Influence Maximization over Large-Scale Social Networks: A Bounded Linear Approach

Qi Liu[1], Biao Xiang[2], Enhong Chen[1]*, Hui Xiong[3], Fangshuang Tang[1], Jeffrey Xu Yu[4]

[1]School of Computer Science, University of Science and Technology of China
[2]Search Technology Center Asia (STCA), Microsoft
[3] Rutgers Business School, Rutgers University
[4] Department of SEEM, Chinese University of Hong Kong
[1]{qiliuql,cheneh}@ustc.edu.cn, fstang@mail.ustc.edu.cn, [2]bixian@microsoft.com, [3]hxiong@rutgers.edu, [4]yu@se.cuhk.edu.hk

ABSTRACT

Information diffusion in social networks is emerging as a promising solution to successful viral marketing, which relies on the effective and efficient identification of a set of nodes with the maximal social influence. While there are tremendous efforts on the development of social influence models and algorithms for social influence maximization, limited progress has been made in terms of designing both efficient and effective algorithms for finding a set of nodes with the maximal social influence. To this end, in this paper, we provide a bounded linear approach for influence computation and influence maximization. Specifically, we first adopt a linear and tractable approach to describe the influence propagation. Then, we develop a quantitative metric, named Group-PageRank, to quickly estimate the upper bound of the social influence based on this linear approach. More importantly, we provide two algorithms *Linear* and *Bound*, which exploit the linear approach and Group-PageRank for social influence maximization. Finally, extensive experimental results demonstrate that (a) the adopted linear approach has a close relationship with traditional models and Group-PageRank provides a good estimation of social influence; (b) *Linear* and *Bound* can quickly find a set of the most influential nodes and both of them are scalable for large-scale social networks.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; H.2.8 [**Database Management**]: Database Application—*Data Mining*

General Terms

Algorithms, Experimentation

Keywords

Social Influence; Linear Approach; Bound; Viral Marketing

*Contact Author.

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2662009 .

1. INTRODUCTION

The diffusion of influence in social networks has provided opportunities for viral marketing, which aims at finding a set of individuals in the network to maximize the word-of-mouth propagation of a brand [24]. In general, there are two challenges for the successful viral marketing in social networks. First, how to model the influence diffusion process in the network? Second, how to design an efficient algorithm to identify which set of nodes to target in the network based on the learned diffusion models [7]?

In the literature, many efforts have been made on the development of influence propagation models for social influence maximization. For instance, both the Independent Cascade (IC) model [14] and the Linear Threshold (LT) model [20] were developed to model the influence propagation. While there are high expectations on efficient and scalable influence computing, limited progress has been made in terms of designing both efficient and tractable algorithms for finding a set of nodes with the maximal social influence. One of the main reasons is that the existing influence models, such as the IC and LT models, usually require to run Monte-Carlo simulation for a significant number of times before the nodes' influence can be computed. This is very time-consuming [10] and is not scalable for large-scale social networks. As a result, some computationally efficient heuristic algorithms based on existing influence models, such as DegreeDiscountIC [10] and PMIA [9], have been proposed to solve the social influence maximization problem for viral marketing. However, as a tradeoff, these heuristic approaches usually sacrifice the effectiveness.

To this end, we provide a bounded linear approach for effective and efficient influence computation and influence maximization. Specifically, our method is based on a linear approach which was preliminarily proposed in [41] for describing the social influence propagation. The unique perspective is that this linear approach assumes the influence flowing into each node is a linear combination of the influence from its neighbors. Therefore, the influence of an arbitrary node set can be linearly computed in a closed form. For leveraging this linear influence approach to the task of social influence maximization, in this paper we first define a quantitative metric, called Group-PageRank. Unlike traditional PageRank algorithm which can only be used to compute the influence of an individual node, Group-PageRank could estimate the influence strength between any node sets in nearly constant time. We show that Group-PageRank is essentially an upper bound of the influence spread under the linear approach. Then, based on linear approach and Group-PageRank, we design two greedy algorithms for viral marketing campaign, *Linear* (based on the original linear approach) and *Bound* (based on Group-PageRank), which can

efficiently find the node set with maximal social influence. Finally, we perform extensive experiments on real-world social network datasets. The experimental results show that: (a) The adopted linear approach could be used to approximate two traditional influence models, i.e., the IC model [14] and the Stochastic model [2], and Group-PageRank is a good estimation of social influence under the linear approach; (b) For social influence maximization, *Linear* and *Bound* can find influential nodes in an efficient way. Actually, *Linear* is more effective while *Bound* is more efficient.

In summary, the main contributions of this paper are as follows.

1. We discover an upper bound, named Group-PageRank (which can be quickly computed), for the social influence estimation under the linear approach. In the experiments, we show that the influence output by Group-PageRank is closely related to the true influence computed by the linear approach.
2. We design two greedy algorithms, *Linear* and *Bound*, by exploiting the properties of the linear influence approach and Group-PageRank, for the social influence maximization problem. In terms of efficiency as well as effectiveness, *Linear* and *Bound* outperform several state-of-the-art algorithms, and are scalable for large-scale social networks.
3. We experimentally show that the linear influence modeling approach adopted in this paper has similar capability as the traditional models (e.g., IC model) for describing the influence propagation in social networks. However, the linear approach is more efficient.

2. RELATED WORK

We discuss the related works on social influence analysis models and the existing strategies for social influence maximization.

Social Influence Models. Social influence modeling has been widely studied in the literature. Some work focuses on inferring the influence probabilities between nodes [17, 36]. For instance, Anagnostopoulos et al. [4] proved the existence of social influence by statistical tests. Gomez-Rodriguez et al. [15] tried to reconstruct the network over which the influence propagates. In addition, there are several models to describe the entire propagation process. For instance, Granovetter et al. [20] proposed the Linear Threshold (LT) model, while Goldenberg et al. [14] proposed the Independent Cascade (IC) model. Let's use IC model for explanation. Under IC model, in each iteration, the activated/influenced nodes have a single chance to influence their neighbors independently with a certain probability. This iterative propagation process will not stop until there is no newly influenced node in an iteration. The IC model with each link sharing the same propagation probability is called the Uniform IC Model, and the one with non-uniform edge weights is called the Weighted Cascade (WC) Model [24].

Both IC model and LT model are descriptive models, and we usually have to run the Monte-Carlo simulation for sufficiently many (e.g., 20,000) times to estimate the nodes' influence. This is very time-consuming and not applicable to large-scale social networks. Thus, Aggarwal et al. [2] proposed a stochastic model to address this scalability issue. Meanwhile, Kimura et al. [25] proposed Shortest-Path model (SPM), Zhang et al. [46] designed probabilistic solutions and Yang et al. [43] designed Gauss-Seidel (GS) algorithm to approximate the influence spread under the IC model with some specific constraints, e.g., the propagation probabilities should be very small. In another direction, PageRank [34] and random walk related algorithms [30, 41], which are quite efficient, have also been proposed for modeling influence propagation and ranking nodes [38]. For instance, by connecting PageRank based methods with the existing social influence analysis, Xiang et al. [41] proposed a linear social influence modeling approach, which could be

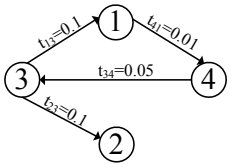


Figure 1: A social network example.

viewed as PageRank with priors by fixing a prior probability α_i to describe with how much probability the node i spreads influence to its neighbors. Here, the prior α_i functions like a supervised label [5] for label propagation and node classifications [48] in the graph. Actually, the defined influence propagation process is similar to that of the Credit Distribution model [18]. However, the further relationship between linear approach and the traditional influence models (e.g., the IC model) is still not carefully studied. Meanwhile, how to use this linear approach for the social influence maximization problem is also underexplored.

Recently, Easley et al. [13], Aggarwal et al. [1] and Chen et al. [7] summarized and generalized many research aspects of social networks. More importantly, they demonstrated that social influence can be further leveraged to deal with some of the real-world application problems (those from online marketing or social security). For instance, exploiting social influence to make better node ranking [30] and more accurate link recommendation [42].

Social Influence Maximization. Among these applications, viral marketing campaign is an important research branch. This application is usually formalized as the social influence maximization problem [7], targeting at finding a small set of influential individuals (called seed nodes) from the network. By triggering a cascade of information propagation that people recommend the product to their friends and the friends' friends, we hope that the product will be adopted by the maximum number of individuals.

With the help of existing social influence models, there are many works which aim at solving this problem, and to the best of our knowledge, social influence maximization could be traced back to Domingoes and Richardson [12, 35]. Kempe et al. formulated it as a discrete optimization problem and they proved that the optimization problem is NP-hard, and presented a greedy approximation algorithm which guarantees that the influence spread result is within $(1 - 1/e) \approx 0.63$ of the optimal result [24]. To address the inefficiency issue, Leskovec et al. [27] presented a "Lazy Forward" scheme (called CELF optimization) which takes advantage of the submodular property of the influence maximization objective to reduce the number of evaluations on the influence spread of individuals. Recently, this scheme is further improved by the CELF++ optimization (exploiting the submodularity to avoid unnecessary re-computation of the marginal gains) [19] and the StaticGreedy algorithm (using snapshots to avoid huge number of Monte-Carlo simulations) [11]. To address the scalability issue, Chen et al. proposed several heuristic methods, including DegreeDiscountIC [10] and PMIA [9], to approximate the social influence propagation using local arborescence structures of each individual. Wang et al. [40] presented a community-based greedy algorithm to find the top-K influential nodes from the selected potential communities. Jung et al. [23] proposed the IRIE algorithm which integrates the advantages of influence ranking (IR) and influence estimation (IE) methods for influence maximization. Similar to our work, Zhou et al. found an upper bound for the influence spread function under IC model [47]. However, their method still requires a number of Monte-Carlo simulations for choosing seed sets.

In summary, a common theme behind the above heuristics is that they avoid Monte-Carlo simulations by exploiting specific aspects of the graph structure and the social influence model to significantly speed up the influence computations [7]. In addition, many researchers also consider some constraints in practice. For instance,

Lee et al. [26], Chen et al. [8], and Goyal et al. [16] all included time constraints into their approximation algorithms. Guo et al. studied the problem of finding the top-K most influential nodes to the target user [21] and Tang et al. [37, 32] modeled social influence at the topic level. Bharathi et al. [6], Wang et al. [39] and Li et al. [31] focused on the influence diffusion or maximization on the competitive, heterogeneous and signed networks, respectively. One step further, Yu et al. [44] studied the way of parallelizing the social influence maximization computation and Liu et al. [33] tried to figure out the "independent influence" of each selected seed. Some of the general techniques and issues with respect to social influence maximization problem were discussed by Chen, Lakshmanan, and Castillo in Chapter 3 of Ref. [7].

Table 1: Several important mathematical notations.

Notations	Description
$f_{\mathcal{S}\to j}$	influence from node set $\mathcal{S}$ to j, j-th entry of $\mathbf{f}_{\mathcal{S}}$
d_j	damping factor for node j
$f_{\mathcal{S}\to\mathcal{T}}$	total influence from $\mathcal{S}$ to the nodes in set $\mathcal{T}$
t_{ij}	transition probability of i on j, (i, j)-th entry of $\mathbf{T}$
fPR_i	influence-PageRank value of node i, i-th entry of $\mathbf{fPR}$
$\mathbf{P}$	$(\mathbf{I} - d\mathbf{T}')^{-1}$, where $\mathbf{I}$ is the identity matrix
$\mathbf{h}_{\mathcal{S}}$	an auxiliary vector satisfying $\mathbf{f}_{\mathcal{S}} = (\mathbf{I} - d\mathbf{T}')^{-1}\mathbf{h}_{\mathcal{S}}$
$f^{A}_{\mathcal{S}\to i}$	the probability of node i being activated under model A (e.g., IC, Linear) when $\mathcal{S}$ is the seed set
$\mathbf{e}_{\mathcal{T}}$	a vector with $e_i = 1$ if $i \in \mathcal{T}$ and $e_i = 0$ otherwise
$GPR(\mathcal{S}, \mathcal{T})$	Group-PageRank value from $\mathcal{S}$ to $\mathcal{T}$
$\Delta_s(\mathcal{S}, \mathcal{T})$	marginal influence increment of $\mathcal{S}$ on $\mathcal{T}$ when adding s to $\mathcal{S}$

3. SOCIAL INFLUENCE COMPUTING

We first present the preliminaries of the linear approach for modeling influence propagations [41]. Since the influence computation is still not efficient enough for the task of social influence maximization, we find an upper bound, called Group-PageRank, to quickly estimate the social influence between any node sets. For better illustration, Table 1 lists some mathematical notations.

3.1 Preliminaries of the Linear Approach

As already mentioned, traditional influence propagation models are usually descriptive and we have to run Monte-Carlo simulation for a significant number of times before the nodes' influence can be computed. This is very time-consuming, especially for the task of social influence maximization, as we have to compute the social influence for many different candidate sets. Thus, we refer to a linear approach for efficient social influence computation [41]. Indeed, to show the rationality of adopting the linear approach, the evidence of the close relationship between this approach and two of the existing models is included in the following experiments.

We could model a social network as a graph $G = (\mathcal{V}, \mathcal{A}, \mathbf{T})$. Here, $\mathcal{V} = \{1, 2, ..., n\}$ is the set of nodes, $\mathcal{A}$ is the set of edges, and $\mathbf{T} = [t_{ij}]$ is the influence transition probability from node i to node j, where t_{ij} is a non-zero value, $0 < t_{ij} \leq 1$ if $(j, i) \in \mathcal{A}$ and 0 otherwise. An example $G = (\mathcal{V}, \mathcal{A}, \mathbf{T})$ is given in Fig. 1, where $\mathcal{V} = \{1, 2, 3, 4\}$, $\mathcal{A} = \{(1, 4), (3, 1), (3, 2), (4, 3)\}$, and $\mathbf{T} = [[0, 0, 0.1, 0], [0, 0, 0.1, 0], [0, 0, 0, 0.05], [0.01, 0, 0, 0]]$ in its matrix representation. The transition probabilities could be pre-learned [17]. Thus, we assume matrix $\mathbf{T}$ is known and $\sum_{i=1}^{n} t_{ij} \leq 1$ [43].

Given a social network G, let's consider a non-empty set of nodes, $\mathcal{S}(\subseteq \mathcal{V})$, and call it the influencer-set. Then we show how to compute the expected social influence of $\mathcal{S}$ by the linear influence modeling approach. Actually, it follows two assumptions given in the literature [1, 14, 18, 20]: (1) A node in the influencer-set $\mathcal{S}$ has the 100% probability to be influenced by $\mathcal{S}$ itself, [1] i.e., each node is on the same line with others in $\mathcal{S}$; (2) The probability that a node not in $\mathcal{S}$ gets influenced depends on its neighbors' influence. Based on the two assumptions, we denote $f_{\mathcal{S}\to j}$ as the final value that a node $j \in \mathcal{V}$ is influenced by the influencer-set $\mathcal{S}$.

[1]This assumption is a special case of that in Ref. [41], i.e., fixing the prior probability α_i=1, since it is easy to be accepted and the linear approach performs well under this setting.

Definition 1. *The influence of the influencer-set $\mathcal{S}$ on a specific node j in G, denoted by $f_{\mathcal{S}\to j}$, is defined below.*

$$f_{\mathcal{S}\to j} = 1, \quad for\ j \in \mathcal{S}, \tag{1}$$

$$f_{\mathcal{S}\to j} = d_j \Sigma_{(j,k)\in\mathcal{A}} t_{kj} f_{\mathcal{S}\to k}, \quad for\ j \notin \mathcal{S}. \tag{2}$$

Here, d_j is the damping factor of j for influence propagation, in the range of $(0, 1)$.

The unique feature of this approach is that the influence to a node j is a linear combination of the influence coming from j's neighbors if $j \notin \mathcal{S}$ (Eq. (2)). It leads to linear efficient iterative algorithms in computing influence propagation. For the damping factor d_j, the smaller d_j is, the more influence will be blocked by node j. For simplicity, the same d_j is chosen for each node [34].

Meanwhile, the influencer-set $\mathcal{S}$ can influence a specific subset of nodes $\mathcal{T}$ ($\subset G$). We denote the influence from $\mathcal{S}$ to $\mathcal{T}$ as $f_{\mathcal{S}\to\mathcal{T}}$, and it is computed as follows.

$$f_{\mathcal{S}\to\mathcal{T}} = \sum_{j\in\mathcal{T}} f_{\mathcal{S}\to j}. \tag{3}$$

By Eq. (3), the influence from $\mathcal{S}$ to $\mathcal{T}$ is the total influence from $\mathcal{S}$ to each node in $\mathcal{T}$. Here, $\mathcal{S}$ can be a single node in G, and $\mathcal{T}$ can be the entire node set of G, i.e., $\mathcal{V}$. In the following, we use $\mathbf{f}_{\mathcal{S}} = [f_{\mathcal{S}\to 1}, f_{\mathcal{S}\to 2}, ... f_{\mathcal{S}\to n}]'$ to denote the influence spread of $\mathcal{S}$. The computation of $f_{\mathcal{S}\to i}$ can be finished efficiently as shown in [41]. Specifically, for node $j \in \mathcal{S}$, $f_{\mathcal{S}\to j} = 1$. Then, for node $j \notin \mathcal{S}$, the influence $f_{\mathcal{S}\to j}$ can be computed iteratively, e.g., $f_{\mathcal{S}\to j}$ in the $(t+1)$-th iteration is $f^{(t+1)}_{\mathcal{S}\to j} = d_j \Sigma_{k=1}^{n} t_{kj} f^{(t)}_{\mathcal{S}\to k}$, and $f^{(t+1)}_{\mathcal{S}\to j}$ will converge to its final solution $f_{\mathcal{S}\to j}$ quickly under the following condition.

$$d_j \leq \frac{1}{\sum_{k=1}^{n} t_{kj}}, \quad for\ each\ node\ j. \tag{4}$$

It is worth noting that Eq. (4) is the Gauss-Seidel convergence condition for Eq. (2). Since $\sum_{k=1}^{n} t_{kj} \leq 1$ [43], Eq. (4) always holds. As a result, the influence of $\mathcal{S}$ can be solved in $O(|\mathcal{A}|)$ time.

3.2 Group-PageRank

From Definition 1, we can see that linear approach is a random-walk-like model and this is similar to PageRank [34] (their differences will be shown later). Actually, Ref. [41] has connected these two types of models together and demonstrated that PageRank value could be used to form upper bounds for the influence of a single node under linear influence approach. However, that upper bound is not good enough for influence maximization. One step further, in this paper, we find another upper bound, called Group-PageRank, which can be viewed as the PageRank value of a set of nodes.

First, we explain why linear influence approach needs an upper bound. By Definition 1, when given $\mathcal{S}$, the influence computation for both $f_{\mathcal{S}\to\mathcal{T}}$ and $f_{\mathcal{S}\to j}$ could be done in linear time (in $O(|\mathcal{A}|)$), and thus significantly outperforms the traditional models. However, this linear approach still can not meet the demand when computing $f_{\mathcal{S}\to\mathcal{T}}$ for any possible influencer-set $\mathcal{S}$ and influencee-set $\mathcal{T}$ over a large-scale social network, e.g., with billions of nodes. As we have to spend $O(|\mathcal{A}|)$ time for each $\mathcal{S}$ and $\mathcal{T}$, and there may be too many $\mathcal{S}$ and $\mathcal{T}$ candidates (i.e., $2^{|\mathcal{V}|}$ for $\mathcal{S}$). Thus, to reduce the time complexity, we propose Group-PageRank.

Then, we introduce Group-PageRank via PageRank. Following [22, 34], PageRank (topic-sensitive PageRank) is computed by

$$PR_i = d \sum_{(j,i)\in\mathcal{A}} w_{ji} PR_j + \frac{(1-d)}{|\mathcal{T}|}\delta_i,$$

where $w_{ji} = \frac{weight(j,i)}{OutWeight(j)}$, $d \in (0,1)$, and $\delta_i = 1$ if $i \in \mathcal{T}$ and 0 otherwise. In terms of PageRank, to compute influence propagation in $G = (\mathcal{V}, \mathcal{A}, \mathbf{T})$, we need to replace the w_{ji} by t_{ij} for the transition probability on each edge (j,i) is t_{ij}, that is

$$fPR_i = d \sum_{j=1}^{n} t_{ij} fPR_j + \frac{(1-d)}{|\mathcal{T}|}\delta_i. \quad (5)$$

fPR_i indicates the node i's importance with respect to social influence and we call it the influence-PageRank of node i to distinguish it from other applications. Clearly, fPR_i for node i can be solved in $O(|\mathcal{A}|)$ time. Let $\mathbf{fPR} = [fPR_1, fPR_2, ..., fPR_n]'$. Eq. (5), for $i = 1, 2, ..., n$, becomes Eq. (6).

$$\mathbf{fPR} = \frac{1-d}{|\mathcal{T}|}(\mathbf{I} - d\mathbf{T})^{-1}\mathbf{e}_{\mathcal{T}}, \quad (6)$$

where $\mathbf{e}_{\mathcal{T}} = [e_1, e_2, ..., e_n]'$, and e_i is 1 if $i \in \mathcal{T}$ and 0 otherwise. Similar to PageRank [34], **fPR** could be solved in $O(|\mathcal{A}|)$ time.

Next, let's go back to the linear influence approach, and the Definition 1 can be represented by one single equation as below.

$$f_{\mathcal{S}\to j} = d \sum_{k=1}^{n} t_{kj} f_{\mathcal{S}\to k} + h_{\mathcal{S},j}, \quad for\ i = 1, 2, ..., n. \quad (7)$$

$h_{\mathcal{S},j}$ is equal to 0 if $j \notin \mathcal{S}$, and is a number to ensure $f_{\mathcal{S}\to j} = 1$, otherwise. By summarizing Eq. (7), for $i = 1, 2, ..., n$, we have

$$\mathbf{f}_{\mathcal{S}} = (\mathbf{I} - d\mathbf{T}')^{-1}\mathbf{h}_{\mathcal{S}}. \quad (8)$$

Now, considering Eq. (6) for influence-PageRank and Eq. (8) for linear influence approach, we could find that there are two matrices $(\mathbf{I} - d\mathbf{T})^{-1}$ and $(\mathbf{I} - d\mathbf{T}')^{-1}$, which are transposes to each other. Besides this, the significant difference is that no entry of **fPR** is given before running Eq. (6) (i.e., PageRank), while in linear approach (Eq. (8)) some of the values (for those nodes belonging to $\mathcal{S}$) of $\mathbf{f}_{\mathcal{S}}$ are fixed as the priors (i.e., 1 in this paper). In summary, with a given matrix $(\mathbf{I} - d\mathbf{T})^{-1}$ and a given vector $\mathbf{e}_{\mathcal{T}}$, influence-PageRank will output the value in each entry of **fPR**. In contrast, with a given matrix $(\mathbf{I} - d\mathbf{T}')^{-1}$ and some priors in $\mathbf{f}_{\mathcal{S}}$, linear approach tries to figure out other values in $\mathbf{f}_{\mathcal{S}}$. Then, a question arises: how to quantitatively measure the connection between linear influence and influence-PageRank? To address this question and to introduce Group-PageRank, we first rewrite Eq. (3) as $f_{\mathcal{S}\to\mathcal{T}} = \mathbf{f}'_{\mathcal{S}}\mathbf{e}_{\mathcal{T}}$. By combining it with Eq. (8) and Eq. (6), we have

$$\begin{aligned} f_{\mathcal{S}\to\mathcal{T}} &= \mathbf{h}_{\mathcal{S}}'(\mathbf{I} - d\mathbf{T})^{-1}\mathbf{e}_{\mathcal{T}} = \mathbf{h}_{\mathcal{S}}'\frac{|\mathcal{T}|}{1-d}\mathbf{fPR} \\ &= \frac{|\mathcal{T}|}{1-d}\sum_{i\in\mathcal{S}} h_{\mathcal{S},i} fPR_i, \end{aligned} \quad (9)$$

which shows that the influence from $\mathcal{S}$ to $\mathcal{T}$ by Definition 1 is proportional to a linear combination of the influence-PageRank. Furthermore, we have

Lemma 1. $h_{\mathcal{S},i} \le 1 - d \cdot \sum_{k\in\mathcal{S}} t_{ki}$.

Proof Sketch: Let $\Gamma = (\mathbf{I} - d\mathbf{T}')$ and $\mathbf{P} = (\mathbf{I} - d\mathbf{T}')^{-1}$. Then, $\mathbf{f}_{\mathcal{S}} = (\mathbf{I} - d\mathbf{T}')^{-1}\mathbf{h}_{\mathcal{S}} = \mathbf{P}\mathbf{h}_{\mathcal{S}}$. Recall that for each node $i \in \mathcal{S}$, $f_{\mathcal{S}\to i} = 1$. We have $\mathbf{P}_{\mathcal{S}\mathcal{S}}\mathbf{h}_{\mathcal{S}\mathcal{S}} = \mathbf{e}$ and $\mathbf{h}_{\mathcal{S}\mathcal{S}} = \mathbf{P}_{\mathcal{S}\mathcal{S}}^{-1}\mathbf{e}$. $\mathbf{P}_{\mathcal{S}\mathcal{S}}$ is the matrix reduced from $\mathbf{P}$ by removing its rows and columns that do not correspond to the members in $\mathcal{S}$, and $\mathbf{h}_{\mathcal{S}\mathcal{S}}$ is reduced from $\mathbf{h}_{\mathcal{S}}$ by removing the entries that do not correspond to the members in $\mathcal{S}$. By rearranging the rows and columns in Γ,

$$\Gamma = \begin{bmatrix} \Gamma_{\mathcal{S}\mathcal{S}} & \Gamma_{\mathcal{S}\overline{\mathcal{S}}} \\ \Gamma_{\overline{\mathcal{S}}\mathcal{S}} & \Gamma_{\overline{\mathcal{S}}\overline{\mathcal{S}}} \end{bmatrix}.$$

Based on the linear algebra theory, we have

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} \mathbf{P}_{\mathcal{S}\mathcal{S}} & \mathbf{P}_{\mathcal{S}\overline{\mathcal{S}}} \\ \mathbf{P}_{\overline{\mathcal{S}}\mathcal{S}} & \mathbf{P}_{\overline{\mathcal{S}}\overline{\mathcal{S}}} \end{bmatrix} = \Gamma^{-1} = \begin{bmatrix} \Gamma_{\mathcal{S}\mathcal{S}} & \Gamma_{\mathcal{S}\overline{\mathcal{S}}} \\ \Gamma_{\overline{\mathcal{S}}\mathcal{S}} & \Gamma_{\overline{\mathcal{S}}\overline{\mathcal{S}}} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathbf{M} & -\mathbf{M}\Gamma_{\mathcal{S}\overline{\mathcal{S}}}\Gamma_{\overline{\mathcal{S}}\overline{\mathcal{S}}}^{-1} \\ -\Gamma_{\mathcal{S}\mathcal{S}}^{-1}\Gamma_{\mathcal{S}\overline{\mathcal{S}}}\mathbf{M} & \Gamma_{\mathcal{S}\mathcal{S}}^{-1} + \Gamma_{\mathcal{S}\mathcal{S}}^{-1}\Gamma_{\mathcal{S}\overline{\mathcal{S}}}\mathbf{M}\Gamma_{\mathcal{S}\overline{\mathcal{S}}}\Gamma_{\mathcal{S}\mathcal{S}}^{-1} \end{bmatrix}, \end{aligned}$$

where $\mathbf{M} = (\Gamma_{\mathcal{S}\mathcal{S}} - \Gamma_{\mathcal{S}\overline{\mathcal{S}}}\Gamma_{\overline{\mathcal{S}}\overline{\mathcal{S}}}^{-1}\Gamma_{\overline{\mathcal{S}}\mathcal{S}})^{-1}$. Thus, $\mathbf{P}_{\mathcal{S}\mathcal{S}} = \mathbf{M}$. In addition, $\mathbf{h}_{\mathcal{S}\mathcal{S}} = \mathbf{P}_{\mathcal{S}\mathcal{S}}^{-1}\mathbf{e} = \Gamma_{\mathcal{S}\mathcal{S}}\mathbf{e} - \Gamma_{\mathcal{S}\overline{\mathcal{S}}}\Gamma_{\overline{\mathcal{S}}\overline{\mathcal{S}}}^{-1}\Gamma_{\overline{\mathcal{S}}\mathcal{S}}\mathbf{e}$. Because $\Gamma_{\mathcal{S}\overline{\mathcal{S}}}\Gamma_{\overline{\mathcal{S}}\overline{\mathcal{S}}}^{-1}\Gamma_{\overline{\mathcal{S}}\mathcal{S}}$ is a nonnegative matrix,[2] we have $\mathbf{h}_{\mathcal{S}\mathcal{S}} \le \Gamma_{\mathcal{S}\mathcal{S}}\mathbf{e}$.

Thus, when $i \in \mathcal{S}$, $h_{\mathcal{S},i} \le 1 - d \cdot \sum_{k\in\mathcal{S}} t_{ki}$, and Lemma 1 holds. □

Definition 2. *We define our Group-PageRank from $\mathcal{S}$ to $\mathcal{T}$ as*

$$GPR(\mathcal{S},\mathcal{T}) = \frac{|\mathcal{T}|}{(1-d)}\left(\sum_{i\in\mathcal{S}} fPR_i - d\sum_{i\in\mathcal{S}}\sum_{k\in\mathcal{S}} t_{ki} fPR_i\right). \quad (10)$$

And we have the following theorem.

Theorem 1. *For the influence from $\mathcal{S}$ to $\mathcal{T}$, $f_{\mathcal{S}\to\mathcal{T}} \le GPR(\mathcal{S},\mathcal{T})$.*

Proof Sketch: Based on Eq. (9) and Lemma 1,

$$\begin{aligned} f_{\mathcal{S}\to\mathcal{T}} &= \frac{|\mathcal{T}|}{1-d}\sum_{i\in\mathcal{S}} h_{\mathcal{S},i} fPR_i \\ &\le \frac{|\mathcal{T}|}{1-d}\sum_{i\in\mathcal{S}}(1 - d\cdot\sum_{k\in\mathcal{S}} t_{ki}) fPR_i \\ &= GPR(\mathcal{S},\mathcal{T}). \end{aligned}$$

□

Actually, GPR is a generalization of the influence-PageRank. When $|\mathcal{S}| = 1$, $GPR(\mathcal{S},\mathcal{T})$ is proportional to fPR_i; When $|\mathcal{S}| > 1$, $GPR(\mathcal{S},\mathcal{T})$ is essentially the collection of each single influence-PageRank (for the nodes in $\mathcal{S}$) with a "discount" by which we mean that the mutual influences between the nodes in $\mathcal{S}$ are removed when estimating the influence spread of $\mathcal{S}$. Therefore, GPR can estimate the importance of any non-empty set of nodes which can be either a single node or a set of nodes. From Eq. (10), we can see that GPR is a combination of the basic elements of **fPR**. This implies the following: to get $GPR(\mathcal{S},\mathcal{T})$ for any $\mathcal{S}$, we can compute the **fPR** in advance (in $O(|\mathcal{A}|)$ time) and maintain it in a look-up table with the size of $|\mathcal{V}|$, and then we only need to take $O(|\mathcal{S}|)$ look-ups and $O(|\mathcal{S}|^2)$ additional computations (Eq. (10)). The computation of $GPR(\mathcal{S},\mathcal{T})$ can be done in near constant time with limited space consumption (as $\mathcal{S}$ are usually small, e.g., $|\mathcal{S}| = 100$).

Thus, if we use $GPR(\mathcal{S},\mathcal{T})$ as the estimation for $f_{\mathcal{S}\to\mathcal{T}}$, this influence computation will then meet the efficiency demand of large-scale online social networks. Group-PageRank has two properties.

- Group-PageRank is an estimation of the influence spread $f_{\mathcal{S}\to\mathcal{T}}$ and is also a very compact (we will show this claim experimentally) upper bound for $f_{\mathcal{S}\to\mathcal{T}}$.
- Group-PageRank can be quickly computed in advance and maintained in a table of length $|\mathcal{V}|$. Then, it only takes $O(|\mathcal{S}|^2)$ to compute Group-PageRank for any small $\mathcal{S}$.

[2] $\Gamma_{\mathcal{S}\overline{\mathcal{S}}}$ is an M-matrix, and its inverse (denoted as $\mathbf{N} = [n_{ij}]$) is nonnegative. Let $\mathbf{K} = \Gamma_{\mathcal{S}\overline{\mathcal{S}}}\mathbf{N}\Gamma_{\overline{\mathcal{S}}\mathcal{S}} = [k_{ij}]$, there is $k_{ij} = \sum_{l\notin\mathcal{S}}\sum_{m\notin\mathcal{S}}(\gamma_{il} n_{lm} \gamma_{mj})$. Because $\gamma_{il} = -t_{il} \le 0$, $\gamma_{mj} = -t_{mj} \le 0$, $n_{lm} \ge 0$, and $k_{ij} \ge 0$. Thus, $\mathbf{K} = \Gamma_{\mathcal{S}\overline{\mathcal{S}}}\Gamma_{\overline{\mathcal{S}}\overline{\mathcal{S}}}^{-1}\Gamma_{\overline{\mathcal{S}}\mathcal{S}}$ is also nonnegative.

4. SOCIAL INFLUENCE MAXIMIZATION

In this section, we show how the linear approach and Group-PageRank can be used to support viral marketing by addressing the influence maximization problem. This aims at finding a set of influencer nodes (e.g., $\mathcal{S}$) to maximize the product's word-of-mouth propagation in the entire or a part of the network. Formally, the influence maximization problem, which is NP-hard [24], is defined as follows.

$$\mathcal{S} = \arg\max_{\mathcal{S}\subset\mathcal{T}} f_{\mathcal{S}\to\mathcal{T}}, \quad s.t.\ |\mathcal{S}| = K,$$

where K is the desired seed set size (e.g., 50). In the following, we use $\mathcal{T}$ and $\mathcal{V}$ interchangeably, because usually we have $\mathcal{T} = \mathcal{V}$.

We adopt the greedy framework proposed by Kempe et al. [24], and the entire process is shown in Algorithm 1. Initially, $\mathcal{S} = \emptyset$. At each iteration, it adds a new node s into $\mathcal{S}$ if s maximizes the increment on influence, $s = \arg\max_{s\in\mathcal{V}\setminus\mathcal{S}} \Delta_s(\mathcal{S},\mathcal{T})$, e.g., $\Delta_s(\mathcal{S},\mathcal{T}) = f_{\mathcal{S}\cup\{s\}\to\mathcal{T}} - f_{\mathcal{S}\to\mathcal{T}}$. This iterative process will continue until the seed set size is up to K.

Algorithm 1: GreedyFramework

1. $\mathcal{S} = \emptyset$;
2. $s = \arg\max_{s\in\mathcal{V}\setminus\mathcal{S}} \Delta_s(\mathcal{S},\mathcal{T})$;
3. $\mathcal{S} = \mathcal{S}\cup\{s\}$;
4. If $|\mathcal{S}| < K$, then go back to step2; else terminate.

For Algorithm 1, we propose two ways to compute $\Delta_s(\mathcal{S},\mathcal{T})$, by *Linear* or by *Bound*.

- *Linear*: $\Delta_s^I(\mathcal{S},\mathcal{T}) = f_{\mathcal{S}\cup\{s\}\to\mathcal{T}} - f_{\mathcal{S}\to\mathcal{T}}$.
- *Bound*: $\Delta_s^{II}(\mathcal{S},\mathcal{T}) = GPR(\mathcal{S}\cup\{s\},\mathcal{T}) - GPR(\mathcal{S},\mathcal{T})$.

By *Linear*, the time cost of $\Delta_s^I(\mathcal{S},\mathcal{T})$ is $O(|\mathcal{A}|)$ due to the computation of $f_{\mathcal{S}\cup\{s\}\to\mathcal{T}}$. By *Bound*, i.e., Group-PageRank, with the help of Eq. (10), we have

$$\Delta_s^{II}(\mathcal{S},\mathcal{T}) = \frac{|T|}{1-d}\left((1-d\sum_{j\in\mathcal{S}} t_{js})fPR_s - d\sum_{j\in\mathcal{S}} t_{sj} fPR_j\right). \quad (11)$$

Since fPR_s for each s is a basic element that can be computed in advance for any $\mathcal{S}$ and s, the computation of $\Delta_s(\mathcal{S},\mathcal{T})$ in Eq. (11) only takes $O(|\mathcal{S}|)$ time. Moreover, the marginal influence increment $\Delta_s(\mathcal{S},\mathcal{T})$ satisfies the submodular property (proof is shown in the following two corollaries), and combining with the monotonicity property (e.g., $f_{\mathcal{S}\cup\{s\}\to\mathcal{T}} \geq f_{\mathcal{S}\to\mathcal{T}}$) we could guarantee that the greedy framework is lazy-forward [27]. Actually, the monotonicity property strictly holds for the linear approach, and we can only prove that the monotonicity property of Group-PageRank holds when $d \leq 0.5$ (Ref. [38] has presented a similar proof strategy). However, the real-world influence transition probabilities (e.g., t_{ij}) are quite small [43, 47] (especially for those between the seed nodes, as the seeds are usually far away from each other), and the monotonicity property generally holds even when $0.5 < d \leq 1$. Meanwhile, considering the positive experimental results, in this paper we simply treat Group-PageRank as monotone and leave the detailed discussion for future work. In the following, let's denote the influencer-set $\mathcal{S}$ in iteration k as $\mathcal{S}_k$.

Corollary 1.

$$\Delta_s^I(\mathcal{S}_0,\mathcal{T}) \geq \Delta_s^I(\mathcal{S}_1,\mathcal{T}) \geq ... \geq \Delta_s^I(\mathcal{S}_K,\mathcal{T}).$$

Proof Sketch [3]:

First, we show Eq. (8) is a linear function.

Second, based on the linear function, we show $f_{\mathcal{S}'\to j} \geq f_{\mathcal{S}\to j}$ where $\mathcal{S}' = \mathcal{S}\cup\{t\}$ and t can be an arbitrary node.

Third, we show $f_{\mathcal{S}'\cup\{s\}\to\mathcal{T}} - f_{\mathcal{S}'\to\mathcal{T}} \leq f_{\mathcal{S}\cup\{s\}\to\mathcal{T}} - f_{\mathcal{S}\to\mathcal{T}}$.

Finally, because

$$\Delta_s^I(\mathcal{S}_i,\mathcal{T}) = f_{\mathcal{S}_i\cup\{s\}\to\mathcal{T}} - f_{\mathcal{S}_i\to\mathcal{T}},$$

we have

$$\Delta_s^I(\mathcal{S}_i,\mathcal{T}) = f_{\mathcal{S}_i\cup\{s\}\to\mathcal{T}} - f_{\mathcal{S}_i\to\mathcal{T}} \geq f_{\mathcal{S}_{i+1}\cup\{s\}\to\mathcal{T}} - f_{\mathcal{S}_{i+1}\to\mathcal{T}} = \Delta_s^I(\mathcal{S}_{i+1},\mathcal{T}).$$

Thus, $\Delta_s^I(\mathcal{S}_0,\mathcal{T}) \geq \Delta_s^I(\mathcal{S}_1,\mathcal{T}) \geq ... \geq \Delta_s^I(\mathcal{S}_K,\mathcal{T})$ holds. □

Corollary 2.

$$\Delta_s^{II}(\mathcal{S}_0,\mathcal{T}) \geq \Delta_s^{II}(\mathcal{S}_1,\mathcal{T}) \geq ... \geq \Delta_s^{II}(\mathcal{S}_K,\mathcal{T}).$$

Proof: First, we prove $GPR(\mathcal{S}\cup\{v\},\mathcal{T}) - GPR(\mathcal{S},\mathcal{T}) \geq GPR(\mathcal{S}'\cup\{v\},\mathcal{T}) - GPR(\mathcal{S}',\mathcal{T})$, where $\mathcal{S}\subseteq\mathcal{S}'$, arbitrary node $v\notin\mathcal{S}'$.

$$\begin{aligned}
&GPR(\mathcal{S}\cup\{v\},\mathcal{T}) - GPR(\mathcal{S},\mathcal{T})\\
=&\frac{|\mathcal{T}|}{1-d}\left(fPR_v + d\sum_{i\in\mathcal{S}}\sum_{k\in\mathcal{S}} t_{ki} fPR_i - d\sum_{i\in\mathcal{S}\cup\{v\}}\sum_{k\in\mathcal{S}\cup\{v\}} t_{ki} fPR_i\right)\\
=&\frac{|\mathcal{T}|}{1-d}\left(fPR_v - d\left(\sum_{i\in\mathcal{S}\cup\{v\}} t_{vi} fPR_i + \sum_{k\in\mathcal{S}} t_{kv} fPR_v\right)\right)\\
\geq&\frac{|\mathcal{T}|}{1-d}\left(fPR_v - d\left(\sum_{i\in\mathcal{S}'\cup\{v\}} t_{vi} fPR_i + \sum_{k\in\mathcal{S}'} t_{kv} fPR_v\right)\right)\\
=&GPR(\mathcal{S}'\cup\{v\},\mathcal{T}) - GPR(\mathcal{S}',\mathcal{T}).
\end{aligned}$$

Then, because

$$\Delta_s^{II}(\mathcal{S}_i,\mathcal{T}) = GPR(\mathcal{S}_i\cup\{s\},\mathcal{T}) - GPR(\mathcal{S}_i,\mathcal{T}),$$

we have

$$\Delta_s^{II}(\mathcal{S}_i,\mathcal{T}) \geq GPR(\mathcal{S}_{i+1}\cup\{s\},\mathcal{T}) - GPR(\mathcal{S}_{i+1},\mathcal{T}) = \Delta_s^{II}(\mathcal{S}_{i+1},\mathcal{T}).$$

Thus, $\Delta_s^{II}(\mathcal{S}_0,\mathcal{T}) \geq \Delta_s^{II}(\mathcal{S}_1,\mathcal{T}) \geq ... \geq \Delta_s^{II}(\mathcal{S}_K,\mathcal{T})$ holds. □

Based on Corollary 1 and Corollary 2, we propose a lazy-forward greedy framework for social influence maximization, as shown in Algorithm 2. Specifically, we use Δ_s for both the upper bound and the real value of the marginal influence increment $\Delta_s(\mathcal{S},\mathcal{T})$, and meanwhile, we use Δ_{max} and s_{max} for the maximal $\Delta_s(\mathcal{S},\mathcal{T})$ and its corresponding node s, respectively. This algorithm starts with $\mathcal{S} = \emptyset$ (initial $\Delta_s = \frac{|\mathcal{T}|}{1-d} fPR_s$). In each iteration, it adds a new node s with the maximal $\Delta_s(\mathcal{S},\mathcal{T})$ into $\mathcal{S}$ until the size of $\mathcal{S}$ is equal to K. We use a priority queue such that $\Delta_s \geq \Delta_{s+1}$ in the queue. For each node s, we compare its upper bound Δ_s with Δ_{max}. There are two cases. First, if $\Delta_s > \Delta_{max}$, we compute its real influence increment for s (either by *Linear* or by *Bound*). If its real increment is still larger than Δ_{max}, the node s is truly a better one, and then we use this as s_{max} and store the real increment into Δ_{max}. Second, if $\Delta_s \leq \Delta_{max}$, then s and all of its successors cannot be better than the current s_{max} for $\Delta_{max} \geq \Delta_s \geq \Delta_{s+1}$, and thus we find the s_{max} and break the loop. We only need to add s into $\mathcal{S}$, and add the real increment by s_{max} into $f_{\mathcal{S}\to\mathcal{T}}$. Finally, we reset $\Delta_{s_{max}} = 0$, i.e., we remove node s_{max} from the candidate list.

In the lazy-forward greedy framework, an influencer-seed set $\mathcal{S}$ with the maximum influence propagation will be found efficiently. We call Algorithm 2 the *Linear* algorithm if it computes the real increment Δ_s by *Linear* (see Function GetDeltaI), and call Algorithm 2 the *Bound* algorithm if it computes Δ_s by *Bound* (see Function GetDeltaII).

[3] Detailed proof is omitted due to the limited space.

Algorithm 2: LinearFramework(G, K, λ)

input : $G(\mathcal{V}, \mathcal{A}, \mathbf{T}), K, \lambda$,
output: $\mathcal{S}$
$\mathcal{S} = \emptyset$;
Compute influence-PageRank vector $\mathbf{fPR} = [fPR_1, ..., fPR_n]'$ (see Section 3.2);
for *each node s in G* **do**
  $\Delta_s = \frac{|\mathcal{T}|}{1-d} fPR_s$; // *Upper bound*
while $|\mathcal{S}| < K$ **do**
  re-arrange the order of nodes to make $\Delta_s \geq \Delta_{s+1}$;
  $\Delta_{max} = 0$;
  for $s = 1$ *to* $n - |\mathcal{S}|$ **do**
    if $\Delta_s > \Delta_{max}$ **then**
      Compute the real increment Δ_s by *Linear* or by *Bound*;
      if $\Delta_s > \Delta_{max}$ **then**
        $\Delta_{max} = \Delta_s$;
        $s_{max} = s$;
    else
      break;
  $\mathcal{S} = \mathcal{S} \cup \{s_{max}\}$;
  $f_{\mathcal{S}\to\mathcal{T}} = f_{\mathcal{S}\to\mathcal{T}} + \Delta_{max}$;
  $\Delta_{s_{max}} = 0$;
return $\mathcal{S}$;

Function GetDeltaI($\mathcal{S}$, s, $f_{\mathcal{S}\to\mathcal{T}}$)

input : $\mathcal{S}$, s, $f_{\mathcal{S}\to\mathcal{T}}$
output: Δ_s
$\alpha = 0$; $\mathcal{S}' = \mathcal{S} \cup \{s\}$;
for *each node i of G* **do**
  if $i \in \mathcal{S}'$ **then**
    $f_{\mathcal{S}'\to i} = 1$;
  else
    $f_{\mathcal{S}'\to i} = 0$;
while $\alpha < MAX\_ITERATIONS$ **do**
  for *each node j in* $\mathcal{V} - \mathcal{S}'$ **do**
    $f_{\mathcal{S}'\to j} = d \sum_{(j,k)\in\mathcal{A}} t_{kj} f_{\mathcal{S}'\to k}$;
  α++;
$f_{\mathcal{S}'\to\mathcal{T}} = 0$;
for *each* $j \in \mathcal{T}$ **do**
  $f_{\mathcal{S}'\to\mathcal{T}} = f_{\mathcal{S}'\to\mathcal{T}} + f_{\mathcal{S}'\to j}$;
return $f_{\mathcal{S}'\to\mathcal{T}} - f_{\mathcal{S}\to\mathcal{T}}$; // Δ_s

Function GetDeltaII($\mathcal{S}$, s, **fPR**)

input : $\mathcal{S}$, s, **fPR**
output: Δ_s
$\Delta_s = fPR_s$;
for *each* $j \in \mathcal{S}$ **do**
  $\Delta_s = \Delta_s - d \cdot t_{js} fPR_s - d \cdot t_{sj} fPR_j$;
return $\Delta_s \cdot \frac{|\mathcal{T}|}{1-d}$;

5. EXPERIMENTAL RESULTS

In our empirical studies, we focus on validating the following performance: (1) The effectiveness and efficiency of the adopted linear social influence modeling approach compared with two of the existing influence models, and the effectiveness of Group-PageRank (Section 5.1); (2) The social influence maximization results of our two algorithms (*Linear* and *Bound*) compared with some of the state-of-the-art solutions (Section 5.2).

The four real-world social network datasets we used are: **Facebook** [29] which is sampled from *Facebook.com*[4], **ca-HepPh** [28] which is a collaboration network from the e-print arXiv covering collaborations between authors whose papers are submitted to *High Energy Physics - Phenomenology category*[5], **web-NotreDame** [3] which is a webpage link network where nodes represent pages from University of Notre Dame and directed edges represent hyperlinks between them[6], and **LiveJournal** [45] which is a friendship network published in July, 2010[7]. The four networks (two directed and two undirected) used cover a variety of networks with sizes ranging from 88K edges to 14M edges. Some basic data statistics about these networks are given in Table 2.

We implemented the approaches in C++ and conducted the following experiments on a server with 2.0GHz Quad-Core Intel Xeon E5410 and 16G memory.

5.1 The Linear Approach & Group-PageRank

In this subsection, we show that the linear influence modeling approach adopted in this paper has similar capability as the traditional models for describing the influence propagation in social networks. Meanwhile, Group-PageRank provides a good estimation of social influence under the linear approach. Furthermore, both linear approach and Group-PageRank are efficient.

[4]http://snap.stanford.edu/data/egonets-Facebook.html
[5]http://snap.stanford.edu/data/ca-HepPh.html
[6]http://snap.stanford.edu/data/web-NotreDame.html
[7]http://socialcomputing.asu.edu/datasets/LiveJournal

Similarity of Influence Vectors. We empirically verify whether the output of linear approach (LA) is similar to two traditional influence models, Independent Cascade (IC) model [14] and Stochastic (ST) model [2]. Specifically, the comparison in this experiment is focused on the similarity of the influence vectors. Suppose $\mathbf{f}_{\mathcal{S}}^{A}$, $\mathbf{f}_{\mathcal{S}}^{B}$ are the influence vectors of influencer-set $\mathcal{S}$ under model A and model B, respectively, where A, B are the model indicators, e.g., LA, IC, or ST. If model A is similar to model B, then $\mathbf{f}_{\mathcal{S}}^{A}$ must be close to $\mathbf{f}_{\mathcal{S}}^{B}$ for any $\mathcal{S}$, and vice versa.

We use the cosine similarity to measure the similarity between $\mathbf{f}_{\mathcal{S}}^{A}$ and $\mathbf{f}_{\mathcal{S}}^{B}$, denoted as $Sim(\mathbf{f}_{\mathcal{S}}^{A}, \mathbf{f}_{\mathcal{S}}^{B})$. Specifically, the formula that we use to measure the similarity between these models is as follows.

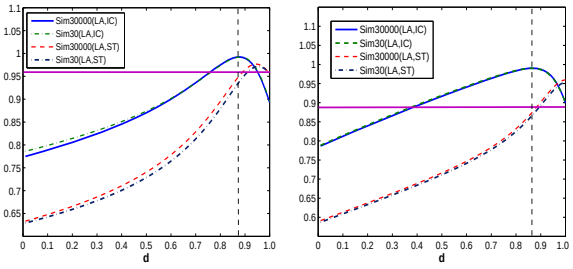
$$Sim(A, B) = \frac{\sum_{\mathcal{S}\subset\mathcal{V}} Sim(\mathbf{f}_{\mathcal{S}}^{A}, \mathbf{f}_{\mathcal{S}}^{B})}{\sum_{\mathcal{S}\subset\mathcal{V}}}. \quad (12)$$

Thus, if $Sim(A, B)$ is close to 1, then model A and B is similar. Because Eq. (12) is very expensive to compute (as there are $2^{|\mathcal{V}|}$ choices for $\mathcal{S}$), we randomly select a certain number of sets as representation to approximate $Sim(A, B)$. Also, since the Monte-Carlo simulation for the IC model is time consuming, we use two small datasets, Facebook and ca-HepPh, to evaluate the similarity between models. The computation is done under the following settings: We randomly select a certain number of influencer-sets (i.e., 30 and 30,000, respectively) with the size ranging in [1,100] as the representation of all influencer-sets. Parameter d ranges in $(0, 1)$, starting from 0.01 and stepping by 0.01. Transition matrix $\mathbf{T}$ is set as the transpose of PageRank matrix $\mathbf{W}$ (same as the transition matrix of WC model [24]), i.e., t_{ij} on edge (j, i) is equal to $\frac{Weight(j,i)}{OutWeight(j)}$.

As shown in Fig. 2, $Sim30000(A, B)$ and $Sim30(A, B)$ are the similarity curves computed using 30,000 sets and 30 sets, respectively. The purple horizontal line shows the similarity between IC and ST on 30,000 sets (i.e., $Sim30000(IC, ST)$). The black vertical dashed line is used to mark the peak point in the $Sim30000$ curve.

Table 2: Statistics of four real-world networks.

Networks	Facebook	ca-HepPh	web-NotreDame	LiveJournal
#Node	4,039	12,008	325,729	2,238,731
#Edge/Arc	88,234	237,010	1,497,134	14,608,137
Type	undirected	undirected	directed	directed



(a) Facebook. (b) ca-HepPh.

Figure 2: The similarity of influence vectors.

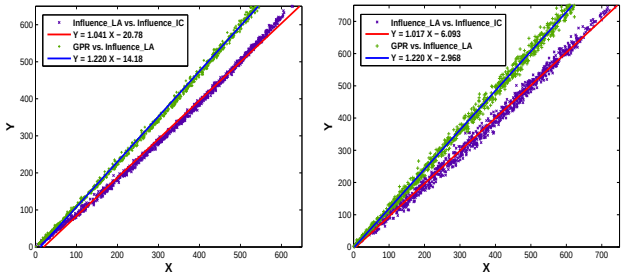
In the figure, we have three observations: First, LA can approach the other two models with a high similarity (larger than 0.99 and 0.96 respectively), while IC model and ST model are less similar to each other (with similarity value 0.96 and 0.89 respectively); Second, the curves of $Sim30000(LA, X)$ and $Sim30(LA, X)$ are very close, where X is either IC or ST. This means the similarities are irrelevant to the number of the sampled sets; Finally, the similarity curves between LA and the other two models all increase firstly and then decrease. The peaks are reached at a value when d is near to the value of 0.85.

These experimental observations show that the linear approach is a similar influence modeling method compared to traditional IC and ST models. Actually, in real applications, we could even replace the IC or ST model (e.g., by simply setting $d = 0.85$) with LA, as these two models are very expensive to compute.

The Influence Computing. In the following, we compare the relations among the exact value of $f_{\mathcal{S}\to\mathcal{T}}^{LA}$, the upper bound $GPR(\mathcal{S},\mathcal{T})$, and $f_{\mathcal{S}\to\mathcal{T}}^{IC}$. If $f_{\mathcal{S}\to\mathcal{T}}^{LA}$ is also close to $f_{\mathcal{S}\to\mathcal{T}}^{IC}$, then LA is really similar to IC model [8] and can be used to substitute for IC in real applications, since LA is much more efficient, which we will illustrate later. We show our experimental results using the four datasets in Table 2. For testing, we randomly select 100 influencer-sets with their sizes ranging in [1,100]. For each selected $\mathcal{S}$, we compute its $f_{\mathcal{S}\to\mathcal{V}}^{LA}$, $GPR(\mathcal{S},\mathcal{V})$, and $f_{\mathcal{S}\to\mathcal{V}}^{IC}$. The final results are shown in Fig. 3, where "Influence_LA" indicates $f_{\mathcal{S}\to\mathcal{V}}^{LA}$, "GPR" means $GPR(\mathcal{S},\mathcal{V})$ and "Influence_IC" indicates $f_{\mathcal{S}\to\mathcal{V}}^{IC}$. Note the x axis is the index of $\mathcal{S}$. We have two observations: 1) Influence_LA and Influence_IC almost overlap each other; 2) On each dataset, GPR is consistently compact to Influence_LA. To further test their quantitative relations, we compute 1,000 groups of results and plot them as pair (Influence_LA vs. Influence_IC) and (GPR vs. Influence_LA) in the coordinates of Fig. 4. The similar results on web-NotreDame and LiveJournal are omitted due to limitations of space. These plots could be well fitted by linear function; and the slopes of these fitting lines are 1.041 and 1.017 for (Influence_LA vs. Influence_IC), 1.220 and 1.220 for (GPR vs. Influence_LA), on Facebook and ca-HepPh, respectively. These results imply that: 1) the influence computation results by LA and IC are almost the same. In other words, LA can be used to substitute for IC if efficiency is the main concern; 2) Group-PageRank is a good estimation of the social influence under LA as a consistently compact upper bound.

Efficiency. The total computing time for $f_{\mathcal{S}\to\mathcal{V}}^{LA}$, $GPR(\mathcal{S},\mathcal{V})$, and $f_{\mathcal{S}\to\mathcal{V}}^{IC}$ on the 100 randomly selected node sets are listed in Table 3. LA is almost 100 times faster than IC. With the help of Group-PageRank, the influence estimation $GPR(\mathcal{S},\mathcal{V})$ can be finished much quicker, e.g., no more than 1 second for small networks.

[8]Combining with the results (Fig. 2) that these two models' influence vectors are similar.



(a) Facebook. (b) ca-HepPh.

Figure 4: The fitting curves for (Influence_LA vs. Influence_IC) and (GPR vs. Influence_LA).

Table 3: Comparison of execution time (Sec.).

	Facebook	ca-HepPh	web-NotreDame	LiveJournal
$f_{\mathcal{S}\to\mathcal{V}}^{LA}$	2.63	2.01	20.17	526.07
$GPR(\mathcal{S},\mathcal{V})$	**0.04**	**0.03**	**0.25**	**5.39**
$f_{\mathcal{S}\to\mathcal{V}}^{IC}$	334.68	996.49	2700.56	12421.45

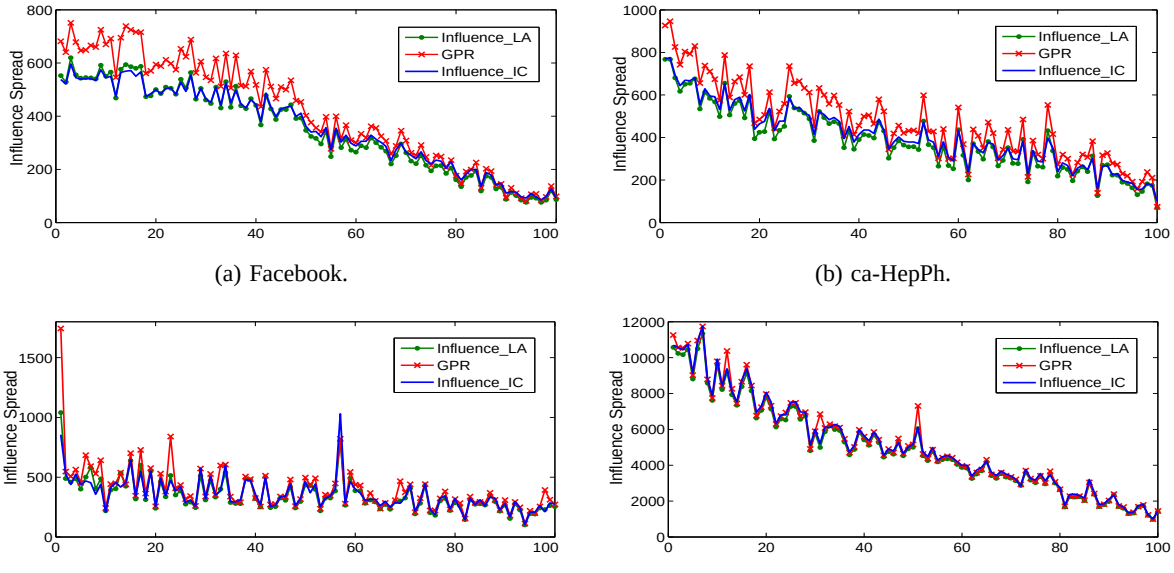
5.2 Social Influence Maximization

In the following, we show that *Linear* and *Bound* are both effective and efficient for solving the social influence maximization problem. To this end, we compare them with several (i.e., 6) state-of-the art algorithms.

- *CELF* is the original greedy algorithm [24] with the CELF optimization of [27], where the number of Monte-Carlo simulations under IC model is set to be 20,000.
- *IRIE* is a scalable algorithm that integrates the advantages of influence ranking (IR) and influence estimation (IE) methods for influence maximization [23].
- *PMIA* is the algorithm proposed in Ref. [9]. According to the authors' suggestions, we select the parameter with the best performance from {1/10,1/20,1/40,1/80,1/160,1/320,1/1280}.
- *PageRank (PR)* algorithm [34], in which we selected top-K nodes with the highest pagerank value.
- *DegreeDiscountIC (DIC)* [10] measures the degree discount heuristic with a propagation probability of $p = 0.01$, which is the same as that used in Ref. [10].
- *Degree (Deg)* algorithm captures the top-K nodes with the highest degree.

Among these algorithms, *Deg*, *DIC* and *PR* are widely used for baselines, and *CELF*, *IRIE* and *PMIA* are three of the outstanding algorithms in terms of both effectiveness and efficiency.

For computing influence maximization, one algorithm will return a set $\mathcal{S}$ with K nodes, and the effectiveness of the algorithm is justified by the influence spread (i.e., $f_{\mathcal{S}\to\mathcal{V}}$, the expected number of nodes that will be influenced) of the chosen $\mathcal{S}$; that is, the bigger the $f_{\mathcal{S}\to\mathcal{V}}$ the better the algorithm. Since IC model is the most widely accepted influence computation model, we run Monte-Carlo simulation under IC model to estimate and compare each $f_{\mathcal{S}\to\mathcal{V}}$. Specifically, the simulation is done as follows (called Weighted Cascade (WC) model [24]): The nodes in $\mathcal{S}$ are viewed as the ones activated at time $t = 0$; Each activated node can influence its neighbors independently; If node i is activated at time t, then it will influence its not-yet-activated neighbor node j at time $t + 1$ (and only time $t + 1$) with transition probability t_{ij} on arc (j, i). As given in Section 5.1, we set the transition probability t_{ij} equal to $\frac{weight(j,i)}{OutWeight(j)}$ which is widely adopted in the literature. The size K of $\mathcal{S}$ in our tests ranges from 5 to 50. We report the best performance of each



(a) Facebook. (b) ca-HepPh.

(c) web-NotreDame. (d) LiveJournal.

Figure 3: The influence spread for $f^{LA}_{\mathcal{S}\to\mathcal{V}}$ (Influence_LA), $GPR(\mathcal{S},\mathcal{V})$ (*GPR*), and $f^{IC}_{\mathcal{S}\to\mathcal{V}}$ (Influence_IC).

Table 4: The Summary.

	Linear	Bound	IRIE	PMIA	PR	DIC	Deg	**Win**
Linear	-	4	4	4	4	4	4	**24**
Bound	0	-	1	4	4	4	4	**17**
IRIE	0	1	-	4	4	4	4	**17**
PMIA	0	0	0	-	1	4	4	**9**
PR	0	0	0	1	-	4	4	**9**
DIC	0	0	0	0	0	-	3	**3**
Deg	0	0	0	0	0	0	-	**0**
Loss	**0**	**5**	**5**	**13**	**13**	**20**	**23**	

algorithm listed by tuning its parameters. Meanwhile, for consistency, we set the parameter d in *PR*, *Linear* and *Bound* (α in *IRIE*) equals to 0.85, which is a widely used value in *PR*.

Effectiveness. Fig. 5 shows the influence spread, where we can see that both *Linear* and *Bound* are effective, as the selected node sets are very influential. Since linear approach is similar to IC model, the performance of *Linear* and *CELF* is also quite similar (almost overlap each other on Facebook). For better illustration, some algorithms are plotted together if they output similar influence spread values, e.g., *Linear* and *CELF* on ca-HepPh. Furthermore, we summarize these results on 4 datasets into Table 4, which means how many times an algorithm A (row) outperforms (with a larger $f_{\mathcal{S}\to\mathcal{V}}$) an algorithm B (column) when $K = 50$. The max value is 4 since we tested 4 datasets, and we do not count the dataset if there is no obvious difference between two algorithms' performance, e.g., *Bound* and *IRIE* on the ca-HepPh and LiveJournal datasets. Thus, the sum of the values in two symmetric entries of Table 4 is less than or equal to 4. In Table 4 the last column shows the summarized number for an algorithm that outperforms the others. Based on Fig. 5 and the numbers in the last column of Table 4: $Linear > Bound \approx IRIE > PMIA > PR > DIC > Deg$, and thus *Linear* is the best. We do not show the results of *CELF* in Table 4, because it only handles two of the small networks and fails when testing web-NotreDame and LiveJournal (Fig. 5).

Efficiency. Fig. 6 shows the computing time, where we do not present the computing time of *DIC* and *Deg* because they are almost 0. In terms of efficiency, the relative performance of the algorithms is given by $DIC \approx Deg > PR \approx Bound \gg Linear > IRIE > PMIA \gg CELF$. The computing time in Fig. 6 are shown in log scale for better illustration. Unfortunately, this also makes

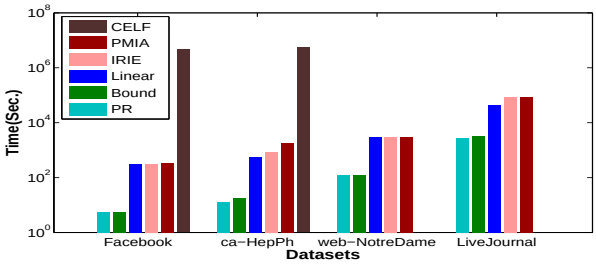
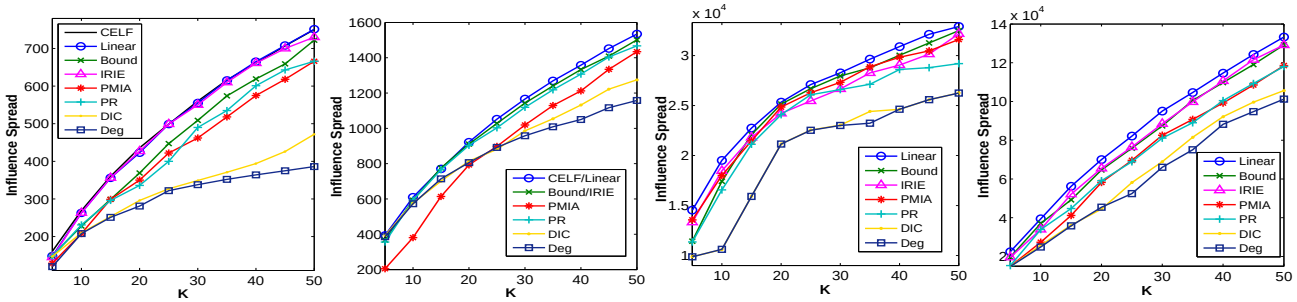


Figure 6: The computational costs (in logs).

the difference among some algorithms, e.g., *Linear*, *IRIE* and *PMIA*, become less obvious (Actually, *Linear* is faster than *IRIE*, and *IRIE* is faster than *PMIA*). Another observation is that the computing time of *Bound* is almost equal to *PR* which means that *Bound* is a linear time algorithm for viral marketing, i.e., with $O(|\mathcal{A}|)$ time for computing vector **fPR**. *Bound* is as scalable as *PR* for large scale networks, and it is more effective (refer to Fig. 5 and Table 4): *PR* may find the top-K most influential individuals. However, it does not consider the "influence overlapping" among selected individuals. Therefore, the top-K most influential individuals selected by *PR* may not lead to the maximization of influence spread. In contrast, *Bound* tackles this issue by including a "discount" (i.e., Group-PageRank) for the mutual influences of the selected seed nodes. Thus, *Bound* outperforms *PR* for handling influence.

In summary, for solving the social influence maximization problem in viral marketing, *Linear* and *Bound* perform consistently well on each network. Specifically, the seed nodes returned by *Linear* could exert the most influence spread, and in contrast, if the company wants to select a fast and also effective algorithm for a large scale social network, then *Bound* will be a better choice. Note that, implementing the algorithms on distributed architectures may further help the companies in viral marketing.

Damping Factor d. Previously, we set d simply equals to 0.85, and Fig. 2 also demonstrates that the output of linear approach is the most similar to that of traditional IC and ST models when d is near to 0.85. More specifically, in the following, we investigate more details on the effect of tuning d in terms of both the running time and the influence spread of the selected seeds (i.e., for social influence maximization) of *Linear* and *Bound*. To this end, we set



(a) Facebook. (b) ca-HepPh. (c) web-NotreDame. (d) LiveJournal.

Figure 5: The influence spread on four datasets.

d ranging from 0.05 to 1, with step 0.05, and compute the corresponding running time and influence spread with K=50.

The upper row of Fig. 7 shows the effectiveness of both *Linear* and *Bound* with different d on four social networks respectively. Here, the x axis is the d value, and the red dashed line is the result of *CELF* (As already mentioned, CELF is too expensive to run on web-NotreDame and LiveJournal). From these figures, we obtain the following observations: 1) The performances of *Linear* and *Bound* increase at first and then decrease, following the same trend in Fig. 2. For instance, the best results on Facebook and ca-HepPh are reached when d is near to 0.9 and 0.8, respectively, which is also close to the optimal d in Fig. 2; 2) When d locates in range [0.6,0.9], the performance stays at a high level (almost better than 95% of CELF results for both *Linear* and *Bound*).

In Fig. 7, the bottom figures show the computing time of *Linear* and *Bound*. The time cost of *Linear* increases while d increases, and when $d \geq 0.9$, the computing time increases significantly. This is because the larger d the less information will be blocked by each node, and therefore, the more nodes social influence will be spread over. Thus, the linear approach converges slowly. However, with the help of Group-PageRank heuristic, the computing time of *Bound* keeps very little.

In summary, as is well known, it is hard to find a specific d which performs the best for all the datasets [23]. However, in terms of both effectiveness and efficiency, we suggest randomly choose a value of d in $[0.6, 0.9]$ for each data set, e.g., 0.85, which is widely used in the research literature of ranking.

6. CONCLUSION

In this paper, we provided a bounded linear approach for effective and efficient influence computation and influence maximization. Specifically, we first adopted a tractable linear approach for describing the influence propagation in social networks. Then, to further address the scalability issues of social influence computing for the social influence maximization problem, we proposed a quantitative metric, named Group-PageRank. It is a tight upper bound of the influence of any node set, and it can be computed in near constant time. Next, we applied both the linear approach and Group-PageRank for solving the social influence maximization problem in viral marketing. Along this line, we proposed two lazy-forward greedy algorithms, *Linear* and *Bound*, based on the linear approach and Group-PageRank, respectively. Finally, the extensive experimental results demonstrated that 1) the linear approach is both flexible and efficient for social influence computing, and Group-PageRank provides a good estimation of social influence under the linear approach; 2) Both *Linear* and *Bound* algorithms could quickly find a set of the influential nodes for viral marketing campaign. For these two algorithms, *Linear* is more effective while *Bound* is more efficient.

Acknowledgements

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the National High Technology Research and Development Program of China (Grant No. 2014AA015203), the Natural Science Foundation of China (Grant No. 71329201), the Fundamental Research Funds for the Central Universities of China (Grant No. WK0110000042) and the Anhui Provincial Natural Science Foundation (Grant No. 1408085QF110). Also, it was supported in part by National Science Foundation (NSF) via grant number CCF-1018151. Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association, CAS.

7. REFERENCES

[1] C. Aggarwal. *Social network data analytics*. Springer-Verlag New York Inc, 2011.

[2] C. C. Aggarwal, A. Khan, and X. Yan. On flow authority discovery in social networks. In *SDM*, pages 522–533. SIAM, 2011.

[3] R. Albert, H. Jeong, and A. Barabási. Internet: Diameter of the world-wide web. In *Nature*, 401(6749):130–131, Nature Publishing Group, 1999.

[4] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *SIGKDD*, pages 7–15. ACM, 2008.

[5] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *SIGKDD*, pages 635–644. ACM, 2011.

[6] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *Internet and Network Economics*, pages 306–311. Springer, 2007.

[7] W. Chen, L. V. Lakshmanan, and C. Castillo. *Information and Influence Propagation in Social Networks*. Morgan and Claypool, 2013.

[8] W. Chen, W. Lu, and N. Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *AAAI*, pages 592–598. AAAI Press, 2012.

[9] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *SIGKDD*, pages 1029–1038. ACM, 2010.

[10] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *SIGKDD*, pages 199–208. ACM, 2009.

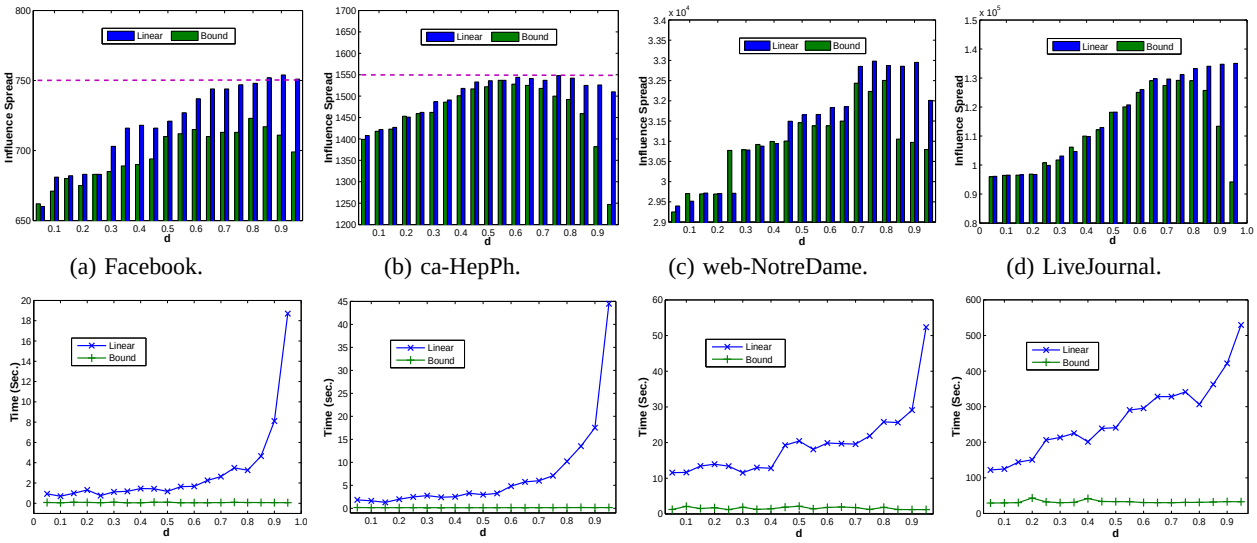
[11] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng. Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In *CIKM*, pages 509–518. ACM, 2013.

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

[13] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge Univ Pr, 2010.

[14] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

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



(a) Facebook. (b) ca-HepPh. (c) web-NotreDame. (d) LiveJournal.

(e) Facebook. (f) ca-HepPh. (g) web-NotreDame. (h) LiveJournal.

Figure 7: The effectiveness and the computing time of *Bound* and *Linear* on four networks with d changing in (0, 1). The top row figures show the effectiveness and the bottom row figures show the computing time.

[16] A. Goyal, F. Bonchi, and L. Lakshmanan. Approximation analysis of influence spread in social networks. *Arxiv preprint arXiv:1008.2005*, 2010.

[17] A. Goyal, F. Bonchi, and L. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, pages 241–250. ACM, 2010.

[18] A. Goyal, F. Bonchi, and L. V. Lakshmanan. A data-based approach to social influence maximization. *PVLDB*, 5(1):73–84, 2011.

[19] A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *WWW*, pages 47–48. ACM, 2011.

[20] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, pages 1420–1443, 1978.

[21] J. Guo, P. Zhang, C. Zhou, Y. Cao, and L. Guo. Personalized influence maximization on social networks. In *CIKM*, pages 199–208. ACM, 2013.

[22] T. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE TKDE*, 15(4):784–796, 2003.

[23] K. Jung, W. Heo, and W. Chen. Irie: Scalable and robust influence maximization in social networks. In *ICDM*, pages 918–923. IEEE, 2012.

[24] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146. ACM, 2003.

[25] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. *PKDD*, pages 259–271, 2006.

[26] W. Lee, J. Kim, and H. Yu. Ct-ic: Continuously activated and time-restricted independent cascade model for viral marketing. In *ICDM*, pages 960–965. IEEE, 2012.

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

[28] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. In *ACM TKDD*, 1(1):2,2007.

[29] J. Leskovec, and J. Julian. Learning to discover social circles in ego networks. In *NIPS*, pages 539–547, 2012.

[30] P. Li, J. X. Yu, H. Liu, J. He, and X. Du. Ranking individuals and groups by influence propagation. In *PAKDD*, pages 407–419. Springer, 2011.

[31] Y. Li, W. Chen, Y. Wang, and Z.-L. Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *WSDM*, pages 657–666. ACM, 2013.

[32] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, pages 199–208. ACM, 2010.

[33] Q. Liu, B. Xiang, L. Zhang, E. Chen, C. Tan, and J. Chen. Linear computation for independent social influence. In *ICDM*, pages 468–477. IEEE, 2013.

[34] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of WWW Conference*. Stanford InfoLab, 1999.

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

[36] K. Subbian, C. Aggarwal, and J. Srivastava. Content-centric flow mining for influence analysis in social streams. In *CIKM*, pages 841–846. ACM, 2013.

[37] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *SIGKDD*, pages 807–816. ACM, 2009.

[38] H. Tong, J. He, Z. Wen, R. Konuru, and C.-Y. Lin. Diversified ranking on large graphs: an optimization viewpoint. In *SIGKDD*, volume 11, pages 1028–1036, 2011.

[39] G. Wang, Q. Hu, and P. S. Yu. Influence and similarity on heterogeneous networks. In *CIKM*, pages 1462–1466. ACM, 2012.

[40] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *SIGKDD*, pages 1039–1048. ACM, 2010.

[41] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang. Pagerank with priors: an influence propagation perspective. In *IJCAI*, pages 2740–2746. AAAI Press, 2013.

[42] D.-N. Yang, H.-J. Hung, W.-C. Lee, and W. Chen. Maximizing acceptance probability for active friending in online social networks. In *SIGKDD*, pages 713–721. ACM, 2013.

[43] Y. Yang, E. Chen, Q. Liu, B. Xiang, T. Xu, and S. Shad. On approximation of real-world influence spread. In *ECML-PKDD*, pages 548–564, 2012.

[44] H. Yu, S.-K. Kim, and J. Kim. Scalable and parallelizable processing of influence maximization for large-scale social networks? In *ICDE*, pages 266–277. IEEE, 2013.

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

[46] M. Zhang, C. Dai, C. Ding, and E. Chen. Probabilistic solutions of influence propagation on social networks. In *CIKM*, pages 429–438. ACM, 2013.

[47] C. Zhou, P. Zhang, J. Guo, X. Zhu, and L. Guo. Ublf: An upper bound based approach to discover influential nodes in social networks. In *ICDM*, pages 907–916. IEEE, 2013.

[48] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.