Influence Maximization in Social Networks When Negative Opinions May Emerge and Propagate[*]

Wei Chen Alex Collins Rachel Cummings Te Ke Zhenming Liu
David Rincon Xiaorui Sun Yajun Wang Wei Wei Yifei Yuan

Abstract

Influence maximization, defined by Kempe, Kleinberg, and Tardos (2003), is the problem of finding a small set of seed nodes in a social network that maximizes the spread of influence under certain influence cascade models. In this paper, we propose an extension to the independent cascade model that incorporates the emergence and propagation of negative opinions. The new model has an explicit parameter called *quality factor* to model the natural behavior of people turning negative to a product due to product defects. Our model incorporates negativity bias (negative opinions usually dominate over positive opinions) commonly acknowledged in the social psychology literature. The model maintains some nice properties such as submodularity, which allows a greedy approximation algorithm for maximizing positive influence within a ratio of $1 - 1/e$. We define a *quality sensitivity ratio (qs-ratio)* of influence graphs and show a tight bound of $\Theta(\sqrt{n/k})$ on the qs-ratio, where n is the number of nodes in the network and k is the number of seeds selected, which indicates that seed selection is sensitive to the quality factor for general graphs. We design an efficient algorithm to compute influence in tree structures, which is nontrivial due to the negativity bias in the model. We use this algorithm as the core to build a heuristic algorithm for influence maximization for general graphs. Through simulations, we show that our heuristic algorithm has matching influence with a standard greedy approximation algorithm while being orders of magnitude faster.

keywords: influence maximization; social networks; negative opinions; independent cascade model;

1 Introduction

Viral marketing, a strategy of conducting product promotions through social influences among individuals' cycles of friends, families, or co-workers, is believed to be a very effective marketing strategy, mainly because it is based on trusted relationships. With the increasing popularity of online social networks such as Facebook, Myspace, and Twitter, the power of viral marketing has more potential than ever before. Therefore, understanding of the effective ways of utilizing viral marketing is crucial.

Motivated by this background, the research community has recently studied the algorithmic aspects of maximizing influence in social networks for viral marketing ([11, 12, 13, 17, 20, 6, 5, 27, 7]). All these works are based on the two basic influence cascade models, namely independent cascade model and linear threshold model, originally defined by Kempe et al. in [11], and their extensions. The essence of the model is that, for a social network modeled as a graph, starting from a small initial set of vertices in the graph (called *seeds*), a stochastic process specifies how influence is propagated from these seeds to their neighbors and neighbors of neighbors, and so on, until the process ends and a portion of the network is activated. The *influence maximization* problem is thus to find an optimal seed set of size at most k such that the expected number of vertices activated from this seed set, referred to as its *influence spread*, is the largest.

However, all of the above works ignore one important aspect of influence propagation that we often experience in the real world. That is, not only positive opinions on products and services that we receive may propagate through the network, negative opinions are also propagating, and are often more contagious and stronger in affecting people's decisions. For example, if you heard from one of your coworkers that she found a cockroach in her meal yesterday in a nearby restaurant, very likely you will avoid this restaurant for a while. Furthermore, you are likely to tell your other friends and co-workers about this, discouraging them to patronize the restaurant even though you did not have this bad experience yourself. In constrast, if you heard good words about the restaurant, you are more likely to visit the restaurant, but probably you will only spread the good words

[*] Author affiliations and emails: W. Chen (contact author), Microsoft Research Asia, China, weic@microsoft.com. A. Collins, Google Inc., U.S.A., aecollin@google.com. R. Cummings, University of Southern California, U.S.A., rcumming@usc.edu. T. Ke, University of California at Berkeley, U.S.A., kete@berkeley.edu. Z. Liu, Harvard University, U.S.A., zliu@fas.harvard.edu. D. Rincon, Universitat Politècnica de Catalunya, Spain, drincon@entel.upc.edu. X. Sun, Shanghai Jiao Tong University, China, sunsirius@sjtu.edu.cn. Y. Wang, Microsoft Research Asia, China, yajunw@microsoft.com. W. Wei, Carnegie Mellon University, U.S.A., weiwei@cs.cmu.edu. Y. Yuan, University of Pennsylvania, U.S.A., yifeiy@cis.upenn.edu. The work was done when all authors were working at or visiting Microsoft Research Asia.

about it after you have a good meal there yourself.

The impact of negative opinions and its asymmetry with positive opinions have long been studied in the social psychology literature (e.g. [22, 26, 1, 25]). In these studies, researchers show that negative impact is usually stronger and much more dominant than positive impact in shaping people's decisions. Marketing literature also addresses negative influence explicitly: people who spread negative opinions are called *detractors* while people spreading positive opinions are called *promoters* (see e.g. [23]). Therefore, when studying influence maximization, it should be important to incorporate the emergence and propagation of negative opinions into the influence cascade model and study its impact together with positive influence. This is exactly the goal of our paper.

In this paper, we first propose a new influence cascade model, the *independent cascade model with negative opinions (IC-N)*, which extends the independent cascade (IC) model of [11], and explicitly incorporates the emergence and propagation of negative opinions into the influence cascade process. The IC-N model is associated with a new parameter q called the *quality factor*. Informally the IC-N model works as follows. Initially, a set of nodes in the network is selected as seeds and are activated (e.g. provided with free trials of the product/service). With probability q each seed turns positive (experiencing good quality of the product/service) and with probability $1 - q$ turns negative (encountered defects). At each time step, a positively activated node in the previous step tries to positively activate each of its non-active neighbors, and if successful (with a success probability) the neighbor is activated (bought the product/service), but it only turns positive with probability q and with probability $1 - q$ it turns negative. Meanwhile a negatively activated node in the previous step also tries to negatively activate its non-active neighbors, and if successful the neighbors become negative (accepted negative opinions and avoiding the product/service). If several nodes try to activate the same node in one step, the order of activation trials is random (See Section 2 for the formal model definition).

The IC-N model captures several phenomena that match our daily experience as well as research results in social psychology. In particular, the product defects are usually the originator of negative opinions, and negative opinions usually dominate positive opinions in decision making and propagation, which is called *negativity bias* in social psychology literature (See Section 2.1 for the conceptual justification of the model.)

For influence maximization, we focus on maximizing the expect number of positive nodes in the network after the cascade, which we refer to as *positive influence spread*, since it is directly related to the revenue generated by the viral marketing effort.

In this paper, we present the following results concerning influence maximization in the IC-N model. First, we study if a universally good quality factor q^* exists such that the optimal seeds selected under q^* is good enough even if the actual quality factor is not q^*. To do so, we define a metric called *quality sensitivity ratio (qs-ratio)* for an influence graph such that a large value of qs-ratio implies that seed selection is sensitive to q. We show that for general graphs, qs-ratio is $\Theta(\sqrt{n/k})$, where n is the number of nodes in the graph and k is the number of seeds to be selected. The result implies that influence maximization algorithms for general graphs need to explicitly incorporate the quality factor, unless one can show specifically that certain graphs of interest have low qs-ratios. Moreover, our proof reveals the seed selection criteria under different quality factors: under a high quality factor we should select seeds with large overall reaches, while under a low quality factor we should select seeds with large immediate neighborhoods. This insight is helpful in understanding and guiding seed selection in general graphs when considering the quality factor.

Second, we study the influence spread mechanism for the IC-N model. We show that positive influence spread in the IC-N model satisfies a diminishing return property called *submodularity*, which immediately results in a $1 - 1/e$-approximation algorithm given the black box access to the influence spread function [11]. On the other hand, computing the *exact* influence spread given a seed set is shown to be #P-hard for general graphs even without negative opinions [5]. It is therefore desirable to know under what circumstances computing the influence spread is no longer intractable with the presence of negative opinions. In Section 4, we show that when the graph is a directed tree, we can compute exact positive influence spread in the IC-N model with a dynamic programming method. The algorithm is much more involved than the straightforward recursive algorithm for the IC model in [5], because the negativity bias feature of the IC-N model makes it necessary to differentiate negative activations from positive activations in the analysis.

Next, we address the practical concern of scaling up the approximation algorithm for finding the seeds. The greedy algorithm with simulation-based influence estimation [11] is slow and not scalable, as already shown in [5, 7]. Instead, we follow the successful approach of [5, 7] to design a heuristic algorithm MIA-N, in which we use local tree structures surrounding a node to represent its local influence and use the above influence computation in trees to achieve fast influence computation and seed selection (Section 5). We conduct experiments using several real-world and synthetic networks and show that (Section 6): (a) quality factor q affects positive influence spread in a superlinear way, (b) our MIA-N algorithm generates influence spread very close to the influence spread of the greedy algorithm, and (c) our MIA-N algorithm is orders of magnitude faster than the greedy algorithm and can be scaled to large graphs of million nodes and vertices. Therefore, our MIA-N algorithm is a good candidate for influence maximization with negative opinions in large-scale real networks.

Finally, we study several further model extensions to IC-N (Section 7). Our results indicate that when adding more parameters to the model, some nice properties such as submodularity no longer holds. This indicates that IC-N model provides a good balance between model expressiveness in covering realistic scenarios and model tractability for efficient algorithms, while if we need to go beyond the IC-N model, some new approach may be required to tackle the influence maximization problem.

As a summary, our paper is the first to incorporate negative opinions emerged due to imperfect product qualities into the influence cascade model and provide detailed studies of influence maximization in this context. Our contributions include (a) proposing the IC-N model that incorporates the emergence and propagation of negative opinions, and showing that it maintains nice properties such as submodularity; (b) studying the quality sensitivity of influence graphs and showing that influence maximization in general graphs may be sensitive to the quality factor; (c) designing an efficient algorithm for computing influence spread in tree structures; (d) designing an efficient heuristic for influence maximization that has influence spread matching the best greedy algorithm while having running time orders of magnitude faster.

Due to space constraints, additional results and some proofs are included in our full technical report [4].

1.1 Related work Domingos and Richardson [9, 24] are the first to study influence maximization as an algorithmic problem. Their methods are probabilistic, however. Kempe, Kleinberg, and Tardos [11] are the first to formulate the problem as a discrete optimization problem. They show that the problem is NP-hard, propose a greedy approximation algorithm, and study generalizations of independent cascade and linear threshold models.

A number of studies [13, 17, 19, 6, 5, 27, 7] aim at improving the efficiency of the greedy algorithm or providing alternative heuristics, while some other work [3] proves that certain formulation of the problem is hard to approximate. Our MIA-N heuristic has a similar structure as the heuristic of [5], but the latter is only for the original IC model without negative opinions, and thus the algorithm is much simpler. Lappas et al. [14] study k-effectors problem, which contains influence maximization (without negative opinions) as a special case. They also use a tree structure to make the computation tractable, and then approximate the original graph with a tree structure. The difference, besides not considering the negative opinion, is that they use one tree structure but our MIA-N algorithm uses multiple local tree structures, one per node to simulate local influence propagations.

Bharathi et al. studies competitive influence diffusion in [2], using an extension of the IC model. The model is for influence diffusion of two or more competing products, and thus it does not have the key futures of our model, such as negative influence emergence due to product defects and negativity bias.

Propagations of negative opinions have been studied extensively in marketing and social science literature, but its algorithmic perspective is rarely touched in the computer science literature. To the best of our knowledge, the only related paper that discusses diffusion of negative opinions is [18]. However, negative opinions in their model are exogenous, and there is no explanation on where negative opinions come from. Moreover, they use the same propagation model for both positive and negative opinions, which ignores negativity bias that have been commonly acknowledged in the social psychology literature. Therefore, their model is closer to the competitive influence diffusion model rather than negative opinion diffusion model. Finally, they use a heat diffusion process, and their focus is not on negative opinion diffusion.

2 Independent Cascade Model with Negative Opinions

We first introduce the independent cascade model with negative opinions (IC-N), and then provide conceptual justifications and some useful properties of the model.

We model a social network as a directed graph $G = (V, E)$, where V is the set of nodes representing individuals and E is the set of directed edges representing relationships among individuals. Each edge of the graph G is associated with a *propagation probability*, which is formalized by function $p : E \to [0, 1]$. We refer to the triple (V, E, p) as an *influence graph*, and also use G to represent it. For a node $v \in V$, let $N^{in}(v)$ and $N^{out}(v)$ denote v's *in-neighbors* and *out-neighbors* respectively.

The dynamic of the IC-N model is as follows. Each node has three states, *neutral*, *positive*, and *negative*. Discrete time steps $t = 0, 1, 2, \ldots$ are used to model dynamic changes in the network. We say that a node v is *activated* at time t if it is positive or negative at time t and neutral at time $t - 1$ (if $t > 0$). The model has a parameter q called *quality factor*, which indicates the probability that a node stays positive after it is activated by a positive in-neighbor. Initially at time $t = 0$, all nodes in a pre-determined *seed set* $S \subseteq V$ are activated, and for each node $v \in S$, with probability q v becomes positive and with probability $1 - q$ v becomes negative. At a time $t > 0$, for any neutral node v, let $A_t(v) \subseteq N^{in}(v)$ be the set of in-neighbors of v that were activated at time $t - 1$. Every node $u \in A_t(v)$ tries to activate v with an independent probability of $p(u, v)$. If one of them is successful, v is activated at step t. Moreover, if v is activated by a negative node u, then v becomes negative; if v is activated by a positive node u, then with probability q v becomes positive while with probability $1 - q$ v becomes negative. To determine which node activates v, we randomly permute all nodes in $A_t(v)$, and let each node in $A_t(v)$ try to activate v following the permutation order until we find the first node u that successfully activates v. Once v is activated and fixed its state (positive or negative), it does not change its state any more. The activation process stops when there is no new activated node in a time step. Note that if $q = 1$, nodes can only be positively activated, and IC-N is reduced to the original independent cascade (IC) model of [11].

The *positive influence spread* of a seed set S in influence graph G with quality factor q is the expected number of positive nodes activated in the graph, and is denoted as $\sigma_G(S, q)$. Given an influence graph $G = (V, E, p)$, a target seed set size k, and a quality factor q, the *influence maximization* problem is to find a seed set S^* of cardinality k such that S^* has the largest positive influence spread in G, i.e. $S^* \in \mathrm{argmax}_{S \subseteq V, |S|=k} \sigma_G(S, q)$.

2.1 Conceptual justification of the model The IC-N model reflects several phenomena of negative influence that match our daily experiences as well as the studies in social psychology. First, negative opinions are originated from imperfect product/service qualities. In the model, when a node v is activated by a positive node u, it means that v is positively influenced by u and subsequently buys the product/service. However, due to defects of the product/service (e.g. the cockroach in the meal), v may dislike the product/service and generate negative opinion about it. The quality factor q reflects the quality of the product, and thus is the property of the product, not the network. Therefore, it is reasonable to use the same q across the network. Typically, before a product is put onto the market, the company will have quality control by testing and/or focus group studies, and thus it is reasonable to assume that an estimate of q is available.

Second, negative and positive influence are asymmetric, and negative influence is more dominant, which is reflected in the IC-N model from two aspects. The first aspect is that, when a node v is negatively activated, it becomes negative with probability one and will stay negative even if it later sees other neighbors turning positive. This reflects the negativity bias and dominance phenomenon studied in social psychology (e.g. [25]) — when combining positive and negative opinions, negative opinions are likely to dominate. The second aspect is that, when v is negatively activated and turns negative, v will also negatively influence its neighbors, even though v does not personally experience the product/service. This is the manifestation of negativity dominance in the domain of contagion, as summarized in [25]: "negative events may have more penetrance or contagiousness than positive events" (e.g. you are likely to spread the bad words about the restaurant even if do not see the cockroach yourself). Note that because of the above negativity bias in the IC-N model, the model is not equivalent as a simpler model in which node activations are first propagated using the IC model and then each node independently decides to be positive or negative based on quality factor q.

Third, we use positive influence spread as our objective since it is directly related to the expected revenue the seller would gain from the viral marketing effort.

We believe our model is a reasonable first-order approximation of the emergence and propagation of negative influence and negativity bias phenomenon. Of course, we may further adjust or extend the model, but we also need to keep model parsimony — the balance between model expressiveness and model simplicity and tractability. In Section 7 we discuss several model extensions and alternatives. Ultimately, statistical analyses on real datasets are needed to validate the model, but this is beyond the scope of this paper and is our future work item.

2.2 Properties of the model We now discuss several key properties of $\sigma_G(S, q)$ to be used in the later sections. Given an influence graph $G = (V, E, p)$, seed set S and quality factor q, let $pap_G(v, S, q)$ denote the "positive activation probability", the probability that node v is positive after the influence cascade from S ends. By the linearity of expectation, it is clear that $\sigma_G(S, q) = \sum_{v \in V} pap_G(v, S, q)$. Let $d_G(S, v)$ denote the graph distance from S to v in G, which is the length of the shortest path from any node in S to v. If there is no path from any node in S to v in G, then $d_G(S, v) = +\infty$. As a convention, $q^{+\infty} = 0$ for all $0 \le q \le 1$ (even when $q = 1$). Let $a_G(S, i)$ denote the number of nodes that are i steps away from set S in G, i.e., $a_G(S, i) = |\{v \mid d_G(S, v) = i\}|$. The following lemma shows a basic property of the IC-N model that leads to many later results.

LEMMA 2.1. *For influence graph $G = (V, E, p)$, suppose that $p(e) = 1$ for all $e \in E$. Then we have for all $v \in V$,*

$$pap_G(v, S, q) = q^{d_G(S,v)+1},$$

and

$$\sigma_G(S, q) = \sum_{i=0}^{n-1} a_G(S, i) q^{i+1}.$$

For any influence graph $G = (V, E, p)$, after we determine all random events on all edges based on their propagation probabilities, we obtain a subgraph $G' = (V', E', p')$, where $V' = V$, $E' \subseteq E$, and $p'(e) = 1$ for all $e \in E'$. G' is obtained with probability $\Pr_G(G') = \prod_{e \in E'} p(e) \cdot \prod_{e' \in E \setminus E'} (1 - p(e'))$. Let Ω_G denote the set of all such subgraphs G'. We say that an edge e is *activated* if e is selected in the random subgraph G'.

An alternative view of the IC-N model is that we first select edges to obtain G', and then influence is propagated on G'. In the graph G', when multiple neighbors of a node

v try to activate v at the same step, we do not need to follow the random permutation order on these neighbors because the first neighbor selected will always activate v. Therefore, in this case we only need to select one of the neighbors of v uniformly at random among all its neighbors activated at the previous step, and the result is the same. We refer to this alternative view as *edge activation view*. Many subsequent results including the following lemma use this alternative view of the IC-N model.

LEMMA 2.2. *Given an influence graph $G = (V, E, p)$, a seed set $S \subseteq V$ and a quality factor q, we have*

$$\begin{aligned}\sigma_G(S, q) &= \mathrm{E}_{G' \leftarrow \Omega_G}[\sigma_{G'}(S, q)] \\ &= \sum_{G' \in \Omega_G} \Pr_G(G') \sigma_{G'}(S, q) \\ &= \sum_{G' \in \Omega_G} \Pr_G(G') \sum_{i=0}^{n-1} a_{G'}(S, i) q^{i+1}.\end{aligned}$$

COROLLARY 2.1. *For any influence graph $G = (V, E, p)$, when fixing a seed set S, function $\sigma_G(S, q)$ on q is monotonically increasing and continuous.*

A set function f on vertices of graph $G = (V, E, p)$ is a function $f : 2^V \to \mathbb{R}$. Set function f is *monotone* if $f(S) \leq f(T)$ for all $S \subseteq T$, and it is *submodular* if $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$ for all $S \subseteq T$ and $u \in V \setminus T$.

THEOREM 2.1. *For any influence graph $G = (V, E, p)$, when fixing a quality factor q, set function $\sigma_G(S, q)$ on S is monotone, submodular, and $\sigma_G(\emptyset, q) = 0$.*

Proof. Notice that

$$\sigma_G(S, q) = \sum_{G' \in \Omega_G} \Pr_G(G') \sum_{v \in V} q^{d_{G'}(S,v)+1}.$$

Define $Q_v(S) = q^{d_{G'}(S,v)+1}$. It is sufficient to show that $Q_v(S)$ is monotone and submodular. Clearly, $Q_v(S)$ is monotone because adding extra elements to the seed set S can only decrease the quantity $d_{G'}(S, v)$. It remains to show that the function is also submodular.

Let $S \subseteq T \subseteq V$ and $u \in V \setminus T$. Clearly, $d_{G'}(S, v) \geq d_{G'}(T, v)$. If $d_{G'}(u, v) \geq d_{G'}(S, v)$, we have $Q_v(S \cup \{u\}) - Q_v(S) = Q_v(T \cup \{u\}) - Q_v(T) = 0$. If $d_{G'}(u, v) \leq d_{G'}(T, v)$, we have $Q_v(S \cup \{u\}) - Q_v(S) = Q_v(T \cup \{u\}) - Q_v(S) \geq Q_v(T \cup \{u\}) - Q_v(T)$ as $Q_v(\cdot)$ is monotonically increasing. The only remaining case is $d_{G'}(T, v) < d_{G'}(u, v) < d_{G'}(S, v)$. In such case, $Q_v(S \cup \{u\}) - Q_v(S) > 0 = Q_v(T \cup \{u\}) - Q_v(T)$. Therefore, $Q_v(\cdot)$ is monotone and submodular. □

With Theorem 2.1, we can apply the result in [21] to obtain a greedy approximation algorithm that achieves $1 - 1/e$ approximation ratio for the influence maximization problem. Algorithm 1 shows the greedy algorithm with a generic monotone and submodular set function f, which would be replaced by $\sigma_G(S, q)$ in our case for any fixed q. The algorithm iteratively selects a new seed u that maximizes the incremental change of f into the seed set S until k seeds are selected.

Algorithm 1 Greedy(k, f)

```
initialize $S = \emptyset$
for $i = 1$ to $k$ do
    select $u = \arg\max_{w \in V \setminus S}(f(S \cup \{w\}) - f(S))$
    $S = S \cup \{u\}$
end for
output $S$
```

The greedy algorithm relies on an efficient computation of $\sigma_G(S, q)$ given set S. However, as pointed out in [5], even when $q = 1$ computing $\sigma_G(S, q)$ is #P-hard. Thus following [11] we use Monte-Carlo simulations of the IC-N model to estimate $\sigma_G(S, q)$. In this case we can achieve an approximation ratio of $1 - 1/e - \epsilon$, where ϵ is small if we use a large number of simulations to estimate $\sigma_G(S, q)$.

The theoretical running time of the greedy algorithm is $O(knmR)$, where k, n, m, and R are the number of seeds, number of nodes, number of edges, and number of simulations, respectively. In the actual implementation used for our experiments, we apply optimization techniques such as the lazy-foward method proposed in [17] to speed up the running time.

3 Quality Sensitivity in Influence Maximization

Since obtaining quality factor q and incorporating it into influence maximization complicates the matter, one may wish to find a constant q^* that is "universally good enough" for a network, in the sense that the optimal seeds found under q^* in the network is reasonably effective regardless of the true value of q. In the rest of this section, we formalize the goal of finding such $q*$ via the notion of *sensitivity*, and show that in general graphs "universally good" q^* may not exist. This suggests that the problem of maximizing positive influence spread in general graphs requires the knowledge of q, unless one can show explicitly that certain graphs have low sensitivities to the quality factor.

Let $\mathcal{S}^*_{G,k}(q) = \mathrm{argmax}_{S \subseteq V, |S|=k} \sigma_G(S, q)$ denote the set of all possible optimal seed sets of size k under a given q, and let $\sigma^*_{G,k}(q)$ denote the maximum positive influence spread with k seeds under q, i.e., $\sigma^*_{G,k}(q) = \max_{S \subseteq V, |S|=k} \sigma_G(S, q)$. The subscripts G and k may be dropped whenever they are clear from the context.

Fix a small constant $c \in (0, 1)$. For a given seed set S of size k, we define the *quality sensitivity ratio (qs-ratio) of S* *for graph G with k seeds* to be the maximum ratio between the optimal influence spread under q and the influence spread of S under q, when q ranges from c to 1, that is,

$$qsr_{G,k}(S) = \max_{q \in [c,1]} \frac{\sigma^*_{G,k}(q)}{\sigma_G(S,q)}.$$

Intuitively, the qs-ratio of seed set S indicates how well S is as a representive under different q: if its qs-ratio is close to 1, then S could be used across different q values (i.e. S is insensitive to q), but if its qs-ratio is large, S is not a good seed set under some q's (i.e. S is sensitive to q). The reason we need a small constant c to bound q away from 0 is because very poor quality is unlikely to happen in practice and mathematically it is a singular point.

Given a quality factor q, we define the *quality sensitivity ratio of* q to be the minimum qs-ratio among all the optimal seed sets under q, that is,

$$qsr_{G,k}(q) = \min_{S \in \mathcal{S}^*_{G,k}(q)} qsr_{G,k}(S).$$

The reason we take the minimum over all optimal seed sets is to (optimistically) consider the best case where some algorithm may find the optimal seed set with the best qs-ratio. Finally, the *quality sensitivity ratio of the influence graph G under target seed set size k* is the minimum qs-ratio among all q values, that is,

$$qsr_{G,k} = \min_{q \in [c,1]} qsr_{G,k}(q). \tag{3.1}$$

The metric $qsr_{G,k}$ indicates that, if we want to use one q value and one optimal seed set S^* under q to work for other possible q values, the best an algorithm can do is to select a q^* that achieves $\min qsr(q)$ and an S^* that achieves $\min_{S \in \mathcal{S}^*(q^*)} qsr(S)$, but in this case there could be some other q' such that the ratio between the optimal influence spread under q' and the influence spread achieved by S^* under q' is $qsr_{G,k}$.

We now give tight upper bounds on both $qsr(q)$ and qsr. Let $n = |V|$ be the number of nodes in the graph. We shall show that for *any graph* and *any* k, the following inequalities hold $qsr_{G,k}(q) \le n/k$ and $qsr_{G,k} \le \sqrt{\frac{n}{ck}}$. On the other hand, we may indeed be able to construct a family of graphs so that $qsr_{G,k}(q) = \Omega(n/k)$ and $qsr = \Omega(\sqrt{\frac{n}{k}})$. These results suggest that there exists a family of influence graphs so that an inappropriate assumption over the value of q will result in the *worst possible* outcome in terms of multiplicative errors, which could be as large as $\Omega(\sqrt{n})$.

LEMMA 3.1. *For any graph G, any integer k, and any $q \in [c,1]$, we have $qsr(q) \le n/k$. Furthermore, for any constant k and $q \in [c,1]$, there exists a family of influence graphs such that $qsr_{G,k}(q) = \Omega(n/k)$. In particular, when the integer k and $q \in [c,1]$ are given, there exists an N and a sequence of graphs $\mathcal{G} = \{G_N, G_{N+1}, G_{N+2}, ...\}$ with $|V(G_i)| = i$ such that for any $G_n \in \mathcal{G}$, we have $qsr_{G_n,k}(q) = \Omega(n/k)$.*

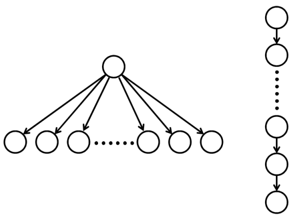


Figure 1: An example of graph to reach large $qsr_{G,k}$ rate. In this example, the value $k = 1$ and the graph only consists of two components $\{\mathrm{C}^1_1, \mathrm{C}^2_1\}$. The component C^1_1 is a star with $\sqrt{n}$ nodes (left diagram); the component C^2_1 is a line with $n - \sqrt{n}$ nodes (right diagram).

LEMMA 3.2. *For any influence graph G and target seed set size k, $qsr_{G,k} \le \sqrt{\frac{n}{ck}}$.*

The proof of Lemma 3.1 is relatively easy while the proof of Lemma 3.2 is more involved, and it uses the monotonicity and continuity of function $\sigma_G(S,q)$ on q as shown by Corollary 2.1.

LEMMA 3.3. *There exists a family of influence graphs $\mathcal{G} = \{G_n\}$ such that $qsr_{G_n,k} = \Omega(\sqrt{n/k})$ for n being sufficiently large.*

Proof. Our family of graphs consists of $2k$ disjoint components $\{\mathrm{C}^1_1, \mathrm{C}^1_2, ..., \mathrm{C}^1_k, \mathrm{C}^2_1, \mathrm{C}^2_2, ..., \mathrm{C}^2_k\}$, in which C^1_i ($i \in [k]$) are stars of size $n_1 = \sqrt{\frac{n}{k}}$ with edge directions pointing away from the center of the star, and C^2_i ($i \in [k]$) are one-way directed lines of size $n_2 = \frac{n}{k} - n_1$. Propagation probabilities on all edges are 1. Figure 1 represents an example of the graph when $k = 1$.

Now the computation of $qsr_{G,k}$ is immediate: when $q = 1$, the set $\mathcal{S}^*_{G,k}$ consists of a unique element S_1, which is the set of all roots of lines; when $q < 1$, the set $\mathcal{S}^*_{G,k}$ consists of a unique element S_2, which is the set of all centers of the stars (for large enough n). Therefore, $qsr_{G,k}(1) = \max_{q \in [c,1]} \frac{\sigma^*_{G,k}(q)}{\sigma_G(S_1,q)} = \frac{1}{4}\sqrt{\frac{n}{k}}$ since $\sigma^*_{G,k}(q) = q^2 n_1 k$ and $\sigma_G(S_1,q) \approx qk/(1-q)$ for $q < 1$ and a sufficiently large n. And for $q < 1$, we have $qsr_{G,k}(q) = n_2/n_1 = \sqrt{\frac{n}{k}} - 1$. Summing up above, we obtain the desired result $qsr_{G,k} = \Omega(\sqrt{n/k})$. □

Remark Notice that the components C^1_i and C^2_i actually suggest two different topologies, in which finding the seed set critically depends on the actual value of q. Lines and stars are extreme examples that yield largest qsr. In fact, when lines are substituted by degree bounded trees (e.g., tree with width $\log n$), the qsr value will still be bad (e.g., $qsr_{G,k} = \tilde{\Omega}(\sqrt{\frac{n}{k}})$ when the tree width is $\log n$). The moral of the lemma is that when the graph contains two different kind of structures, where one structure has fast neighborhood growth initially but small overall reach and the other structure has slow neighborhood growth but large overall reach, the optimal choice of the seed set may critically depend on the product's quality. With high quality factor, we prefer to choose structures with a large reachable set, but with a low quality factor, we prefer to choose structures that have large immediate neighborhood, since when influence are propagated in multiple hops, it is likely that someone in the chain will dislike the product if the quality factor is low.

Summing up above, we have the following theorem.

THEOREM 3.1. *For any influence graph G and target seed set size k, we have $qsr_{G,k} \leq \sqrt{\frac{n}{ck}}$, and for any $q \in [c, 1]$, $qsr(q) \leq \frac{n}{k}$. Moreover, there exist families of graphs such that the above upper bound is tight up to a constant factor.*

Since the qs-ratio for general graphs could be quite large as shown by the above theorem, it is worthwhile to invest in algorithms that explicitly incorporate quality factor q. In practice, q could be estimated by quality testing and focus group studies, and thus it is reasonable to assume that an estimate on q is available for influence maximization.

4 Computing Influence in Arborescences

As pointed out in [5], computing influence spread in a general influence graph in the IC model is #P-hard. In this section, we show an efficient algorithm to compute influence spread in tree structures. This algorithm will be used in Section 5 to derive an efficient heuristic for influence maximization.

An in- (or out-) arborescence is a directed tree where all edges point into (or away from) the root. Consider an arborescence $A = (V, E, p)$ with p as the propagation probability function on edges. Fix a seed set $S \subseteq V$ and a quality factor q. We study the algorithm that computes the positive influence spread $\sigma_A(S, q)$ in A. Since A, q, and S are fixed in this section, we will omit them in our notations.

For any $u \in V$, let $pap(u)$ denote the *positive activation probability* of u, which is the probability that u is positive after the influence cascade ends in A. It is clear that $\sigma_A(S, q) = \sum_{u \in V} pap(u)$, so we focus on the computation of $pap(u)$.

If A is an out-arborescence, the computation is straightforward and is summarized by the following lemma.

LEMMA 4.1. *For an out-arborescence A and a node u in A. Let $path(u)$ denote the path from seed s in S to u in A that has the minimum length among all such paths ($\emptyset$ if no such path exists). Let $E(path(u))$ denote the edge set of the path and $|path(u)|$ is the length of the path. Then we have $pap(u) = \prod_{e \in E(path(u))} p(e) \cdot q^{|path(u)|+1}$ if $path(u) \neq \emptyset$, and otherwise $pap(u) = 0$.*

With the above lemma, it is easy to see that we can compute the positive influence spread $\sigma_A(S, q)$ in one traversal of the out-arborescence. On the contrary, computing the positive influence spread in an in-arborescence is more involved. For the rest of this section, let A be an in-arborescence, and we focus on computing $pap(u)$ in A.

Let $ap(u)$ denote the *activation probability* of u, which is the probability that u is activated (positive or negative) after the influence cascade ends in A. As described already in [5], computing $ap(u)$ (or equivalently $pap(u)$ when $q = 1$) is easily done using the following recursive formula $ap(u) = 1 - \prod_{w \in N^{in}(u)}(1 - ap(w)p(w, u))$, with the boundary condition $ap(s) = 1$ for all $s \in S$, and $ap(u) = 0$ for all non-seed leaves u. However, once negative opinions may emerge in the network ($q < 1$), the situation changes significantly for computing $pap(u)$.

Suppose now that some of u's in-neighbors are positive and some are negative. Because of the negativity bias in the IC-N model, in particular negative neighbors will only make u negative while positive neighbors may make u positive or negative, the influence result on u depends on the order of the activation attempts of u's neighbors. This order is affected by two factors: (a) the time steps at which neighbors of u are activated, and (b) the random permutation among the neighbors who are activated at the same time step. A direct recursive formulation of $pap(u)$ requires a summation of all possible combinations of u's neighbors activation steps and all possible random permutations, which is exponential to the size of the graph and the number of seeds. In the following, we use the dynamic programming method to give an efficient algorithm to compute $pap(u)$. The computation is divided into two steps.

Computing $ap(u, t)$. Let $ap(u, t)$ denote the probability that u is activated at step t, for any integer $t \geq 0$. Thus we have $ap(u) = \sum_{t \geq 0} ap(u, t)$. The following lemma shows a recursive formula for $ap(u, t)$.

LEMMA 4.2. *For any $u \in V$ and any integer $t \geq 0$, we have*

$$
ap(u, t) = \tag{4.2}
$$

$$
\begin{cases}
1 & t = 0 \wedge u \in S, \\
0 & t = 0 \wedge u \notin S, \\
0 & t > 0 \wedge u \in S, \\
\begin{array}{l}\prod_{w \in N^{in}(u)}[1 - \sum_{i=0}^{t-2} ap(w, i)p(w, u)] \\ - \prod_{w \in N^{in}(u)}[1 - \sum_{i=0}^{t-1} ap(w, i)p(w, u)]\end{array} & t > 0 \wedge u \notin S.
\end{cases}
$$

Proof. The cases of $t = 0$ or $u \in S$ are trivial. Consider the case $t > 0$ and $u \notin S$. For an in-neighbor $w \in N^{in}(u)$,

$ap(w,i)p(w,u)$ is the probability that w is activated at step i and edge (w,u) is also activated, which means u will be activated in step $i+1$ if u is not already activated. Since the events of w being activated at a step i for different i's are mutually exclusive, $1-\sum_{i=0}^{t-2} ap(w,i)p(w,u)$ is the probability that u is not activated by w at step $t-1$ or earlier. Thus $\prod_{w\in N^{in}(u)}[1-\sum_{i=0}^{t-2} ap(w,i)p(w,u)]$ is the probability that u is not activated (by any of its in-neighbors) at step $t-1$ or earlier. Note that as the convention, $\sum_{i=0}^{-1} ap(w,i)p(w,u) = 0$ so the above is still true for $t = 1$. Similarly, $\prod_{w\in N^{in}(u)}[1-\sum_{i=0}^{t-1} ap(w,i)p(w,u)]$ is the probability that u is not activated (by any of its in-neighbors) at step t or earlier. Therefore, their difference is exactly the probability that u is activated at step t, which is $ap(u,t)$. □

The recursive computation given in Formula (4.2) can be easily carried out by using the dynamic programming method and traversing the arborescence from the leaves to the root. Let h be the height of the arborescence A, $k = |S|$ be the number of seeds, $n = |V|$ be the number of nodes in A, and ℓ be the number of possible steps that the root of A could be activated in A. It is straightforward to see that $\ell \le \min(k,h)$. Therefore, computing all $ap(u,t)$'s for all $u \in V$ and all possible t's using Formula (4.2) and dynamic programming takes $O(\ell n) = O(\min(k,h)n)$ time.

Computing $pap(u,t)$. Let $pap(u,t)$ denote the probability that u is activated and turns positive at step t, for any integer $t \ge 0$. The following lemma shows that $pap(u,t)$ can be easily derived from $ap(u,t)$.

LEMMA 4.3. *For any $u \in V$ and any integer $t \ge 0$, we have*

$$pap(u,t) = ap(u,t)\cdot q^{t+1}. \tag{4.3}$$

With Formula (4.3), we obtain the positive activation probability $pap(u) = \sum_{t\ge 0} pap(u,t)$, and the influence spread $\sigma_A(S) = \sum_{u\in V} pap(u)$. Therefore, we obtain the following result.

THEOREM 4.1. *Formulae (4.2),(4.3) together provide an efficient computation of influence spread in an in-arborescence A, with time complexity $O(\ell n) = O(\min(k,h)n)$, where ℓ, k, h, and n are the number of possible steps in which the root of A could be activated, the number of seeds, the height of A, and the number of nodes in A, respectively.*

5 MIA Algorithm for IC-N

The greedy algorithm (Algorithm 1) is slow because it lacks of an efficient way of computing the positive influence spread given a seed set. In this section, we develop a heuristic algorithm that uses arborescences to approximate local influence regions of the node, and uses the algorithm of Section 4 to compute influence spread efficiently in arborescences. The key points are that influence from a node is typically restricted to the local neighborhood region of the node, and that the computation of influence spread could be performed efficiently by the algorithm in Section 4.

For a path $P = \langle u = p_1, p_2, \ldots, p_m = v\rangle$, we define the *positive propagation probability* of the path, $ppp(P)$, as

$$ppp(P) = \Pi_{i=1}^{m-1} p(p_i, p_{i+1})\cdot q^m.$$

Intuitively the probability that u activates v through path P and makes v positive is $ppp(P)$, because it needs to activate all nodes along the path and all nodes along the path turn positive. To approximate the actual expected influence within the social network, we propose to use the *maximum influence path* (*MIP*) to estimate the influence from one node to another. Let $\mathcal{P}(G,u,v)$ denote the set of all paths from u to v in influence graph G.

DEFINITION 1. (MAXIMUM INFLUENCE PATH) *For influence graph G, we define the* maximum influence path *$MIP(u,v)$ from u to v in G as*

$$MIP(u,v) = \arg\max_P\{ppp(P) \mid P \in \mathcal{P}(G,u,v)\}.$$

Ties are broken in a predetermined and consistent way, such that $MIP(u,v)$ is always unique, and any subpath in $MIP(u,v)$ from x to y is also the $MIP(x,y)$. If $\mathcal{P}(G,u,v) = \emptyset$, we denote $MIP(u,v) = \emptyset$.

Note that for each edge (u,v) in the graph, if we add a distance weight $-\log(p(u,v)q)$ on the edge, then $MIP(u,v)$ is simply the shortest path from u to v in the weighted graph G. Therefore, the maximum influence paths and the later maximum influence arborescences directly correspond to shortest paths and shortest-path arborescences, and thus they permit efficient algorithms such as Dijkstra algorithm to compute them.

For a given node v in the graph, we propose to use the *maximum influence in-arborescence* (*MIIA*), which is the union of the maximum influence paths to v,[1] to estimate the influence to v from other nodes in the network. We use an *influence threshold* θ to eliminate MIPs that have too small propagation probabilities. Symmetrically, we also define *maximum influence out-arborescence* (*MIOA*) to estimate the influence of v to other nodes.

DEFINITION 2. (MAXIMUM INFLUENCE IN(OUT)-ARBORESCENCE) *For an influence threshold θ, the* maximum influence in-arborescence *of a node $v \in V$, $MIIA(v,q,\theta)$, is*

$$MIIA(v,q,\theta) = \cup_{u\in V, ppp(MIP(u,v))\ge\theta} MIP(u,v).$$

[1] Since we break ties in maximum influence paths consistently, the union of maximum influence paths to a node does not have undirected cycles, and thus it is indeed an arborescence.

The maximum influence out-arborescence $MIOA(v, q, \theta)$ *is:*

$$MIOA(v, q, \theta) = \cup_{u \in V, ppp(MIP(v,u)) \geq \theta} MIP(v, u).$$

Intuitively, $MIIA(v, q, \theta)$ and $MIOA(v, q, \theta)$ give the local influence regions of v, and different values of θ controls the size of these local influence regions. Given a set of seeds S in G and the in-arborescence $MIIA(v, q, \theta)$ for some $v \notin S$, we approximate the IC-N model by assuming that the influence from S to v is only propagated through edges in $MIIA(v, q, \theta)$. With this approximation, we can calculate the probability that v is activated given S exactly, using the algorithm given in Section 4. We refer to our model of restricting influence through local arborescences as the MIA model.

Let $\mu(S, q)$ denote the positive influence spread of S in our MIA model, in influence graph G with quality factor q. Let $pap(v, S, A, q)$ be the positive activation probability of v in in-arborescence A with seed set S and quality factor q. Then we have

$$\mu(S, q) = \sum_{v \in V} pap(v, S, MIIA(v, q, \theta), q). \tag{5.4}$$

We are interested in finding a set of seeds S of size k such that $\mu(S, q)$ is maximized. As already pointed out in [5], results in [11, 10] imply that maximizing $\mu(S, q)$ is still hard, even to any approximation factor within $1 - 1/e + \epsilon$ for any $\epsilon > 0$.

Nevertheless, we have that $\mu(S, q)$ for any given q is still submodular and monotone, because every $pap(v, S, MIIA(v, q, \theta), q)$ is submodular and monotone. Therefore, the greedy Algorithm 1 with influence spread computed by algorithm in Section 4 achieves $1 - 1/e$ approximation ratio for the influence maximization problem in the MIA model. The important point of the algorithm is that, when a new seed u is selected, we only need to update the incremental influence spread of nodes $w \in MIIA(v, q, \theta)$ where $v \in MIOA(u, q, \theta)$, since other nodes are not affected by the selection of u. The full pseudocode of the algorithm mostly deals with how incremental influence spread of every node is initialized and updated and is omitted due to space constraint. We denote the full algorithm as MIA-N.

THEOREM 5.1. *Algorithm* MIA-N *finds a seed set S of size k, the influence spread of which is guaranteed to be within $1 - 1/e$ of the optimal influence spread in the MIA model.*

Running time. We discuss the running time of algorithm MIA-N. Let $n = |V|$ be the number of nodes in the graph. Let $n_i = \max_{v \in V}\{|MIIA(v, q, \theta)|\}$ and $n_o = \max_{v \in V}\{|MIOA(v, q, \theta)|\}$. Let h_{max} denote the maximum height among all $MIIA(v, q, \theta)$'s. Computing $MIIA(v, q, \theta)$ and $MIOA(v, q, \theta)$ can be done using efficient implementations of Dijkstra's shortest-path algorithm. Assume the maximum running time to compute $MIIA(v, q, \theta)$ (resp. $MIOA(v, q, \theta)$) for any $v \in V$ is t_i (resp. t_o). Notice that $n_i = O(t_i)$ and $n_o = O(t_o)$.

The initialization part of MIA-N needs to compute $MIIA(v, q, \theta)$ and $MIOA(v, q, \theta)$ for all $v \in V$. We only need to compute and store all $MIOA(v, q, \theta)$'s using the Dijkstra shortest-path algorithm, since $MIIA(v, q, \theta)$ can be easily obtained from $MIOA(v, q, \theta)$'s. Initializing incremental influence spread is done by computing $pap(u, \{v\}, MIIA(u, q, \theta), q)$ for all $u \in MIOA(v, q, \theta)$ with Lemma 4.1, which takes $O(|MIOA(v, q, \theta)|)$ time. We use a max-heap to store incremental influence spread of every node, which takes $O(n)$ time. Therefore, initialization takes $O(nt_o)$ totally.

The main part of MIA-N has k iterations, each of which selects a new seed u and then updates the incremental influence spread for every $w \in MIIA(v, q, \theta)$ where $v \in MIOA(u, q, \theta)$, so total number of updates in each iteration is $O(n_o n_i)$. In each update, $pap(v, S \cup \{w\}, MIIA(v, q, \theta), q)$ with the new seed set S needs to be computed, which uses the algorithm in Section 4 and takes $O(\min(k, h_{max})n_i)$ time. Updating the entry on the max-heap takes $O(\log n)$ time. Hence the running time for the main loop is $O(kn_o n_i(\min(k, h_{max})n_i + \log n))$. Therefore, the total running time of MIA-N is $O(nt_o + kn_o n_i(\min(k, h_{max})n_i + \log n))$.

Since propagation probability along a path drops exponentially fast in general, for large n and a reasonable range of θ values, n_i, n_o, and t_o are significantly smaller than n, and thus our algorithm should have good efficiency, as demonstrated by our experiments.

6 Experiments

We implement both the greedy algorithm and the MIA-N algorithm, and conduct experiments on these two algorithms using three real-world networks as well as synthetic networks. We are interested in comparing both the influence spread and the running time of the two algorithms. We do not include other heuristics such as degree or distance centrality based heuristics or PageRank style algorithms, because none of them takes into account the quality factor q in the IC-N model, and thus by our quality sensitivy study they cannot be applied as a general solution to all social networks.

6.1 Experiment setup

Dataset. We use three real-world networks of increasing sizes in our experiments. The first dataset, NetHEPT, is an academic collaboration network extracted from the "High Energy Physics - Theory" section (form 1991 to 2003) of the e-print arXiv (http://www.arXiv.org). The nodes in NetHEPT are authors and an edge between u and v means u and v coauthored a paper (we allow multiple edges between a pair of nodes). The second dataset, WikiVote, is

Table 1: Statistics of the three real-world networks.

Dataset	NetHEPT	WikiVote	Epinions
number of nodes	15K	7K	76K
number of edges	31K	101K	509K
average degree	4.12	26.64	13.4
maximal degree	64	1065	3079
number of connected components	1781	24	11
largest component size	6794	7066	76K
average component size	8.55	296.46	6.9K

Note: Directed graphs are treated as undirected graphs in these statistics.

a voting history network from Wikipedia [16], where nodes represent Wikipedia users, and a directed edge from u to v means v voted on u (for promoting u to adminship). The third dataset, Epinions, is a Who-trust-whom network of Epinions.com [15], where nodes are members of the site and a directed edge from u to v means v trusting u (and thus u has influence to v). Note that for WikiVote and Epinions, we reverse the edge directions from the original graphs, since we are studying influence and we interpret v voting u or trusting u as u having an influence on v. Basic statistics about these networks are given in Table 1. We also use synthetic power-law degree graphs generated by the DIGG package [8] to test the scalability of our algorithm with different sized graphs of the same feature.

For propagation probability on edges, we use the *weighted cascade* model proposed in [11]. In this model, $p(u, v)$ for an edge (u, v) is $1/d(v)$, where $d(v)$ is the in-degree of v.

Algorithms. We evaluate both MIA-N and the Greedy algorithm. For the greedy algorithm, we use the lazy-forward optimization of [17] to speed up the computation. For each candidate seed set S, 20000 simulations are run to obtain an accurate estimate of the influence spread. For MIA-N, the θ parameter is chosen as $1/160$ for all of our tests. A method of choosing θ is given in [5], and for IC-N the method is the same. To obtain the influence spread of the MIA-N algorithm, for each seed set, we run the simulation on the networks 20000 times and take the average of the influence spread, which matches the accuracy of the greedy algorithm.

The experiments are run on a server with 2.33GHz Quad-Core Intel Xeon E5410 and 32G memory running on Microsoft Windows Server 2003.

6.2 Experiment results

Quality factor on influence spread. We first run the greedy algorithm on NetHEPT to select up to 50 seeds, with the quality factor q taking values from 0.5 to 1. Figure 2 shows the result of this test. Clearly, when q increases, the positive influence spread increases in a superlinear trend. For example, when q doubles from 0.5 to 1, the influence spread increases about 7.2 times (averaging from $k = 1$ to $k = 50$). The reason is due to negativity bias — if the product quality drops, the negative influence would be more dominant, and the loss in positive influence spread is more than the simple proportion of those people directly experiencing the slip of product quality. Therefore, the result suggests that maintaining a high product quality is very important in achieving a high influence spread.

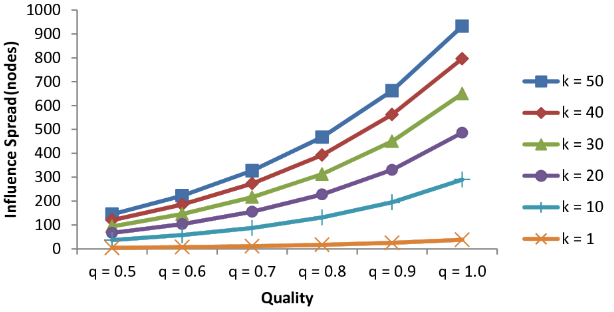


Figure 2: Influence Spread vs. the quality factor for the NetHEPT network.

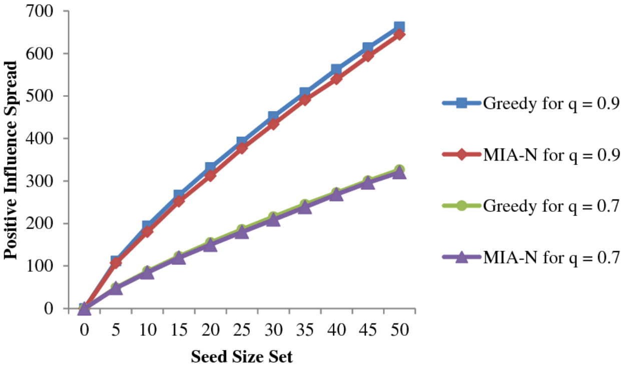
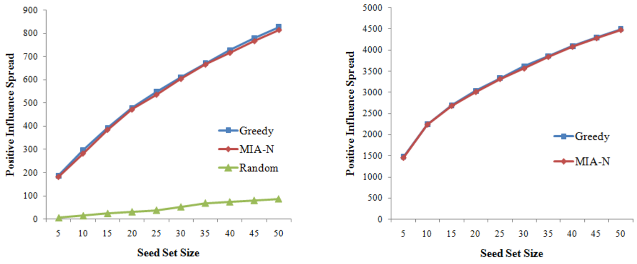


Figure 3: Positive influence spread for NetHEPT.

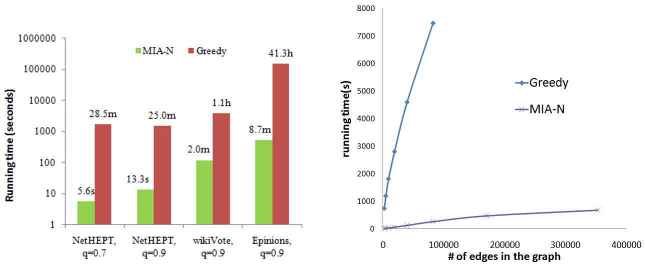
Positive influence spread and running time on real-world datasets. Figures 3 and 4 show the influence spread results for the three networks. For ease of reading, the legend of each figure lists the algorithms in the same order as their corresponding influence spread with 50 seeds. All figures show that the performance of MIA-N consistently matches the performance of the greedy algorithm in all



(a) WikiVote

(b) Epinions

Figure 4: Positive influence Spread for WikiVote and Epinions, for $q = 0.9$.



(a) real-world graphs

(b) synthetic graphs

Figure 5: Running time results. (a) running time for three real-world networks; (b) scalability test on synthetic power-law graphs. All use $q = 0.9$.

three networks, and for different quality factors (tested for NetHEPT for $q = 0.7$ and $q = 0.9$). Figure 4(a) also shows the influence spread of randomly selecting seeds, which is significantly worse than the greedy algorithm and MIA-N. This is consistent with previous reported results, and we omit reporting results of random seed selection for other datasets. On the other hand, Figure 5(a) shows that in all cases, our MIA-N algorithm is orders of magnitude faster than the greedy algorithm (the speedup is 307,112,33,285 times, respectively).

Scalability of MIA-N. We further test the scalability of MIA-N algorithm by using a family of synthetic power-law graphs generated by the DIGG package [8]. We generate graphs with doubling number of nodes, from $2K$, $4K$, up to $256K$, using power-law exponent of 2.16. Each size has 10 different random graphs and our running time result is the average among the runs on these 10 graphs. We run both the greedy algorithm and MIA-N to select 50 seeds for each graph. The result in Figure 5(b) clearly shows that our MIA-N scales almost linearly with the size of the graph, and scales much better than the greedy algorithm (e.g. MIA-N only takes 11 minutes to finish in a graph of $256K$ nodes and $353K$ edges while the greedy algorithm takes more than 2 hours to finish a graph four times smaller). The greedy algorithm has a much steeper curve mainly because it requires a large number of simulations to estimate influence spread accurately. Reducing the number of simulations in the greedy algorithm will significantly reduce its accuracy, as already reported in similar earlier work [5, 7], and we omit the report here.

Quality sensitivity. In general it may be intractable to compute the quality sensitivity of an influence graph. For the tested graphs we obtain the qs-ratios in some restricted cases and also use MIA-N to test their sensitivity. Our results (see [4] for more details) indicate that these influence graphs are not sensitive to the quality factor. However, this does not mean that MIA-N is not useful. On the contrary, without MIA-N, we cannot efficiently check if a large influence graph is quality sensitive. Since obtaining qs-ratio directly seems to be intractable, we propose that MIA-N is an efficient tool to check the quality sensitivity of a given influence graph. If the result from MIA-N indicates that the graph is not quality sensitive, then we do not need to obtain the quality factor of the product; otherwise we do need to obtain a good estimate of the quality factor and use MIA-N with the quality factor estimate to achieve a better influence maximization result.

7 Further Model Extensions

We further extend the IC-N model and study different optimization objectives. In particular, we have considered the following four model extensions: (a) allowing each node to have a different quality factor to model the situation where different individuals have different tendency of turning negative to a product; (b) allowing negative influence to propagate through an edge with higher probabilities to further strengthen negativity bias; (c) allowing different propagation delays along different edges to model the nonuniform interaction frequency between individuals; and (d) using other objectives such as maximizing the difference or the ratio between positive and negative influence spread.

For each of the alternatives, we investigate whether the important properties of monotonicity and submodularity still hold for the objective function. Our results show that, except for some extreme cases, none of these models could maintain these properties. Therefore, we see that introducing the quality factor q seems to reach a boundary, from which introducing further parameters will both complicate the model and make it much less tractable. If we do need to introduce more parameters to make the model more realistic, new techniques are needed to tackle the influence maximization problem for these models. Our results thus suggest that the IC-N model provides a good balance between the expressiveness of the model in covering realistic scenarios and the tractability of the model in allowing efficient algorithms. We include a number of results on these models in [4].

All of the above topics are interesting ones for future research. We hope that our study could motivate more work on the algorithmic aspects of social influence propagations that include both positive and negative opinions.

Acknowledgment

This project was initiated as a summer project in the program of Research in Industrial Projects for Students (RIPS) 2009, jointly hosted by Institute for Pure and Applied Mathematics (IPAM), University of California at Los Angeles and Microsoft Research Asia (MSRA). We thank IPAM and MSRA for the generous support of this project.

References

[1] R. F. Baumeister, E. Bratslavsky, and C. Finkenauer. Bad is stronger than good. *Review of General Psychology*, 5(4):323–370, 2001.

[2] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*, pages 306–311, 2007.

[3] N. Chen. On the approximability of influence in social networks. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms*, pages 1029–1037, 2008.

[4] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. Technical Report MSR-TR-2010-137, Microsoft Research, Oct. 2010.

[5] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large scale social networks. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.

[6] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009.

[7] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 10th IEEE International Conference on Data Mining*, 2010.

[8] L. Cowen, A. Brady, and P. Schmid. DIGG: DynamIc Graph Generator. http://digg.cs.tufts.edu.

[9] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[10] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[11] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[12] D. Kempe, J. M. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. In *Proceedings of the 32nd International Conference on Automata, Languages, and Programming*, pages 1127–1138, 2005.

[13] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 259–271, 2006.

[14] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila. Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1059–1068, 2010.

[15] J. Leskovec. Epinions social network. http://snap.stanford.edu/data/soc-Epinions1.html.

[16] J. Leskovec. Wikipedia vote network. http://snap.stanford.edu/data/wiki-Vote.html.

[17] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 420–429, 2007.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 233–242, 2008.

[19] R. Narayanam and Y. Narahari. Determining the top-k nodes in social networks using the shapley value. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1509–1512, 2008.

[20] R. Narayanam and Y. Narahari. A shapley value based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering*, 2010. to appear. Online version available at http://clweb.csa.iisc.ernet.in/nrsuri/social-networks-nrsuri-2010.pdf.

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

[22] G. Peeters and J. Czapinski. Positive-negative asymmetry in evaluations: The distinction between affective and informational negativity effects. *European Review of Social Psychology*, 1:33–60, 1990.

[23] F. F. Reichheld. The one number you need to grow. *Harvard Business Review*, Dec. 2003.

[24] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 61–70, 2002.

[25] P. Rozin and E. B. Royzman. Negativity bias, negativity dominance, and contagion. *Personality and Social Psychology Review*, 5(4):296–320, 2001.

[26] S. E. Taylor. Asymmetrical effects of positive and negative events: The mobilization-minimization hypothesis. *Psychological Bulletin*, 110(1):67–85, 1991.

[27] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.