Maximizing Diffusion on Dynamic Social Networks

Jared Sylvester

University of Maryland
Department of Computer Science
jsylvest@umd.edu

15 May 2009

Abstract

The influence maximization problem is an important one in social network analysis, with applications from marketing to epidemiology. The task is to select some subset of the nodes in the network which, when activated, will spread the activation to the greatest portion of the rest of the network as quickly as possible. Since exact solutions are computationally intractable, greedy approximation algorithms have been developed. However, such methods have only been tested on static social networks: those in which the edges do not change while diffusion is occurring on the network. This is despite the fact that many social networks exhibit strongly dynamic behavior. Applying the heuristics used for static networks to dynamic ones is not straightforward since the metrics typically used to judge the influence of nodes are not well defined when edges are changing. This paper examines the use of several potential dynamic measures for use with greedy approximation algorithms. Both linear threshold and independent cascade models of diffusion are studied, and networks are formed using random, preferential attachment and proximity-based paradigms.

1. Introduction

Social networks are a very useful paradigm for understanding a wide variety of problems from the spread of infectious diseases to the diffusion of information and ideas [Abrahamson & Rosenkopf, 1997; Granovetter, 1973, 1983; Newman et al., 2006]. Techniques applicable to social networks can also be useful for physical networks and other graph-based problems, for instance locating sources of contamination in a water supply [Habiba et al., 2008]. While these problems have been widely studied in the past, almost all efforts focus on static social networks. That is, networks in which the ties between actors remain constant throughout the period being studied.

For many situations this is a highly unrealistic assumption. Students interact with different peers when their class schedules change or when they return home for breaks; the social network of an adult changes when they have children, take a new job or relocate; the network of interaction among animals changes as they migrate. Despite this there has been little work done on the behavior of dynamic networks.

Part of the difficulty in addressing dynamic networks is that they are more difficult to measure than static networks. Standard measures of distance, centrality and clustering break down when network ties change.

For instance, dynamic networks were shown to speed diffusion of innovation over static networks, for both proximity-based and random networks [Stonedahl et al., 2008]. Despite this the authors admit they do not yet have have a way of formalizing the behavior of these dynamic networks in more detail because they lack a well-defined notion of distance, centrality or clustering in the presence of network dynamism.

Brandes [2008] lists a variety of centrality measurements for networks. These include both refinements and generalizations of the more standard notions of centrality. Also presented are algorithms to calculate all of the measures presented. Brandes specifically mentions centrality measures for dynamic networks as one of two major goals, mentioning that it has relevance to clustering algorithms presented in Newman & Girvan [2004].

A lot of the work on dynamic networks has dealt with computer networks, especially ad-hoc wireless communications networks [Do, 2008; Wang & Crowcroft, 1992]. Nonetheless, these domains have different needs than social network analysis. Wang, for instance, is mostly concerned with traffic analysis within a network. Social networks analysis rarely concerns itself with the level of use on a link, rather it focuses on whether links exist and whether they are being utilized at all. (For instance, a virus is either passed from one person to another or it is not.)

Leskovec et al. [2007] presents a study of social networks, in this case blog citation networks, which incorporates some sense of temporality. The network in their model remains static, but each link is weighted with the weight representing how much time passes between the appearance of a story on one blog and it being linked by the citing blog.

One of the most important problem in social networks is influence maximization, as it is a generalization underlying many real-world applications, such as designing viral marketing campaigns. The influence maximization problem is attempting to select the subset of nodes in the social graph which, when activated, will spread their activation to the most other nodes in the graph. Given a network $G = (G_1, \ldots, G_T)$ which is dynamic over T time steps, choose a set of vertices A_0 with $|A_0| = k$ to seed with activation which will maximize the total activation $|A_T|$ of the network at time T.

In the context of viral marketing, one might give free samples of a new product to a select group of people thought to be highly influential in the hopes that they will recommend it to their contacts, who in turn will recommend it to theirs, hopefully causing a chain of recommendations to cascade through the network, resulting in as many people as possible having heard favorable accounts of the new product.

The problem of influence maximization has been widely studied in static networks [Abrahamson & Rosenkopf, 1997; Kempe et al., 2003; Valente, 1995]. Because the problem of selecting the optimal initial subset is NP-Hard, approximation algorithms must be used. The standard approach is a greedy one, whereby each node is measured and ranked by some centrality metric. The highest ranking k nodes are then selected for initial activation. Though this method is extremely simple, it is also very effective [Kempe et al., 2003].

Despite its effectiveness, this approach has not been thoroughly applied to dynamic networks, at least in part because of the aforementioned lack of metrics to measure the influence of a node in a dynamic setting. One potential measure, called "temporal path betweenness," is proposed in Habiba et al. [2007]. Temporal path betweenness relies on calculating the shortest temporal paths, which are those paths through the graph which are time respecting. By considering a dynamic network as the union of the graphs representing the network at each time step, and labeling the resulting multigraph's edges with the time step in which they are extant, a temporal path on the multigraph is a path for which the labels of each consecutive edge are increasing. The temporal betweenness of a node is the fraction of the total number of pairwise shortest paths on which it appears.

Some variations on this theme are possible. The lengths of these temporal paths can be defined either as the number of edges comprised ("shortest link path"), or the total time elapsed along the path ("shortest simple temporal path"). Each of these measures can also be used to calculate a corresponding notion of centrality.

Habiba et al. have thus far applied temporal path betweenness to problems of maximizing and blocking spread in networks with good results [Habiba & Berger-Wolf, 2007; Habiba et al., 2008]. In all of these applications the authors use real-world network data and the independent cascade model of diffusion. Though promising, the only citations thus far of Habiba et al's techniques are in ad hoc vehicular networks [Do, 2008]. (Do presents their measures as being potentially applicable, but does not seem to have used them in the experiments presented.)

In this paper temporal path betweenness centrality is applied to artificial dynamic networks of three different types to asses it's usefulness as a heuristic for greedy influence maximization techniques. Additionally, the average degree of a node, a weighted count of total neighbors, and a novel notion of clustering coefficients in dynamic settings are all considered. Both the independent cascade and linear threshold models of network diffusion are used.

2. Methods

2.1. Network Structure

In order to isolate the effects network structure may have on the dynamic metrics, artificially generated networks are used. Three different types of network dynamics are used: random networks, scale-free networks, and proximity-based networks. In all cases only the edges of the network are added or removed; nodes remain throughout the experiment. Note that the graphs used are all undirected and unweighted.

After initializing the network according to one of the following paradigms, the dynamics are allowed to run for 25 times steps. After that period the network is measured to asses the centrality of each node. The k nodes with the highest centrality are selected to seed the network with activity. Activity is diffused through the network according to either the Linear Threshold or Independent Cascade models. While activity is diffusing the network structure continues being updated. After a predetermined length of time activity ceases and the diffusion across the network is measured.

2.1.1. Random Networks

Random networks are generated by randomly selecting any two nodes in the graph and adding an edge between them until enough edges have been added to satisfy the desired degree of edge density. At each time step the graph is updated by removing some fixed proportion of edges and replacing them with a new edge so that the number of total edges remains constant.

2.1.2. Scale-free Networks

Scale-free networks are generated in much the same way as random networks. We begin by selecting one node randomly and uniformly, but instead of selecting the node it will be attached to using a uniform distribution, as in the random networks above, the neighboring node is selected with a preferential attachment model. We use a modification of the Barabási-Albert model [Barabasi & Albert, 1999]. The probability that a node i will be selected as the neighbor is given by

$$p_i = \frac{k_i}{\sum_j k_j} \tag{1}$$

where k_j is the number of neighbors of node j. The modification of the Barabási-Albert model arises because unlike BA, the number of nodes in our networks remains constant. Thus rather than adding a new node and using preferential attachment to select it's neighbor the first node is selected evenly from among all the nodes in the graph.

Like the random networks described above a certain proportion of all edges are removed from the graph each time step and replaced by the same number of newly generated edges. New tial attachment rules as were used to generate the initial network configuration.

2.1.3. Proximity Networks

Proximity networks encapsulate a degree of spatial interaction between the nodes, and are modeled on the spatial proximity networks in Stonedahl et al. [2008]. Each node is considered to be a mobile agent on a toroidal surface. At each time step the agents update their positions using Brownian acceleration. After updating nodes' positions an edge is added to the graph for every pair of nodes within some fixed distance of each other. This constant radius is chosen such that the expected number of edges in the graph averaged throughout time is the same as the other two network dynamics used.

2.2. Diffusion

We consider each individual node as being either active or inactive. Only the progressive case is considered here: nodes can switch from being inactive to active, but can not switch back to inactivity once becoming active. As mentioned previously, both the Linear Threshold and Independent Cascade models of diffusion across networks were used.

The Linear Threshold Model has many variations, but it originated with Granovetter and Schelling [Granovetter, 1978; Schelling, 1978]. Each node i is given a threshold value of $\theta = 1/2$. This is the proportion of i's neighbors which must become active for i to activate. Activation in the Linear Threshold Model is deterministic. At each time step all active nodes remain active, and all inactive nodes calculate the fraction of their neighbors which are active. Any nodes with neighboring activity exceeding their threshold become active at the end of the time step. Granovetter's original model gave each node it's own threshold uniformly samples from $[0, 1]$. Setting each node's threshold to an identical value as we do gives similar results [Berger, 2001]

The second model of diffusion used is the Independent Cascade Model [Goldenberg et al., 2001]. When a node first becomes active it has the chance to probabilistically activate each of it's neighbors. Each inactive neighbor will become active at the end of the current time step with a probability $p = 1/3$. Whether or not a node spreads it's activation successfully it can make no further attempts to activate its neighbors. If a node n has more than one newly activated neighbor they are each given a chance to spread activation to n. The process continues until no more activations can occur.

2.3. Measurements

As mentioned, various measurements were used to choose the initial activation set. A description of each follows.

2.3.1. Average Degree

The mean of a node's degree in each time step in the initialization period. Despite being a somewhat naive approach the degree of a node is an effective heuristic in static networks [Kempe et al., 2003]. It is a simple and obvious extension to generalize the degree across the initialization period, and should be useful as a baseline comparison method.

2.3.2. Weighted Neighbor Count

Another simple measure being used for the sake of comparison, it is a count of the number of nodes a given node was connected to at any point during the initialization period, weighted by the number of time steps two nodes were connected.

We use $\delta_{i,j,t}$ to indicate whether an edge exists between nodes i and j at time step t.

$$\delta_{i,j,t} = \begin{cases} 1 & \text{if } (i,j) \in E_t \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Then we let $NC_i(j)$ be the number of time steps that nodes i and j were connected.

$$NC_i(j) = \sum_t \delta_{i,j,t} \tag{3}$$

We use this to calculate the weighted neighbor count NC_i of node i as follows:

$$NC_i = \sum_j NC_i(j)\alpha^{NC_i(j)} \tag{4}$$

The intuition behind this measure is that two nodes which are often connected in a social network may have a greater likelihood of staying connected in the future, thus a node which is well connected by this metric may be likely to stay well connected. This assumption may hold for the proximity network dynamics, since it can be expected that nodes in the same region of the spatial environment are likely to remain close to each other on average, and thus remain connected. Because there is no mechanism in the random and preferential attraction networks to favor edge longevity it is likely that weighted neighbor count will perform relatively worse on these networks.

2.3.3. Temporal Betweenness Centrality

Introduced by Habiba et al. [2007], Temporal Betweenness Centrality measures the number of “shortest temporal paths” which pass through a node. This is an abstraction of the standard betweenness centrality metric to dynamic networks in order to cpature the notion that edges must exist in the “right time” as well as the “right place” in order to be effective at spreading activation. (See discussion of shortest temporal paths in the Introduction for more.)

Letting the number of shortest temporal paths between nodes s and t be given by g_{st}, and the number of those paths which pass through node u be given by $g_{st}(u)$. The temporal betweenness of node u is then defined as

$$B_T(u) = \sum_{s \neq t \neq u} \frac{g_{st}(u)}{g_{st}} \tag{5}$$

Note that path length is computed based on the total time taken to traverse it (at a rate of one edge per time step) rather than the number of edges in the path. This incorporates a notion of delay into the distance metric.

Calculating the shortest paths is accomplished by forming a new directed acyclic graph based on the undirected, dynamic graph representing the social network. For every node n and every time step t a proxy node n_t is created in the D.A.G. which represents that node at that moment in time. An edge is added from every node's proxy at time t to the same node's proxy at time $t+1$. Traversing this edge represents remaining at the same node node for two consecutive time steps in the original graph. Additionally, for every edge in the original graph $e = (u, v),\ e \in E_t$ — where E_t is the set of edges extant at time t — an edge is added to the D.A.G. from u_t to v_{t+1} and from v_t to u_{t+1}. This represents the ability to move from either u to v or vice versa, as well as the one time step delay that this incurs.

By encapsulating the dynamic nature of the graph in this newly constructed directed acyclic graph we gain the ability to ignore the temporality of the original graph and do our calculations just as we would for a static graph. (See Brandes [2008] for details on efficient calculation of betweenness.)

2.3.4. Temporal Clustering Coefficient

The Temporal Clustering Coefficient is an extension of the standard clustering coefficient, which measures how many of a node's neighbors are also neighbors of each other. One could measure this for any one time step of the network, but this would not capture any of the temporality of the network.

In order to account for the dynamism of the network we extend the standard definition to count any connections between neighbors in the previous c time steps. That is, of all of the current neighbors of node n, how many times in the previous c time steps do edges exist between them? This figure is divided by the total number of edges that could have existed between this group, which is given by $\frac{1}{2}c(k)(k-1)$, where k is the number of neighbors of n.

This definition accounts for situations in which a node's neighbors are not currently linked, but often are and have been recently. Consider, for instance, building a social network based on the conversations of office workers. Measuring the standard clustering coefficient on a Saturday would yield very low results despite the fact that for the previous five days the network was densely connected. The temporal clustering coefficient accounts for this sort situation.

Depending on the network structure it may be optimal to select the most highly clustered or the least highly clustered nodes, so in all experiments we choose an activation set using both. Highly clustered nodes may be desirable in some circumstances because diffusion can spread through tight cluster very rapidly in the first few time steps, quickly acting to increase the activated portion of the network. On the other hand, poorly clustered nodes may be beneficial since a low clustering coefficient may indicate that they have ties to many different neighborhoods in the graph, which presents the chance to spread activation throughout the network.

2.4. Random

For purposes of baseline comparison, all networks were also activated with an initial set of randomly selected nodes.

3. Results

3.1. Linear Threshold Model

For experiments with the Linear Threshold Model networks of 64 nodes were used. The initialization period lasted 25 time steps, the diffusion period lasted between 200 and 300 steps, and the threshold for activation was set at $\theta = 1/2$ for all nodes. Edge density was set at 10% of all possible edges. One hundred random trials for each combination of network dynamics and centrality measure were conducted and the results averaged.

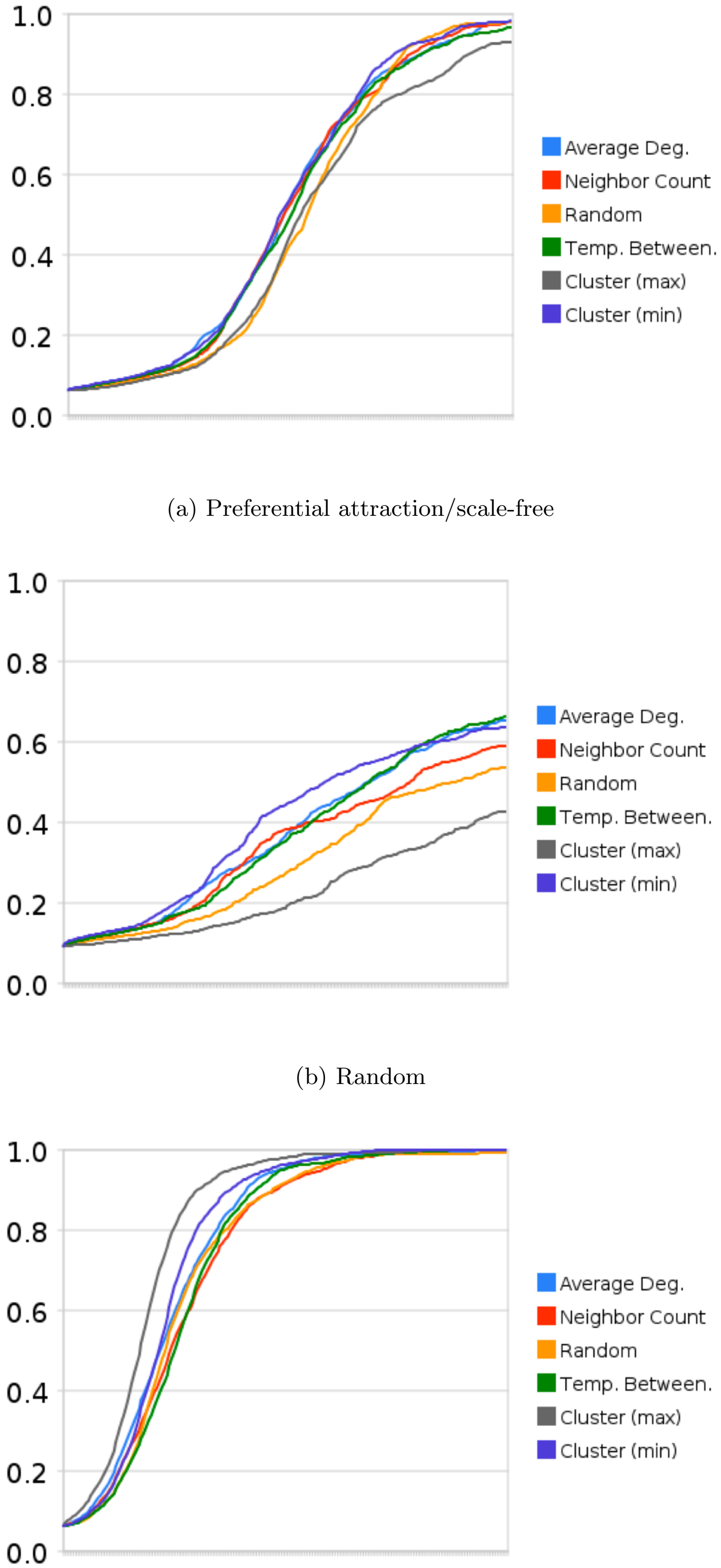
For the first set of experiments, results of which can be seen in Figure 1, the initial seed group was 4 nodes and 10% of edges were culled and replaced every time step. There is very little differentiation between the metrics used, but minimizing temporal clustering is a clear loser. Early in the diffusion period random selections are also poor, but by later in the diffusion period it performs as well as any other technique.

The random networks proved more resistant to diffusion no matter what heuristic was used. This is likely because no particular nodes are likely to be particularly central to the network since edges are evenly distributed, in contrast to the preferential attraction system. The best and worst measures were both clustering coefficient, with the minimal clustering method doing the best and maximal clustering doing the worst. Average degree and temporal betweenness tie for the next best results.

Neighbor count out performs random, but does the worst other than maximal clustering. A mentioned previously, there is no reason to expect that a long-lived node in a random graph will continue to persist, so it is reasonable that the weighted neighbor count performs poorly in this situation. It performed better in the preferential attraction networks. This is likely because while there is no mechanism to cause long-lasting edges to remain, any edges removed from influential nodes with a high degree (and therefor likely to have a high weighted neighbor count as well) have a better than average chance of being added back in in later time steps.

In proximity-based networks both maximal clustering coefficient has the highest performance by a wide margin, followed closely by minimal clustering. Random selections and weighted neighbor count perform the worst. It is unclear why both highly clustered and poorly clustered nodes would be good choices in this situation.

Figure 2 shows the effect that varying the size of the initial activation set has on the diffusion on preferential attraction networks. Other than the size of the activation seed, all other parameters remain the same. Figure 2(a) shows the default of $|A_0| = 4$, while in Figure 2(b) $|A_0| = 8$ and in Figure 2(c) $|A_0| = 16$. (Note that Figure 1(a) and Figure 2(a) represent the same data.) As the size of the initial set becomes larger the result of the different metrics become more varied. In the first case Average Degree, Neghbor Count and Temporal Betweenness are all indistinguishable, but in the final case they are clearly ordered. We hypothesize this is occurring because the same nodes are



(a) Preferential attraction/scale-free

(b) Random

(c) Proximity

Figure 1. LTM results: proportion of nodes activated in 200 time steps. (Best viewed in color.)

ranked in the top positions by each metric, and as a result they are choosing the same initial activation set. However, moving further down the various centrality rankings the lists that each metric generates begin to become different, which allows some metrics to out perform others on larger seed sets.

Another set of experiments was done varying the rate at which edges were replaced, but leaving all other paramters the same. The results for preferential attraction networks can be seen in Figure 2. Figure 3(a) shows the default rate of 10% of edges being culled and replaced each time step, while in Figures 3(b) and 3(c) the rate is 5% and 2.5%, respectively. (Note again that Figure 1(a) and Figure 3(a) represent the same data.) As the rate at which edges are replaced decreases and the network becomes more static the results of the metrics become more differentiated. In the latter case, maximal clustering stands out as worse than random selections, which are in turn worse than minimal clustering. The remaining three options are not significantly different from each other. It is likely that each metric's results are more tightlty grouped with higher turnover of the edges because the resulting graphs are in essense too dynamic. They change so quickly that no observation of the initial 25 time steps will tell one much about the state of the graph after 100 or more time steps of diffusion.

3.2. Independent Cascade Model

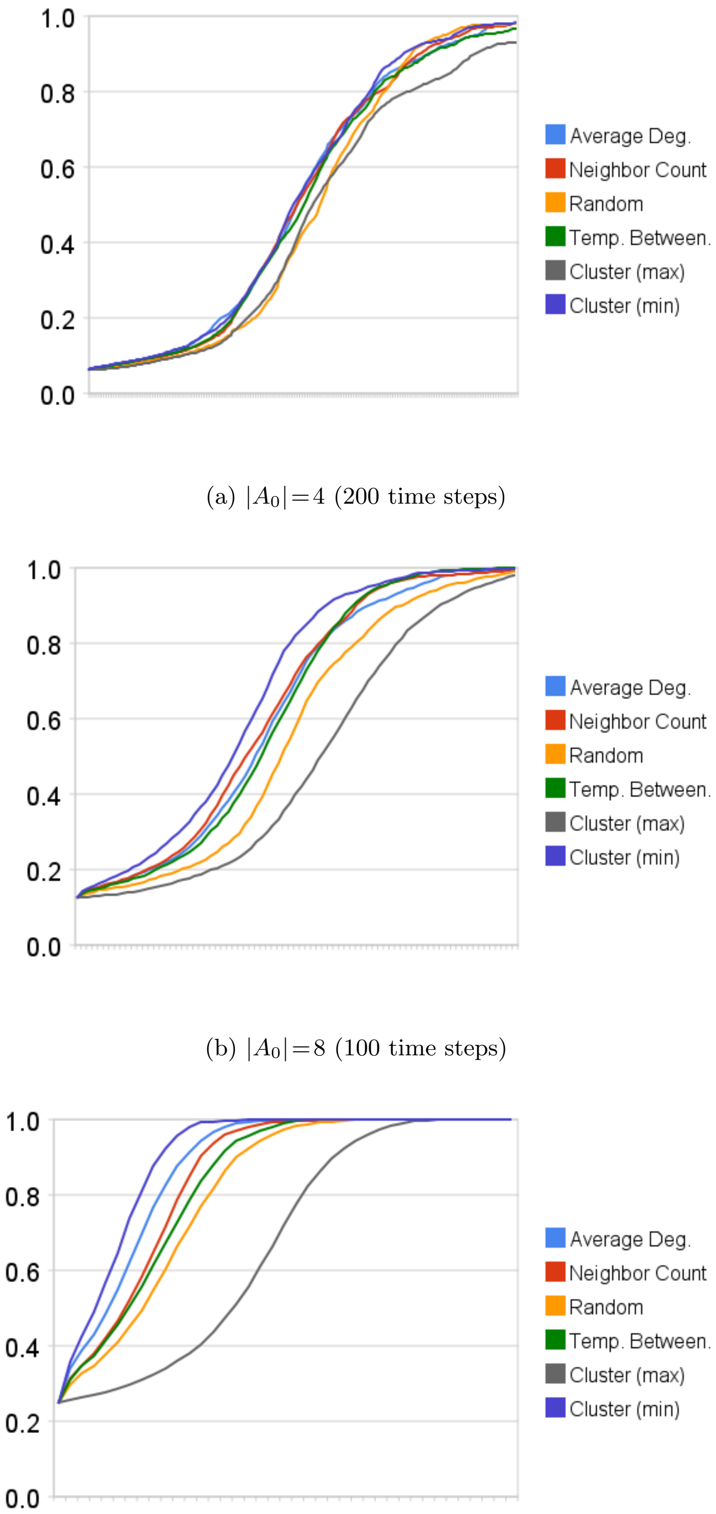
For experiments with the Independent Cascade Model networks of 64 nodes were used, with an initial seed set of size 3 and an edge density of 10%. The probability of activation spreading to an inactive node from an active neighbor was $1/3$. The initialization period lasted for 25 steps and the diffusion period ran for 50 steps, though the networks almost always stabilized before that. For each set up 100 random trials of the simulation were run and the results averaged.

Results for scale-free, random and proximity networks can be seen in Figures 4(a), 4(b), and 4(c), respectively. Shown is the proportion of the network activated at each time step.

For scale-free networks the maximum clustered nodes perform the worst. Minimally clustered nodes perform indistinguishably from a random selection. Average degree, neighbor count and temporal betweenness are also indistinguishable from each other. The only trend that can be drawn from the random graphs is that maximal clustering is the worst performer. No other other deviated from the performance of random selection. There was more differentiation in the proximity-based networks. Again, maximal temporal clustering was the worst performer, followed by minimal clustering. However, no metric out-perfromed random selection. (Weighted neighbor count does no worse than random though.)

4. Discussion

No metric emerged as a clear winner. Average degree, temporal betweenness and weighted neighbor count were consistently the best measures, however there was little differentiation between them. This may be because the networks were changing too rapidly for statistics computed on the initialization period to carry

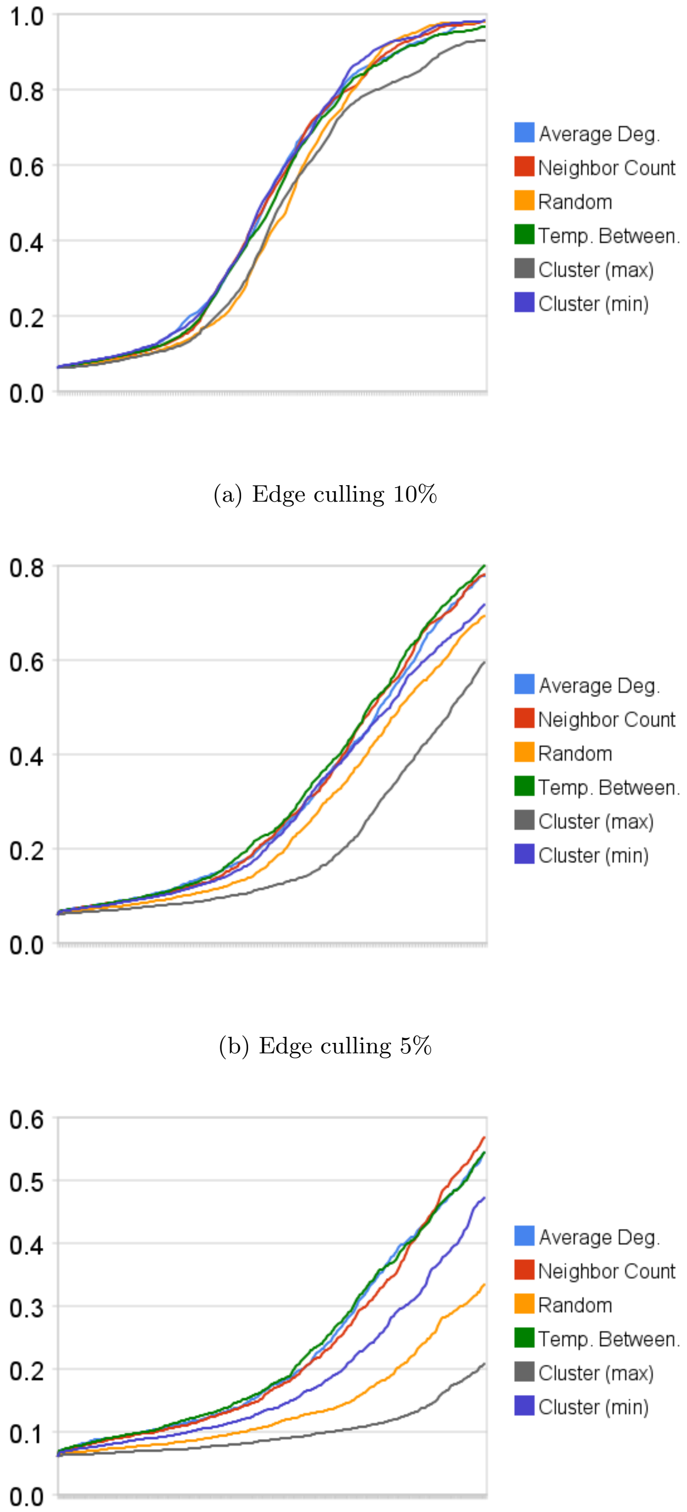


(a) $|A_0|=4$ (200 time steps)

(b) $|A_0|=8$ (100 time steps)

(c) $|A_0|=16$ (50 time steps)

Figure 2. LTM results for scale-free networks with various sizes of the initial activation set. (Best viewed in color.)

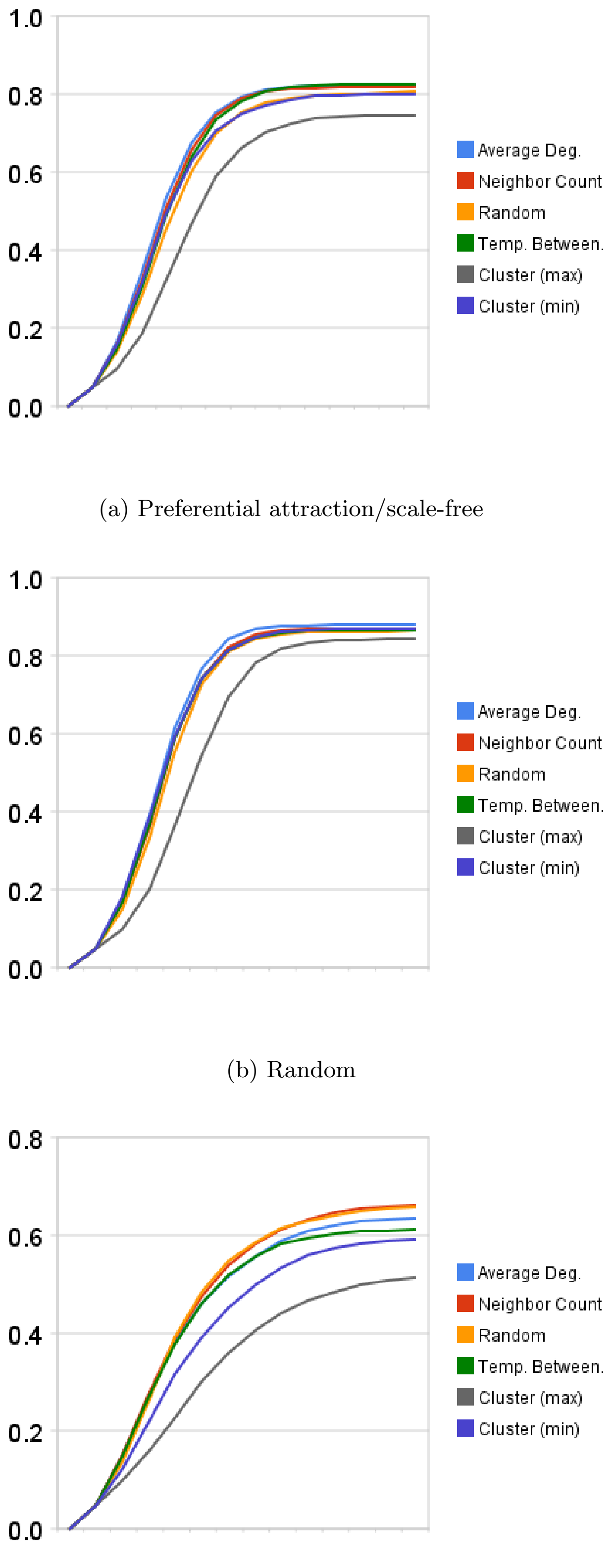


(a) Edge culling 10%

(b) Edge culling 5%

(c) Edge culling 2.5%

Figure 3. LTM results for scale-free networks with varying degree of edge dynamism. (Best viewed in color.)



(a) Preferential attraction/scale-free

(b) Random

(c) Proximity

Figure 4. ICM results: proportion of nodes activated in 15 time steps. (Best viewed in color.)

much information about the network hundreds of time steps later. Another possibility is that the initial activation sets chosen by the different measures had a high degree of overlap.

In general, choosing the nodes with the lowest clustering coefficient outperformed those with the highest clustering coefficient. Minimal clustering coefficient selection were mixed: in most scenarios they outperformed random selections but not the other metrics, but in a few situations they performed the best of all methods.

Weighted neighbor count performed better in preferential attraction networks than in random or proximity-based networks. This is likely because while there is no bias to keeping edges of highly-connected nodes, when such nodes are removed there is a higher probability that new edges will soon be added back into that node,

Experiments up to this point were limited by computational resources, though improvements continue to be made in processing efficiency as the simulation code base is refined. As efficiency improves it would be worthwhile to study much larger networks, with larger initial activation sets. A larger real-world data set would also be helpful.

There are a number of other extensions to be made in the future. Refinements can be made to the current network dynamics, such as adding a more realistic movement system to the proximity networks, or biasing edge culling to favor the continued existence of long-lived edges. It would also be interesting to look at an oscillating or rhythmic network structure: one in which there are predictable cycles of connectivity.

It may be worthwhile to change the Linear Threshold model to Granovetter's original formulation with each node having it's own threshold. There was not opportunity to dig deeper into Independent Cascade Models. At the very least it would be worthwhile to study the effect of initial activation set size with ICM diffusion.

Also it would be a good idea to use a greedy algorithm that reevaluates the centrality of each node after removing prior picks from the graph. This was not done in this version due to computational time limitations.

Some changes could be made to the metrics studied here, and some new metrics could be introduced. Another potential heuristic, Load Centrality, was defined by Goh et al. [2001]. Load centrality is very similar to temporal betweenness centrality, but may yield slightly different results. The temporal clustering coefficient is admittedly a bit crude at this point. It would be useful to weight more recent connections between neighbors more heavily than connections further in the past.

It would also be good to examine the different sets being selected by each metric in more detail. Specifically, are various measures ending up with similar performance because they are equally good at selecting activation sets, or because they are all selecting the same activation sets?

References

Abrahamson, E., & Rosenkopf, L. (1997). Social network effects on the extent of innovation diffusion: A computer simulation. *Organization Science*, *8*, 289–309.

Barabasi, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*, 509–511.

Berger, E. (2001). Dynamic models of constant size. *Journal of Combinatorial Theory Series B*, *83*, 191–200.

Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, *30*, 136–145.

Do, Y. (2008). *Centrality Analysis in Vehicular Ad Hoc Networks*. Technical Report 1 Ecole Polytechnique Federale de Lausanne.

Goh, K. I., Kahng, B., & Kim, D. (2001). Universal behavior of load distribution in scale-free networks. *Physical Review Letters*, *87*, 278701.

Goldenberg, J., Libai, B., & Muller, E. (2001). Talk of the network: A complex system look at the underlying process of word-of-mouth. *Marketing Letters*, *12*, 211–223.

Granovetter, M. (1973). The strength of weak ties. *American Journal of Sociology*, *78*, 1360–1380.

Granovetter, M. (1978). Threshold models of collective behavior. *American Journal of Sociology*, *83*, 1420–1443.

Granovetter, M. (1983). The strength of weak ties: A network theory revisited. *Sociological Theory*, *1*, 201–233.

Habiba, & Berger-Wolf, T. (2007). *Maximizing the Extent of Spread in a Dynamic Network*. Technical Report 20 Rutgers DIMACS.

Habiba, Tantipathananandh, C., & Berger-Wolf, T. (2007). *Betweenness Centrality Measure in Dynamic Networks*. Technical Report 19 Rutgers DIMACS.

Habiba, Yu, Y., Berger-Wolf, T., & Saia, J. (2008). Finding spread blockers in dynamic networks. In *Proc. 2nd Int'l Conf. on Advances in Social Network Mining & Analysis* (pp. 55–76).

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proc. 9th ACM Int'l SIGKDD Conf.* (pp. 137–146).

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proc. 13th ACM Int'l SIGKDD Conf.* (pp. 420–429).

Newman, M., Barabasi, A.-L., & Watts, D. J. (2006). *The Structure and Dynamics of Networks*. Princeton University Press.

Newman, M., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*, 026113.

Schelling, T. (1978). *Micromotives and Macrobehavior*. Norton.

Stonedahl, F., Rand, W., & Wilensky, U. (2008). Multi-agent learning with a distributed genetic algorithm. In *Proc. Autonomous Agents & Multi-Agent Systems Conf.: ALAMAS+ALAg Workshop*.

Valente, T. W. (1995). *Network Models of the Diffusion of Innovations*. Hampton Press.

Wang, Z., & Crowcroft, J. (1992). Analysis of shortest-path routing algorithms in a dynamic network environment. *ACM SIGCOMM Computer Communication Rev.*, *22*, 63–71.