Soc. Netw. Anal. Min. (2015) 5:14
DOI 10.1007/s13278-015-0252-6

ORIGINAL ARTICLE

A model-independent approach for efficient influence maximization in social networks

Hemank Lamba[1] · **Ramasuri Narayanam**[2]

Received: 30 October 2014 / Accepted: 25 April 2015 / Published online: 20 May 2015

Abstract The well-known influence maximization problem (Kempe et al., in proceedings of the 9th SIGKDD international conference on knowledge discovery and data mining (KDD), pp 137–146, 2003) (or viral marketing through social networks) deals with selecting a few influential initial seeds to maximize the awareness of product(s) over the social network. As it is computationally hard (Kempe et al., in proceedings of the 9th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp 137–146, 2003), a greedy approximation algorithm is designed to address the influence maximization problem. However, the major drawback of this greedy algorithm is that it runs extremely slow even on network datasets consisting of a few thousand nodes and edges (Leskovec et al., in proceedings of the 13th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp 420–429, 2007; Checn et al., in proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp 937–944, 2009). Several efficient heuristics have been proposed in the literature (Checn et al., in proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp 937–944, 2009) to alleviate this computational difficulty; however, these heuristics are designed for specific influence propagation models such as linear threshold model and independent cascade model. This motivates the strong need to design an approach that not only works with any influence propagation model, but also efficiently solves the influence maximization problem. In this paper, we precisely address this problem by proposing a new framework which fuses both link and interaction data to come up with a backbone for a given social network, which can further be used for efficient influence maximization. We then conduct thorough experimentation with several real-life social network datasets such as DBLP, Epinions, Digg, and Slashdot and show that the proposed approach is efficient as well as scalable.

Keywords Social networks · Influence maximization · Sparsification

1 Introduction

Most of the recent online social media collect a huge volume of data not just about who is linked with whom (*aka link data*), but also about who is interacting with whom (*aka interaction data*). The presence of both *variety* and *volume* in these datasets pose new challenges while conducting social network analysis. In particular, we present a general framework to deal with both variety and volume in the data for a key social network analysis task, namely viral marketing. The phenomenon of viral marketing is to exploit the social interactions among individuals to promote awareness for new products. There are two well-known operational models in the literature that capture the underlying dynamics of the information propagation in viral marketing. They are the linear threshold (LT) model (Schelling 1978; Kempe et al. 2003) and the independent cascade (IC) model (Goldenberg et al. 2001; Kempe et al. 2003).

✉ Ramasuri Narayanam
nrsuri@gmail.com; ramasurn@in.ibm.cm

Hemank Lamba
hlamba@cs.cmu.edu

[1] Carnegie Mellon University, Pittsburgh, USA

[2] IBM Research, Bangalore, India

Domingos and Richardson (2001) posed a fundamental algorithmic problem in the context of viral marketing with single product as follows. We are given the information about the extent individuals influence each other. We would like to market a new product that we hope will be adopted by a large fraction of individuals in the network. One of the key issues in viral marketing is to select a set of influential individuals (also called as *initial seeds*) in the social network and give them free samples of the product (or simply promotional offers on the product) to trigger a cascade of influence over the network. The problem is, given a value for k, how should we choose a set of k influential individuals so that the cascade of influence over the network is maximized? Hereafter, we refer to this problem as *influence maximization problem* (Kempe et al. 2003). It is shown to be an NP-hard problem and is well studied in the literature in the context of a single product (Domingos and Richardson 2001; Richardson and Domingos 2002; Kempe et al. 2003; Leskovec et al. 2007; Chen et al. 2009), multiple independent products (Datta et al. 2010), and two competing companies (Bharathi et al. 2007; Borodin et al. 2010). For more details, we refer to Sect. 2 on the related work.

As the influence maximization problem is computationally hard, Kempe et al. (2003) proposed a greedy approximation algorithm and it achieves an approximation guarantee of $\left(1-\frac{1}{e}\right)$. However, the severe drawback of this greedy algorithm is that it runs extremely slow even on network datasets consisting of a few thousand nodes and edges (Leskovec et al. 2007; Chen et al. 2009). To circumvent the difficulties associated with the computational aspects of the greedy algorithm, several efficient heuristics have been proposed in the literature (Chen et al. 2009) to address the influence maximization problem. However, a major inadequacy of these efficient heuristics is that they are designed for specific influence propagation models such as linear threshold model and independent cascade model.

This motivates the strong need to design an approach that not only works with any influence propagation model, but also efficiently solves the influence maximization problem taking into account heterogeneous data sources that a social network provides. The primary contribution of this paper is to address this important research gap, and our proposed approach complements the existing approaches in the literature for solving the influence maximization problem. In what follows, we briefly describe our approach in this paper.

1.1 Our approach

There are two key tasks associated with our proposed approach. We refer to the first task as *graph sparsification* and the second task as *influence maximization on sparse graphs*. In what follows, we now briefly describe each of the two tasks.

Graph sparsification In recent times, the proliferation of user activity on web-based social communities, microblogging sites, and other media such as Slashdot, Wikipedia, Facebook, Twitter, and Epinions (Leskovec et al. 2010; Kunegis et al. 2009) have offered a tremendous scope to mine not only data about the link structure among the users, but also the rich information present in the content of interactions among users. The data about the link structure among users is popularly known as *link data* and the data about user interactions is known as *interaction data*.[1] A practical example for the link data is whether user x appears in the friends list of user y in online social communities such as Facebook and that of the interaction data is whether user x sends a message to user y (e.g., sending a wall post in Facebook).

The major challenge here lies in dealing with the huge *volume* and a wide *variety* of datasets. While volume throws computational challenges, variety poses the challenge of data curation and information aggregation. The core idea of the sparsification task is to derive a backbone of the given social network by leveraging the link data and the interaction data (if it is available). At high level, our approach comprises four steps: (1) for each of the link data and the interaction data, we induce ranking of the neighboring nodes for every node. Higher ranked neighbor means stronger association between these two nodes; (2) since different data sources convey different kinds of information at varying magnitude, we compute the relative importance of these data sources after having derived the neighborhood rankings; (3) next, we aggregate different rankings of the neighborhood (along with their weights) induced by different data sources to arrive at a single aggregated ranking; (4) finally, we apply a sparsification trick to derive a sparse representation of the given graph by dropping edges that are less informative.

Influence maximization using sparse graphs For a given integer value for k, we determine the top-k initial seeds for the influence maximization problem by using the sparse graph obtained during the first task. To show the efficacy of the proposed approach, we then conduct thorough experimentation with several real-life social network datasets such as Digg, Epinions, Slashdot, Amazon, and DBLP.

1.2 Novelty of the paper

We believe that our contributions lead to the following novelty:

[1] Some authors call link data as *static data* and interaction data as *dynamic data* or *trace data*.

- Our approach is capable of handling volume as well as variety of the data while solving the influence maximization problem. Even in the absence of the interaction data, our approach also works with only the link data.
- The proposed approach is also very simple and extendable. At the core of our approach lies the notion of rank ordering the neighboring nodes for every node in a given graph. This rank ordering can be considered as a common currency for capturing the information contained in the link data and the interaction data. For any given new type of data source, one just needs to figure out the way to compute ranking of neighborhood and rest of the framework remains unchanged.
- The proposed method to determine the sparse graph of the given social network is independent of the information propagation model. The proposed approach can be considered as a pre-processor step for the given dataset. Once the sparse graph is computed from the social network, one can just ignore the huge volume of the heterogenous data sources and just focus on this extracted structure as far as the influence maximization problem is concerned.

Outline of the paper In Sect. 2, we present a brief review of the relevant work in the literature. In Sect. 3, we present the details of the graph sparsification task. Then we present the approach to determine the initial seeds for the influence maximization problem using the sparse graph in Sect. 4. We next present a thorough experimental validation of our approach in Sect. 5.

Note We also use the phrase *data sources* to refer to both the link data and the interaction data in the rest of the paper.

2 Relevant work

We first briefly mention the models for diffusion of information. There are two well-known operational models in the literature that capture the underlying dynamics of the information propagation in viral marketing. They are the linear threshold model (Schelling 1978; Granovetter 1978; Kempe et al. 2003) and the independent cascade model (Goldenberg et al. 2001; Kempe et al. 2003). We now present the most related work on the influence maximization problem in the literature and we categorize this relevant work into three major categories. In what follows, we discuss each of these three categories of work in detail.

Influence maximization with single product Domingos and Richardson (2001), Richardson and Domingos (2002) were the first to study the influence maximization problem as an algorithmic problem. They modeled social networks as Markov random fields where the probability of an individual adopting a technology (or buying a product) is a function of both the intrinsic value of the technology (or the product) to the individual and the influence of neighbors. The computational aspects of the influence maximization problem were investigated by Kempe et al. (2003). The authors show that the optimization problem of selecting the most influential nodes is NP-hard and derive the first provable approximation guarantees for the proposed algorithm. Recall that the objective function, $\sigma(.)$, for information diffusion is the expected number of nodes (i.e., $\sigma(S)$) that become active at the end of the diffusion process for a given set of initial active nodes (i.e., S). The authors first show that this objective function is a sub-modular function under both the linear threshold model and the independent cascade model. A function $g(.)$ is called sub-modular if it satisfies $g(S \cup \{i\}) - g(S) \geq g(T \cup \{i\}) - g(T)$ for all elements i and all pairs of sets $S \subseteq T \subseteq N$ where N is the set of nodes in the graph.

The authors (Kempe et al. 2003) then propose a greedy approximation algorithm for the influence maximization problem and show that this greedy algorithm achieves an approximation guarantee of $\left(1 - \frac{1}{e}\right)$ where $e = \sum_{r=1}^{\inf} \frac{1}{r!}$. We note that the running time of this greedy approximation algorithm (Kempe et al. 2003) is $O(knRm)$ as mentioned in Chen et al. (2009). Here R is the number of repetitions of each experiment, n is the number of nodes, and m is the number of edges in the graph.

Leskovec et al. (2007) proposed an efficient algorithm for the influence maximization problem based on the sub-modularity of the underlying objective function that scales to large problems and is reportedly 700 times faster than the greedy algorithm of Kempe et al. (2003). Chen et al. (2009) present an efficient algorithm to find the initial seeds in a social network and this algorithm improves upon the greedy algorithm of Kempe et al. (2003) and also the algorithm of Leskovec et al. (2007) in terms of its running time. The authors also design a new heuristic algorithm, which they call degree discount heuristic that achieves much better influence spread than classic degree and centrality-based heuristics. They also note that the performance of this heuristic algorithm is comparable to that of the greedy algorithm, while its running time is much less than that of the greedy algorithm.

Even-Dar and Shapira (2007) studied the influence maximization problem in the context of probabilistic voter model. Kimura and Saito (2006) proposed a shortest path-based influence cascade model and designed efficient algorithms for finding the most influential nodes. Ramasuri and Narahari (2011) and Chen et al. (2011) show a few advances in the literature in the context of the influence maximization problem. We note that the ideas similar to

those in the context of influence maximization problem are utilized to design immunization strategies in the context of virus propagation (Gao et al. 2011). Goyal et al. (2011) proposed a new approach that leverages the traces of past action propagations while solving the influence maximization problem.

Influence maximization with multiple products Datta et al. (2010) considered the influence maximization problem for *multiple independent products*. Specifically, given the social network and t products along with their seed requirements, we want to select seeds for each product that maximize the overall influence. The authors designed two efficient algorithms which they call as GREEDY and FAIRGREEDY. The GREEDY algorithm is based on the simple greedy hill climbing technique and results in a $\frac{1}{3}$approximation to the optimum. They designed efficient and scalable heuristics for the problem.

Viral marketing in the context of competing companies Another important branch of viral marketing is to study the algorithmic problem of how to introduce a new product into the market in the presence of a single or multiple competing products already in the market. Indeed, this problem has been the subject of interest in the literature (Bharathi et al. 2007; Borodin et al. 2010; Carnes et al. 2007; He et al. 2012; Budak et al. 2011) where the authors present competitive extensions for the linear threshold model and independent cascade model.

Our work in this paper is similar in spirit to that of Mathioudakis et al. (2011). The authors (Mathioudakis et al. 2011) proposed an efficient algorithm to determine the backbone of an influence network, given a social network and a log of past propagations. Our approach is very different from Mathioudakis et al. (2011) in two fundamental aspects:

- our work does not require any past propagations as in Mathioudakis et al. (2011) and
- our proposed approach is independent of the information propagation model, whereas the work (Mathioudakis et al. 2011) assumes the independent cascade (IC) model to be the underlying information propagation model.

3 Our proposed approach: graph sparsification

In this section, we describe our four-step approach to discover the sparse graph of the social network. We begin with defining the notion of the sparse graph.

Definition 1 (*Sparse graph*) Let $G = (V, E)$ be a given directed/undirected social network. The sparse graph of a social network is a subgraph $G' = (V, E')$ so that we have $|E'| \subset |E|$, and the results of the influence maximization that we perform on G' is a good approximation of the result of the influence maximization problem when performed on the original graph G.

Thus, to discover the sparse graph of a social network, we need to systematically delete a set of edges from the original given network. We identify this set of edges, i.e., $E \backslash E'$, in four steps—(1) ranking the neighborhood for each data source, (2) computing relative importance of rankings, (3) rank aggregation, and (4) graph sparsification. In what follows, we describe each of these steps in detail.

3.1 Ranking the neighborhood

For each data source D_s (link as well as interaction), we rank order the set of neighboring nodes, say $N(i)$, for each node $i \in V$. We denote this rank list by $R_i^s(\cdot)$ and the rank of a neighboring node j by $R_i^s(j)$. This rank list is a reflection of which edges are relatively more informative, as far as the corresponding data source is concerned. In other words, if j_1 and j_2 are two neighboring nodes for the node i and we have $R_i^s(j_1) < R_i^s(j_2)$, then it would mean that the edge (i, j_1) should be given priority over the edge (i, j_2) when it comes to removing an edge for the purpose of computing the sparse graph. Bear in mind that this recommendation is based on the data source D_s. For a different data source, say D_t, it could well be possible that we have $R_i^t(j_1) > R_i^t(j_2)$.

Now, the question is "how to generate such a ranked list for a given data source D_s and a node $i \in V$". As far as link data is concerned, it comprises a graph structure and the edge weights. For link data, one can choose any structural property, for example *centrality measure*, as a criterion to rank order the neighborhood. For example, if one uses betweenness centrality as a criterion, then the node $j \in N(i)$ having the highest value of betweenness centrality would acquire rank one. In our experiments, we have used 11 different such measures to induce 11 different rankings of neighborhood for each node $i \in V$.

When it comes to a interaction data source, one can induce a ranking of the neighborhood either by simply counting the frequency of the neighboring node appearing in the interaction data or by applying a sophisticated *learning to rank* method (Burges et al. 2005). However, the key idea behind any scheme is to assign higher ranking to a neighbor $j_1 \in N(i)$ as compared to the neighbor $j_2 \in N(i)$ if j_1 appears to be interacting relatively more with node i than j_2 with i as per the given interaction data.

It is crucial to mention that the ranking criterion for link as well as interaction data are typically independent of the influence maximization problem.

3.2 Computing the relative importance of rankings

Suppose, in the previous step, we generated k different ranked lists $R_i^1(\cdot), R_i^2(\cdot), \ldots, R_i^k(\cdot)$ for any node $i \in V$. In this step, we assess the relative importance of these ranked lists so that we can avoid introducing bias in the sparse graph of the social network by having considered ranked lists of complementary nature multiple times. This step is analogous to the feature selection step in learning tasks.

Like feature selection problem, there can be multiple ways of assigning the relative importance to the ranked list. However, we suggest one approach based on the idea of spectral clustering. For this, we define a distance $dist(R^s, R^t)$ between two different ranked lists $R^s(\cdot)$ and $R^t(\cdot)$ as follows.

$$dist(R^s, R^t) = \frac{1}{|V|} \sum_{i \in V} dist(R_i^s, R_i^t), \tag{1}$$

where $dist(R_i^s, R_i^t)$ is the distance between two different permutations of the nodes in the neighborhood $N(i)$ of the node i. There are several different ways for computing $dist(R_i^s, R_i^t)$, including *Spearman footrule* distance and *Kendall tau* distance (Dwork et al. 2001). We, in our experiments, used Kendall tau distance.

While there can be many ways to assign the relative importance to the ranked lists from a given matrix of pairwise distances, we highlight two such approaches for the sake of illustration. In the first approach, we use techniques of spectral clustering or graph cut to identify clusters of ranked lists where each cluster represents the set of similar ranked lists. We choose a centroid element as a representative element from each of these clusters and assign its relative importance as 1 and for others as 0. In the second approach, we construct a complete graph whose nodes represent ranked lists and whose edge weights represent the corresponding distance between the lists. For each node of this graph, we compute the closeness centrality. Closeness centrality contains information as to how close a particular node is to all the other nodes in the graph. Therefore, high closeness centrality will indicate that the node is quite close to other nodes and thus is not that unique. Therefore, we assign $[closeness\ centrality]^{-1}$ as its relative importance score.

3.3 Rank aggregation

In this step, we aggregate multiple ranked lists into one single ranked list to achieve an overall ordering of the neighborhood for every node. We also utilize the relative importance (aka weights) of the ranked list in the aggregation step. The rank aggregation is a well-researched topic in Social Choice Theory and AI communities. There are a large number of ways to perform rank aggregation. These methods differ in terms of their underlying objectives. For our purpose, we prefer to use the well-known *Borda rule* (Dwork et al. 2001), because it is simple, intuitive, and computationally easy to perform. Additionally, this rule can be augmented to take into account weights of the ranked lists as well.

As per Borda rule, if $R_i^1(\cdot), R_i^2(\cdot), \ldots, R_i^k(\cdot)$ are the k different rankings of the neighborhood for node $i \in V$, with the corresponding weights being $w_i^1, w_i^2, \ldots, w_i^k$, then the aggregated ranking $R_i(\cdot)$ is achieved by arranging them in a decreasing order of their *Borda score*. The Borda score of a neighboring node $j \in N(i)$ is given by

$$\alpha_i(j) = \sum_{t=1}^{k} w_i^t \, b\left(R_i^t(j)\right), \tag{2}$$

where $b\left(R_i^t(j)\right)$ is the Borda score function which assigns scores of $m-1, m-2, \ldots, 1, 0$ to the elements at the rank positions $1, 2, \ldots, m$, respectively, in any given ranked list of size m.

3.4 Graph sparsification

In this step, we leverage aggregated ranked ordering of the neighborhood $N(i)$ for the purpose of deciding which of the edges incident on node i can be removed to compute the sparse graph. We essentially retain certain fractions of edges incident on every node $i \in V$. Note, the aggregated ranked list $R_i(\cdot)$ represents the importance of every edge (i.e., neighboring node) incident on node i. Therefore, it would be apt to retain those edges that appear in the top part of the aggregated ranked list. For this, we appeal to the method suggested by Satuluri et al. (2011) where we retain $[deg(i)]^e$ number of top ranked edges (as per the list $R_i(\cdot)$) incident on node i. Here, $deg(i)$ denotes the degree of the vertex i and $0 \leq e \leq 1$. One can also use more sophisticated sampling-based techniques to select the edges.

4 Our proposed approach: influence maximization on sparse graphs

The second task of our proposed approach is to solve the influence maximization problem on the sparse graph. We measure the effectiveness of the initial seeds using the expected number of nodes in the social network that become active when we use these top k seeds as the initial active nodes (Kempe et al. 2003). In this paper, we refer to this as *reach*.

We work with both the LT model and the IC model. We consider two algorithms to determine the initial seeds for the influence maximization problem, namely the CELF

algorithm (Leskovec et al. 2007) and the degree discount heuristic (Chen et al. 2009). In particular, we deal with the following three configurations to address the influence maximization problem: (1) the CELF algorithm with LT model which we refer to as *CELF–LT*, (2) the CELF algorithm with IC model which we refer to as *CELF–IC*, and (3) degree discount heuristic with IC model which we refer to as *DDIC*. In each of these three configurations, for a given integer constant k, we first determine the top-k initial seeds using the sparse graph. Then we compute the reach of these top-k nodes on the original graph by running Monte Carlo simulations. We finally compare this value with the reach of the top-k initial seeds determined using the original graph itself.

5 Experiments

In this section, we conduct thorough experimentation of the proposed approach to reveal its novelty. We first describe the datasets and then present the experimental setup. We next present the experimental results.

5.1 Datasets

We conduct experiments on seven well-known network datasets. These datasets are HepTh (Newman 2001), Digg (Lerman and Ghosh 2010), FilmTip, Epinions (Richardson et al. 2003), Amazon (Yang and Leskovec 2012), DBLP (Yang and Leskovec 2012), Netscience (Newman 2006), and Slashdot (Leskovec et al. 2010). For the first four of these datasets, we do not have knowledge about the ground-truth community structure; and for the last three datasets, we do know the ground-truth community structure. Further, among these seven datasets, only Digg and FlimTip datasets have both link and interaction data. All the remaining five datasets have only link data. Table 1 provides a brief summary of these datasets and we now describe them as follows: (1) HepTh dataset is a co-authorship network of researchers in physics. (2) Digg dataset is a friendship network of users on Digg.com. (3) Epinions dataset relates to *who-trust-whom* type relationship among users of a general purpose consumer review site Epinions.com. (4) FilmTip dataset is a friendship network data. (5) Amazon dataset relates to the following feature of Amazon—*who bought this item also bought this*. In this kind of dataset, if a product i is frequently co-purchased with product j, the graph contains an undirected edge from i to j. Each product category defined by Amazon naturally forms a ground-truth community. (6) DBLP dataset is a co-authorship network of researchers in computer science. In this dataset, two authors are connected if they publish at least one paper together. We obtain the ground-truth community structure of this network based on the publication venue. That is, all the authors who published in a certain journal/conference can be considered as one community. (7) Netscience dataset is a co-authorship network of scientists working on network theory (Newman 2006). The vertices of the network represent authors of papers, and edges join every pair of individuals whose names appear together as authors of a paper. (8) Slashdot is a technology-related news website and it is known for its specific user community. In 2002, Slashdot introduced the Slashdot Zoo feature which allows users to tag each other as friends or foes.

Table 1 Datasets used in the experiments

Dataset	Nodes	Edges	Type
Digg	*68,634*	*1,242,544*	*Directed*
FilmTip	*39,581*	*72,312*	*Undirected*
Amazon	334,863	925,872	Undirected
DBLP	317,080	1,049,866	Undirected
Epinions	75,879	508,837	Directed
HepTh	10,748	52,293	Directed
NetScience	1589	2742	Directed
Slashdot	82,168	948,464	Directed

The datasets in italics are the ones that have the interaction data as well

5.2 Criteria for ranking neighbors

In our experiments, as far as the link data is concerned, we use the following 12 criteria to rank the neighbors of each node in the network. We denote these 12 ranking criteria as *static criteria*. These criteria are applicable only for those datasets where links are directed (e.g., in the case of HepTh, Epinion, and Digg datasets). For an undirected dataset, different subsets of these criteria collapse to one single criterion and hence the remaining number of different criteria becomes 6 (as described later). For a directed network, we adhere to the following convention when it comes to using these criteria: *if there is an edge from node a to node b, we say that node a is a follower of node b and node b is a friend of node a*. For undirected networks, if there is an edge between nodes a and b, it just means that nodes a and b are neighbors. For any node i, $d_{in}(i)$ and $d_{out}(i)$ represent its indegree and outdegree, respectively.

We start with defining the first five criteria that are based on local graph structure for any node.

1. Average indegree: It is the arithmetic mean of the number of incoming edges for each a and b, given by $[d_{in}(a) + d_{in}(b)]/2$.

2. Average outdegree: It is the arithmetic mean of the number of outgoing edges for each a and b, given by $[d_{out}(a) + d_{out}(b)]/2$.

3. Common followers: It is the number of common followers of a and b, given by $|d_{in}(a) \cap d_{in}(b)|$.

4. Common friends: It is the number of common friends of a and b, given by $|d_{out}(a) \cap d_{out}(b)|$.

5. Common neighbors: It is the number of common neighbors of a and b, given by $|(d_{out}(a) \cap d_{out}(b)) \cup (d_{in}(a) \cap d_{in}(b))|$.

We now introduce a few criteria based on network proximity measures (Lermann et al. 2012). These criteria measure the likelihood of a message originating from node a reaching node b regardless of the path it takes.

6. Common neighbors metric: Let us define $C = d_{out}(a) \cap d_{in}(b)$ and $C' = d_{in}(a) \cap d_{out}(b)$. Now, the common neighbors metric is the average number of nodes in C and C'. That is, $0.5|C| + 0.5|C'|$.

7. Jaccard coefficient: $0.5\frac{|d_{out}(a)\cap d_{in}(b)|}{|d_{out}(a)\cup d_{in}(b)|} + 0.5\frac{|d_{out}(b)\cap d_{in}(a)|}{|d_{out}(b)\cup d_{in}(a)|}$.

8. Adamic Adar score: $0.5\sum_{z\in C}[\log d(z)]^{-1} + 0.5\sum_{z'\in C'}[\log d(z')]^{-1}$.

We next consider a few more criteria where the proximity score is dependent not only on the neighborhood overlap, but also on different types of interactions that can occur between network nodes (Lerman and Ghosh 2010).

9. Conservative metric: $0.5\sum_{z\in C}\frac{1}{d_{out}(a)d_{out}(z)} + 0.5\sum_{z'\in C'}\frac{1}{d_{out}(b)d_{out}(z)}$.

10. Conservative attention limited metric: $0.5\sum_{z\in C}\frac{1}{d_{out}(a)d_{in}(z)d_{out}(z)d_{in}(b)} + 0.5\sum_{z\in C}\frac{1}{d_{out}(b)d_{in}(z)d_{out}(z)d_{in}(a)}$.

11. Non-conservative metric: $0.5|C| + 0.5|C'|$.

12. Non-conservative attention limited metric: $0.5\sum_{z\in C}\frac{1}{d_{in}(z)d_{in}(b)} + 0.5\sum_{z\in C'}\frac{1}{d_{in}(z)d_{in}(a)}$.

For undirected networks, it is easy to verify that some of the above criteria would produce the same ranking of neighbors. Therefore, for undirected networks, these 12 criteria would boil down to following 6 criteria: (1) average degree, (2) common neighbors mMetric, (3) Jaccard coefficient, (4) Adamic Adar score, (5) conservative metric, and (6) conservative attention limited metric.

As far as the interaction data are concerned, we also consider a few other criteria and refer to them as dynamic criterion. Recall that only Digg and FilmTip datasets have interaction data. For Digg dataset, we use the method proposed in Goyal et al. (2010). We use the Jaccard coefficient metric $\frac{A_{a2b}}{A_{a|b}}$ where A_{a2b} are the actions (here votes on stories) performed by user a and seen by user b, and $A_{a|b}$ are unique actions performed by either a or b. Similarly, for the FilmTip dataset, we work with three dynamic criteria as follows: (1) it is the jaccard coefficient between the movies rated by user a and b. It is denoted by $\frac{M_a\cap M_b}{M_a\cup M_b}$, where M_a is the set of movies rated by user a and M_b is the set of movies rated by b; (2) we divide the movie ratings into three categories, namely bad (ones with score of 1 or 2), neutral (those with score 3) and good (ones with the score of 4 or 5). The metric is denoted by the number of movies rated in the same category by both of the users; and (3) the third criterion measures the exact similarity of users. It is denoted by the number of movies rated exactly the same by both of the users.

5.3 Experimental setup

We conducted all our experiments on a PC with Linux OS, 3.05 GB of RAM and Intel Core i7 CPU Q720 1.2 GHz. As already mentioned earlier, we consider various algorithms and heuristics to determine the initial seeds and the various datasets to work with. Table 2 shows the list of algorithms that we ran on different datasets. For every instance of the IC model, we consider a constant probability $p = 0.01$ except for for NetScience dataset where we set $p = 0.25$.

5.4 Experimental results

We categorize the experimental results into various subsections as follows. We first present the results obtained using the CELF algorithm and then those obtained using the degree discount heuristic.

CELF Algorithm We ran the CELF algorithm with both LT model and IC model on three datasets, namely FilmTip, HepTh, and Netscience. Figure 1 shows the reach obtained using the CELF with LT model on Film-

Table 2 Listing of dataset and algorithm (or heuristic) pair used to determine the initial seeds

Dataset	DDIC	CELF–IC	CELF–LT
Digg	✔	×	×
FilmTip	✔	✔	✔
Amazon	✔	×	×
DBLP	✔	×	×
Epinions	✔	×	×
HepTh	✔	✔	✔
NetScience	✔	✔	✔
Slashdot	✔	×	×

Tip, HepTh and Netscience datasets. For the IC model, we set $p = 0.01$ for FilmTip and HepTh datasets and we set $p = 0.25$ for Netscience dataset. Similarly, Fig. 2 shows the reach obtained using the CELF with IC model on FilmTip, HepTh and Netscience datasets. The reach of the initial seeds using the CELF with both LT and IC models computed with the sparse graphs is almost similar to that of the original graph.

Results based on DDIC Since applying greedy algorithm on larger datasets takes a long time, we used the degree discount heuristic on larger datasets to measure the reach of the initial seeds computed using the sparse graphs. The results of the degree discount heuristic with independent cascade probability 0.01 on DBLP, Amazon, Epinions and FilmTip are shown in Fig. 3. We plot only reach of the seeds for the sparse files with $e = 0.1$, $e = 0.5$, and $e = 0.9$, and the rest are omitted in the interest of space constraints. Similar results were obtained even for other datasets. From the figure we can see that the seed quality obtained using sparse graphs is equivalent and in some cases even outperforms the original seeds. On Amazon and FilmTip, even by using just 34.23 % and 39.56 % edges ($e = 0.1$), respectively, we obtain a solution that is matchable with the original.

Analysis of running time Previous results show the efficacy of the initial seeds computed using the sparse graphs in terms of *reach*. We now compare and contrast the running time taken by the CELF algorithm on the sparse graphs and the original graph, respectively, for each dataset to determine the top-100 initial seeds. Figure 4 shows the running of the CELF algorithm with the LT model on FilmTip,

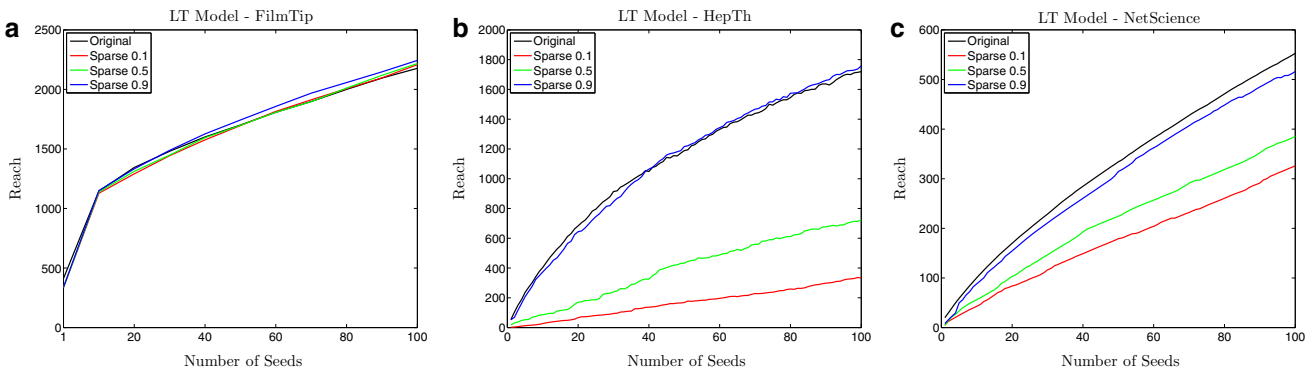

Fig. 1 Reach of the initial seeds obtained using CELF with LT model on **a** FilmTip, **b** HepTh, **c** NetScience

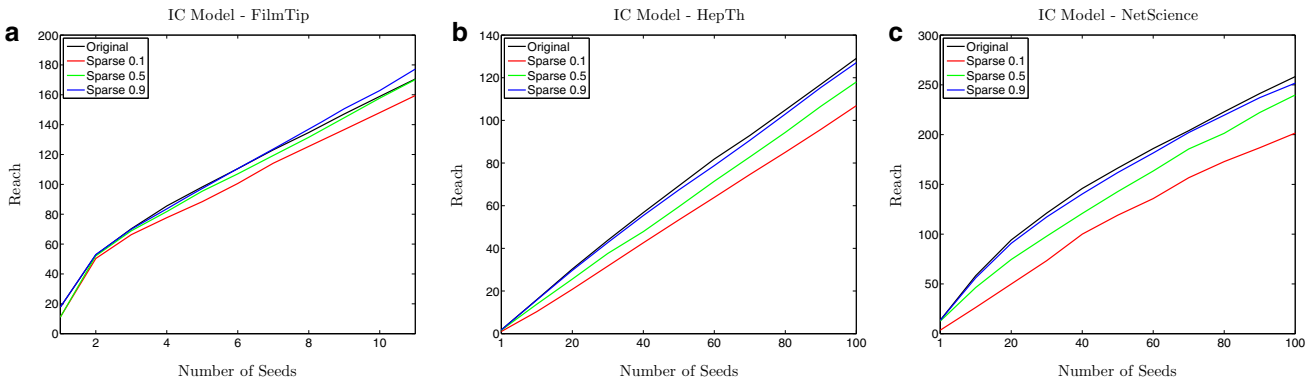

Fig. 2 Reach of the initial seeds obtained using the CELF with the IC model on **a** FilmTip, **b** HepTh, **c** NetScience

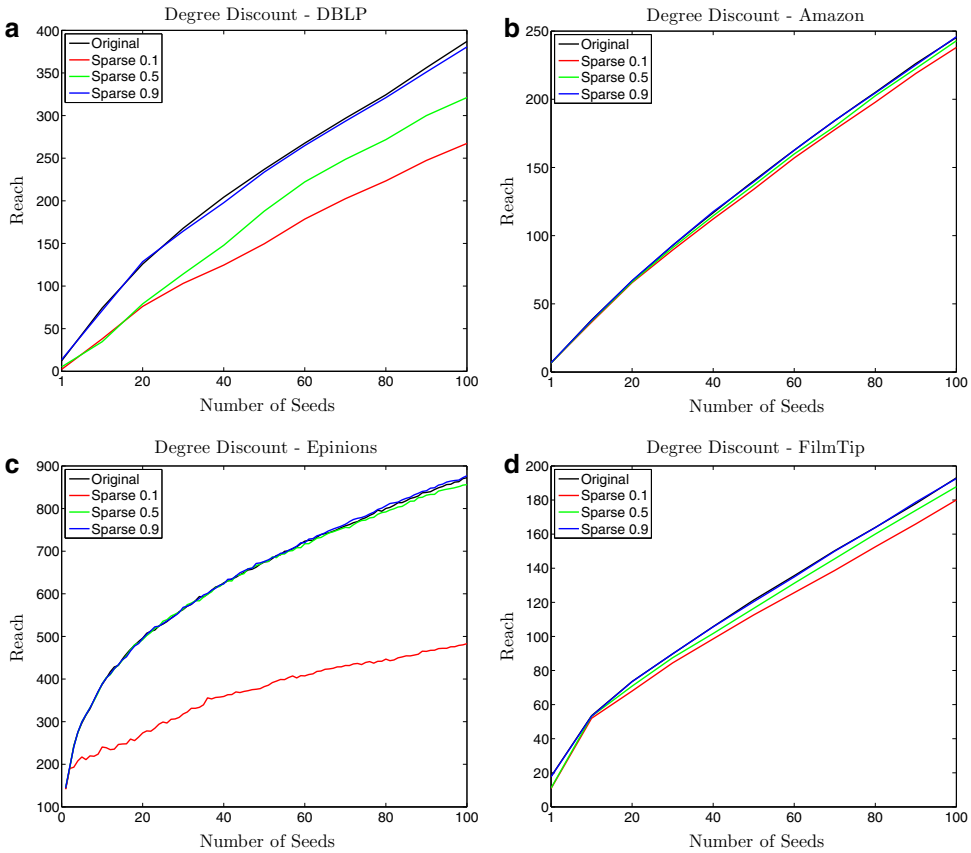

Fig. 3 Degree discount results: **a** DBLP, **b** Amazon, **c** Epinions, **d** FilmTip

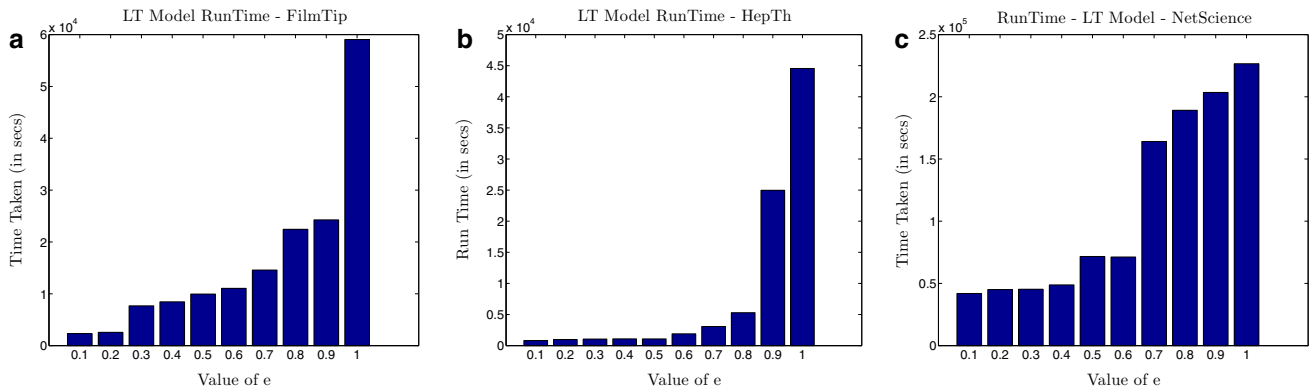

Fig. 4 LT Model time analysis: **a** FilmTip, **b** HepTh, **c** NetScience

HepTh, and Netscience datasets to determine the top-100 initial seeds. Similarly, Fig. 5 shows the running of the CELF algorithm with IC model on FilmTip, HepTh, and Netscience datasets to determine the top-100 initial seeds.

From these results, it is easy to see that the time taken by sparse graphs to determine the top-100 initial seeds is much lesser than the time taken by the original graph. In other words, the proposed method to determine the sparse graphs essentially retains very rich information as far as the original graph is concerned while dropping a significant number of edges. These experimental results reveal that the sparse graphs obtained using the proposed approach not

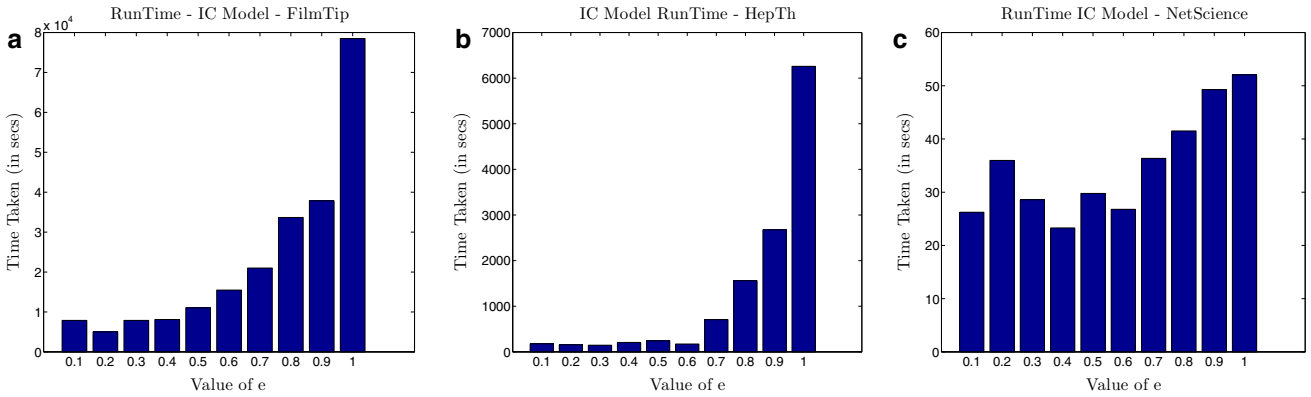

Fig. 5 IC Model time analysis: **a** FilmTip, **b** HepTh, **c** NetScience

only produce the initial seeds with high quality, but also lead to significant speed-up in terms of computation time.

References

Bharathi S, Kempe D, Salek M (2007) Competitive influence maximization in social networks. In: Proceedings of the 3rd workshop on internet and network economics (WINE), pp 306–311

Borodin A, Filmus Y, Oren J (2010) Threshold models for competitive influence in social networks. In: Proceedings of the 6th workshop on internet and network economics (WINE), pp 539–550

Budak C, Agrawal D, Abbadi AE (2011) Limiting the spread of misinformation in social networks. In: Proceedings of the 20th international conference on world wide web (WWW), pp 665–674

Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G (2005) Learning to rank using gradient descent. In: Proceedings of the 22nd ICML, pp 89–96

Carnes T, Nagarajan C, Wild S, van Zuylen A (2007) Maximizing in uence in a competitive social network: a follower's perspective. In: Proceedings of the 9th international conference on electronic commerce (ICEC), pp 351–360

Cornuejols G, Fisher M, Nemhauser G (1977) Location of bank accounts to optimize oat: an analytic study of exact and approximate algorithms. Manag Sci 23:789–810

Chen W, Collins A, Cummings R, Ke T, Liu Z, Rincon D, Sun X, Wang Y, Wei W, Yuan Y (2011) Influence maximization in social networks when negative opinions may emerge and propagate. In: Proceedings of SIAM SDM

Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 937–944

Datta S, Majumder A, Shrivastava N (2010) Viral marketing for multiple products. In: Proceedings of IEEE ICDM, pp 118–127

Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of the 7th SIGKDD international conference on knowledge discovery and data mining (KDD), pp 57–66

Dwork C, Kumar R, Naor M, Sivakumar D (2001) Rank aggregation methods for the web. In: Proceedings of WWW, pp 613–622

Even-Dar E, Shapira A (2007) A note on maximizing the spread of influence in social networks. In: Proceedings of the 3rd workshop on internet and network economics (WINE), pp 281–286

Gao C, Liu J, Zhong N (2011) Network immunization and virus propagation in email networks: experimental evaluation and analysis. Knowl Inf Syst 27(2):253–279

Goldenberg J, Libai B, Muller E (2001) Talk of the network: a complex systems look at the underlying process of word-of-mouth. Mark Lett 12(3):211–223

Goyal A, Bonchi F, Lakshmanan LVS (2010) Learning influence probabilities in social networks. In: WSDM, pp 241–250

Goyal A, Bonchi F, Lakshmanan LVS (2011) A data-based approach to social influence maximization. In: PVLDB, pp 73–84

Granovetter M (1978) Threshold models of collective behavior. Am J Sociol 83:1420–1443

He X, Song G, Chen W, Jiang Q (2012) Influence blocking maximization in social networks under the competitive linear threshold model. In: Proceedings of the 12th SIAM international conference on data mining (SDM)

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: Proceedings of the 9th SIGKDD international conference on knowledge discovery and data mining (KDD), pp 137–146

Kimura M, Saito K (2006) Tractable modles for information diffusion in social networks. In: Proceedings of 10th European conference on principles and practice of knowledge discovery in databases (PKDD), pp 259–271

Kunegis J, Lommatzsch A, Bauckhage C (2009) The slashdot zoo: Mining a social network with negative edges. In: Proceedings of 18th WWW, pp 740–750

Lerman K, Ghosh R (2010) Information contagion: an empirical study of spread of news on Digg and Twitter social networks. In: Proceedings of 4th international conference on weblogs and social media (ICWSM)

Lermann K, Intagorn S, Kang JH, Ghosh R (2012) Using proximity to predict activity in social networks. In: Proceedings of the 21st international world wide web conference (poster)

Leskovec J, Huttenlocher D, Kleinberg J (2010) Signed networks in social media. In: Proceedings of the 28th ACM SIGCHI conference on human factors in computing systems (CHI), pp 1361–1370

Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N (2007) Cost-effective outbreak detection in networks. In: Proceedings of the 13th SIGKDD international conference on knowledge discovery and data mining (KDD), pp 420–429

Mathioudakis M, Bonchi F, Castillo C, Gionis A, Ukkonen A (2011) Sparsification of influence networks. In: Proceedings of the 17th SIGKDD international conference on knowledge discovery and data mining (KDD)

Newman MEJ (2001) The structure of scientific collaboration networks. Proc Natl Acad Sci 98:404–409

Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. Phys Rev E 74(3):036104

Newman MEJ (2009) The structure of scientific collaboration networks. Proc Natl Acad Sci 98:404–409

Ramasuri N, Narahari Y (2011) A shapley value based approach to discover influential nodes in social networks. IEEE Trans Autom Sci Eng 8(1):130–147

Richardson M, Agrawal R, Domingos P (2003) Trust management for the semantic web. In: Proceedings of ISWC

Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th SIGKDD international conference on knowledge discovery and data mining (KDD), pp 61–70

Satuluri S, Parthasarathy V, Ruan Y (2011) Local graph sparsification for scalable clustering. In: Proceedings of SIGMOD, pp 721–732

Schelling T (1978) Micromotives and macrobehavior. W.W Norton and Company, New York

Yang J, Leskovec J (2012) Defining and evaluating network communities based on ground-truth. In: Proceedings of ICDM