Limiting the Spread of Misinformation in Social Networks

Ceren Budak Divyakant Agrawal Amr El Abbadi
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5110, USA
{cbudak, agrawal, amr}@cs.ucsb.edu

ABSTRACT

In this work, we study the notion of competing campaigns in a social network and address the problem of *influence limitation* where a "bad" campaign starts propagating from a certain node in the network and use the notion of limiting campaigns to counteract the effect of misinformation. The problem can be summarized as identifying a subset of individuals that need to be convinced to adopt the competing (or "good") campaign so as to minimize the number of people that adopt the "bad" campaign at the end of both propagation processes. We show that this optimization problem is NP-hard and provide approximation guarantees for a greedy solution for various definitions of this problem by proving that they are submodular. We experimentally compare the performance of the greedy method to various heuristics. The experiments reveal that in most cases inexpensive heuristics such as degree centrality compare well with the greedy approach. We also study the influence limitation problem in the presence of missing data where the current states of nodes in the network are only known with a certain probability and show that prediction in this setting is a supermodular problem. We propose a prediction algorithm that is based on generating random spanning trees and evaluate the performance of this approach. The experiments reveal that using the prediction algorithm, we are able to tolerate about 90% missing data before the performance of the algorithm starts degrading and even with large amounts of missing data the performance degrades only to 75% of the performance that would be achieved with complete data.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Theory

Keywords

social networks, information cascades, misinformation, competing campaigns, submodular functions, supermodular functions

1. INTRODUCTION

Online social networks have many benefits as a medium for fast, widespread information dissemination. They provide fast access to large scale news data, sometimes even before the mass media as in the case of the announcement of death of Michael Jackson [34]. They also serve as a medium to collectively achieve a social goal. For instance with the use of group and event pages in Facebook, events such as "Day of Action" protests reached thousands of protestors [16]. While the ease of information propagation in social networks can be very beneficial, it can also have disruptive effects. One such example was observed during the recent shootings at Fort Hood, Texas, when a soldier inside the base sent out messages via Twitter as the event unfolded. Her incorrect reports of multiple shooters and shooting locations quickly spread through the social network and even to the mass media where it was reported on television broadcasts [22]. Another example is the spread of misinformation on swine flu in Twitter [35]. The spread of misinformation in this case reached a very large scale causing panic in the population. Although social networks are the main source of news for many people today, they are not considered reliable due to such problems.

Clearly, in order for social networks to serve as a reliable platform for disseminating critical information, it is necessary to have tools to limit the effect of misinformation. In the presence of a misinformation cascade, we aim to find a near-optimal way of disseminating "good information" that will minimize the devastating effects of a misinformation campaign. For instance in the case of [35, 22], we seek ways of making sure that most of the users of the social network hear about the correct information before the bad one, making social networks a more "trustworthy" or "reliable" source of information. In addition to the implication our work has in limiting the effect of misinformation, our methods can be applied to any two simultaneously spreading competing campaigns.

In this work, we study the problem of minimizing the number of people that adopt the misinformation and prove that even though the general problem does not exhibit the submodular property, certain restricted versions of it are in fact submodular. We exploit this property to provide efficient solutions with approximation bounds. We also evaluate the performance of our algorithm on a number of close-knit regional networks obtained from the Facebook social network comparing its performance with some well-known heuristics including degree centrality. We show that in many cases, heuristics have performance comparable to the more computationally intense greedy method. Since in the real world, decisions about how to deploy a limiting campaign need to be made with incomplete data, we also consider the case where the states of only a fraction of the nodes in network can be observed. We show that, although the naive solution to the optimization problem in this setting is intractable, using matrix tree theorem [27] and the fact that the specific problem is supermodular [39], a polynomial time solution can be used where *polynomial time* is defined in terms of calls to an *oracle function*. However, this solution is still expensive

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

for large scale social networks, so we propose a prediction method that is based on generating random spanning trees on a set of *likely to have been infected nodes* to predict the missing information. We show that in most cases, this method has good performance, i.e. decisions made as to who to first influence by the limiting campaign under uncertain data still result in effective inoculation.

We start with a brief overview of information propagation in social networks in Section 2. In Section 3, we introduce our model of communication and formalize the influence limitation problem. In Section 4 we prove NP-hardness and submodularity of influence limitation. Submodularity guarantees approximation bounds for a greedy algorithm presented in Section 4.3. In Section 5, we provide the experiments that compare the performance of the greedy solution with various heuristics. In Section 6, we explore the problem in the presence of missing information and propose an algorithm for predicting the missing data and limiting the spread of misinformation in that setting. Finally, Section 7 concludes the paper.

2. RELATED WORK

The identification of *influential users* in a social network is a problem that has received significant attention in recent research. For the *influence maximization problem*, given a probabilistic model of information diffusion such as the Independent Cascade Model, a network graph, and a budget k, the objective is to select a set A of size k for initial activation so that the expected value of $f(A)$ (size of cascade created by selecting set A) is maximized [12, 37]. Early works relied on heuristics such as node degree and distance centrality [42] to select the set A. Although the problem of finding an optimal solution in this model is NP-hard, there is a greedy algorithm that yields a spread that is within $1 - 1/e$ of optimal [24]. This solution depends on Monte Carlo simulations which are computationally expensive. Work has been done to improve the performance of this greedy algorithm [8, 31, 26, 9], but scalability remains a significant challenge. In addition to the scale issues, these definitions of influential users ignore certain aspects of real social networks such as the existence of competing campaigns. In this work we consider different models of communication that incorporate different aspects of real social networks. Similar to [24, 31], we identify a problem that involves detecting "influential nodes" and study the feasibility of a solution to this problem. However, our problem formulation is more general since we model the existence of competing cascades dissipating in a network.

The existence of competing campaigns has been captured by a number of studies recently. Dubey et al. [13] study the problem as a network game focusing on quasi-linear model and consider various cost, benefit and externality functions for competing firms. They study the existence of Nash Equilibrium (NE) and show that it is unique if there is enough competition between firms or if their valuations of clients are anonymous. Bharathi et al. [4] augment the Independent Cascade Model to capture the existence of competing campaigns in a network. Their diffusion model is similar to ours and captures the timing issues that are crucial in competing campaigns optimization problems. The algorithmic problem defined in [4] is: when there is more than one campaign dissipating in a network and each campaign can select a set of early adopters so as to maximize their benefit, what is the best strategy for the players? This work studies the problem from both the first and last player's perspectives and shows that the problem of selecting the early adopters for the last player is submodular. They also introduce a fully polynomial time approximation scheme for the first player when the network structure is a tree. Carnes et al. [7] consider the same problem from the last player's perspective and use a diffusion model where nodes of the network choose the campaign to adopt w.r.t. their distance to the early adopters of the campaigns and another model where the nodes make a uniform random choice among its active neighbors. The experimental results show that the greedy approach performs better than the heuristics. They also experimentally show that the best strategy for the first player is to choose high degree nodes. Kostka et al. [29] study competing campaigns as a game theoretical problem and show that being the first player, i.e. the first to decide, is not always advantageous. Both [7, 4] use diffusion models where the two campaigns propagate exactly the same way, i.e. the probability of diffusion on a certain edge is the same for both campaigns and both start at the same time. In our work, we study the case where the competing campaigns have different acceptance rates and one is in response to the other, and therefore the campaign of the last player is started with a delay. Also, different from previous work we address the problem of influence *limitation* as opposed to *maximization*.

The problem of limiting the effect of misinformation in a social network can be seen as similar to the problem of epidemics and inoculation. There are many studies on the spread of infections and immunization [41, 3, 25]. A recent work on identifying influential people in a social network [28] uses SIS (susceptible-infected-susceptible), SIR (susceptible-infected-recovered) models [2, 11, 21] and concludes that the influence of a node is more dependent on its location in the network than the number of connections it has. This work captures the notion of being "immunized" but the immunization is limited to the node that is inoculated by external means. Conversely, we consider the case where once a node is inoculated, it can inoculate more people (by virally spreading the "good" information). Inoculation has also been studied in the game theory literature. Meier et al. [33] study inoculation games in social networks. The problem is posed in terms of virus propagation where the owner of each node decides whether or not to protect itself. Here inoculation has a direct effect only on the inoculated node, meaning that the "good" information does not propagate. The decision to "protect" oneself is a distributed process, each node decides for itself and aims to maximize its own function whereas we consider the problem of finding the best solution for the community.

3. DIFFUSION OF MISINFORMATION

A social network can be modeled as a directed graph $G = (N, E)$ consisting of nodes *N* and edges *E*. In the context of influence spread, *N* can be viewed as the users of the social network. A node *w* is a *neighbor* of a node *v* if and only if there is $e_{v,w} \in E$, an edge from *v* to *w* in *G*. In addition to this, $p_{v,w}$ is assigned to each edge $e_{v,w}$ which is used to model the direct influence v has on w.

3.1 Diffusion Models

Independent cascade model (ICM) is one of the most basic and well-studied diffusion models that has been used in different contexts [14, 32, 17, 19]. In the *ICM*, a process starts with an initial set of active nodes A_0, and unfolds in discrete steps. When node *v* first becomes active in step *t*, it has a single chance to activate each currently inactive neighbor *w*; it succeeds with probability $p_{v,w}$. If *v* succeeds, then *w* will become active in step *t + 1*; but whether or not v succeeds, it cannot make any further attempts in subsequent rounds. The process runs until no more activations are possible. If w has incoming edges from multiple newly activated nodes, their attempts are sequenced in an arbitrary order.

We now introduce the *Multi-Campaign Independent Cascade Model (MCICM)* which models the diffusion of two cascades evolving simultaneously in a network. Let C (for "campaign") and L (for "limiting campaign") denote the two cascades. The initial set of active nodes for cascade L (C) is denoted by A_L (A_C). When a node

v first becomes active in campaign L (or C) in step *t*, it has a single chance to activate each currently inactive neighbor *w* in campaign L (or C) and it succeeds with probability $p_{L,v,w}$ (or $p_{C,v,w}$) given that no neighbor of *w* tries activating *w* in the competing campaign at the same step. We also refer to $p_{L,v,w}$ (or $p_{C,v,w}$) as the probability of the edge $e_{v,w}$ being *live*. If there are two or more nodes trying to activate *w* at a given time step, at most one of them can succeed. In independent cascade, when *w* has several newly activated neighbors, their attempts are sequenced in arbitrary order. However in our studies, we will assume that there is a natural order to the two campaigns, more specifically one is "good" while the other is the "bad" campaign and if the "bad information" and the "good information" reach a node *w* at the same step, "good information" takes effect. Once a node becomes active in one campaign, it never becomes inactive or changes campaigns and the process continues until there is no newly activated node in either campaign.

We also consider another model of diffusion in which the probabilities of each edge being *live* is independent of the campaign. In this setting we only associate one probability $p_{v,w}$ with each edge $e_{v,w}$. No matter which information reaches a node v, v forwards this information to its neighbor w with probability $p_{v,w}$. Although this model is not a perfect fit for the inoculation of misinformation, it is a good fit for modeling competing campaigns where the two information cascades are more likely to be of similar "quality" and the nodes would agree to the campaign that reaches out to them first. Consider for example two news articles L and C about the same event spreading in a social network. The probability of a user forwarding article L and C is more dependent on the news itself rather than which agency the news is from. Similar to the *Multi-Campaign Independent Cascade* model, there are three states a node can be in; *inactive, in campaign L, in campaign C* and once a node becomes active in either L or C, it cannot change its state. As before, we assume that in the case of simultaneous trials of activation at a node, campaign L is ordered before C. We call this model *Campaign-Oblivious Independent Cascade (COICM)*. *COICM* is similar to the diffusion model used in [4]. However we assume that one of the campaigns is prioritized over the other in the case of simultaneous activation trials whereas independent and exponentially distributed continuous random variables are assigned to each edge as delay in [4] to ensure there are no simultaneous activation trials. The earlier studies using similar diffusion models support the validity of *MCICM* and *COICM*. However whether such models reflect the real influence spread in social networks is still an open problem. In future work, we plan to investigate this problem by studying the behavior in real social networks.

3.2 Problem Definition

While a substantial amount of research has been done in the context of influence maximization, a problem that has not received much attention is limiting the influence of a misinformation campaign. One strategy to deal with a misinformation campaign is to limit the number of users who are willing to accept and spread it. We will assume the *Multi-Campaign Independent Cascade Model* described in Section 3.1 as the model of communication. W.l.o.g. we will assume that the spread of influence for campaign C starts from one node n_a and is detected with delay r and at that point campaign L is initiated. However the algorithms can be easily extended to the case where C starts from a set of nodes and the proofs of NP-hardness and submodularity still hold for this case.

Depending on the context that the influence limitation problem is introduced in, we need to consider different objective functions. The objective can be to try and "save" as many nodes as possible, to limit the lifespan of the "bad" information campaign or to maximize the effect of the "good" campaign in the presence of the "bad" campaign. In this paper, we will focus on minimizing the number of nodes that end up adopting campaign C when the information cascades from both campaigns are over. We refer to this problem as the *eventual influence limitation problem (EIL)*.

4. EVENTUAL INFLUENCE LIMITATION

Given a network and the *Multi-Campaign Independent Cascade Model* defined in Section 3.1, suppose that a campaign C spreading bad information is detected with delay r. Given budget k, select A_L as *seeds* for initial activation with the limiting campaign L such that the expected number of nodes that adopt campaign C, $\sigma(A_C)$ is minimized. Let $IF(A_C)$ denote the influence set of C in the absence of L, *i.e* the set of nodes that would accept campaign C if there were no limiting campaign. We define the function $\pi(A_L)$ to be the size of the subset of $IF(A_C)$ that campaign L prevents from adopting campaign C. Then, the influence limitation problem is equivalent to selecting A_L such that the expectation of $\pi(A_L)$ is maximized. Note that we are not necessarily interested in the number of *inoculated* nodes but the *inoculated* nodes that would be *infected* otherwise. We will refer to this set of nodes as *saved*.

We now outline a solution to a simplified version of this problem where there is only a single source of information for C, meaning $|A_C| = 1$. We refer to this node as the *adversary node* or n_a As it may be much easier to convince a user of the truth than a falsehood, we also assume that the *limiting campaign information* is accepted by users with probability 1 ($p_{L,v,w} = 1$ if there is an edge from v to w and $p_{L,v,w} = 0$ otherwise). We refer to this notion as *high-effectiveness property*. Even with these restrictions, *EIL* is NP-hard and therefore finding the optimal solution is expensive. However as we will establish in Section 4.2, the problem is submodular which guarantees that we can provide approximation bounds with a simple hill climbing approach. Later we will investigate a more general form of this problem where we allow arbitrary values for $p_{L,v,w}$ and show that this problem is no longer submodular.

4.1 NP-Hardness of EIL

THEOREM 4.1. *EIL is NP-hard even with the high effectiveness property.*

PROOF. Consider an instance of the NP-complete Set Cover problem, defined by a collection of subsets $S_1, S_2, ..., S_m$ for a universe set $U = \{u_1, u_2, ..., u_n\}$; we wish to know whether there exist k of the subsets whose union is equal to U. We show that this can be viewed as a special case of *EIL*. Given an arbitrary instance of the Set Cover problem, we define a corresponding directed bipartite graph with $n+m+1$ nodes: there is a node *i* corresponding to each set S_i, a node *j* corresponding to each element u_j, and a directed edge (i, j) whenever $u_j \in S_i$. In addition, there is an adversary node a and a directed edge (a, j) for all u_j with activation probability $p_{a,j} = 1$. The Set Cover problem is equivalent to deciding if there is a set A_L of k nodes in this graph with $\pi(A_L) \geq n+k$ when we become aware of campaign C at time step 0 (when a itself is active in campaign C but has not contacted any of its neighbors yet). Note that for the instance we have defined, activation is a deterministic process, as all probabilities for adversary to infect its neighbors are 0 or 1. Initially activating the k nodes corresponding to sets in a Set Cover solution results in saving all n nodes corresponding to the ground set U, and if any set A_L of k nodes has $\pi(A_L) \geq n+k$, then the Set Cover problem must be solvable. □

4.2 Submodularity of EIL

A function $f(.)$ is said to be submodular or have "diminishing returns" if it satisfies the following property: $f(S \cup v) - f(S) \geq$

$f(T \cup v) - f(T)$, for all elements v and all pairs of sets $S \subset T$, i.e. the marginal gain from adding an element to a set S is at least as high as the marginal gain from adding the same element to a superset of S. As proved by Nemhauser, Wolsey, and Fisher [10, 36], for submodular and monotone functions, the greedy hill-climbing algorithm of starting with the empty set, and repeatedly adding an element that gives the maximum marginal gain approximates the optimum solution within a factor of $(1 - 1/e)$. Here we will prove that the influence limitation problem is submodular when the *limiting campaign* has the *high effectiveness property*. We omit the proofs of monotonicity of EIL due to space limitations but to give an intuition for monotonicity we note that having more nodes to initially activate in campaign L can never have a *negative effect* under the models of diffusion we study.

Since influence spread over G is a stochastic process, the influence function for a set of nodes is tricky to define. Following the same approach presented in [24], we view an event of a newly activated node v attempting to activate its neighbor w and succeeding with $p_{C,v,w}$ as flipping a coin with bias $p_{C,v,w}$. It does not matter whether the coin is flipped at the moment when v tries to activate w, or if it was pre-flipped and stored to be examined at the time when v tries to activate w. So while considering a specific instance of influence spread, we can pre-flip all the coins to determine which edges of the graph G are *live* (meaning if the start node of this edge were to be activated, it would succeed in activating its neighbor) or *blocked* (meaning the attempt would be unsuccessful). In this setting, the spread of "bad campaign" C can be modeled as graph $G' = (N', E')$. where N' represents the set of nodes that are reachable from adversary node n_a via live edges and E' represents the set of live edges amongst the nodes in N'.

Consider the graph of 10 nodes represented in Figure 1(a). Assume that by pre-flipping the coins, we end up with probabilities such that the solid lines are *live* edges and dotted lines are *blocked* edges. In this case a campaign starting from adversary node 0 would reach nodes $0, 1, 2, 3$ if there was no limiting campaign. A first look at this graph (or the general *EIL* in general) suggests that in order to save node 3, we need to make sure both 1 and 2 should be saved (or 3 should be saved directly). Superficially, it would seem that submodularity is no longer viable. Since saving only 1 or 2 would not be sufficient to save 3, but their combination would. However, a closer look at this problem reveals that we do not need to secure all the possible paths to a node from an adversary but just the shortest path. If L can reach 3 before C, 3 can never be infected. For instance, for the campaign in Figure 1(a), if campaign L reaches node 1 by $r = 1$, it will be saved. In this case the good campaign will reach node 3 at $r = 2$ and even if node 2 is not saved, that still guarantees that node 3 will be saved. Next we provide the formal proof of submodularity for *EIL*. Note that the proof depends on the *high-effectiveness property* of the good campaign. Later on, we will show that when this property does not hold, *EIL* is not, in general, a submodular function.

CLAIM 1. *In MCICM with the high effectiveness property a node w can be saved if and only if $\exists$ v such that $v \in A_L$ and $|SP_G(v, w)| + r \leq |SP_{G'}(n_a, w)|$ where $SP_G(v, w)$ denotes a shortest path from node v to w in graph G.*

PROOF. *1. If $\exists$ v such that $v \in A_L$ and $|SP_G(v, w)| + r \leq |SP_{G'}(n_a, w)|$, then w is saved:* Assume that such a v exists but w could not be saved. This is only possible if the bad campaign C reaches w strictly before L since otherwise w would be saved at $ts = |SP_G(v, w)| + r$. So there must exist a path $P_{G'}(n_a, w)$ from n_a to w such that $|P_{G'}(n_a, w)| < |SP_G(v, w)| + r$. Since $|SP_G(v, w)| + r \leq |SP_{G'}(n_a, w)|$, $|P_{G'}(n_a, w)| < |SP_{G'}(n_a, w)|$. This means there is a shorter path from n_a to w in G' than the shortest path which is a contradiction.

2: If $\nexists$ v such that $|SP_G(v, w)| + r \leq |SP_{G'}(n_a, w)|$, then w cannot be saved: Assume contrary, i.e. $\nexists$ v s.t. $SP_G(v, w) \leq SP_{G'}(n_a, w)$ and w is saved. If w is saved, at least one of the nodes in one of those shortest paths must have been activated in L since otherwise C would propagate on one of those paths to w and infect it. W.l.o.g. let a shortest path from n_a to w consist of nodes $n_a, n_1, n_2 ..., n_i, w$ and $n_j \in SP_{G'}(n_a, w)$ be a node activated in L, n_j can only be activated in L if L reaches n_j at $ts \leq j$ because $SP_{G'}(n_a, j) = n_a, n_1, n_2, ..., n_{j-1}, n_j$. Therefore $\exists$ $v \in A_L$ s.t. $|SP_G(v, n_j)| + r \leq |SP_{G'}(n_a, n_j)|$. Since $|SP_G(n_j, w)| \leq |SP_{G'}(n_j, w)|$, $|SP_G v, w| + r \leq |SP_G(v, n_j)| + SP_G(n_j, w)| + r \leq |SP_{G'}(n_a, n_j)| + |SP_{G'}(n_j, w)| \leq |SP_{G'}(n_a, w)|$. This contradicts with the initial assumption that $\nexists$ v s.t. $|SP_G(v, w)| + r \leq |SP_{G'}(n_a, w)|$. □

THEOREM 4.2. *EIL is submodular when the limiting campaign L has high-effectiveness property*

PROOF. Consider the *inoculation graph* $G'' = (N'', E'')$ s.t. $N'' = \{u | u \in N \wedge u \notin I\}$ and $E'' = \{(u, v) | v \in S_u\}$ where $S_u = \{v | v \in N' \wedge |SP_G(u, v)| + r \leq |SP_{G'}(n_a, v)\}|$ and I is the set of nodes that are infected by time step r. Based on Claim 1, the *EIL* when L has *high-effectiveness property* is equivalent to maximizing the number of nodes reachable from the set A_L in G'' and as established in [24], this is submodular. □

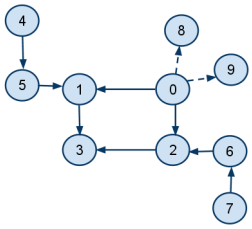
Unfortunately, the general *EIL* problem where L does not have the *high-effectiveness property* is not in general submodular. Consider the graphs in Figure 1. Assume an instance of *EIL* where G'' representing the spread of influence for the good campaign L consists of nodes 1,2,5,6 and the edges $e_{5,1}, e_{6,2}$. In this case, $f(5) = 1, f(6) = 2, f(5, 6) = 1, 2, 3$ since by using 5(6) as a seed, L can save node 1(2). (Since $e_{1,3}(e_{2,3})$ is not a live edge for L, the good campaign will never reach node 3, and 3 will be infected by node 2(1) at the next time step.) On the other hand if $A_L = \{5, 6\}$ both 2,3 will be saved and since these are the only two nodes that could infect 1, node 1 will also be saved. This example shows that *EIL* without the *high-effectiveness property* is not in general submodular.

Finally, consider the *Campaign-Oblivious Independent Cascade* introduced in Section 3.1 where the probabilities on the edges are campaign-independent. In this case we associate only one probability $p_{v,w}$ with each edge $e_{v,w}$. This model fits competing campaigns where the two campaigns are trying to get users to adopt very similar products or ideas. In this case users are as likely to adopt campaign L as they would adopt campaign C. Note that, this model does not rely on either one of the campaigns being good or bad and therefore can be applied to any two competing campaigns.

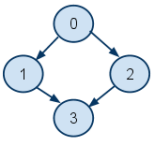
CLAIM 2. *EIL is submodular for Campaign-Oblivious Independent Cascade Model.*

PROOF. Since a node can only be activated in one campaign, an edge $e_{v,w}$ will only be visited at most once. Therefore, using the same idea presented in 4.2, we can pre-flip all the coins to determine which edges are *live* or *blocked* for an instance of influence dissemination from campaigns C and L. Consider the graph $G^{live} = (N, E')$ where E' is the set of *live* edges in E. Both L and C can be modeled as propagating on this graph. Let N' denote the nodes that are reachable from adversary n_a via *live* edges. In this case, influence limitation problem is equivalent to maximizing the number of nodes reachable from A_L in G'' where $G'' = (N'', E'')$ s.t. $N'' = \{u | u \in N \wedge u \notin I\}$ and $E'' = \{(u, v) | v \in S_u\}$ where

$S_u = \{v | v \in N' \wedge |SP_{G^{live}}(u,v)| + r \leq |SP_{G^{live}}(n_a,v)|\}$ and I is the set of nodes that are infected by time step r. Since reachability problem is submodular [24], so is *EIL* on *COICM*. □



(a) Spread of C. Solid lines are *live* and dotted lines are *blocked* edges. The adversary node 0, without an opposing campaign, reaches 1,2 and 3



(b) The shortest path structure for spread of influence for the bad campaign

Figure 1: General Influence Spread

4.3 Possible Solutions for EIL

Since *EIL* for $MCICM$ where the limiting campaign has *high effectiveness property* or for $COICM$ in general are submodular and monotone, the hill climbing approach provides a $(1-1/e)$ approximation [10, 36] for these problems. Figure 2 provides this greedy algorithm that yields an A_L for which $\pi(A_L)$ is within $1-1/e$ of optimal. The algorithm works for a given graph G, a set of adversaries S_a, *delay* r and budget k, i.e. number of nodes to initially activate in campaign L. According to our problem definition S_a consists of only one node n_a, however the proofs can easily be generalized to hold for multiple initial adversaries. Since independent cascade is a stochastic process, computing π for a given set of nodes requires running a large number of simulations ($\#_{sim}$) as demonstrated in steps (6,7). The procedure *InfLimit (G,S_a,r,S,v)* decides *liveness* of edges in G based on the probability associated with that edge and simulates the influence limitation given that the set of adversaries that C starts from is S_a, the adversary campaign is caught with delay *r*, the nodes we have already chosen to initially activate in campaign L is *S* and the node that we are evaluating the influence of is v. This method returns the marginal gain of v i.e. number of people v could save but the set S could not, where the nodes that could be *saved* by v are the nodes that have a shorter path from v than of any node in set S_a.

Considering the large scale of social networks today and the complexity of the *EIL*, even the greedy approach that is a polynomial time algorithm is too costly to be used in real social networks. Therefore, we seek alternatives that can potentially compare well with the greedy approach which, as we have proved, is guaranteed to be a good approximation. We consider three different heuristics. The first heuristic we consider is the *degree centrality* which has been used in early work to target "influential people" [42].

1: {Given (G,A_C,r,k) where $G = (N, E)$, A_C is the set of adversaries and r is the delay and k is the number of seeds}
2: Initialize A_L to $\emptyset$, R to $\#_{sim}$
3: **for** $i = 1$ to k **do**
4: **for** each vertex $v \in N - A_L$ **do**
5: $s_v = 0$
6: **for** j = 1 to R **do**
7: $s_v += InfLimit(G, S_a, r, A_L, v)$
8: $s_v = s_v/R$
9: $A_L = A_L \cup \{argmax_{v \in V - A_L}\{s_v\}\}$
10: Output A_L

Figure 2: Greedy algorithm to select the set for initial activation

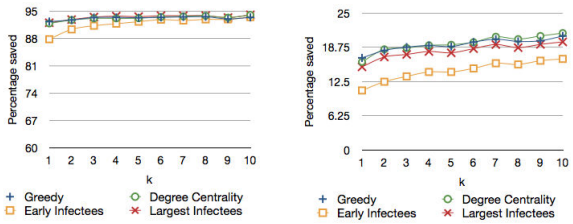
The second heuristic we consider is called *early infectees* and entails choosing *seeds* that are expected to be infected at time step r. This is equivalent to reaching out to nodes that would be infected early on but after L is started, since those nodes are likely to create a large cascade for campaign C. In order to calculate this heuristic, we run $\#_{sim}$ simulations of infection spread from S_a and select nodes A_L in decreasing order, where the nodes are ordered w.r.t. the number of simulations they were infected at time step r.

The third heuristic is *largest infectees*. This heuristic is very similar to the *early infectees* but rather than choosing the nodes that are expected to be infected early on, it chooses seeds that are expected to infect the highest number of nodes if they were to be infected themselves. In this case we only consider nodes that would be infected after time step r. In order to calculate this heuristic, we run $\#_{sim}$ simulations of infection spread from S_a and at each simulation we increase the value val_i of a node n_i that is infected after time step r by the number of nodes n_j s.t. n_i is on the shortest path from an adversary in S_a to n_j. We select nodes A_L in decreasing order of val_i. Note that both *early infectees* and *largest infectees* are computationally more intensive to compute than degree centrality. However they are still far less expensive than the greedy method that involves shortest path computations. Though not directly applicable due to different natures of the problems, the large body of research in influence maximization [8, 26] can be leveraged from to obtain a larger pool of heuristics for *EIL*. We leave a more extensive evaluation including such heuristics as future work.

5. EVALUATION

Here we evaluate the performance of the greedy algorithm w.r.t. the three heuristics. Note that since influence propagation is a stochastic process, in order to evaluate value of each seed set with an error ϵ with high probability, we need to perform Monte Carlo simulations polynomial in $1/\epsilon$ and the number of nodes of the network [23]. This is one of the major scalability issues inherent in this type of problem. However, in our specific problem each simulation involves the expensive computation of shortest paths which is crucial to *EIL* and this makes *EIL* even more computationally intense then the influence maximization problems [24, 31]. We ran experiments choosing the adversary uniformly randomly. As part of our experiments, we also evaluated how factors like the degree centrality of the adversary, *delay* of campaign L, and the weight distribution for $p_{C,v,w}$ and $p_{L,v,w}$ influence our choice of best fit algorithm. This requires performing the computationally expensive simulations for each choice of such parameters. Taking these factors into consideration we performed experiments on 4 regional network graphs obtained from Facebook that exhibit properties such as power-law degree distribution, high clustering and positive assortativity [43]. The data sets are as follows: 2009 snapshot of Santa Barbara regional network with 26455 nodes and 453132 edges (bi-directional edges count as two edges); 2008 snapshot of the same network with 12814 nodes and 184482 edges; 2009 snapshot of the Monterey Bay regional network with 14144 nodes and 186582 edges; and 2008 snapshot of the same network with 6117 nodes and 62750 edges.

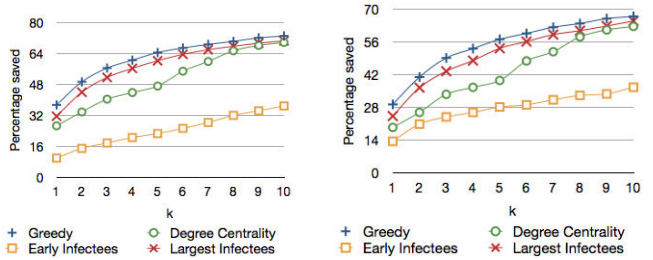
In Figure 3 we present our evaluation of the 4 methods on MCICM when L has the *high effectiveness property* and $p_{C,v,w}$ values of 0.1 using the Santa Barbara 2008 data set. The y-axis represents the percentage of the population that was *saved*. The x-axis represents the number of nodes that are initially activated in L. Figure 3(a) demonstrates the case where $delay = 20\%$ i.e. the ratio of the delay of the algorithm L to the duration of the campaign C is 0.2. In this case, all of the methods perform well, *saving* a large portion of the population. Figure 3(b) shows the rapid decay of the



(a) $delay = 20\%$ (b) $delay = 50\%$

Figure 3: Evaluation on SB08 for MCICM with high effectiveness property

performance of all the approaches in the case where delay is 50%. Here we omit the case where delay is 70% since all of the algorithms were doing poorly, especially degree centrality had near-0 value. We conducted the same experiments on the other data sets for which the result were similar. We also conducted experiments setting $p_{C,v,w}$ values to 0.5 instead of 0.1. Though the percentage saved is smaller for all algorithms, the trend w.r.t. increasing k was similar. Due to space limitations, we omit the graphs for these experiments. It is evident that for MCICM when L has the *high effectiveness property*, the biggest determining factor is how late the limiting campaign L is started. When L is started early, all the methods perform well whereas when the delay is large, all the algorithms perform poorly. For larger delays, greedy performs better than the other algorithms but the portion of the population saved is so small in all cases that this improvement is not significant.

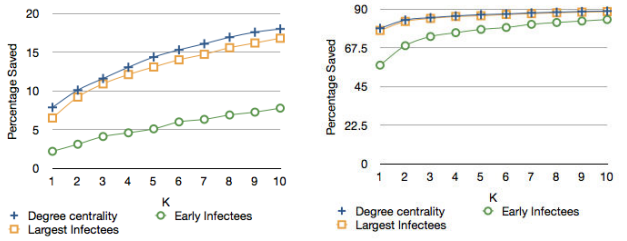


(a) $delay = 20\%$ (b) $delay = 20\%, deg(n_a) \geq 40$

Figure 4: Evaluation on SB08 for COICM

In Figure 4 we present our evaluation of the 4 methods on COICM using the Santa Barbara 2008 data set and setting $p_{v,w}$ values to 0.1. Figure 4(a) shows that when the delay is 20% both the *largest infectees* and *degree centrality* heuristics perform similar to the greedy method. Due to space limitations we omit the graph presenting the performance of the approaches when the delay is large. However as an example we note that when the delay is 50% and the number of seeds is 10, the greedy method performs 8%, 36% and 116% percent better than *largest infectees*, *degree centrality* and *early infectees* respectively. The reason for the decay of performance of *degree centrality* heuristic w.r.t. the delay is that degree centrality is purely a structural heuristic so the expectation of being infected is not computed for the seeds in A_L. When L is started too late, the highly connected nodes and their neighbors are more likely to have already been activated in campaign C. Comparing Figure 3(a) and 4(a), we observe the importance of *high effectiveness* since for the latter an average of 72% of the population can be saved with 10 seeds whereas the former shows consistent savings of 90-95% even with only one seed. Figure 4(b) presents the case where the delay of L is 20% and the adversary that C starts from, has degree ≥ 40. All methods are less effective when the start node of C is a highly connected node, since a highly connected adversary is likely to infect more people early on in which case when L is started a large portion of the population is already infected.

Next, we evaluate MCICM where L does not have the *high effectiveness property*. In this case, the greedy algorithm is too costly to perform since many of the optimizations we performed for the earlier two cases cannot be applied. Considering the results obtained from the earlier two sets of experiments, we conclude that, at least for close-knit social networks, the heuristics introduced above result in a good performance. Therefore we evaluated how well they perform on a slightly larger social network to see if there was consistency in their behavior. Figure 5 presents the test results for the Santa Barbara 2009 data set. Figure 5(a) presents the case where the limiting campaign is started early (delay is 20%) but campaign C is more dominant ($0 \leq p_{C,v,w} \leq 0.5$ for all edges) than L ($0 \leq p_{L,v,w} \leq 0.1$ for all edges) in the sense that nodes are more likely to adopt C than L. Figure 5(b) presents the opposite case where L is more dominant than C. In both cases, the degree centrality and *largest infectees* heuristics have similar behavior while *early infectees* performs worse than both. The savings are much larger for Figure 5(b) compared to Figure 5(a). We also note the similarity of Figure 5(b) with Figure 3(a), and claim that even if campaign L does not have the *high effectiveness property*, if it is more dominant than C, it is still likely to *save* a large population.



(a) $delay = 20\%$, $0 \leq p_{C,v,w} \leq 0.5$, $0 \leq p_{L,v,w} \leq 0.1$ (b) $delay = 20\%$, $0 \leq p_{C,v,w} \leq 0.1$, $0 \leq p_{L,v,w} \leq 0.5$

Figure 5: Evaluation on SB09 for MCICM

There are crucial lessons we can extract from the tests we performed: 1) In almost all cases, *largest infectees* performs comparable with the greedy algorithm while being far less computationally intense. The *early infectees* heuristic, on the other hand, performs poorly since it *strictly* targets nodes that are expected to be infected at time step r. In many cases even the simpler heuristic of degree centrality is a better alternative. 2) Parameters such as the delay of L, the connectedness of the adversary n_a are crucial to identify correctly to choose the right method for determining influential nodes for limiting a bad campaign C. For instance, when the delay is large, degree centrality is not a good option whereas it performs well for small delays. Having sufficient information about such parameters can help identify the best method for EIL.

6. EIL WITH INCOMPLETE DATA

So far, we have focused on the problem where A_C and the delay r are known. Therefore, we can provide an approximation algorithm with error bounds for the expected case. However, such precise data is not easy to attain. Practically, decisions must be made in the face of missing information. Therefore, we study a more realistic formulation of the *EIL* where the information about the current state of the nodes is *incomplete* and an *approximate* value for the number of currently infected nodes is known. The question we address is: "Can effective inoculation be performed in the presence of incomplete data and how fast does the performance degrade w.r.t. the amount of missing data?"

Consider a specific instance of propagation of a bad campaign C.

Assume that this process is detected at round r (value of r is unknown) at which point the limiting campaign L is to be started. Let the set of *active*, *inactive* and *newly activated* nodes for campaign C at round r be denoted Λ, Σ and Ξ respectively. Assume further that we are given the sets Λ_{given} and Σ_{given} where $\Lambda_{given} \subset \Lambda$ and $\Sigma_{given} \subset \Sigma$, i.e. we know for *only a subset* of nodes if campaign C reached out to and activated them by the time campaign L is to be started. Note that we assume Ξ is completely unknown, i.e. the current infectors are unknown and the value of r is not given.

The main idea we employ in this section is to provide a good prediction of the sets Λ, Σ and Ξ given Λ_{given}, Σ_{given} and c_a (an *approximate* value for $|\Lambda|$). Let the predicted sets be Λ_{pred}, Σ_{pred}, Ξ_{pred}. We then use these sets to create a new instance of ***EIL*** to provide a solution to the influence limitation problem under uncertain data. In Section 6.1, we introduce our prediction algorithm. Later in Section 6.2, we present the solution to the influence limitation problem under uncertain data using the results of the prediction method. Finally, Section 6.3 presents an evaluation of the methods.

6.1 Prediction Algorithms

6.1.1 Identifying Λ and Σ

The first step of the prediction is to predict Λ and Σ. Identifying Λ and Σ is crucial for three reasons: 1) They will be used to further identify the *newly activated* nodes Ξ, 2) nodes that are predicted to be already *active* in campaign C will be eliminated from the set of nodes to save, so inaccurately predicting *inactive* nodes to be *active* might result in not saving nodes that could be saved otherwise and 3) predicting *active* nodes to be *inactive* might result in targeting nodes that provide no savings which is a waste of resources.

Since we are given Λ_{given}, a subset of the nodes active in C and c_a, the total number of nodes active in C, our aim in this section is to find the other $c_a - |\Lambda_{given}|$ nodes that are *most likely* to be *active* in campaign C. This can be posed as an optimization problem, more precisely finding $\Lambda_{add}{}^*$, the set of $c_a - |\Lambda_{given}|$ nodes that maximizes the number of possible scenarios of spread of C including all nodes in Λ_{given} and no node from Σ_{given}.

Define $G_{add} = (N_{add}, V_{add})$ where $N_{add} = \Lambda_{add} \cup \Lambda_{given}$ and $E_{add} = \{(u,v) | (u,v) \in E \wedge u \in N_{add} \wedge v \in N_{add}\}$. For a set Λ_{add}, the number of cascade scenarios including nodes in Λ_{add} and Λ_{given} (and no other node) can be calculated as the number of spanning trees in $G_{add}{}'$ where $G_{add}{}'$ is the connected component of G_{add} that includes all the nodes in Λ_{given} (if no such component exists $G_{add}{}'$ is an empty graph). This follows from the fact that the spread of a cascade starting from one node under ***MCICM*** or ***COICM*** follows a tree structure. Therefore, the value function of a set Λ_{add} can be computed by counting the number of such spanning trees it would be able to produce and offsetting the value of each spanning tree by the multiplication of the weights of the edges of that spanning tree (to favor more likely scenarios of cascades) in the following way:

$$f(\Lambda_{add}) = \sum_{T \in T(G_{add}{}')} \prod_{(u,v) \in T} p_{C,u,v} \quad (1)$$

where $|\Lambda_{add}| = c_a - |\Lambda_{given}|$ and $T(G_{add})$ is the set of possible spanning trees in $G_{add}{}'$. The $c_a - |\Lambda_{given}|$ nodes that are most likely to be *active* can be detected by solving the optimization problem: $\Lambda_{add}{}^* = \arg\max f(\Lambda_{add})$. Note that the number of spanning trees of a graph can be exponential and therefore enumerating them is infeasible. Luckily, Kirchoff [27] introduced a method for counting the spanning trees which later was developed in a computationally useful form in [6]. For counting the number of spanning trees of a graph G with n nodes, the algorithm uses an nxn matrix $D = d_{i,j}$ called the *degree matrix*. The entries of this matrix are: $d_{i,i} = \sum w_{j,i}$, $d_{i,j} = -w_{i,j}$ if n_i and n_j are neighbors and $d_{i,j} = 0$ otherwise. Deleting the nth row and column from D, we get the *reduced* matrix D'. The determinant of D' is then the number of directed spanning trees in G [6]. Creating one such D matrix for graph $G_{add}{}'$ and setting the weights $w_{i,j}$ to $p_{C,i,j}$, we can find the number of spanning trees and compute the value of a specific Λ_{add} set. Same observation has been stated in a recent study for inferring networks of diffusion [18]. However this does not solve all scalability issues as there are $\binom{|N|-|\Lambda_{given}|-|\Sigma_{given}|}{|\Lambda_{add}|}$ possible Λ_{add} sets to evaluate.

The next observation we make is that $\Lambda_{add}{}^*$ can be found using a *supermodular* function. A function $f(.)$ is supermodular if it satisfies: $f(S \cup v) - f(S) \leq f(T \cup v) - f(T)$, for all elements v and all pairs of sets $S \subset T$ [39]. We will show $f(\Lambda_{add})' = \sum_{T \in T(G_{add}{}')} \prod_{(u,v) \in T} p'_{C,u,v}$ is supermodular where $p'_{C,u,v} = p_{C,u,v}/p_{min}$ and p_{min} is the smallest non-zero $p_{C,u,v}$ in G (to set $p'_{C,u,v} \geq 1$ for each edge). Note that as we are augmenting each probability with the same value, *maximizing $f(\Lambda_{add})'$ for a fixed size $|\Lambda_{add}|$ also maximizes $f(\Lambda_{add})$*. Consider two sets S and T s.t. $S \subset T$ and let S' (T') denote the subset of nodes in S (T) that form a connected component that include all Λ_{given}. We define similar subsets $S_u{}'$ ($T_u{}'$) for $S \cup u$ (or $T \cup u$). Supermodularity in the general case can be shown by a proof that shows each spanning tree "lost" for the set $S \cup u$ by removing node u can be augmented by the set of nodes in $T_u{}' - S_u{}'$ and shown to be lost when node u is removed from the set $T \cup u$. As the augmented tree has a higher weight, the loss in larger for $T \cup u$. The details of this proof is omitted due to space limitations. Instead we will demonstrate for the case $S_u{}' - S' - u = T_u{}' - T' - u = \emptyset$, $f(\Lambda_{add})'$ is supermodular:

THEOREM 6.1. *$f(\Lambda_{add})'$ is a supermodular function.*

PROOF. Consider two sets S and T s.t. $S \subset T$. Let the number of spanning trees (offset by the multiplication of the edges $p'_{C,u,v}$) induced by the set $S \cup n_i$ (or $T \cup n_i$) that has n_i as a leaf be $K_{S,1}$ ($K_{T,1}$) and the number of spanning trees induced by the set $S \cup n_i$ ($T \cup n_i$) that has n_i as an internal node be $K_{S,2}$ ($K_{T,2}$). Let $\sum_{v \in S} p'_{C,v,n_i}$ (or $\sum_{v \in T} p'_{C,v,n_i}$) be denoted by $deg_S(n_i)$ ($deg_T(n_i)$). Note that since $S \subset T$, $deg_S(n_i) \leq deg_T(n_i)$. By removing n_i from set $S \cup n_i$, f' will decrease from $K_{S,1} + K_{S,2}$ to $\frac{K_{S,1}}{deg_S(n_i)}$, whereas the reduction from set $T \cup n_i$ to T will reduce f' from $K_{T,1} + K_{T,2}$ to $\frac{K_{T,1}}{deg_T(n_i)}$. Since every spanning tree that involves nodes in $S \cup n_i$ and has n_i as an internal node can be augmented with the nodes in the set $T - S$ while still forming a spanning and since $p'_{C,u,v} \geq 1$ for each edge, we have $K_{S,2} \leq K_{T,2}$. Similarly, $K_{S,1} \leq K_{T,1}$. Factoring in the fact that $1 - \frac{1}{deg_S(n_i)} \leq 1 - \frac{1}{deg_T(n_i)}$, we can conclude that $K_{S,2} + (1 - \frac{1}{deg_S(n_i)})K_{S,1} \leq K_{T,2} + (1 - \frac{1}{deg_T(n_i)})K_{T,1}$. This shows that $f(S \cup n_i)' - f(S)' \leq f(T \cup n_i)' - f(T)'$. □

Our goal is to maximize a supermodular function with cardinality constraints, as we are seeking to detect a set of $c_a - |\Lambda_{given}|$ nodes that would maximize $f(.)'$. As evident from the definition, if a function $f(.)$ is supermodular, $-f(.)$ is submodular and therefore maximizing a supermodular function is equivalent to minimizing a submodular function. Unlike maximizing submodular functions, maximizing supermodular functions (or minimizing submodular functions) in general is shown to be polynomial-time solvable [20]. However maximizing supermodular functions with cardinality constraints are known to be NP-hard. The problem of minimizing a general submodular function under a cardinality constraint

1: {Given (Λ_{given},Σ_{given},G,c_a) where $G = (N, E)$ is the network graph, Λ_{given},Σ_{given} are the incomplete sets of active and inactive nodes and c_a is an approximate value of $|\Lambda|$ }
2: $\Lambda_{pred} = \Lambda_{given}$
3: Create a refined graph G' that consists of nodes in $N - \Sigma_{given}$
4: Select a node n_i at random from Λ
5: T_{stein} = min Steiner tree rooted at n_i in G' covering Λ_{given}
6: N_{stein} = nodes in T_{stein}
7: $\Lambda_{pred} = \Lambda_{pred} \cup N_{stein}$
8: **while** $|\Lambda_{pred}| \leq c_a$ **do**
9: $N_{choose} = n_i \in N - \Sigma_{given} - \Lambda_{pred}$
10: $\Lambda_{pred} = \Lambda_{pred} \cup \{argmax_{n \in N_{choose}} \{deg(n)_{\Lambda_{pred}}\}\}$
11: Output Λ_{pred}

Figure 6: Heuristic to identify Λ_{pred}

is known to be inapproximable within $o(\sqrt{n/logn})$ [38]. Svitkina et al. also provide a sampling-based solution that provides approximation guarantees of the same order as this lower bound [38]. The algorithm is a $(5\sqrt{\frac{n}{lnn}}, \frac{1}{2})$ bicriteria decision procedure. That is, given a feasible instance, it outputs a set U with $f(U) \leq 5\sqrt{\frac{n}{lnn}}B$ and $w(U) \geq W/2$ with probability at least p, where the cost of high p is more iterations. Although it is hard to even approximate an arbitrary supermodular function with cardinality constraints, certain functions can be solved exactly in polynomial time. Maximizing $f(\Lambda_{add})'$ can be easily reduced to a supermodular knapsack problem with cardinality constraints which is shown to be one such function [15]. Unfortunately, *polynomial time* in this setting is defined based on the number of calls to an *oracle* function which answers whether a given set X belongs to the base polyhedron associated with function f. We also note that even for the case where there is a polynomial time algorithm for maximizing $f(\Lambda_{add})'$, this problem is still computationally expensive for today's online social networks since the value function depends on computing a determinant which is an $O(n^3)$ problem.

As the optimization problem for identifying Λ and Σ proves to be computationally expensive, we next investigate heuristics to tackle this problem. A good heuristic should 1) provide a coherent answer, i.e. the nodes in Λ_{pred} should form a connected component so that there is at least one arborescence (directed spanning tree) in the subgraph induced by nodes in Λ_{pred}, 2) have as many arborescences as possible, therefore getting close to the optimal solution. Figure 6 provides details of the heuristic we used to predict Λ_{pred}. As our first goal is to provide a *feasible* solution, in the first step, we choose a Steiner tree in G that covers all nodes in Λ_{given} to make sure the infected nodes will form a connected component (Lines 2-7). Since we are aiming for a minimum Steiner tree *rooted at a specific node*, this computation can be easily done in polynomial time whereas the general minimum Steiner tree problem is NP-hard. As the next step we choose additional nodes which will induce a large number of arborescences. We do this by adding nodes that are connected to the largest number of nodes that are already predicted to be *active*. Note that, the more incoming edges a node has from *active nodes*, the more chances for it to be activated. Clearly, choosing nodes this way will result in choosing nodes that have a large number of *possible parents* in a directed tree. After identifying the set Λ_{pred}, we predict the *inactive* nodes to be: $\Sigma_{pred} = N - \Lambda_{pred}$. Section 6.3 provides an evaluation of this heuristic.

6.1.2 Identifying Ξ_{pred}

The first, rather naive approach we implemented to predict Ξ_{pred} was to select the most central node in set Λ_{pred}; i.e. the node that has the shortest average path to all the other nodes in Λ_{pred} and to perform a breadth-first-search from this node in G_{pred} (subgraph of G containing only the nodes in Λ_{pred} and their interconnects) to create a tree of information spread and to use the leaves of that tree as the *newly activated* nodes. Our experiments revealed that the influentials identified using this method have poor performance which led us to identify the next method of prediction.

Considering an influence epidemic starting from a single node under *IC, MCICM* or *COICM* diffusion models forms a tree since a node can be activated by at most one node in campaign C, we investigate the performance of generating a random spanning tree on graph G_{pred} to identify Ξ_{pred}. By constructing random spanning trees in a finite space of ST spanning trees on graph G_{pred} and a probability distribution γ on ST, we can sample a spanning tree st with probability $\gamma(st)$. Therefore, this method is more likely to pick scenarios of information diffusion that are more likely to happen. The algorithms for constructing random spanning trees can be categorized into two families, ones that are based on computing determinant [30] and ones that are based on random walks on the graph. Random walk based algorithms can be further categorized as ones that work for undirected graphs [5, 1] and directed graphs [45, 44]. Of those algorithms all except for [44] run within the cover time, i.e. the expected time it takes for the random walk to reach all the vertices. In this work, we use the algorithm provided in [44] which generates random spanning trees within mean hitting time (faster than the cover time) and provides solutions for directed graphs. Using this algorithm, a random spanning tree can be sampled according to the probability distribution on all spanning trees with probabilities proportional to the weights of the trees. Since we have no means of distinguishing the *importance* of edges, in our experiments we assigned the same weight to all the edges of the network which results in a unweighted graph but given more data, for instance the relative importance of friendships, this method will bias towards more likely scenarios where information flows on edges with higher weights. After generating a random spanning tree on G_{pred}, we select the leaves of the tree as the *newly activated nodes*. Note that this method is a heuristic. One could enumerate all possible spanning trees on G_{pred} and identify nodes that repeatedly appear at the highest depth of the tree as *newly activated* nodes. However this is a computationally expensive solution. We leave finding a scalable solution for Ξ while providing accuracy guarantees as an open problem.

6.2 Predictive Hill Climbing Approach (PHCA)

After constructing the sets Λ_{pred}, Σ_{pred}, Ξ_{pred}, we identify A_{LP}, the set of k nodes to influence by campaign L in graph G, in the following manner: Create graph $G_{reduced} = (N_{reduced}, E_{reduced})$ where $N_{reduced} = \Sigma_{pred} \cup \Xi_{pred}$ and $E_{pred} = \{(u, v)|(u, v) \in E \wedge u \in N_{reduced} \wedge v \in N_{reduced}\}$. We then use the algorithm presented in Figure 2 on graph $G_{reduced}$ where $A_C = \Xi_{pred}$ and delay $r = 0$. This method would provide $1 - 1/e$ approximation guarantees for the case where $\Lambda_{pred} = \Lambda$, $\Sigma_{pred} = \Sigma$ and $\Xi_{pred} = \Xi$. We call this method *Predictive Hill Climbing Approach (PHCA)*. The true value of A_{LP} ($\pi(A_{LP})$) can be computed by calculating the expected number of nodes A_{LP} would save in G_{real} where $G_{real} = (N_{real}, E_{real})$ s.t. $N_{real} = \Sigma \cup \Xi$ and $E_{real} = \{(u, v)|(u, v) \in E \wedge u \in N_{real} \wedge v \in N_{real}\}$ where the set of adversaries is $A_C = \Xi$ and delay r is 0.

6.3 Evaluation of PHCA

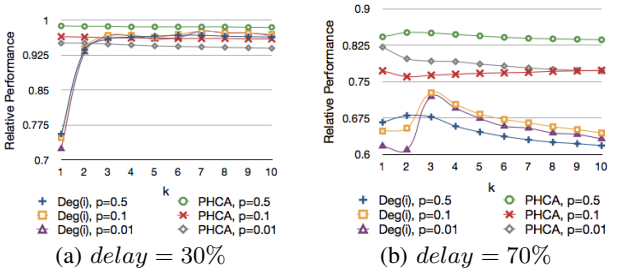
Accuracy, precision and recall statistics of the prediction algorithm are given in Table 1. *Accuracy* refers to the ratio of the nodes whose true states are correctly identified. *Precision* refers to the ratio of nodes that are *active* (or for Y: *newly activated*) to those that are identified as *active* and *recall* refers to the ratio of nodes identified as *active* to the total number of *active* nodes. Amount of unknown data is modeled by using the parameter p_{known} which denotes the probability that the state of a node is known. Decreas-

Table 1: Prediction Statistics

		Accuracy	Recall	Precision
$p_{known} = 0.5$	X	0.964	0.796	0.801
	Y	0.945	0.369	0.356
$p_{known} = 0.1$	X	0.933	0.614	0.617
	Y	0.936	0.227	0.224
$p_{known} = 0.01$	X	0.919	0.567	0.569
	Y	0.929	0.194	0.188

ing values of p_{known} would result in larger amounts of missing information. The prediction algorithm provides good accuracy (especially since many nodes of the graph are *inactive*) but precision and recall numbers decay with the amount of unknown data. Note, however, that we are interested in the performance of EIL under uncertain data rather than the *accuracy, precision* and *recall* statistics. We will now show that poor recall statistics do not necessarily translate to poor performance for *PHCA*.

We have studied 165,643 cascade scenarios under the *MCICM* with *high effectiveness property* choosing the starter of campaign C uniformly randomly and setting $p_{C,v,w}$ values to 0.1 in the Monterey Bay 2008 network . Each of these scenarios required performing in the order of 100,000 simulations to retrieve and evaluate influential nodes selected by each method. Figure 7 presents the results. In the presence of missing information, where the newly activated nodes in the network are unknown, using the greedy algorithm is not possible without prediction. In this case, the best available base-line algorithm to compare our method to is the degree centrality heuristic where seeds are chosen from the high degree nodes that are *not known to be infected*. The X-axis in both Figures 7(a) and 7(b) presents the number of seeds whereas the Y-axis presents the *relative performance* of the algorithm w.r.t. the performance of the greedy method with complete data, i.e. ratio of the number of the nodes saved using the respective method to the number of nodes that would be saved by the greedy method were we to have complete data (Λ, Σ, Ξ). Figure 7(a) presents the performance of the *predictive hill climbing approach(PHCA)* and the degree centrality(Deg(i)) heuristic under various amounts of missing information for the case where the limiting campaign L is started with 30% delay. Figure 7(b) provides similar data for the case when the delay is 70%. As it is evident from both figures, the performance of *predictive hill climbing approach* is mostly resilient to missing information. Especially when delay is small, *PHCA* performs within 96-90% of the performance with complete data and when delay is large it drops to 75% in the worst case ($p_{known} = 0.01$). The performance of the degree centrality heuristic without using prediction is not as robust and has consistently worse performance than *PHCA*. When the delay is small and a large number of seeds can be chosen, this heuristic performs well but when the delay is large, the performance compared to the greedy method (with complete data) fluctuates and consistently underperforms.



(a) $delay = 30\%$ (b) $delay = 70\%$

Figure 7: Influence Limitation using incomplete data

We also considered another variation on *EIL* with missing data where an outsider can observe "new activity" in the network, i.e. one can detect only the newly activated nodes. This problem formulation fits systems such as Twitter [40] where one can easily retrieve new tweets on a topic but retrieving the entire history of tweets on that topic is impractical. Interestingly, our experiments on the Facebook Monterey Bay 2008 network revealed that, even without using a prediction algorithm, i.e. assuming all the nodes except the newly activated ones are inactive, the limiting campaign performs within 99% of what would be achieved with complete data when the delay r is small, and within 92-96% when the delay is large. However, we believe deeper analysis and experimentation is needed to generalize this finding before concluding of a prediction method is not necessary for this case. In future work, we plan to investigate this problem further.

7. CONCLUSION

In this work we performed an extensive study of the problem of limiting the spread of misinformation in a social network. We investigated efficient solutions to the following question: Given a social network where a (bad) information campaign is spreading, who are the *k* "influential" people to start a counter-campaign if our goal is to minimize the effect of the bad campaign? We call this *eventual influence limitation* problem. We proved that this problem is NP-hard and therefore an exact solution is infeasible for large scale social networks. We also showed that two variations of this problem on two different communication models are submodular and therefore a greedy method is guaranteed to provide a $1/(1-e)$ approximation. Although the greedy algorithm is a polynomial time algorithm, it is still too costly for large scale social networks. Therefore, we also experimentally studied the performance of the greedy algorithm, comparing it with 3 different heuristics one of which is degree centrality. We showed that in many cases the performance of heuristics, even the simple degree centrality heuristic, is comparable to the greedy algorithm. We explored different aspects of the problem such as the effect of starting the limiting campaign early/late, or the properties of the adversary and how prone the population is to accepting either one of the campaigns.

We also studied the more realistic problem of influence limitation in the presence of missing information. We introduced an algorithm called *predictive hill climbing approach* which first predicts the current state of *all* the nodes of the network given the states of a fraction of the nodes and then uses the hill climbing approach to choose the set of "influentials" using the predicted data. We introduced an optimization algorithm to choose the set of nodes that are most likely to have been infected by the "bad campaign". We show that, even though the naive solution to this problem is exponential, using matrix-tree theorem, and supermodularity of the specific problem at hand, one could provide a polynomial time algorithm for this problem. However, the method requires using an *oracle function* that is not available for our problem. Therefore we seek heuristics to solve the prediction problem. Our method relies on generating random spanning trees to capture "likely cascade scenarios". Our experiments show that for most cases, the *predictive hill climbing approach* provides good performance, within 96-90% of the performance that would be achieved with no missing information. Although for small delays the performance is consistently within 96-90%, for large delays the performance degrades to 75% when the amount of missing information increases dramatically, i.e. states of the nodes are known with only 0.01 probability.

8. ACKNOWLEDGMENTS

This work is partially supported by NSF Grant IIS-0847925. We would also like to thank OIT at UCSB for TRITON support and Manuel Gomez-Rodriguez for his comments.

9. REFERENCES

[1] D. J. Aldous. The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM J. Discret. Math.*, 3(4):450–465, 1990.

[2] R. M. Anderson and R. M. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, USA, September 1992.

[3] N. T. J. Bailey. *The mathematical theory of infectious diseases and its applications*. 1975.

[4] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *WINE*, pages 306–311, 2007.

[5] A. Broder. Generating random spanning trees. In *SFCS '89*, pages 442–447, 1989.

[6] R. L. Brooks, C. Smith, A. Stone, and W. Tutte. The dissections of rectangles into squares. *Duke. Math. J.*, 7:312–340, 1940.

[7] T. Carnes, C. Nagarajan, S. M. Wild, and A. van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *ICEC '07*, pages 351–360, New York, NY, USA, 2007. ACM.

[8] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, 2009.

[9] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, pages 88–97, 2010.

[10] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

[11] O. Diekmann and J. A. P. Heesterbeek. *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation*. Wiley, 1 edition, May 2000.

[12] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, pages 57–66, 2001.

[13] P. Dubey, R. Garg, and B. D. Meyer. Competing for customers in a social network: The quasi-linear case. In *WINE*, pages 162–173, 2006.

[14] R. Durrett. Lecture notes on particle systems and percolation. Wadsworth Publishing, 1988.

[15] G. Gallo and B. Simeone. On the supermodular knapsack problem. *Mathematical Programming*, 45:295–309, 1989.

[16] J. Garrison and C. Knoll. Prop. 8 opponents rally across california to protest gay-marriage ban. *Los Angeles Times*, November 2008.

[17] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 2001.

[18] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD*, pages 1019–1028, 2010.

[19] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 1978.

[20] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[21] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Rev.*, 42(4):599–653, 2000.

[22] K. M. Heussner. Ft. hood soldier causes stir on twitter. *ABS News (online)*, November 2009.

[23] D. Kempe. *Structure and Dynamics of Information in Networks*. 2010.

[24] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[25] J. O. Kephart and S. R. White. Measuring and modeling computer virus prevalence. In *SP '93*, page 2, Washington, DC, USA, 1993. IEEE Computer Society.

[26] M. Kimura, K. Saito, and R. Nakano. Extracting influential nodes for information diffusion on a social network. In *AAAI*, pages 1371–1376, 2007.

[27] G. Kirchhoff. Über die auflösung der gleichungen auf welche man sei der untersuchung der linearen vertheilung galvanischer strome gefuhrt wind. *Ann. Phys. Chem*, 72:497–508, 1847.

[28] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse. Identifying influential spreaders in complex networks. Jan 2010.

[29] J. Kostka, Y. A. Oswald, and R. Wattenhofer. Word of mouth: Rumor dissemination in social networks. In *SIROCCO*, pages 185–196, 2008.

[30] V. G. Kulkarni. Generating random combinatorial objects. *Journal of Algorithms*, 11(2):185 – 207, 1990.

[31] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *KDD*, pages 420–429, 2007.

[32] T. M. Liggett. Interacting particle systems, 1985.

[33] D. Meier, Y. A. Oswald, S. Schmid, and R. Wattenhofer. On the windfall of friendship: inoculation strategies on social networks. In *EC '08*, pages 294–301, 2008.

[34] Michael jackson on tmz, iran on twitter. `http://www.blogher.com/spreading-news`.

[35] E. Morozov. Swine flu: Twitter's power to misinform. *Foreign Policy*, April 2009.

[36] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[37] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[38] Z. Svitkina and L. Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. In *FOCS*, pages 697–706, 2008.

[39] D. M. Topkis. *Supermodularity and Complementarity*. Princeton University Press, first edition edition, April 1998.

[40] Twitter. `http://www.twitter.com`.

[41] C. Wang, J. C. Knight, and M. C. Elder. On computer viral infection and the effect of immunization. In *ACSAC '00*, page 246, Washington, DC, USA, 2000. IEEE Computer Society.

[42] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.

[43] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *EuroSys*, pages 205–218, 2009.

[44] D. B. Wilson. Generating random spanning trees more quickly than the cover time. In *STOC '96*, pages 296–303, New York, NY, USA, 1996. ACM.

[45] D. B. Wilson and J. G. Propp. How to get an exact sample from a generic markov chain and sample a random spanning tree from a directed graph, both within the cover time. In *SODA '96*, pages 448–457, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.