Circle-Based Tipping Cascades in Social Networks

Paulo Shakarian
Network Science Center and
Dept. of Electrical Engineering
and Computer Science
U.S. Military Academy
West Point, NY 10996
paulo@shakarian.net

Luke Gerdes
Network Science Center and
Dept. Behavorial Science and
Leadership
U.S. Military Academy
West Point, NY 10996
luke.gerdes@usma.edu

Hansheng Lei
Network Science Center and
Dept. of Electrical Engineering
and Computer Science
U.S. Military Academy
West Point, NY 10996
hansheng.lei@usma.edu

ABSTRACT

Traditional models of cascades in social networks are based on the idea that individuals become "activated" once a certain number of their friends are active. However, some recent work has suggested that in some cases individuals become active based on the receipt of social signals from *circles* of their friends. In this paper, we introduce a deterministic model of this circle-based activation process and study the problem of finding a minimal subset of initially-activated individuals to produce a cascade that activates the entire population. We design decomposition-based heuristics for this problem and experimentally evaluate them on various simulated and real-world datasets. Our empirical results not only provide insight into the development of heuristics for this problem, but also shed light on the effect the number of circles has upon the cascades.

Categories and Subject Descriptors

Applied Computing [**Law, social and behavioral sciences**]: Sociology

General Terms

Algorithms, Experimentation

Keywords

complex networks, network diffusion

1. INTRODUCTION

Classic models of diffusion in social networks [4, 9] often view a node as "active" when it adopts a certain new characteristic based its neighbors possession of the same characteristic. However, a recent empirical study [8] suggests that there are some cases in which individuals' activation depends on the number of previously active friends *in unique friendship circles.*

Consider a simple example. Joe has an old college friend who recommends an online news article. According to the circles approach, an identical recommendation from a second old college friend would not increase Joe's likelihood of reading the piece. However, if a friend from a different sphere of Joe's life, such as a co-worker or neighbor, recommended the same article, then the additional influence could potentially lead Joe to read the piece.

In this paper, we make three main contributions. First (Section 2), we introduce our new model of tipping based on the circles-based pattern of influence described in the previous example. Our approach extends the deterministic version of the linear threshold model of [4]. Second (Section 3), we introduce our class of decomposition-based heuristics for the problem. Finally, (Section 4) we perform an experimental evaluation. From a performance standpoint, we find experimentally that our heuristics can quickly find small seed-sets under this model and outperformed the standard centrality measures. However, in addition, we observe a linear relationship between the average number of circles per individual and the seed-set size (as determined by our heuristic) necessary to activate the entire network.

2. MODEL

Throughout this paper we assume the existence of a *social network,* $G = (V, E, L, \mathit{lf})$, where V is a set of vertices and E is a set of directed edges, L is a set of labels and $\mathit{lf}\colon E \to 2^L$. We will use the notation n and m for the cardinality of V and E respectively. For a given node $v_i \in V$, the set of incoming neighbors is η_i^{in}, and the set of outgoing neighbors is η_i^{out}. The cardinalities of these sets (and hence the in- and out-degrees of node v_i) are k_i^{in}, k_i^{out} respectively. If the network is bi-directional, we shall use the notation k_i for both in- and out-degree as they are the same in this case. For some $\ell \in L$, we shall also use $\eta_i^{in}(\ell)$ to denote the set $\{v_j \in \eta_i^{in} | \ell \in \mathit{lf}((v_j, v_i))\}$ (the set of all incoming neighbors with label ℓ). We will use analogous notation for the outgoing neighbors with a particular label.

By labeling the edges, we have essentially imposed a community structure on the neighborhood of each individual node. For instance, a given node v_i has neighbors in the following circles: $\mathsf{cir}_i = \{\ell | \eta_i^{in}(\ell) \neq \emptyset\}$. We shall use the notation $\lambda_i = |\mathsf{cir}_i|$.

It is relatively simple to engineer a function lf based on certain properties of the subgraph induced by a node's immediate neighborhood - for instance by using a community-finding algorithm or the identification of connected components. One such method was recently proposed in [8]. In this paper, we are concerned with some initial set of "activated" nodes in a cascade process. While previous work [2, 4] was

The authors are supported by the Army Research Office (project 2GDATXR042), the U.S. Air Force, and the Minerva Initiative. The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the funders, the U.S. Military Academy, the U.S. Army, the U.S. Air Force, or the U.S. Department of Defense.
.

concerned with nodal activation based on a threshold of activated neighbors, here we are concerned about a threshold of activated circles. Hence, we must then define what it means for a circle to be activated. In this paper, we shall say that a circle is activated if at least one individual in that circle is active (we shall leave other alternatives to future work). Therefore, for a given node v_i and set of activated nodes V', we shall define v_i's active circles as follows:

$$\mathsf{act}_i(V') = \{\ell \in \mathsf{cir}_i | \eta_i^{in}(\ell) \cap V' \neq \emptyset\}$$

We now define the threshold vector which specifies, for each node, the number of circles that must be activated for it to become activate as well.

Definition 2.1 (Threshold Vector). *A **threshold vector**, κ is a vector of size n s.t. each component $\kappa_i \in \{0, \ldots, \lambda_i\}$.*

We now define an *activation function* that, given an initial set of active nodes, returns a set of active nodes after one time step.

Definition 2.2 (Activation Function). *Given a social network G and threshold vector, κ, an **activation function** A_κ maps subsets of V to subsets of V, where for some $V' \subseteq V$,*

$$A_{G,\kappa}(V') = V' \cup \{v_i \in V \text{ s.t. } |\mathsf{act}_i(V')| \geq \kappa_i\} \tag{1}$$

We also note that the activation function can be applied iteratively, to model a diffusion process. Hence, we shall use the following notation to signify multiple applications of A (for natural numbers $t > 1$).

$$A^t_{G,\kappa}(V') = \begin{cases} A_{G,\kappa}(V') & \text{if } t = 1 \\ A_{G,\kappa}(A^{t-1}_{G,\kappa}(V')) & \text{otherwise} \end{cases} \tag{2}$$

Clearly, when $A^t_{G,\kappa}(V') = A^{t-1}_{G,\kappa}(V')$ the process has converged. Further, this always converges in no more than n steps, since the process must activate at least one new node in each step prior to converging. Based on this idea, we define the function Γ which returns the set of all nodes activated upon the convergence of the activation function.

Definition 2.3 (Γ Function). *Let t be the least value such that $A^t_{G,\kappa}(V') = A^{t-1}_{G,\kappa}(V')$. We define the function $\Gamma_{G,\kappa} : 2^V \to 2^V$ as follows.*

$$\Gamma_{G,\kappa}(V') = A^t_{G,\kappa}(V') \tag{3}$$

We now have all the pieces to introduce our problem: finding the minimal number of nodes that are initially active to ensure that the entire set V becomes active.

Definition 2.4 (The CirMS Problem). *The Circle-Based Minimum Seed (CirMS) Problem is defined as follows: given a social network G, threshold vector, κ, return $V' \subseteq V$ s.t. $\Gamma_{G,\kappa}(V') = V$, and there does not exist $V'' \subseteq V$ where $|V''| < |V'|$ and $\Gamma_{G,\kappa}(V'') = V$.*

We note that by assigning each edge a single unique label, we can embed the minimum seed problem for the standard deterministic tipping model, which is NP-Complete [2, 4], providing us with the following result:

Theorem 2.1 (Complexity of CirMS). *CirMS is NP-Complete.*

3. APPROACH

Due to the NP-Completeness of the CirMS Problem, a general approach that provides an exact solution in polynomial time is not likely to exist. However, we note that under the standard deterministic tipping model (the special case of CirMS where each edge label is unique), the decomposition heuristic of [7] has been shown to provide small seed-sets in polynomial time as well as the ability to scale to very large social networks. Hence, our first algorithm, CirSeedDecomp (Algorithm 1), is based on this idea. However, unlike the algorithm of [7], our approach iteratively peels nodes from the network based on the number of adjacent circles, rather than adjacent nodes. This distinction provides us a new and different selection criterion. Further, we also introduce an improvement to the algorithm not presented in [7]. Later, we show experimentally that this modification leads to significantly improved solution quality.

To illustrate how CirSeedDecomp works, consider the bidirectional network in Figure 1. Let us consider where for each node v_i, $\kappa_i = \lceil \frac{\lambda_i}{3} \rceil$. The algorithm assigns each node a "distance to being tipped" (*dist*), which is the difference between the number of adjacent labels and the number of labels required to tip the node. For Figure 1, *dist* is 0 for nodes $1, 2, 8, 9, 12$; 1 for nodes $3, 4, 5, 10, 11$; and 2 for nodes $6, 7$. The algorithm removes a node of minimal non-negative distance from the network. Thus, if the algorithm removes node 11 first, the distance of node 10 will update to 0, and the distance of node 12 will update to -1 (thereby making it part of the solution). The process then continues, with the removal of nodes $8, 10, 9, 1, 2, 3, 7, 4, 6$. Hence, $\{5, 12\}$ is returned as a solution. Using a simple proof by induction, we can show that the solution returned by the algorithm will always create a cascade that causes the entire population to activate, though this solution is not necessarily a minimal sized seed set. Further, we can also show that this algorithm runs in polynomial time.

Proposition 3.1. *If $V' \subseteq V$ is a set returned by CirSeedDecomp, then $\Gamma_{G,\kappa}(V') = V$.*

Proposition 3.2. *CirSeedDecomp runs can be implemented in $O(mL\log(n))$ time, where m is the number of nodes, L is the maximum number of labels per edge, and n is the number of nodes.*

Improvement. However, we note that in our above example that CirSeedDecomp provides a redundant element in the solution, because node 12 is not required to activate the entire network. Fortunately, the algorithm provides an opportunity for further improvement. When confronted with more than one node of the same minimal non-negative distance to being tipped, there is no criteria by which to select a "best" minimal node. Hence, we introduce a "tie-breaker" heuristic. In this case, we replace line 5 with the following:

```
1: Let $V_{\min} = \{v_i | dist_i \text{ is minimal}\}$
2: For a given edge $(v_i, v_j)$, let $RED((v_i, v_j)) = \{\ell_q \in lf((v_i, v_j)) | LBL_{j,q} = 1\}$
3: For a given node $v_i$, let $REM(v_i) = \{(v_i, v_j) \in E | dist_j - |RED((v_i, v_j))| < 0\}$
4: Let $v_i$ be the element of $V_{\min}$ where $|REM(v_i)|$ is minimal
```

This modification requires the algorithm to examine all candidates for removal (minimal, non-negative distance) in

Line 1. For each candidate node, the algorithm identifies the set of neighbors that would have a negative distance if the node in question were removed (Lines 2-4). Our intuition is that once a node receives a negative distance, it will be part of the solution, and our modification attempts to delay that from occurring. Thus, according to our approach, node 11 would not be immediately removed, ensuring that node 12 would keep a non-negative distance at that point.

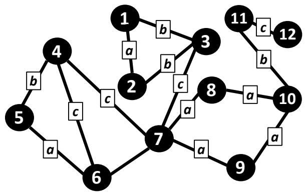

Figure 1: Example labeled network.

Algorithm 1 CirSeedDecomp

```
Require: Threshold vector, $\kappa$ and directed social network
    $G = (V, E, L, lf)$
Ensure: $V'$
 1: For each vertex $v_i$, $dist_i = \lambda_i - \kappa_i$
 2: Let $LBL$ be a matrix array where component $LBL_{i,j}$
    corresponds to vertex $v_i$ and label $\ell_j$. Initially, set
    $LBL_{i,j} = |\eta_i^{in}(\ell_j)|$.
 3: $FLAG = TRUE$.
 4: while $FLAG$ do
 5:   Let $v_i$ be the element of $V$ where $dist_i$ is minimal.
 6:   $FLAG = (dist_i \neq \infty)$
 7:   if $FLAG$ then
 8:     Remove $v_i$ from $G$
 9:     for $v_j$ in $\eta_i^{out}$ do
10:       for $\ell_q \in lf((v_i, v_j))$ do
11:         $LBL_{j,q} = LBL_{j,q} - 1$
12:         if $LBL_{j,q} = 0$ then
13:           $dist_j = dist_j - 1$
14:         end if
15:       end for
16:       If $dist_j < 0$ then $dist_j = \infty$
17:     end for
18:   end if
19: end while
20: return All nodes left in $G$.
```

4. EXPERIMENTAL RESULTS

Our implementation was written in Python 2.7 using the NetworkX library. [1] Our implementation consisted of approximately 500 lines of code. The code used a binomial heap library written by Björn B. Brandenburg available from http://www.cs.unc.edu/∼bbb/. The experiments were run on a computer equipped with an Intel X5677 Xeon Processor operating at 3.46 GHz with a 12 MB Cache running Red Hat Enterprise Linux version 6.1 and equipped with 70 GB of physical memory.

We used three existing datasets: an academic collaboration network for High Energy Physics [5] consisting of $8,638$ nodes and $49,612$ edges; an e-mail network [3] consisting of $1,133$ nodes and $10,902$ edges; and a sample of the Douban social network [10] consisting of $154,908$ nodes and $654,324$ edges. As these were undirected networks, we treated their edges as bi-directional. We also utilized only the greatest connected component. To obtain edge labels, we partitioned the network using the Louvain algorithm [1] and assigned the outgoing edges of each node the number of the community identified by the Louvain method. In this section, we shall use the symbol $\delta = \frac{1}{n}\sum_i \frac{\lambda_i}{k_i^{in}}$, the average ratio of adjacent node circles over in-degree, for a given network. For the academic collaboration network, $\delta = 0.486$, for the e-mail network, $\delta = 0.460$, and for the social network, $\delta = 0.894$. The number of communities found in the e-mail, collaboration, and social networks were 10, 51, and 84 respectively.

Solution Quality and Comparison. We evaluated the CirSeedDecomp algorithm, both with and without the tie-breaker heuristic (which we shall refer to as "TB"). These evaluations considered the size of the seed-sets returned on our datasets under different threshold settings (see Figure 2). We explored "integer thresholds" where every node, v_i, had κ_i was set to $\min(j, \lambda_i)$ where j varied for each trial (though was the same for all nodes) and was a natural in the interval $[1, 10]$. Additionally, we explored "fractional thresholds" where for every node, v_i $\kappa_i = \lceil \frac{j\lambda_i}{10} \rceil$ where again j varied on each trial (though was the same for all nodes) and was a natural in $[1, 10]$. Once we determined the seed-set for both CirSeedDecomp as well as CirSeedDecomp with TB, we then evaluated the outcome of our diffusion model for a similarly-sized set of the top nodes as determined by nodal degree and PageRank [6]. For our integer threshold tests, we found that the addition of the TB heuristic consistently reduced the size of the seed-set by about half as compared to CirSeedDecomp alone. For these trials, beyond the trivial threshold of 1, we found that similarly-sized seed-sets produced by degree and PageRank would not lead to the activation of the entire population (except in the case of the social network when the tie breaker was not applied). Particularly striking, in the cases where the tie-breaker heuristic was used, similarly-sized sets of top nodes, as determined by degree and PageRank, resulted in 10 to 30 percent of nodes remaining unactivated. Moreover, these results held even in the case of the social network, and we obtained similar results when using fractional thresholds, especially when the fraction of activated neighbors exceeded 0.5 (not pictured).

Runtime. We also examined the runtime of the CirSeedDecomp heuristic, both with and without the tie-breaker heuristic (Table 1). We found that the extra over-head for maintaining the information required to determine the tie-breaker was expensive in our current implementation. As a result, we are currently exploring more efficient data structures to maintain this information.

Varying the Number of Circles Per Node. We generated 96 Erdos-Reyni random graphs each containing 100 nodes. We the labeled the edges based on the results of a hierarchical clustering algorithm, which partitioned nodes into structural trees. By cutting these trees at varying heights, we were able to generate 10 variations on community struc-

[1] http://networkx.github.io/

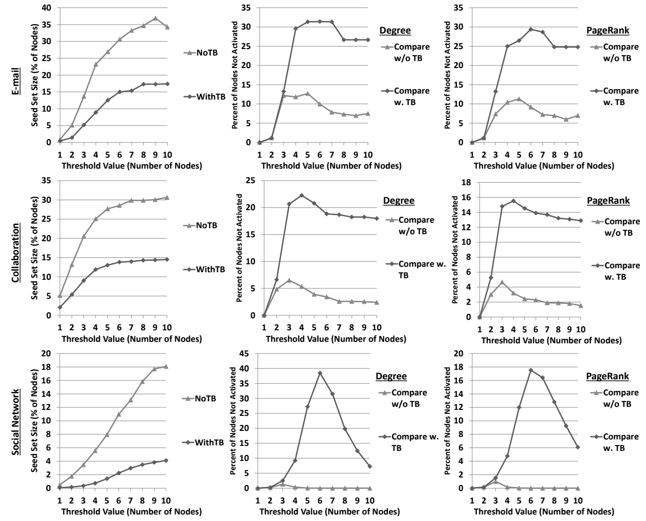

Figure 2: Performance of heuristics (integer thresholds) on the e-mail network (top), collaboration network (bottom), and social network (bottom row) in terms of seed-set size (left) and comparisons to degree (middle) and PageRank (far right).

Table 1: Average Runtime in Seconds

Dataset	CirSeedDecomp	With Tie-Breaker
Social Network	6.863	7204.800
E-mail	0.033	5.840
Collaboration	0.275	14.209

ture and social circles from each of the 96 clustering. Because the underlying structure of each test-network remained unchanged, we held nodal degree constant, while varying the number of circles to which nodes held ties, thereby causing the quantity δ to vary. This approach to permuting community structure allowed us to study the effect on δ on the seed set size returned by our heuristic. In this experiment, we studied the 96 graphs and the 10 label settings with respect to two often-studied thresholds: majority ($\forall v_i, \kappa_i = \lceil \frac{\lambda_i}{2} \rceil$) and full thresholds ($\forall v_i, \kappa_i = \lambda_i$). We found a strong linear relationship between delta and the size of the seed set (as returned by our heuristic with the tie-breaker). Results are depicted in Figure 3. For the majority thresholds, $R^2 = 0.908$. For the full thresholds, $R^2 = 0.977$.

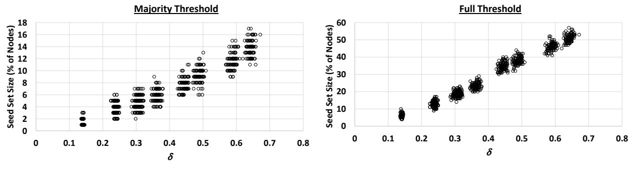

Figure 3: The relationship between δ and the seed set size for 96 Erdos-Reyni graphs.

5. CONCLUSION

In this paper, we introduced a new diffusion model for social networks where nodes are tipped based on the number of activated communities rather than neighbors. We studied the problem of identifying a minimal seed-set necessary to activate the entire network, and we introduced an efficient heuristic that we show to perform well in practice. We also studied the effect that the number of circles per node has on the size of a seed-set found by our heuristic. We intend to further study the behavior of our decomposition heuristic in order to improve it in terms of both runtime and solution quality. We also plan to examine extensions to the model, including a probabilistic version.

6. ACKNOWLEDGMENTS

The authors are supported by the Army Research Office (project 2GDATXR042), the U.S. Air Force, and the Minerva Initiative. The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the funders, the U.S. Military Academy, the U.S. Army, the U.S. Air Force, or the U.S. Department of Defense.

7. REFERENCES

[1] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008, 2008.

[2] P. Dreyer and F. Roberts. Irreversible -threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Applied Mathematics*, 157(7):1615 – 1627, 2009.

[3] R. Guimerà, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Phys. Rev. E*, 68:065103, Dec 2003.

[4] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, New York, NY, USA, 2003. ACM.

[5] J. Leskovec. Stanford network analysis project (SNAP), 2012.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, 1998.

[7] P. Shakarian and D. Paulo. Large social networks can be targeted for viral marketing with small seed sets. *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1–8, 2012.

[8] J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg. Structural diversity in social contagion. *Proceedings of the National Academy of Sciences*, 2012.

[9] D. J. Watts and P. S. Dodds. Influentials, networks, and public opinion formation. *Journal of Consumer Research*, 34(4):441–458, 2007.

[10] R. Zafarani and H. Liu. Social computing data repository at ASU, 2009.