CUTTINGEDGE: Influence Minimization in Networks

Elias Khalil, Bistra Dilkina, Le Song
Georgia Institute of Technology
801 Atlantic Drive - Atlanta, GA 30332
{ekhalil3,bdilkina,lsong}@cc.gatech.edu

Abstract

The diffusion of undesirable phenomena over social, information and technological networks is a common problem in different domains. Domain experts typically intervene to advise solutions that mitigate the spread in question. However, these solutions are temporary, in that they only tackle the spread of a specific instance over the network, failing to account for the complex modalities of diffusion processes. We propose an optimization formulation for the problem of minimizing the spread of influence in a network by removing some of its edges. We show that the corresponding objective function is supermodular under the linear threshold model, allowing for a greedy approximate solution with provable guarantees. Preliminary experiments on real and synthetic network data show that our method significantly outperforms other common heuristics.

1 Introduction

The diffusion of ideas and influence is a topic of study across many disciplines ranging from viral marketing [6] and population epidemics [11], to social media [15] and other fields. A central element in any diffusion process is the *communication channel* along which the spread occurs. Recent efforts in network analysis, as well as the proliferation of real-world data of diffusion on the web, have established the network as a strong abstraction tool for spread phenomena: nodes in the underlying directed graph represent the individuals or entities in a given system, and edges represent the presence of a medium of communication between the nodes. Based on this representation, a number of methods have been devised with the goal of finding a set of k nodes whose adoption of a given idea would result in *maximizing* the spread of this idea in the network [6, 12]. This particular problem has been extended for different diffusion models [18], and increasingly efficient solutions are being proposed for it [7].

However, not much attention has been accorded to the study of negative phenomena that propagate in networks. Diseases that spread via contagion in societies, rumors that diffuse through blogs and news websites, and computer viruses that propagate in computer networks and the internet are all examples of processes, where the object of diffusion is considered harmful, hence undesirable. An obvious counter-measure in those situations is to call upon domain experts, e.g biologists and epidemics experts create new health and immunization measures, the New York Times or other reputable media outlets deny rumors and fake news, etc. But such measures may suffer from being too case-specific (tailored for one undesirable diffusion instance) rather than general, as well as too heuristic, in that they do not model cascading behavior properly, and rather make use of human judgment exclusively.

Adopting a more prinicipled computational approach to the problem, we ask the following question: what *structural* changes to the *network topology* would result in suppressing the influence of an undesireable diffusion process, in the best possible way? First, we consider the *linear threshold* model as a diffusion model [10]. This model is widely adopted by sociologists as representative of adoption dynamics, where each node or individual has a *threshold*, representing the fraction of its neighbors or connections that must adopt a certain idea before they adopt it themselves. Moreover, we do not make any assumptions about the nodes that initiate the diffusion in the network, i.e each node is equally likely to cause the initial spread. The spread susceptability of a network is defined as the sum of each node's individual expected influence.

The problem can then be formulated as follows: Given a network $G(V, E)$, a vector of diffusion probabilities w and a budget k, find the set of k edges E^*, such that the spread susceptability of the network $G^*(V, E \backslash E^*)$, resulting from removing $E^*$from E, is minimized.

Although the constrained optimization problem we propose is NP-hard, we show that the optimal solution can be approximated to a constant factor. This is due to the fact that under the linear threshold model the objective is monotonincally decsreasing and *supermodular*, a result we will prove in detail in this paper. Therefore, the *greedy* algorithm, based on iteratively adding to the chosen set the next best edge to remove in terms of marginal decrease in susceptability, until k edges have been selected, produces solutions that are within $(1 - 1/e)$ of the optimal value [17, 14]. We also prove that other network manipulation operations, such as adding edges, deleting nodes and adding nodes, are supermodular, as described in the Appendix 5.

We conduct computational experiments on both synthetic and real-world network data to evaluate our greedy method, comparing it to other heuristics that rely solely on the structural properties of the network (shortest paths, eigenvalues, degrees, etc), not making use of the probabilistic diffusion information on the edges. Experiments show that our method significantly outperforms the other heuristics.

Related Work: The topic of manipulating network structure to impact diffusion processes has been recently explored in [20, 19, 13, 2]. Several studies consider manipulating nodes. Sheldon et. al. [19] solve the problem of adding nodes to the network to maximize spread under the Independent Cascade Model using Sample Average Approximation combined with Mixed Integer Programming. Bogunovic [2] addresses the problem of finding the minimum set of nodes to block to guarantee a desired level of containment of the spread under the Independent Cascade model.

Kimura et. al. [13] attempt to solve the same edge-based influence minimization problem as ours for the Independent Cascade model by removing edges greedily, which we show does not yield approximations with guarantees. They compare their method to two other heuristics based on out-degree and edge betweenness centrality [8], which we will use as baselines in our study as well. Tong et. al. [20] also consider removing edges from the network but under a different cascade model, for which the eigenvalue of the adjacency matrix determines the epidemic threshold.

While the Independent Cascade model has been well studied, fewer have considered the Linear Threshold model. Chen et. al. [5] study influence maximization under the Linear Threshold Model and show that computing exact influence in general networks is $\#P-hard$. They propose a more scalable method for estimating influence under the linear threshold model than using Monte-Carlo simulations, an issue we intend to tackle in future work.

2 Cascade Models

An *influence graph* is a weighted directed graph $G = (V, E, w)$, where V is a set of n vertices (nodes) and E is a set of m directed edges, and $w : V \times V \to [0, 1]$ is a weight function such that $w(u, v) = 0$ if and only if $(u, v) \notin E$. Under the *Linear Threshold* (LT) model, we additionally have the requirement that $\sum_{u \in V} w_{uv} \leq 1$. Each time a cascade is propagated, every vertex v first independently selects a threshold θ_v uniformly at random in the range $[0, 1]$, corresponding to the lack of knowledge of users true thresholds. A cascade proceeds in discrete time steps $t = 0, 1, 2, 3, \ldots$ starting with a set of activated nodes $A = S_0$ where S_i denotes the set of nodes activated upto time t. An inactive node v becomes activate at time $t + 1$ if:

$$\sum_{u \in S_t} w_{u,v} \geq \theta_v \ .$$

The process terminates if no more activations are possible. Under the *Independent Cascade* (IC) model, started with a set of activated nodes A at time $t = 0$. At each discrete time step $t = 0, 1, 2, \ldots$ each newly activated node v is given a single attempt at activating each of its still inactive neighbors

u with probability of success w_{vu}, independently of the history this far. If v succeeds then u is newly activated at time $t+1$.

Given an influence graph $G = (V, E, w)$ and an initial set of active nodes $A \subset V$, we define the *influence* function $\sigma(A, G)$ as the expected number of active nodes at the end of the random diffusion process (for either of the *independent cascade* or *linear threshold* models).

Kempe et al.[12] showed that the linear threshold model is equivalent to the reachability in the following set of random graphs, called *live-edge graphs*: Given an influence graph $G = (V, E, w)$, for every node $v \in V$, select at most one of its incoming edges at random, where each edge $(u; v)$ is selected with probability $w(u; v)$, and no edge is selected with probability $1 - \sum_{u \in V} w(u; v)$. The random graph X generated by this process consists of all vertices in V and all selected edges, called *live*. Let us denote by $\mathcal{X}_G$ the set of all possible live-edge random graphs that can be generated from G. Kempe et al. [12] show that:

Proposition 1 *[Claim 2.6 of [12]]: Given an influence graph G and an initial set A, the distribution of the set of active nodes in G starting with A under the linear threshold model is the same as the distribution of the set of nodes reachable from A in the random graphs $\mathcal{X}_G$.*

Let us denote the set of all reachable nodes in X when starting from a set A by $r(A, X)$. Notice that the generation process for live-edge graphs guarantees that each node $v \in V$ has at most one parent in any live-edge graph$X$. Given our live-edge graph generation process, it is easy to see that the probability of a random live-edge graph $X = (V, E_X) \in \mathcal{X}$ is:

$$\Pr[X|G] = \prod_{v:(u;v) \in E_X} w(u; v) \prod_{v:\nexists(u;v) \in E_X} \left(1 - \sum_{(u;v) \in E} w(u; v)\right).$$

We can re-write the probability of a live-edge graph so that to isolate the contribution of each node: $\Pr[X|G] = \prod_{v \in V} p(v, X, G)$, where:

$$p(v, X, G) = \begin{cases} w(u, v) & \text{when } \exists(u, v) \in E_X \\ 1 - \sum_{(u,v) \in E} w(u, v) & \text{when } \nexists(u, v) \in E_X \end{cases}$$

For a subset $V' \subseteq V$, we will use the shorthand $p(V', X, G) = \prod_{u \in V'} p(u, X, G)$.

Clearly from Proposition 1 it follows that

$$\sigma(A, G) = \sum_{X \in \mathcal{X}_G} \Pr[X|G] \cdot r(A, X).$$

3 Deleting edges

We define the *susceptibility* of a graph G to diffusion as the sum of the expected influence of each node when it is the single source for a cascade, more precisely $\sigma(G) = \sum_{a \in V} \sigma(a, G)$.

In our setting, we are interested in manipulating the underlying influence graph in order to minimize its susceptibility to diffusions. In particular, we address the question of which set of k edges to delete such that the resulting graph has minimum susceptibility.

Given an influence graph $G = (V, E, w)$, deleting a set of edges $S \subseteq E$ results in an influence graph $G_S = (V, E \backslash S, w_S)$ with edge weights $w_S(u, v) = w(u, v)$ for edges $(u, v) \in E \backslash S$ and $w_S(u, v) = 0$ for $(u, v) \in S$. Our optimization problem is the following:

$$S^* = \arg \min_{S \subseteq E: |S| = k} \sum_{a \in V} \sigma(a, G_S)$$

We will show that $f_a(S) = \sigma(a, G_S)$ is a monotone and supermodular function. Since we will be modifying the set of edges in the influence graph, in a slight abuse of notation we will use $\mathcal{X}_{E \backslash S}$ instead of $\mathcal{X}_{G_S}$, $\Pr[X|E \backslash S]$ instead of $\Pr[X|G_S]$. From our earlier definition:

$$f_a(S) = \sigma(a, G_S) = \sum_{X \in \mathcal{X}_{E \backslash S}} \Pr[X|E \backslash S] \cdot r(a, X).$$

First, given an influence graph $G = (V, E, w)$, any edge set $S \subseteq E$ and an edge $e = (u, v) \in E \backslash S$, let us consider the sets of live-edge graphs $\mathcal{X}_{E\backslash S}$ and $\mathcal{X}_{E\backslash\{S\cup e\}}$ with respect to the influence graphs G_S and $G_{S\cup e}$ respectively. Let us partition the set $\mathcal{X}_{E\backslash S}$ into three subsets according to the live edge selected for node v: 1) the set of live-edge graphs $\mathcal{X}^{e}_{E\backslash S}$, where edge $e = (u, v)$ is selected for v ; $\mathcal{X}^{\bar{e}}_{E\backslash S}$, where another edge $\bar{e} = (y, v)$ was selected for v; $\mathcal{X}^{\emptyset}_{E\backslash S}$, where no edge was selected for v. This partition is illustrated in Figure 1 .

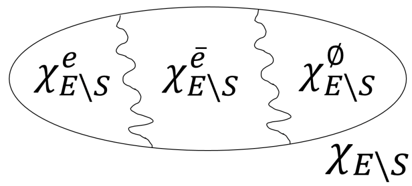


Figure 1: Venn diagram for the set of live-edge graphs $\mathcal{X}_{E\backslash S}$

The following two propositions characterize the relationship between $\mathcal{X}_{E\backslash S}$ and $\mathcal{X}_{E\backslash\{S\cup e\}}$.

Proposition 2 *Given an influence graph $G = (V, E, w)$, any edge set $S \subseteq E$ and an edge $e = (u, v) \in E\backslash S$, then $\mathcal{X}_{E\backslash\{S\cup e\}} = \mathcal{X}^{\bar{e}}_{E\backslash S} \cup \mathcal{X}^{\emptyset}_{E\backslash S}$.*

Proof It is easy to see that any live-edge graph $X \in \mathcal{X}_{E\backslash\{S\cup e\}}$ is also in $\mathcal{X}_{E\backslash S}$ since $S \subseteq S \cup e$. The set of live-edge graphs $\mathcal{X}_{E\backslash S}\backslash\mathcal{X}_{E\backslash\{S\cup e\}}$ are all the live-edge graphs that contain the edge e, namely $\mathcal{X}^{e}_{E\backslash S}$. ■

Proposition 3 *Given an influence graph $G = (V, E, w)$, any edge set $S \subseteq E$ and an edge $(u, v) \in E\backslash S$, there is a one-to-one mapping from $\mathcal{X}^{e}_{E\backslash S}$ to $\mathcal{X}^{\emptyset}_{E\backslash S}$, where $X \in \mathcal{X}^{\emptyset}_{E\backslash S}$ corresponds to $X_{(u,v)} = (V, E_X \cup (u, v)) \in \mathcal{X}^{e}_{E\backslash S}$.*

Now we are ready to show the following theorem:

Theorem 4 *f_a is a monotone decreasing function.*

Proof Given the influence graph $G = (V, E, w)$, we need to show that for any set $S \subseteq E$ and $e = (u, v) \in E\backslash S$:

$$f_a(S) - f_a(S \cup e) = \sum_{X\in\mathcal{X}_{E\backslash S}} \Pr[X|E\backslash S] \cdot r(a, X) - \sum_{X\in\mathcal{X}_{E\backslash\{S\cup e\}}} \Pr[X|E\backslash S \cup e] \cdot r(a, X) \geq 0$$

Using Proposition 2, we can rewrite the above equation as:

$$\begin{aligned} f_a(S) - f_a(S \cup e) &= \sum_{X\in\mathcal{X}^{e}_{E\backslash S}} \Pr[X|E\backslash S] \cdot r(a, X) \\ &+ \sum_{X\in\mathcal{X}^{\emptyset}_{E\backslash S}} (\Pr[X|E\backslash S] - \Pr[X|E\backslash S \cup e]) \cdot r(a, X) \\ &+ \sum_{X\in\mathcal{X}^{\bar{e}}_{E\backslash S}} (\Pr[X|E\backslash S] - \Pr[X|E\backslash S \cup e]) \cdot r(a, X) \end{aligned} \quad (1)$$

The probability of a live-edge graph $X \in \mathcal{X}_{E\backslash\{S\cup e\}}$ differs between the two influence graphs G_S and $G_{S\cup e}$ only in the calculation concerning node v, since all other nodes have the same set of possible parents with the same set of weights, i.e. we have $p(V\backslash v, X, G_S) = p(V\backslash v, X, G_{S\cup e})$.

For $X \in \mathcal{X}^{\bar{e}}_{E\backslash S}$, the probability is the same under both influence graphs, $p(v, X, G_S) = p(v, X, G_{S\cup e}) = w(\bar{e})$.

For all $X \in \mathcal{X}^{\emptyset}_{E\backslash S}$, the probability of selecting no edge for v differs between the two influence graphs. In particular, it easy to see that $p(v, X, G_S) = (1 - \sum_{(y,v)\in E\backslash S\cup e} w(y,x) - w(u;v)) = p(v, X, G_{S-cupe}) - w(u;v)$. Hence, $\Pr[X|E\backslash S] - \Pr[X|E\backslash S \cup e]) = -w(u,v)\cdot p(V\backslash v, X, G_S)$.

We can re-write Eq. 1 as:

$$f_a(S) - f_a(S\cup e) = \sum_{X\in\mathcal{X}^{e}_{E\backslash S}} \Pr[X|E\backslash S]\cdot r(a,X) + \sum_{X\in\mathcal{X}^{\emptyset}_{E\backslash S}} -w(u,v)\cdot p(V\backslash v, X, G_S)\cdot r(a,X) + 0$$

Using Proposition 3, for each $X \in \mathcal{X}^{\emptyset}_{E\backslash S}$ the corresponding live-edge graph in $\in \mathcal{X}^{e}_{E\backslash S}$ is $X_{(u,v)} = (V, E_X \cup (u,v))$ and it has probability $\Pr[X_{(u,v)}|E\backslash S] = w(u,v)\cdot p(V\backslash v, X_{(u,v)}, G_S)$. Hence:

$$f_a(S) - f_a(S\cup e) = \sum_{X\in\mathcal{X}^{\emptyset}_{E\backslash S}} \Pr[X_{(u,v)}|E\backslash S]\cdot \big(r(a, X_{(u,v)}) - r(a,X)\big) \tag{2}$$

Since the live-edge graph $X_{(u,v)}$ has one more edge than X, clearly $r(a, X_{(u,v)}) - r(a,X) \geq 0$, which completes the proof. ■

Given an influence graph $G = (V, E, w)$, $S \subset E$ and $e = (u,v), g = (u',v') \in E\backslash S$, we will establish the supermodularity of f_a, by showing that $f_a(S) - f_a(S\cup e) \geq f_a(S\cup g) - f_a(S\cup g\cup e)$. Let $T = S \cup g$. From Eqn. 2 in the proof of Thm. 4, we know that we need to take into account live-edge graphs in $\mathcal{X}^{\emptyset}_{E\backslash T}$ and in $\mathcal{X}^{\emptyset}_{E\backslash S}$.

Proposition 5 *There exists a family of sets $P = \{\Phi_i : \Phi_i \subseteq \mathcal{X}^{\emptyset}_{E\backslash S}\}_{i=1}^{t}$ that partitions $\mathcal{X}^{\emptyset}_{E\backslash S}$ into t disjoint subsets, where $t = |\mathcal{X}^{\emptyset}_{E\backslash T}|$.*

Proof Since $S \subset T$, every live-edge graph $X_i \in \mathcal{X}^{\emptyset}_{E\backslash T}$ is also in $\mathcal{X}^{\emptyset}_{E\backslash S}$. For each X_i, we create a corresponding $\Phi_i \subset \mathcal{X}^{\emptyset}_{E\backslash S}$ in the following manner. Recall that $g = (u';v')$. If node v' has a parent in X_i, then $\Phi_i = \{X_i\}$. Otherwise if v' has no parent in X_i, then $\Phi_i = \{X_i, X'_i\}$, where $X'_i = (V_i, E_i \cup g)$. X'_i is a valid live-edge graph in $\mathcal{X}^{\emptyset}_{E\backslash S}$ since v' had no parent in X_i and $g \in S$.

It is easy to see that sets Φ_i are pairwise disjoint, since each set contains a distinct X_i and all X'_i are obtained by extending the dinstinct X_i by $g \notin T$.

We show that $\cup_{i=1}^{t}\Phi_i = \mathcal{X}^{\emptyset}_{E\backslash S}$ by contradiction. Let us assume $\exists H = (V_H, E_H) \in \mathcal{X}^{\emptyset}_{E\backslash S}$such that $H \notin \Phi_i, \forall i = 1,..,t$. If H does not contain g then all edges in H are in $S\backslash g = T$, and it is easy to see that $\exists X_i \in \mathcal{X}^{\emptyset}_{E\backslash T}$ such that $X_i = H$, hence $H \in \Phi_i$. Otherwise, if H contains g, then the graph $H'' = (V_H, E_H\backslash g)$ is a valid live-edge graph where v' has no parent. Then similarly, $\exists X_i \in \mathcal{X}^{\emptyset}_{E\backslash T}$ such that $X_i = H''$. Since H'' does not contain a live edge for v', then $\Phi = \{X_i = H'', X'_i\}$, where $X'_i = (V_{H''}, E_{H''} \cup g) = H$. Hence, we have a contradiction in both cases. ■

Proposition 6 *For all $X_i \in \mathcal{X}^{\emptyset}_{E\backslash T}$ and the corresponding $\Phi_i \subset \mathcal{X}^{\emptyset}_{E\backslash S}$, $\Pr[X_i|E\backslash T] = \sum_{Y\in\Phi_i} \Pr[Y|E\backslash S]$.*

Proof Recall that $T\backslash S = g = (u',v')$. The statement holds true trivially in the case when v' has a parent in X_i and hence $\Phi_i = \{X_i\}$.

When v' has no parent in X_i, $\Phi_i = \{X_i, X'_i\}$, where $X'_i = (V_i, E_i\cup g)$. We consider the contribution of the node v' to the probability of the relevant live-edge graphs, since all other nodes contribute the same amount in across all cases considered:

$$\begin{aligned} p(v', X_i, E\backslash T) &= 1 - \sum_{(x,v')\in E\backslash T} w(x, v') \\ p(v', X_i, E\backslash S) &= 1 - \sum_{(x,v')\in E\backslash S} w(x, v') = 1 - \sum_{(x,v')\in E\backslash T} w(x, v') - w(u', v') \\ p(v', X_i', E\backslash S) &= w(u', v') \end{aligned}$$

$$\implies p(v', X_i, E\backslash T) = p(v', X_i, E\backslash S) + p(v', X_i', E\backslash S)$$

$$\implies Pr[X_i|E\backslash T] = Pr[X_i|E\backslash S] + Pr[X_i'|E\backslash S]$$

■

Theorem 7 *The function f_a is supermodular.*

Proof Given an influence graph $G = (V, E, w)$, $S \subset E$ and $e = (u, v), g = (u', v') \in E\backslash S$, we will establish the supermodularity of f_a, by showing that $(f_a(S) - f_a(S \cup e)) \geq (f_a(T) - f_a(T \cup e))$ where $T = S \cup g$. From Eqn. 2 in the proof of Thm. 4, we know that:

$$f_a(S) - f_a(S \cup e) = \sum_{X \in \mathcal{X}^{\emptyset}_{E\backslash S}} \mathrm{Pr}[X_{(u,v)}|E\backslash S] \cdot \big(r(a, X_{(u,v)}) - r(a, X)\big)$$

$$f_a(T) - f_a(T \cup e) = \sum_{X \in \mathcal{X}^{\emptyset}_{E\backslash T}} \mathrm{Pr}[X_{(u,v)}|E\backslash T] \cdot \big(r(a, X_{(u,v)}) - r(a, X)\big) \, .$$

For $t = |\mathcal{X}^{\emptyset}_{E\backslash T}|$, using Prop. 5 we can write:

$$f_a(S) - f_a(S \cup e) = \sum_{i=1}^{t} \sum_{X \in \Phi_i} \mathrm{Pr}[X|E\backslash S] \cdot \big(r(a, X_{(u,v)}) - r(a, X)\big) \tag{3}$$

Then we need only compare $f_a(S) - f_a(S \cup e)$ and $f_a(T) - f_a(T \cup e)$ component-wise for each $X_i \in \mathcal{X}^{\emptyset}_{E\backslash T}, i = 1, \ldots, t$. Clearly, when $\Phi_i = \{X_i\}$, the two are equal. When $\Phi_i = \{X_i, X_i'\}$, we need to show that:

$$\mathrm{Pr}[X_i|E\backslash S] \cdot \big(r(a, X_{i,(u,v)}) - r(a, X_i)\big) + \mathrm{Pr}[X_i'|E\backslash S] \cdot \Big(r(a, X'_{i,(u,v)}) - r(a, X_i')\Big)$$

$$\geq \mathrm{Pr}[X_i|E\backslash T] \cdot \big(r(a, X_{i,(u,v)}) - r(a, X_i)\big)$$

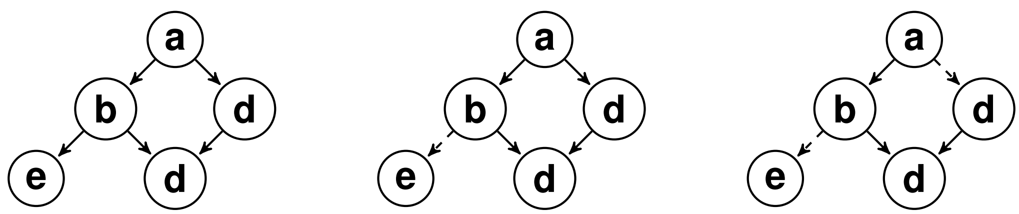
Based on Prop. 6 we know that $Pr[X_i|E\backslash T] = Pr[X_i|E\backslash S] + Pr[X_i'|E\backslash S]$. Then to establish the above inequality, it suffices to show that $r(a, X'_{i,(u,v)}) - r(a, X_i') \geq r(a, X_{i,(u,v)}) - r(a, X_i)$. Recall that $X_i' = (V_i, E_i \cup g)$. Since live-edge graphs are constructed in a way that each node has at most one incoming edge, each reachable node x has a unique path from the source a to x. Also a reachability path in $X_{i,(u,v)}$ is clearly also present in $X'_{i,(u,v)}$. Therefore if removing $e = (u; v)$ from $X_{i,(u,v)}$ results in unreachability of some nodes in X_i then those same nodes become unreachable when removing $e = (u; v)$ from $X'_{i,(u,v)}$. In addition, removing $e = (u; v)$ from $X'_{i,(u,v)}$ might disconnect some additional nodes whose path from the source a includes g. Hence, the reduction in reachable from nodes when removing $e = (u; v)$ from $X'_{i,(u,v)}$ is same or larger than the reduction when removing $e = (u; v)$ from $X_{i,(u,v)}$. This completes the proof. ■

Greedy approximation algorithm

Since the susceptibilty of a graph is a linear combination of $f_a, \forall a \in V$, then it is supermodular itself. In fact, it is established that *minimizing a supermodular function f* is equivalent to *maximizing the submodular function* $-f$. The classical result by Nemhauser et. al. [17] shows that this type of optimization problem can be approximated to a constant factor of $(1 - 1/e)$ using a simple greedy approach on the input set. In our case, the input set is the set of all edges in a graph G, E. Starting with an empty set of edges S_0, at each iteration i of the greedy algorithm, we add to our result set the edge e maximizing the *marginal gain* $\Delta(e|S_{i-1}) = \sigma(G_{S_{i-1}}) - \sigma(G_{S_{i-1} \cup e})$, where S_i is the result set up till the i-th iteration. The greedy CUTTINGEDGE algorithm (Alg. 1) runs in k steps, where k is the budget.

Input: $G(V, E), k$
Output: E^*
for *i=1 to k* **do**
  E^*=$E^* \cup$ $\mathrm{argmax}_{e \in E \setminus E^*} \Delta(e|S_{i-1})$
end

Algorithm 1: CUTTINGEDGE



(a) Original graph (b) Graph with S removed (c) Graph with T removed

Figure 2: Example graph where the IC model is not supermodular.

Unfortunately, this positive algorithmic result under the Linear Threshold model does not carry over to the Independent Cascade model.

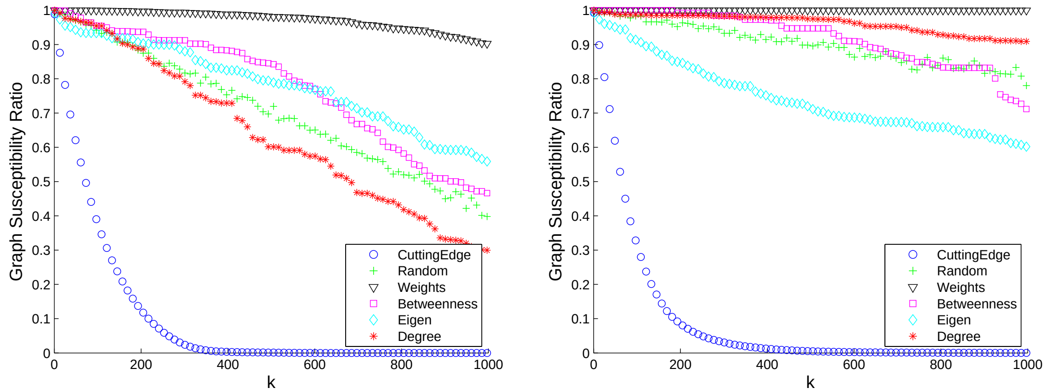
Theorem 8 *The function f_a is not supermodular under the Independent Cascade model.*

Proof We give a counter-example to prove the above. Consider the graph illustrated in Fig. 2(a) as our original influence graph $G = (V, E, w)$ with all weights equal to 1. Hence, in this trivial setting there is always only one possible cascade. Let $S = \{(b; e)\}, T = S \cup (a; d)$, and $e = (a; b)$. The resulting graphs after removing S and T are illustrated in Fig. 2 (b) and (c) respectively. The influence of node a after removing S is 3, and adding e to S results in influence of 2. Hence the marginal gain of adding e to S is 1. The influence of node a after removing T is 2, and adding e to T reduces the influence to 0, with a marginal gain of 2. Hence adding e to the smaller set S results in a larger marginal gain, violating the supermodularity property. ■

4 Experiments and Results

We evaluate the effectiveness of our algorithm CUTTINGEDGE on both synthetic and real-world networks, and compare the quality of the solution it provides against other heuristic algorithms.

Since evaluating the true expected susceptability of a graph has been shown to be $\#P$-hard problem [4], we use the usual Monte Carlo based approach and approximate it by the average susceptability over a large sample of live-edge graphs. Using the greedy approach in a naive way will result in evaluating marginal gain for each candidate edge at every iteration. Instead, we use a technique called *lazy evaluation* [16], which avoids computing the function for all edges and has been shown to result in significant speed-ups over the naive evaluation.



(a) ForestFire, 500 nodes, 1691 edges

(b) MemeTracker, 861 nodes, 5000 edges

Figure 3: Comparison of methods for (a) ForestFire, (b) MemeTracker. Lower is better.

Synthetic networks *Forest fire* is a generation model that produces networks mimicking the structure of growing networks [1]. The model produces realistic networks in terms of heavy-tailed degree distributions, community structure and other network properties. We generate a 500-node 1691-edge ForestFire network using parameters: forward burning probability 0.3, and backward burning probability 0.25. As for the diffusion probabilities on the edges, each is chosen uniformly at random, subject to the condition $\sum_{u \in V} w_{uv} \leq 1$ for any node v in the network. We use 5000 live-edge graph samples to estimate influences for this network.

Real-world networks In order to evaluate our method in a relvant application context, we consider the publicly available MemeTracker network [15]. This is a who-copies-from-whom dataset, where each node u is a news media site or blog, and each edge $e(u, v)$ represents the recorded event of v copying u. These edges are inferred from actual hyperlink cascade traces using a network inference algorithm, NetInf [9]. To assign probabilities on the edges, we make use of the median transmission time, also provided as part of the dataset. Let $\tilde{t}_{u,v}$ be the median transmission time between two nodes u and v, then we set $w_{u,v} =\propto \dfrac{1}{\tilde{t}_{u,v}}$, rewarding smaller transmission times with higher diffusion probabilities, and vice versa. We assign a probability of 0.2 to the event that a node v does not adopt despite its in-neighbors influence, such that $\sum_{u \in V} w_{u,v} + 0.2 = 1$. We also use 5000 live-edge graph samples to estimate influences in this case.

Heuristics To evaluate the quality of the solution provided by CuttingEdge, we compare it against other heuristic metrics that are based on the structure of the network irrespective of the dynamics entailed by the diffusion model. These heuristic strategies can be described as follows: (1) select k edges uniformly at random (referred to as 'Random'), (2) select the k edges that cause the maximum decrease in the leading eigenvalue of the network when removed from it (referred to as Eigen') [21, 20], (3) select the k edges with highest *edge betweenness centrality*, where this measure is defined for edge e as the sum of the fraction of all-pairs shortest paths that pass through e (referred to as 'Betweenness') [3], (4) select the k edges whose destination nodes have the highest out-degree (referred to as 'Degree'), (5) select the k edges with highest diffusion probability (weight) $w_{u,v}$, where an edge goes from node u to v (referred to as 'Weights'). Note that all three methods 'Eigen','Betweenness' and 'Degree' are weighted, where the diffusion probability of each edge is also its weight in the corresponding adjacency matrix.

Results To evaluate the solution quality for CuttingEdge and the other heuristics, we compute the ratio of the graph susceptibility after the given edge set is removed from the graph, to the graph susceptibility when no edge is removed. Clearly, the smaller the ratio is, the more effective a method is. For the ForestFire network, CuttingEdge completely mitigates diffusion after almost 400 edges (23% out of 1691 edges) are removed from the network. This can be explained by the fact that even for a large number of sample cascades, the number of edges that are actually live across all cascades is less than the number of edges in the network. While the heuristics perform poorly relative to our method, 'Degree' fares best amongst those, surpassing both supposedly "smarter" methods 'Betweenness' and 'Eigen'. Perhaps surprisingly, 'Random' also provides a better solution than most heuristics, despite its results being averaged over multiple random edge sets for each k. As seen in the figure, 'Random' does not decrease strictly monotonically due to the repeated random choosing of edges, as opposed to the other methods which are all incremental (i.e the set of edges chosen for a given k includes all edges chosen for $k-1$). Removing the edges with highest diffusion probability as in 'Weights' barely decreases the susceptibility even for a large k.

As for the MEMETRACKER dataset results, CUTTINGEDGE is able to fully mitigate diffusion by removing only around 500 edges, or 10% of the network's 5000 edges. Already for $k = 100$, the gap in the susceptibility ratio between CUTTINGEDGE and the other heuristics is significant, around 0.4 as compared to 0.95 for the most competitive heuristic here, 'Eigen'.

5 Conclusion

We have presented an optimization formulation for the problem of edge-based influence minimization in networks. Under the linear threshold model, we prove that the objective function is supermodular, allowing for an approximation algorithm that yields solutions with strong guarantees. Our first experimental results on both synthetic and real-world networks demonstrate our method's effectiveness in mitigating diffusion processes, one that is unmatched by other commonly used eigenvalue and centrality-based heuristics.

Possible areas of future work include more extensive experimentation on a wider array of datasets. Also, we intend to research algorithmic methods to make our method scalable to networks with millions of nodes and edges. More broadly, the class of problems at the intersection of network manipulation and diffusion processes remains very challenging and interesting.

Acknowledgements. We thank Prof. Hanghang Tong for providing us with his code for the 'Eigen' method.

References

[1] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[2] Ilija Bogunovic. *Robust Protection of Networks against Cascading Phenomena*. PhD thesis, Master Thesis ETH Zürich, 2012, 2012.

[3] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[4] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD: ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 1029–1038, 2010.

[5] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM: IEEE International Conference on Data Mining*, pages 88–97, 2010.

[6] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *KDD: ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 57–66, 2001.

[7] Nan Du, Le Song, Hongyuhan Zha, and Manuel Gomez Rodriguez. Scalable influence estimation in continuous time diffusion networks. In *NIPS: Advances in Neural Information Processing Systems*, page To Appear, 2013.

[8] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *PNAS: Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[9] Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *KDD: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1019–1028, 2010.

[10] Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, pages 1420–1443, 1978.

[11] Herbert W Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000.

[12] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[13] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(2):9, 2009.

[14] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3, 2012.

[15] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD: ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 497–506, 2009.

[16] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *KDD: ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 420–429, 2007.

[17] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

[18] Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML: International Conference on Machine Learning*, pages 561–568, 2011.

[19] Daniel Sheldon, Bistra Dilkina, Adam N Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla P Gomes, David Shmoys, William Allen, Ole Amundsen, et al. Maximizing the spread of cascades using network design. In *UAI: Conference in Uncertainty in Artificial Intelligence*, pages 517–526, 2010.

[20] Hanghang Tong, B Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, and Christos Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *CIKM: ACM International Conference on Information and Knowledge Management*, pages 245–254, 2012.

[21] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *IEEE International Symposium on Reliable Distributed Systems*, pages 25–34, 2003.

Appendix

We show that the supermodularity we proved for deleting edges holds as well for the three other possible network manipulation operations: adding edges, deleting nodes and adding nodes.

Given an influence graph $G'(V, E', w')$ and the *complete* influence graph $G = (V, E, w)$ (where every two nodes are connected in both directions), such that $E' \subseteq E, w' \subseteq w$, we would like to add k edges from $E \backslash E'$ such that G''s susceptibility to diffusion is maximized. Our optimization problem is the following:

$$S^* = \arg \max_{S \subseteq E \backslash E' : |S| = k} \sum_{a \in V} \sigma(a, G_S),$$

where $G_S = (V, E' \cup S, w' \cup w_S)$.

We will show that $g_a(S) = \sigma(a, G_S)$ is a monotone and supermodular function. From our earlier definition:

$$g_a(S) = \sigma(a, G_S) = \sum_{X \in \mathcal{X}_{E \cup S}} \Pr[X | E \cup S] \cdot r(a, X).$$

Theorem 9 *The function g_a is supermodular.*

Proof Now consider the graph $G = (V, E, w)$ and $G'(V, E', w'), E' \subseteq E$. Let $S \subseteq T \subseteq E \backslash E'$, and $e \in E \backslash (E' \cup T)$.

Then, adding edges S to G' results in the same expected influence as removing edges $E \backslash (S \cup E')$ from G. Namely, if $A = E \backslash (S \cup (E' \cup e))$, then $g_a(S) = f_a(A \cup e)$. Also, $g_a(S \cup e) = f_a(A)$. Analogously, if $B = E \backslash (T \cup (E' \cup e))$, then $g_a(T) = f_a(B \cup e)$. Also, $g_a(T \cup e) = f_a(B)$. Note that $B \subseteq A$. But we know from 7 that:

$$f_a(B) - f_a(B \cup e) \geq f_a(A) - f_a(A \cup e),$$

which implies that:

$$\begin{aligned} g_a(T \cup e) - g_a(T) &\geq g_a(S \cup e) - g_a(S), \\ g_a(S) - g_a(S \cup e) &\geq g_a(T) - g_a(T \cup e). \end{aligned}$$

Since $S \subseteq T$, then g_a is supermodular, completing the proof. ∎

Now define $E_S \subseteq E$, for any set of nodes $S \subseteq V$, as the set of edges having as source or target a node in S. Also, for any node $v \in V \notin S$, define $E_v^S = (v, u) \in E | u \notin S$.

Then, let the function describing the graph susceptibility in the event of node deletion be defined as:

$$h_a(S) = \sum_{X \in \mathcal{X}_{E \backslash E_S}} \Pr[X | E_S] \cdot r(a, X).$$

Theorem 10 *The function h_a is supermodular.*

Proof Let $G = (V, E, w)$ be the complete influence graph, and $B = A \cup u$, where $A, B, u \in V$. Also let $v \in V \backslash B$ and $E_v^B = \{e_1, e_2, ..., e_k\}$. From 7, we can write:

$$\begin{aligned} f_a(E_A) - f_a(E_A \cup e_1) &\geq f_a(E_B) - f_a(E_B \cup e_1) \\ f_a(E_A \cup e_1) - f_a(E_A \cup e_1 \cup e_2) &\geq f_a(E_B \cup e_1) - f_a(E_B \cup e_1 \cup e_2) \\ &\;\;... \\ f_a(E_A \cup e_1 \cup ... \cup e_{k-1}) - f_a(E_A \cup e_1 \cup ... \cup e_{k-1} \cup e_k) &\geq f_a(E_B \cup e_1 \cup ... \cup e_{k-1}) - f_a(E_B \cup e_1 ... \cup e_{k-1} \cup e_k) \end{aligned}$$

Adding all these equations together, we obtain:

$$f_a(E_A) - f_a(E_A \cup E_v^B) \geq f_a(E_B) - f_a(E_B \cup E_v^B)$$

If the edges (u, v) and (v, u) are both not in E, then $E_v^A = E_v^B$, and the proof is complete. Even if either one or both of these two edges appear in E, h_a is still supermodular. We show that for the case where $(u, v), (v, u) \in E$.

In that case, $E_v^A = E_v^B \cup (u, v) \cup (v, u)$, and f_a is monotone decreasing, implying that $f_a(E_A \cup E_v^A) \leq f_a(E_A \cup E_v^B)$, and consequently $f_a(E_A) - f_a(E_A \cup E_v^A) \geq f_a(E_A) - f_a(E_A \cup E_v^B)$. Since $f_a(E_A) = h_a(A)$, $f_a(E_A \cup E_v^A) = h_a(A \cup v)$ (and the same for B instead of A), we finally get:

$$h_a(A) - h_a(A \cup v) \geq h_a(B) - h_a(B \cup v)$$

which comletes this proof.

Adding nodes is also supermodular by a similar proof based on g_a instead of f_a. ■