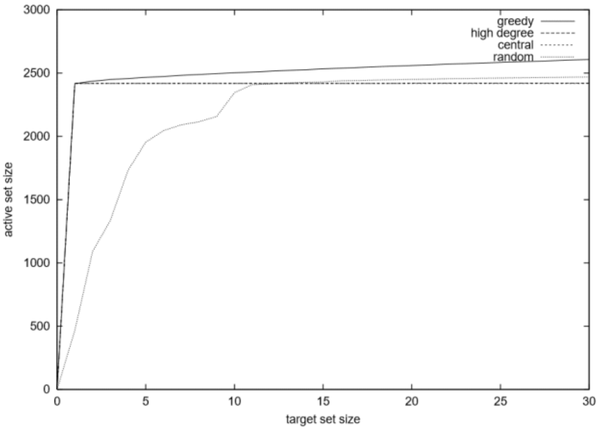


Figure 3: Independent cascade model with probability 1%

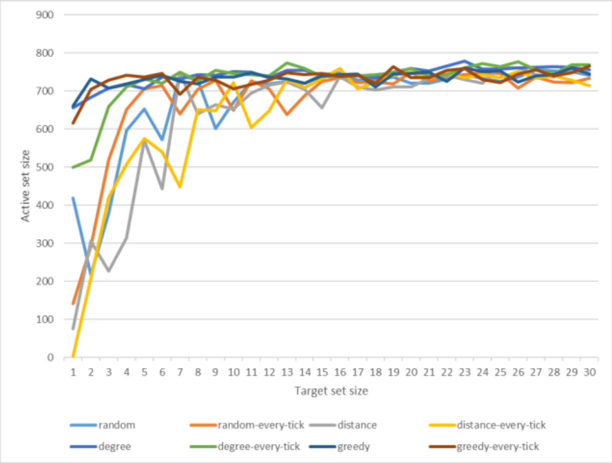
Figure 4: Independent cascade model with probability 10%

3.2. Finding influence-max to interest-threshold ratio

As our influence model differs from the model of Kempe et al (Kempe, Kleinberg, & Tardos, 2003), in order to ensure that all our values are justified, experiments were run to obtain the values for which our model has similar results to Kempe. These values were then used when testing any expansions of the model.

The values determining the influence spread in our model are the interest-max and influence-max values. The ratio of these values, along with the ratio of links to nodes, effectively determines the average possibility for a node to be influenced.

Using the algorithms described in model section, we obtained results from tests using a Monte Carlo simulation style using different random seeds (Mooney, 1997). Kempe used a graph of 10,748 nodes, and edges between about 53,000 pairs of nodes, to generate results. Due to time restrictions, a reduced amount maintaining the same ratio was selected: 1074 nodes and 5300 links.



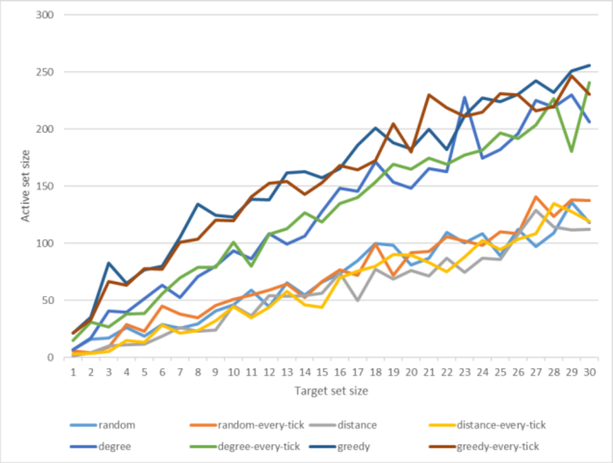
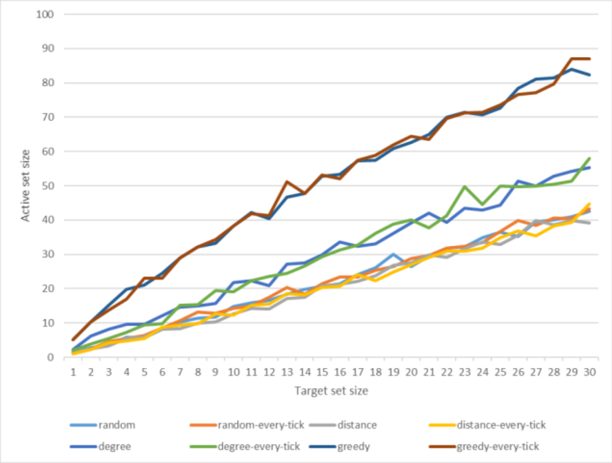


Figure 5: Influence Spread

Interest-max 50, Influence-max 25 – 2:1 ratio

Figure 6: Influence Spread

Interest-max 75, Influence-max 25 – 3:1 ratio



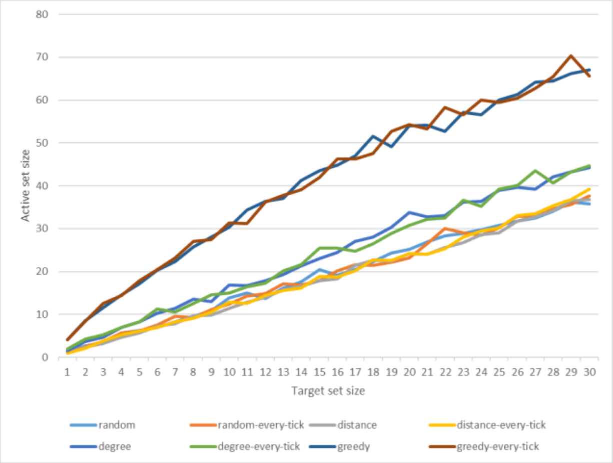


Figure 7: Influence Spread

Interest-max 200, Influence-max 25 – 8:1 ratio

Figure 8: Influence Spread

Interest-max 300, Influence-max 25 – 12:1 ratio

Figure 5 shows that with a ratio of 2:1, nodes activating too easily causes there to be almost no distinguishable difference between the different algorithms on the larger initially active target sets. The slopes are not similar to those of Kempe, and the number of active agents is much higher, relatively.

Figure 6 shows that increasing the ratio to 3:1 causes the difference between the algorithms to become more distinguishable, however the active set size is still higher, relatively, than the results from Kempe.

Figure 7 shows that a ratio of 8:1 appears similar to results of Kempe, both in slope and the relative number of nodes activated.

Figure 8 shows that with a ratio of 12:1 the results follow a similar pattern to the results of Kempe, however the active size set is relatively lower.

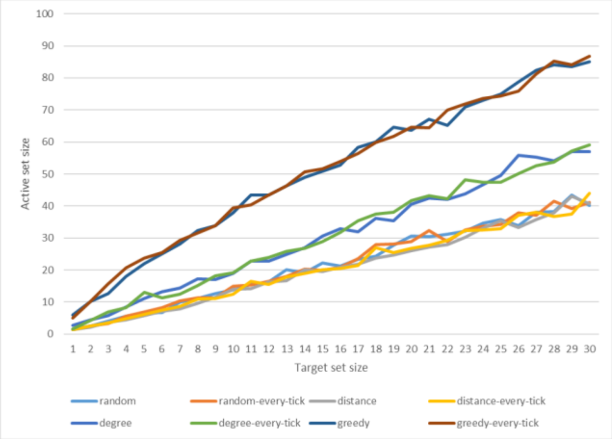


Figure 9: Influence spread

Interest-max 400, Influence-max 50 – 8:1 ratio

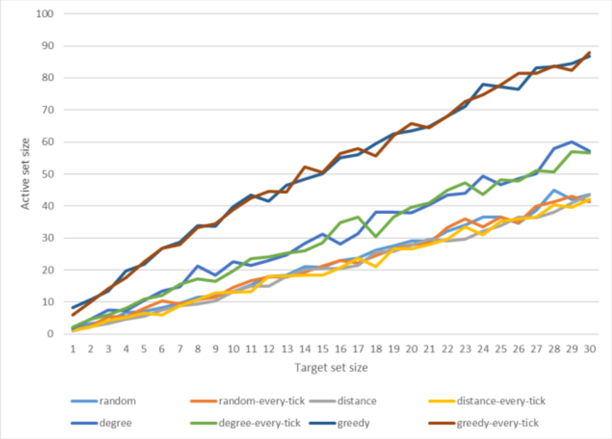


Figure 10: Influence spread

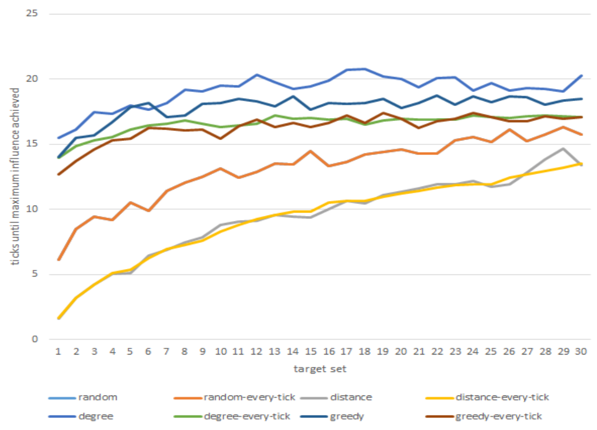
Interest-max 800, Influence-max 100 – 8:1 ratio

Both Figure 9 and Figure 10 shows very similar results to both Figure 7 and the results from Kempe, using different interest-max and influence-max values while maintaining the same 8:1 ratio. From this, we can conclude that the ratio of interest-max and influence-max is the determining factor in influence spread, and not the values themselves. This ratio can be used as a baseline when expanding the model.

3.3. Speed of spread

As agents do not lose interest, the spread of influence is progressive, therefore the value of algorithm-every-tick does not affect the total number of agents eventually activated. Our results reflect this, but suggest that the spread of influence is faster when the sorting algorithm is used to select targets both initially and every time a node decides which follower to influence.

We ran Monte Carlo simulations for each algorithm with both algorithm-every-tick values, and the resulting set of active nodes were the same size, however the algorithms took less time to reach the resulting set when the algorithm was applied every tick.



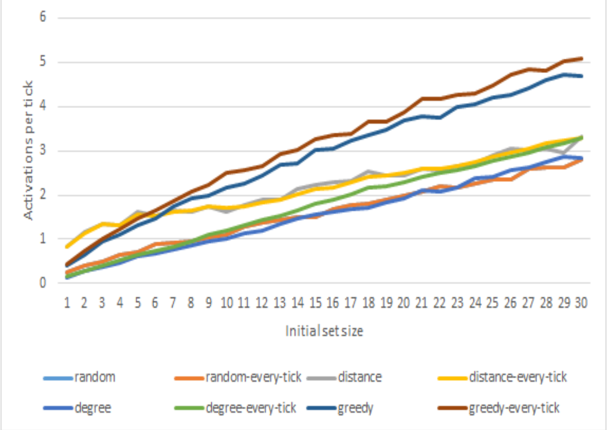


Figure 11: Speed of influence spread

Average activations per tick

Figure 12: Speed of influence spread

Ticks until maximum influence is achieved

Figure 11 shows that distance centrality selection reaches the maximum influence much faster than other algorithms, especially at low numbers of initially targeted nodes.

Figure 12 shows the average number of activations per tick increases as the initial set size increases.

Distance centrality is particularly interesting as it is highly effective on small numbers of initially active agents, but becomes less effective, relatively, as the size of the initially active set increases. This behaviour could potentially be explained by clustering within the network; tightly connected clusters of nodes within the network that the algorithm does not take into consideration; targeting several nodes that have high distance centrality in the same cluster quickly activates the rest of the cluster, but may fail to activate nodes in the rest of the network. As the initial set size increases, the activations per tick begin to converge with the results of the degree and random algorithms. This may be because the initially active agents are from different clusters, or there are enough active agents to break out from the cluster into the general populace.

The random algorithm completes very quickly for small numbers of initially active agents as it does not perform very well in general. Failure of the initial nodes to activate a significant number of followers quickly stops the influence from spreading.

The degree centrality and the greedy hill climbing algorithms perform better than the random algorithm, and naturally take longer to complete as the influence continues to spread.

4. Extended Model - Deactivation

In the extended model we introduce the concept of node deactivation. This is achieved by decreasing the interest level of each node by a definable amount (interest-decrease) every tick. An active node deactivates when their interest level reaches zero. Once deactivated, their interest-threshold is increased by a definable amount (threshold-increase). The deactivated node is removed from the influenced followers list of the nodes it was following, allowing the node to again be influenced and activated. When deactivation is enabled, the model stops when all nodes have deactivated, this allows us to see how long interest remains in the network over time (ticks).

- deactivation-enabled - toggles node depreciation and deactivation
- interest-decrease - the amount the interest level of each node decreases every tick
- threshold-increase - the amount the interest-threshold of a node increases when deactivated

When deactivation is enabled, we ensure that each initially active node has an interest level high enough to at least attempt to influence each of their followers once before they deactivate. This is ensured by the follow equation:

*interest level = interest-threshold + (follower count * interest-decrease)*

4.1. Experiments

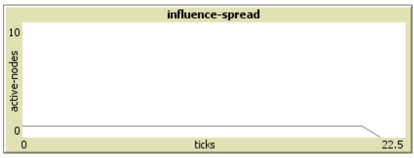


Figure 13

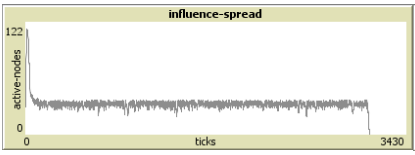


Figure 14

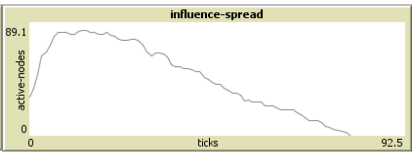


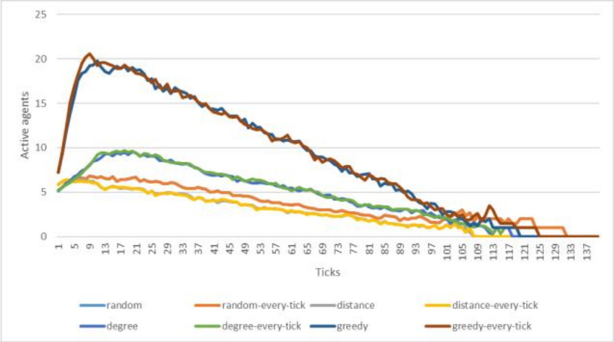
Figure 15

Figure 13 shows a state where a node was initially active, failed to influence any followers, then deactivated.

With a threshold increase of zero, cyclical tendencies often occur. Figure 14 shows a simulation taking over 3000 ticks for all the nodes to finally deactivate.

Figure 15 initially spikes as nodes are influenced and activated, then briefly shows some cyclical properties as nodes continue to be activated and others start to deactivate. It then shows a gradual decline until all nodes have deactivated, with some smaller spikes as some nodes in are activated or reactivated.

4.2. Results



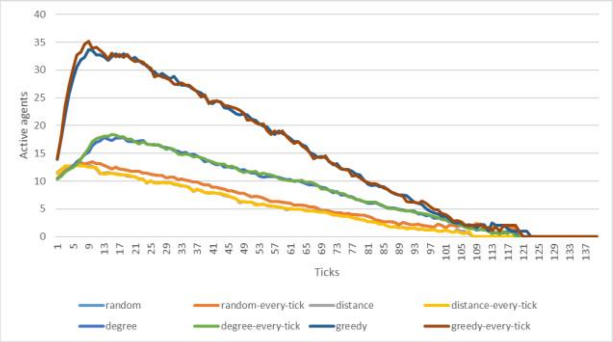
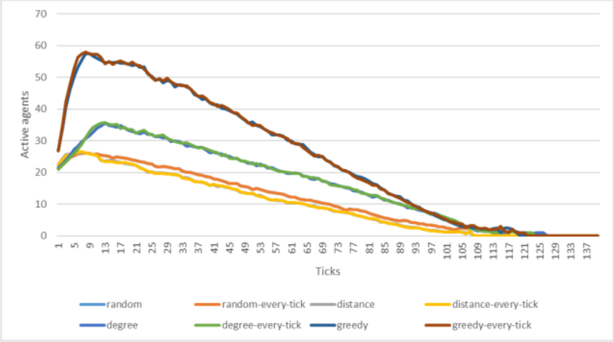


Figure 16: Influence spread

Deactivation enabled, 5 initially active

Figure 17: Influence spread

Deactivation enabled, 10 initially active



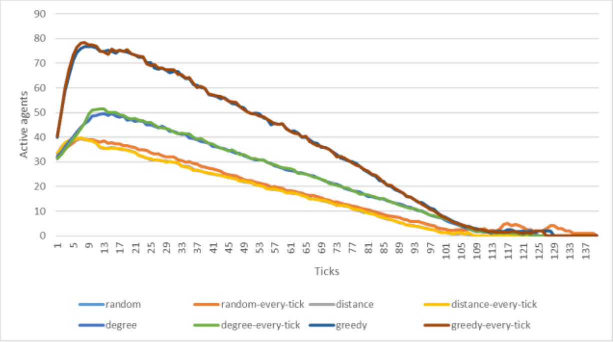


Figure 18: Influence spread

Deactivation enabled, 20 initially active

Figure 19: Influence spread

Deactivation enabled, 30 initially active

Monte Carlo simulations were performed on the extended deactivation model, and the results were averaged. The results in Figures 16 to 19 show that each of the algorithms follow a similar path for different numbers of initially active agents: first, a sharp increase in active agents, then a slow linear decline.

The hypotheses still holds, with the algorithms performing as expected, similar to the standard model. The greedy algorithm still performs best, with the high degree algorithm still performing significantly worse than greedy, but better than the distance centrality and random algorithms. Again, the distance centrality algorithm performs poorly, similar to the results found in the original model.

The graph show similar values for the number of ticks until maximum influence peaks, with slight fluctuations in the time taken before all nodes deactivate. The higher the initial active count, the higher the peak of active nodes, with the ratio of peak active users declining slowly compared to the number of initially active agents; this would manifest in the real world as diminishing returns on advertising spend.

In some of the graphs, there are outliers in the later ticks due to the reduced number of samples at that value as other simulations complete; if more samples were to be taken, we would expect to see these late spikes disappear.

5. Critique

5.1. Ambiguity in Kempe's paper

Due to ambiguity or lack of detail in Kempe's paper when describing the greedy hill climbing algorithm, we made assumptions when calculating the marginal influence of nodes and whether or not the algorithms are applied in the agent decision making process when deciding which follower to influence. Before taking the model further by adding deactivation, we made sure the ratios of values used were backed by the Kempe's results.

The description of the distance centrality algorithm is also ambiguous; we assume that the average influence on followers is used, but this could be incorrect, and another potential reason for the unexpectedly poor performance. Distance centrality could be calculated using the sum of influence on followers, instead of the average, and the results compared to those of Kempe.

5.2. Network structure

Although our values for the number of nodes and links were proportional to those used in Kempe's paper, the randomly generated structure of our network based on these values may not coincide with the structure of real social networks. If clusters occur more often in our network than Kempe's, it could explain the poor performance of the distance centrality algorithm. A potential expansion would be to recreate the network using data from real life social networks such as Facebook or Twitter (Leskovec, n.d.).

6. Further Investigations

Currently we look at the four algorithms discussed in Kempe's paper, however there may be algorithms that perform even better. The effectiveness of the eigenvector algorithm could be evaluated, for example. Eigenvector centrality incorporates the importance of the followers of a user (Liu, Abbasi, Zafarani, State, & Tempe, 2014), and is a good candidate for an effective algorithm.

This paper assumes nodes can only be targeted/activated before the first tick, during setup. In reality, advertisers may spread their advertising over a longer time span. Finding the optimal times to advertise/activate agents, especially when deactivation is enabled, would not only be interesting to explore, but could have commercial value.

In our model, each node can only attempt to influence a single follower each tick; to bring this closer to reality, where multiple people can see the same status or tweet at the same time, the model could be modified to be slightly closer to a linear threshold model, where each tick, the influence level of each user has a chance of being increased by the influence of the neighbouring nodes.

In our model, agents are only influenced when they are inactive. To bring the model closer to reality, active nodes should still be able to be influenced by other neighbours (increasing their interest further above the threshold). While this would not make sense when deactivation is disabled (as it would just slow the rate of influence), if deactivation were enabled, the active agents could stay active longer due to the extra influence, and deactivate less (which in turn would reduce the time taken until the average interest-threshold increases), possibly leading to more cyclical behaviour.

Algorithms could be used to identify clusters of nodes, and experiments carried out on algorithms that only target a select number of nodes from each cluster to see if this gives a better spread of influence throughout the network as a whole (Huang, et al., 2010).

7. Conclusion

7.1. Spread of influence

This paper argues that a greedy hill climbing algorithm could be used in a real life network to obtain maximum influence throughout a network. Our results show that the greedy algorithm performed the best, followed by the high degree centrality algorithm. We suggest that social networks such as Facebook and Twitter have enough computing power and necessary data on the approximate interest and influence levels of their users that they could compute the greedy algorithm in a reasonable amount of time.

7.2. Speed of spread

We have shown that distance centrality is highly effective on small numbers of initially active agents, but becomes less effective, relatively, as the size of the initially active set increases. Targeting several nodes that have high distance centrality in the same cluster quickly activates the rest of the cluster, but may fail to activate nodes in the rest of the network.

7.3. Depreciation

We demonstrate that with the addition of interest depreciation, deactivation, and reactivation, the effectiveness of the different approximation algorithms remain the same. We also show that cyclical behaviour can be observed under certain conditions.

7.4. Summary

We conclude that in most cases, targeting individuals with high degree centrality is the most effective way to spread influence throughout a network. Large organisations such as Facebook now have extensive data on how their members interact with one another, however; using this data could allow for our greedy hill climbing algorithm to be used, which would be more effective.

8. References

Domingos, P. (2005). Mining social networks for viral marketing. *IEEE intelligent systems*, 80.

Huang, J., Sun, H., Han, J., Deng, H., Y, S., & Liu, Y. (2010). SHRINK. *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*.

Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*.

Leskovec, J. (n.d.). *Stanford Large Network Dataset Collection*. Retrieved from Stanfod University: https://snap.stanford.edu/data/

Liu, H., Abbasi, M., Zafarani, R., State, A., & Tempe. (2014). *Social media mining: An introduction.* United Kingdom: Cambridge University Press.

Mooney, C. Z. (1997). *Monte Carlo simulation, Vol. 116.* Thousand Oaks, CA: Sage Publications (CA).