Computer Networks 75 (2014) 544–559

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Whom to follow: Efficient followee selection for cascading outbreak detection on online social networks

Junzhou Zhao [a,*], John C.S. Lui [b], Don Towsley [c], Xiaohong Guan [a]

[a] MOEKLINNS Lab, Xi'an Jiaotong University, China
[b] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
[c] School of Computer Science, University of Massachusetts at Amherst, United States

ARTICLE INFO

Article history:
Received 30 November 2013
Received in revised form 8 July 2014
Accepted 11 August 2014
Available online 5 October 2014

Keywords:
Follow model
Followee selection
Outbreak detection
Submodularity

ABSTRACT

Online social networks (OSNs), such as Twitter and Sina Weibo, have become important platforms for generating and spreading information on the Internet. On these OSNs, the "follow model" has become a popular way to discover information; i.e., a user subscribes to content generated by others by following them as information sources. The content producers are called followees. Due to human beings' limited attention capacity and the constraints imposed by OSNs, a user can only follow a few followees. The question then arises: which subset of followees shall we follow so that we can discover the most information in an OSN in a timely fashion? To solve this problem, we present a randomized method that does not require complete OSN data and is well suited for third parties who do not own OSN data. Our method is based on the birthday paradox and is mathematically tractable for analysing its solution quality and computational efficiency. Moreover, we find that the power-law structure of real-world OSNs can further improve the solution quality of our method. Experiments conducted on two real datasets demonstrate that our method can create a good trade-off between solution quality and computational efficiency.

1. Introduction

As platforms for communicating with friends, updating status and sharing information, online social networks (OSNs), such as Twitter and Sina Weibo, have become extremely popular. These platforms provide users with near-real-time services that can be accessed across multiple devices at any Internet-enabled venue. Due to their large user bases and ubiquitous services, microblogs, where users act as sensors reporting events happening around them, become essential news sources, i.e., the so-called social media [1,2]. In fact, social media have attracted surveillance from conventional media outlets to discover breaking news and from governments to detect signals of riots. For example, the death of Osama bin Laden was first reported on Twitter rather than traditional news media [3], and, during the period of England riots, people used social media to organize [4].

The emergence of social media has changed the way we discover information. Traditional ways, such as information retrieval, rely on user-specified queries (e.g., keywords) to retrieve the information from indexed data [5–7]. However, specifying explicit queries might be difficult, as keywords for time-evolving and emerging information are highly dynamic [8] and unpredictable (e.g., the death of bin Laden [3]). In recent years, the follow model [9] has become a convenient way to discover information. A microblog user, say, Alice, obtains information mainly from her timeline, which comprises tweets generated by

* Corresponding author. Tel.: +86 29 8266 3330; fax: +86 29 8266 4603.
E-mail address: junzhouzhao@gmail.com (J. Zhao).

http://dx.doi.org/10.1016/j.comnet.2014.08.024
1389-1286/

users she follows, called her *followees*[1] in the follow model. Once a new tweet is posted by a user, it spreads to the user's followers and their followers iteratively, depending on whether users retweet the message. Finally, the tweet appears in Alice's timeline with a certain *probability* and *time delay*. Such probability and time delay mainly depend on which subset of followees Alice chooses to follow. By choosing different followees as her information sources, Alice can discover varied information with different time delay from the aggregated tweets in her timeline.

In the current information era, common goals that people want to achieve are to discover as much information as possible, i.e., maximize information coverage, and to obtain the information as soon as possible, i.e., minimize time delay. To achieve this, it seems that if Alice can follow all microblog users, then she will discover all information with zero time delay. However, due to human beings' limited attention capacity [11] and the constraints imposed by OSNs (e.g., a user on Twitter and Sina Weibo can follow at most 2000 users, in general [12,13]), Alice can only follow a few followees (or budgeted followees). Consequently, a problem arises: how to optimally choose these budgeted followees as information sources to maximize information coverage and minimize time delay?

The above problem is challenging. Selecting a subset of items from a population to maximize some specified utility function is a classical combinational optimization problem that has been studied for decades, e.g., the set cover problem [14], knapsack problem [15], influence maximization problem [16,17], and sensor placement problem [18,19]. These problems have been proven to be NP-hard, and we can only obtain suboptimal solutions using approximate algorithms. However, as we will see in Section 3, these algorithms cannot be applied to our problem and they do not scale to handle modern large-scale OSNs which have hundreds of millions of users.

Another challenge we face is that we are constrained to solving the problem from the perspective of a third party. A third party does not own OSN data. OSN companies own users' data, but there is a lack of cooperation from OSN companies due to user privacy and business secrecy concerns. Thus, third parties can only use the public APIs to crawl the data. However, OSN companies usually impose barriers to limit large-scale crawling by third parties [20] and restrict the request rate of APIs. For example, Twitter and Sina Weibo allow a user to issue at most 350 and 150 requests per hour, respectively [21,22]. As a result, it is practically impossible for third parties to crawl the complete data, and one has to consider the query cost (i.e., the number of API calls) while achieving the goal of selecting budgeted followees. We would like to note that most of the existing works [18,19,23] have ignored this second challenge and assumed that the complete data are available in advance. This assumption limits the practical application of existing methods, and the goal of this work is to fill this research gap.

[1] The Oxford dictionary defines a followee as a person who is being tracked on a social media website or application [10].

In this work, we present a framework to select a subset of users as followees to maximize the information coverage and minimize the time delay from incomplete data obtained via graph sampling methods. Our method guarantees both solution quality and computational efficiency (under the worst-case situation) that enable it to be used in large-scale OSNs. (Note that the proposed approach does not replace but instead supplements existing methods, and we elaborate on this point in Section 8). The basic idea behind our method is based upon the *birthday paradox*, which states that with more than 50% chance, there will be a birthday match among a handful of 23 people, and, for merely 70 people, the chance of matching increases to 99.9%. In our scenario, the randomized greedy algorithm, which is the main component of our framework, chooses one user in each iteration from a set of user samples, which is substantially smaller than the population, and the user samples contain at least one optimal followee with high probability according to the birthday paradox. Due to the significant reduction of search space, we achieve a major speedup in obtaining the solution. The quality of the final solution can be proven to be lower bounded when user samples are chosen uniformly at random in each iteration, which actually is the worst-case situation because we do not use any strategy in sampling (see Section 5).

Moreover, we find that if we bias user samples toward high degree nodes (i.e., users) in the network using graph sampling methods such as random walk [24], we need fewer user samples than when using uniform sampling, thereby improving efficiency. We present an in-depth analysis in Section 6 and reveal another important finding. For power-law networks, information cascades are not uniformly dispersed among nodes, but rather, high degree nodes are more likely to be infected by an information diffusion process than low degree nodes. Therefore, if the sampling is biased toward high degree nodes, we achieve a higher probability of detecting information cascades. This is related to the generalized birthday paradox, i.e., when people's birthdates are not uniformly distributed, the probability of matching increases [25]. Our numerical solutions in Section 6 and experiments in Section 7 both demonstrate the finding.

The rest of the paper is structured as follows. We review the related literature in Section 2 and formulate the problem in Section 3. Then, we motivate a randomized method through empirical observations in Section 4. The detailed analysis of our method is given in Section 5. In Section 6, we study how the power-law structure of real-world networks can benefit this method. We conduct experiments on real datasets to validate the method in Section 7 and conclude in Section 8.

2. Related work

Both Twitter and Sina Weibo provide the "whom-to-follow" services to recommend "interesting persons" to users [26]. This function is related to a large body of research on link prediction [27]. However, algorithms in link prediction are mainly based on common friends, shared interests, and

other factors [28–30], which are not directly related to our goal.

Optimal sensing [19,31] is the problem of selecting a subset of informative observations or sensors from a domain to maximize some utility for environment monitoring [31,32] or event detection [18]. For OSNs like Twitter and Sina Weibo, their ubiquitous services enable users to report events happening around them at any Internet-enabled venue at any time by tweeting. Because OSN users' behaviours are exactly like physical sensors reporting measurements of environments, they can be considered as social sensors [2]. Therefore, our followee selection problem is related to this research area. Leskovec et al. [18] studied the optimal sensing problem for city water monitoring (i.e., select a few locations to install sensors in order to detect water pollutions) and blog selection (i.e., select a subset of blogs to read to catch the most stories). They proposed the Cost-Effective Lazy Forward (CELF) approach to find the approximate solution. CELF is similar to the Accelerated Greedy (AG) approach posed by Minoux [33], which exploits the submodularity of utility functions. It is important to note the fundamental contrast between our setting and theirs: their methods are designed based on the necessity of complete data, which is impractical for third parties on contemporary large-scale OSNs, and their methods do not guarantee computational efficiency, which we will analyse in detail in Section 3.

Recently, several empirical studies have leveraged the friendship paradox to select users for the purpose of predicting contagious outbreaks in a university [34], a city [35], and detecting events on Twitter [36]. In this method, a user set is returned by repeatedly selecting a random friend of a randomly sampled user, and friendship paradox [37,38] guarantees selected users to have high degrees on average. Intuitively, high degree nodes in a network are easier to be infected by contagions than low degree nodes because of higher contact rate. That is why this approach works. However, this approach only uses topology information of a network and does not use contagion data on the network. For example, in Twitter, besides connections among users, we also know from history data which user retweeted which tweet at what time, and such information can be used to obtain better solutions. In addition, these empirical studies lack an analysis of bounding their solution quality with respect to the optimal solution. We fill this gap via a randomized method that is mathematically tractable to analyse.

It is also worth noting that our randomized method is motivated by an optimization method called ordinal optimization (OO) [39], which has been widely applied in the optimization of discrete event dynamic systems [40], power systems [41], and other areas [42]. OO considers the problem of searching for an optimal strategy in a very large strategy space (which is similar to our setting in dealing with a large-scale OSN). Other than finding an optimal strategy, OO defines a "good enough" subset which contains acceptable good strategies, and softens the goal to find one strategy in this good enough set. Often, a little softening of the goal, a major speedup in search can be achieved. We were inspired by this goal softening idea and have designed a randomized method, which can trade off between solution quality and computational efficiency.

3. Problem definition

In this section, we first introduce some terminology and notations (a notation table can be found in Appendix A). Then, we formulate the problem and analyse various state-of-the-art methods. Finally, we introduce two real-world datasets as our ground-truth data.

3.1. Terminology and notations

We model an OSN by a graph, $G(V, E, C)$, with $|V| = n$ nodes and $|E| = m$ edges. Each node represents a user, and each edge represents a relation between two users. We assume G is undirected for ease of presenting our idea.

Here, C denotes a set of *information cascades* on the network. Information cascades (or *cascades* for short) are phenomena in which actions or ideas become widely adopted by people due to the influence of others (typically, people are influenced by their neighbours in the network) [43]. For example, if a tweet is retweeted by many Twitter users or many users tweet the same hashtag, then we say it forms a cascade because many people have adopted the same action. If Alice also retweets the tweet or tweets the same hashtag, we say that Alice has joined the cascade. Here, we simplify a cascade $c \in C$ by a vector of user join times, $[t_{uc}]_{u \in V}$, where t_{uc} denotes the time that user u joined c, and $t_{uc} = \infty$ if u never joins c. The *start time* of cascade c is the earliest join time, denoted by t_c, i.e., $t_c = \min_{u \in V} t_{uc}$. Then, the time delay for user u to join c is $t_{uc} - t_c$. In our previous example, if Alice follows u, then she can obtain c from her timeline with time delay $t_{uc} - t_c$. Hence, different followees will experience different time delays for discovering cascades. Furthermore, we define the *size* of a cascade as the number of unique users that join it, i.e., $size(c) = |\{u : t_{uc} < \infty \wedge u \in V\}|$, and it will be used to indicate the importance of c. In other words, the larger the size of a cascade is, the more important it is among all cascades.

3.2. Followee selection for cascading outbreak detection

Having defined the terminology, our problem becomes which subset of users Alice should follow so that she can obtain as many important cascades as possible in her timeline, and with time delays as small as possible. In other words, if Alice chooses the right followees, she can discover the majority of important cascades with small average time delay in her timeline. We call this problem the cascading outbreak detection problem, and formally define it as follows.

Definition 1 (*Cascading outbreak detection problem*). Find a subset S of V containing at most B nodes to maximize a prespecified reward function $F(S)$ without the complete knowledge of G. Here, $B < n$ is a given budget, and $F : 2^V \mapsto \mathbb{R}_{\geqslant 0}$ is a non-decreasing submodular function, such that $F(S) \geqslant 0, \forall S \subseteq V$ and $F(\emptyset) = 0$.

If a set function F satisfies $F(S)+F(T) \geqslant F(S \cup T) + F(S \cap T)$ for all $S, T \subseteq V$, then F is *submodular* [44]. The submodular objective function captures the notion of a utility function with diminishing returns, and it arises naturally in many applications [14,17–19]. As an example (one which is used in our experiments), we consider

$$F(S) = \sum_{c \in C(S)} \frac{size(c)}{1 + \min_{u \in S}\{t_{uc} - t_c\}}, \quad (1)$$

where $C(S)$ denotes the set of cascades joined by users in S. $F(S)$ yields large values when users in S join many important cascades with small time delay. It is easy to verify that Eq. (1) meets all of the conditions in Definition 1.

The above definition contains a major difference from those in existing works [18,19]; i.e., here, we do not have the complete knowledge of G, neither topology nor cascades. In practice, we are only allowed to explore G by querying each user by account ID one at a time through the OSN APIs, e.g., `SELECT * FROM USERS WHERE UID = ID`, equivalently.[2] When a user u is queried, the cascades he has joined and his neighbours are returned simultaneously. We assume a query incurs one unit query cost. Therefore, existing works [18,19] actually implicitly assume that we have collected the complete data (by traversing the whole account ID space). As a result, their methods are often unfeasible and suffer from drastically expensive query costs.

3.3. Submodularity and greedy algorithm (GA)

Even when the complete data are available, maximizing a submodular function is proven to be NP-complete [44,14]. The non-decreasing submodular property of F allows us to use a greedy algorithm (GA) to obtain an approximate solution that is at least $1 - 1/e \approx 63\%$ of the optimal solution [44]. GA is well studied and it can be stated as follows. It runs for at most B rounds to obtain a set S of size $|S| \leqslant B$. In each round, it selects a node $s \in V \setminus S$ that maximizes the *reward gain*, $\delta_s(S) \triangleq F(S \cup \{s\}) - F(S)$, and inserts s into S in this round. This process repeats for at most B rounds until $|S| = B$ or $\delta_s(S) = 0$.[3] The computational complexity of this algorithm is $O(nB)$. However, for a very large population n, even GA is still inefficient.

To improve its efficiency, Minoux [33] proposed the *Accelerated Greedy* (AG) approach to reduce the calculations of $\delta_s(S)$ in each round by further utilizing the submodularity of F (also called the *Cost-Effective Lazy Forward* approach, or CELF, in [18]). The basic idea is that the reward gain of a node in the current round cannot be higher than its gain in previous rounds, i.e., if $k > l$, then $\delta_s(S_k) \leqslant \delta_s(S_l),\ \forall s \in V \setminus S_k$ (where S_k is the set of selected nodes after round k). Unfortunately, AG and CELF do *not* guarantee an improvement in computational efficiency, and, in the worst case, it is as inefficient as the naive greedy approach [33]. To make matters even worse, GA is not suitable for our setting, as it requires complete knowledge of G, which is practically impossible to acquire for real-world OSNs.

3.4. Two ground-truth datasets as testbeds

Before we move forward, we introduce two real-world datasets that will be used as ground-truth data in this work (see Table 1). Sina Weibo (http://weibo.com), which is similar to Twitter, is one of the most popular microblogging sites in China. We collect a small portion of the Weibo network using the breath-first-search method along the follow relationships. We store the tweets of each user that were tweeted between January 1, 2012 and September 1, 2012. URL links and hashtags contained in these tweets are extracted and considered the representation of cascades. In other words, if two tweets tweeted by two users contain the same hashtag or URL link, then the two users have joined the same cascade. Another dataset is from Twitter. This dataset contains a large fraction of network and tweet data from Twitter in June 2009, which are from [1,45] respectively. Similar to our handling of Weibo, URL links and hashtags contained in tweets are extracted to form cascades.

4. Motivation for a randomized framework and empirical evidence

In this section, we first discuss the motivation behind a randomized framework, which is simple at this stage and will be enriched in the following sections. Later, we conduct measurements on two real-world datasets to support our claim.

4.1. A randomized framework

We consider a simple randomized framework comprising the following two steps:

Step 1: *A set of nodes is randomly sampled from the network.*
Step 2: *Followees are chosen from these sampled nodes.*

If we sample all of the nodes in the network in Step 1, the above framework becomes solving the original problem with complete data. Therefore, this framework is general. Here, we are interested in how Step 1 impacts the quality of solutions obtained in Step 2. Suppose the optimal followees are uniformly distributed in the network; then, after Step 1, these optimal nodes are included in samples (uniformly sampled) with the same probability. As we will show in the next section, we can prove that the final solution quality is lower bounded for this case. However, if we are able to sample the optimal nodes with higher probability than that of uniform sampling, then the quality of samples obtained during Step 1 will be improved, and we can obtain a much-improved solution in Step 2.

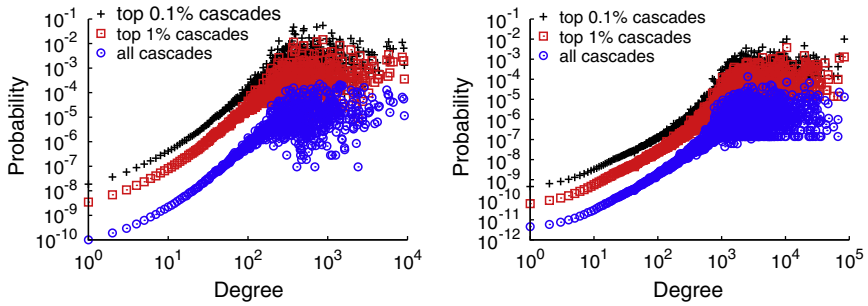
Here, we claim that good followees are correlated with nodes of high centrality in networks. Therefore, they can be sampled with higher probability if we prefer to sample high centrality nodes. The reason for this is that high

[2] In Sina Weibo, given a user ID, we obtain his tweets and neighbours via accessing http://weibo.com/u/ID.

[3] Once $\delta_s(S) = 0$, GA obtains the optimal solution [44].

Table 1
Ground-truth datasets.

Dataset	Nodes	Edges	Cascades	Time
Sina Weibo	339,130	1,697,888	11,439,756	January–August 2012
Twitter	1,705,243	21,639,326	7,077,596	June 2009



(a) SinaWeibo (b) Twitter

Fig. 1. Probability of a user joining a cascade, in the top 0.1%, 1% largest and all cascades, respectively.

centrality nodes are located at important positions of the network and they are more likely to participate in cascades. Although there are many node centrality measures [46], we use the degree centrality due to its simplicity. To support our claim, we conduct measurements on two real-world datasets in the follow subsection.

4.2. Empirical observations on Sina Weibo and Twitter

We first show that high degree nodes can detect many cascades. We measure the probability of a node with degree d joining a cascade, which can be expressed as follows

$$\text{Probability}(d) = \frac{1}{n_d|C'|}\sum_{c\in C',u\in V}\mathbf{1}\{t_{uc} < \infty \wedge d_u = d\}.$$

Here, $\mathbf{1}\{\cdot\}$ is the indicator function, n_d is the number of nodes with degree d, d_u denotes the degree of node u, and $C' \subseteq C$. We set C' to be the top 0.1% largest cascades, top 1% largest cascades and all cascades. Fig. 1 shows the probability with respect to degree d. From the figure, it is clear that higher degree nodes are more likely to join cascades than smaller degree nodes. Therefore, choosing higher degree nodes as followees can result in the discovery of more cascades.

We next show that high degree nodes can discover cascades in a timely fashion. We measure the probability of a node with degree d joining a cascade within some time delay Δt. That is

$$\text{Probability}(d) = \frac{1}{n_d|C|}\sum_{c\in C,u\in V}\mathbf{1}\{t_{uc} - t_c \leqslant \Delta t \wedge d_u = d\}.$$

We set Δt to be 1 day, 1 h, and 30 min. Fig. 2 shows the results. Generally speaking, high degree nodes are more likely to join cascades earlier than small degree nodes (although with a large variance). Therefore, high degree nodes can discover cascades in a timely manner.

In conclusion, empirical evidence indicates that there does exist a correlation between good information sources and high centrality nodes in OSNs; therefore, our previous claim is supported. Now, we are ready to show that such a simple randomized framework can guarantee both solution quality and computational efficiency due to submodularity of the reward function and the birthday paradox. The detailed analysis will be covered in the next section.

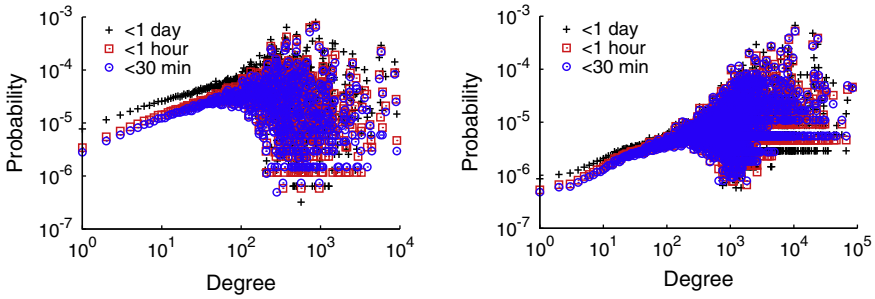
5. Analysing the randomized framework under the worst-case situation

In this section, we study the solution quality and computational efficiency of the randomized framework introduced in the previous section. Our purpose is to demonstrate that this simple randomized framework can lower bound the solution quality and guarantee computational efficiency. We analyse its solution quality lower bound by randomizing a well-studied greedy algorithm (GA) and show its computational efficiency by exploiting the birthday paradox.

5.1. Randomizing the greedy algorithm

Because we are allowed to query nodes in an OSN through their account IDs, if we know the scope of the account ID range,[4] we can randomly generate test IDs in this range and easily test their validity by polling them on OSNs. This way, we can construct a set of node samples $X \subseteq V$ with

[4] For example, a valid Weibo user has an identity code of 10 digits ranging from "1000000000" to "5058913818" by March 25, 2014.



(a) Sina Weibo (b) Twitter

Fig. 2. Probability of a user joining a cascade within 1 day, 1 h and 30 min, respectively.

sufficient size. To imitate GA, we consider executing Steps 1 and 2 iteratively, and this forms our *randomized greedy algorithm (RG)*, which is described in Algorithm 1.

Algorithm 1. Randomized greedy algorithm (RG)

Input: Budget B.
Output: Selected nodes S.
$S = \emptyset$;
while $|S| < B$ **do**
  Generate node samples $X \subseteq V \setminus S$;
  Choose a node $s^* = \arg\max_{s \in X} \delta_s(S)$;
  if $\delta_{s^*}(S) > 0$ **then** $S = S \cup \{s^*\}$;
  else break;
end

In the original GA, at round k, one selects a node s_k^* from $V \setminus S_{k-1}$ to maximize the reward gain $\delta_s(S_{k-1})$. In RG, at each round k, we generate a set of node samples $X_k \subseteq V \setminus S_{k-1}$ and select s_k^* from X_k to maximize $\delta_s(S_{k-1})$. As $X \to V \setminus S$ in each round,[5] the performance of RG can be arbitrarily close to that of GA.[6] RG has two advantages over GA: (1) it does not require complete knowledge of G; and (2) it is $n/|X|$ times faster than GA.

It is known that GA has an approximation factor $1 - 1/e$, i.e., the value of the GA solution is at least $1 - 1/e \approx 0.63$ times the optimal value. The following theorem states that RG has a similar performance bound.

Theorem 1 (*Lower bound on solution quality of RG*). *Let OPT denote $K \leqslant B$ optimal nodes of the problem, and $S_k = \{s_1^*, \ldots, s_k^*\}$ be the set of nodes obtained by RG after k rounds, $1 \leqslant k \leqslant B$. Suppose X is uniformly sampled from V; then, there exists a constant $\lambda \geqslant |X|/n$, s.t.*

$$\mathbb{E}[F(S_B)] \geqslant \mathbb{E}[F(S_K)] \geqslant \left(1 - \frac{1}{e^{\lambda}}\right) F(OPT).$$

Proof. By utilizing the non-decreasing property of reward function F, we have

$$\begin{aligned} F(OPT) - F(S_{k-1}) &\leqslant F(OPT \cup S_{k-1}) - F(S_{k-1}) \\ &= F(OPT \setminus S_{k-1} \cup S_{k-1}) - F(S_{k-1}). \end{aligned} \tag{2}$$

Assume $OPT \setminus S_{k-1} = \{z_1, \ldots, z_J\}, J \leqslant K$, let $j = 1, \ldots, J$, and define

$$Z_j \triangleq F(S_{k-1} \cup \{z_1, \ldots, z_j\}) - F(S_{k-1} \cup \{z_1, \ldots, z_{j-1}\}). \tag{3}$$

Then Ineq. (2) becomes

$$F(OPT) - F(S_{k-1}) \leqslant \sum_{j=1}^{J} Z_j. \tag{4}$$

In order to obtain the expectation of $F(S_{k-1})$, we require the expectation of Z_j. According to Eq. (3) and the submodularity of F, we have

$$\mathbb{E}[Z_j] \leqslant \mathbb{E}[F(S_{k-1} \cup \{z_j\}) - F(S_{k-1})] = \mathbb{E}\left[\delta_{z_j}(S_{k-1})\right]. \tag{5}$$

In fact, $\mathbb{E}\left[\delta_{z_j}(S_{k-1})\right]$ is upper bounded by $\mathbb{E}\left[\delta_{s_k^*}(S_{k-1})\right]/\lambda$ where $0 < \lambda \leqslant 1$ is a constant. To see this, we define two sets C_1 and C_2 by

$$C_1 \triangleq \{X : \exists x \in X \wedge \delta_x(S_{k-1}) \geqslant \delta_{z_j}(S_{k-1})\} \quad \forall j,$$
$$C_2 \triangleq \{X : \forall x \in X \wedge \delta_x(S_{k-1}) < \delta_{z_j}(S_{k-1})\} \quad \forall j.$$

Let $\lambda \triangleq P(X \in C_1)$, and note that

$$\lambda = P(X \in C_1) \geqslant P(z_j \in X) = \frac{|X|}{n - k + 1} \geqslant \frac{|X|}{n}. \tag{6}$$

Now,

$$\begin{aligned} \mathbb{E}\left[\delta_{s_k^*}(S_{k-1})\right] &= \mathbb{E}[F(S_{k-1} \cup \{s_k^*\}) - F(S_{k-1})] \\ &= P(X \in C_1)\mathbb{E}[F(S_{k-1} \cup \{s_k^*\}) | X \in C_1] \\ &\quad + P(X \in C_2)\mathbb{E}[F(S_{k-1} \cup \{s_k^*\}) | X \in C_2] \\ &\quad - \mathbb{E}[F(S_{k-1})] \\ &= P(X \in C_1)\mathbb{E}\left[\delta_{s_k^*}(S_{k-1}) | X \in C_1\right] \\ &\quad + P(X \in C_2)\mathbb{E}\left[\delta_{s_k^*}(S_{k-1}) | X \in C_2\right] \\ &\geqslant P(X \in C_1)\mathbb{E}\left[\delta_{s_k^*}(S_{k-1}) | X \in C_1\right] \\ &\geqslant \lambda\mathbb{E}\left[\delta_{z_j}(S_{k-1})\right]. \end{aligned}$$

Finally, from Eq. (5) we get

$$\mathbb{E}[Z_j] \leqslant \mathbb{E}\left[\delta_{z_j}(S_{k-1})\right] \leqslant \frac{1}{\lambda}\mathbb{E}\left[\delta_{s_k^*}(S_{k-1})\right].$$

[5] We sometimes omit subscript k if there is no ambiguity.
[6] However, we cannot guarantee whether RG is better or worse than GA due to the approximation nature of GA.

Combining with Eq. (4), we get the following iterative formula,

$$F(OPT) - \mathbb{E}[F(S_{k-1})] \leqslant \sum_{j=1}^{J} \mathbb{E}[Z_j] \leqslant \frac{K}{\lambda} \mathbb{E}[F(S_k) - F(S_{k-1})],$$

from which we can derive the following relationship,

$$\mathbb{E}[F(S_k)] \geqslant \left[1 - \left(1 - \frac{\lambda}{K}\right)^k\right] F(OPT).$$

Finally, letting $k = K$, and using the non-decreasing property of F, we have

$$\mathbb{E}[F(S_B)] \geqslant \mathbb{E}[F(S_K)] \geqslant \left[1 - \left(1 - \frac{\lambda}{K}\right)^K\right] F(OPT) \geqslant \left(1 - \frac{1}{e^{\lambda}}\right) F(OPT). \quad \square$$

Theorem 1 exploits the non-decreasing submodular property of the reward function and illustrates that the proposed randomized framework can obtain quality guaranteed solutions when samples are uniformly picked from the OSN. Due to the lack of complete data, the solution quality lower bound for RG is smaller than that of GA, and, more importantly, Theorem 1 describes the method to improve the solution quality of RG. The performance lower bound is related to the parameter λ, which relates to the probability of including an optimal node in the sample set X (see Ineq. (6)). Therefore, if we can increase this probability, we can obtain better solutions. As discussed in Section 4, this can be achieved by including high centrality nodes (e.g., high degree nodes) in samples because high centrality nodes are more likely to be optimal nodes; then, the solution quality lower bound of RG increases.

Because RG is $n/|X|$ times faster than the naive GA, a too-large sample size will harm RG's computational efficiency. In the following, we show that the sample size needs not be very large according to the birthday paradox argument.

5.2. Quantifying the sample size via cover ratio analysis

To determine the sample size, we need a relation between $F(S)$ and $|X|$. However, establishing such a relation is non-trivial. Here, we determine $|X|$ by quantifying the overlap between samples $\bigcup_{k=1}^{K} X_k$ and OPT. Let η denote the ratio of nodes in OPT covered by samples $\bigcup_{k=1}^{K} X_k$, i.e., $\eta \triangleq \frac{1}{K}|\bigcup_{k=1}^{K} X_k \cap OPT|$. Intuitively, if the samples used in RG can cover a large fraction of nodes in OPT, we can find a good solution with high probability. To show that RG does guarantee a lower bound for the cover ratio η using only moderate-sized samples, we first study the probability that at least x nodes in the set OPT fall into set X in one round of RG. Remember that $|V| = n$ and $|OPT| = K$; therefore,

$$Prob\{|X \cap OPT| \geqslant x\} = \sum_{i=x}^{K} \frac{\binom{K}{i}\binom{n-K}{|X|-i}}{\binom{n}{|X|}}.$$

Let $x = 1$ and the expression above becomes

$$Prob\{|X \cap OPT| \geqslant 1\} = 1 - (1 - K/n)^{|X|}.$$

If we want to guarantee that the above probability is at least p, i.e., that X can cover at least one node in OPT with probability at least p, we finally obtain

$$|X| \geqslant \left\lceil \frac{\ln(1-p)}{\ln(1-K/n)} \right\rceil.$$

Fig. 3 shows the relation between the smallest sample size $|X|$ and K/n with $p = 0.90, 0.95$ and 0.99. One interesting observation is that sample size drops quickly when K/n varies from 0 to 0.01; this shows that we do *not* need very large samples if OPT is moderate in size. For example, if we want to make sure that at least one node in OPT falls in X when $K/n = 0.01$, then one only needs to generate 458 samples, and the resulting success probability is greater than 0.99. This is a counter-intuitive result that is related to the birthday paradox.

The above result can be generalized to the situation of covering at least one node in a subset of a set of size n, where the subset contains a fraction α of the nodes of the set. To guarantee that this event occurs with probability at least p, we can use $n_X(\alpha, p)$ samples, where

$$n_X(\alpha, p) = \left\lceil \frac{\ln(1-p)}{\ln(1-\alpha)} \right\rceil.$$

Now we show that RG can bound the cover ratio η with only moderate-sized samples in each round. It is important to note that as the RG algorithm proceeds, the fraction of uncovered nodes in OPT shrinks, and therefore one needs more samples to guarantee the cover probability p. Nevertheless, the following theorem shows that even if we use fixed size samples in each round, the cover ratio η after K rounds is still bounded.

Theorem 2 (*Lower bound on the cover ratio*). *Given α and p, if we use $n_X(\alpha, p)$ samples in each round of RG to cover K optimal nodes, then, after K rounds in RG, we have*

$$\mathbb{E}[\eta] \geqslant \begin{cases} 1 - \frac{1}{e^{p\beta}}, & \beta \leqslant 1, \\ 1 - \frac{1}{\beta e^{1-(1-p)\beta}}, & \beta > 1, \end{cases} \quad \text{where} \beta = \frac{K}{\alpha n}.$$

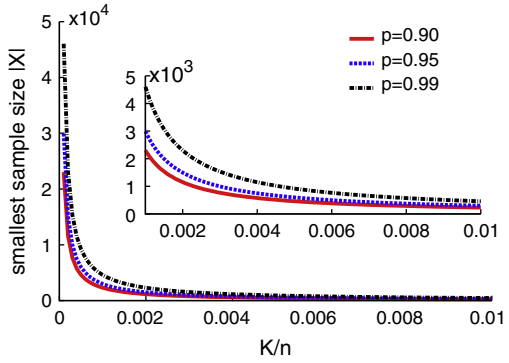


Fig. 3. The smallest sample size. Inset shows the same plot ranging from 0.001 to 0.01.

Proof. Let y_k denote the number of nodes in *OPT* that have been covered after the k-th round and $\bar{y}_k$ be its expectation. Further, let q_k denote the probability that an uncovered node in *OPT* will be covered in the k-th round. Then $\bar{y}_k$ satisfies

$$\bar{y}_k = \bar{y}_{k-1} + q_k. \tag{7}$$

- **Case 1:** $\beta \leqslant 1$

  In this case, $\alpha n \geqslant K$, it means that using $n_X(\alpha, p)$ samples can only guarantee that the probability of covering a larger set of size $\alpha n \geqslant K$ is larger than or equal to p. At round k, an uncovered node in *OPT* belongs to a set of size $\alpha n = K/\beta$ with probability $(K - \bar{y}_{k-1})/\alpha n = \beta(K - \bar{y}_{k-1})/K$. Therefore,

  $$q_k \geqslant p\beta \frac{K - \bar{y}_{k-1}}{K}.$$

  Substituting this into Eq. (7), yields

  $$\bar{y}_k \geqslant \bar{y}_{k-1} + p\beta \frac{K - \bar{y}_{k-1}}{K},$$

  which can be solved to yield

  $$\begin{aligned} \bar{y}_k &\geqslant \left(1 - \frac{p\beta}{K}\right)^k \bar{y}_0 + K\left[1 - \left(1 - \frac{p\beta}{K}\right)^k\right] \\ &= K\left[1 - \left(1 - \frac{p\beta}{K}\right)^k\right] \geqslant K\left(1 - \frac{1}{e^{kp\beta/K}}\right). \end{aligned} \tag{8}$$

  Setting $k = K$, yields

  $$\mathbb{E}[\eta] = \frac{\bar{y}_K}{K} \geqslant 1 - \frac{1}{e^{p\beta}}.$$

- **Case 2:** $\beta > 1$

In this case, using $n_X(\alpha, p)$ samples can guarantee the probability of covering a set of size $\alpha n < K$. Since $|OPT| = K$, for the first few rounds in RG, we are always able to cover at least one node in *OPT* with probability greater than or equal to p, until $|OPT \setminus S_{k^*}| = \alpha n$, where k^* can be determined by

$$k^* = \frac{K - \alpha n}{p} = \frac{K(\beta - 1)}{\beta p}.$$

Hence, in the first k^* rounds, we have covered $K - \alpha n$ nodes in *OPT* with probability at least p. After k^*, we can use only $K - k^*$ rounds to cover the remaining $\alpha n = K/\beta$ nodes with sample size $n_X(\alpha, p)$. This situation has been discussed in Case 1 (replacing K by K/β and k by $K - k^*$ in (8)). Therefore,

$$\bar{y}_k \geqslant K - \frac{K}{\beta} + \frac{K}{\beta}\left(1 - \frac{1}{e^{1-(1-p)\beta}}\right) = K - \frac{K}{\beta e^{1-(1-p)\beta}}.$$

Hence,

$$\mathbb{E}[\eta] = \frac{\bar{y}_K}{K} \geqslant 1 - \frac{1}{\beta e^{1-(1-p)\beta}}. \quad \square$$

We depict the relation between $\mathbb{E}[\eta]$ and β in Fig. 4. When β increases (or α decreases and, therefore, sample size increases), we observe that the cover ratio increases. As an illustration, to cover a set containing K nodes, if we set $\alpha = K/n$ and $p \approx 1$, we can obtain a cover ratio of 63%; if we reduce α to $K/(2n)$, the cover ratio increases to 82%. In practice, we can use Theorem 2 to determine the sample size guaranteeing a proper coverage on *OPT*, and the sample size needs not be too large. Therefore, our randomized framework is computationally efficient.

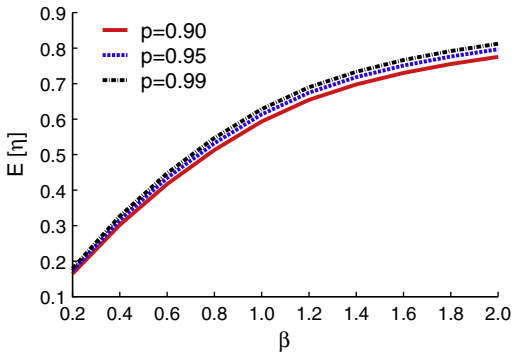


Fig. 4. Cover ratio lower bound.

In conclusion, our analysis in this section demonstrates that the randomized framework can guarantee both solution quality and computational efficiency under the condition that user samples are sampled uniformly at random. Note that this condition actually is the worst-case situation because we do not use any strategy in sampling nodes. In fact, we can explore structural properties of OSNs to obtain high quality samples, which we discuss in the next section.

6. Benefiting from power-law networks

Power-law networks (i.e., the fraction of degree-d nodes in the graph is proportional to $d^{-\gamma}$, where γ is a constant) are very common in real world such as online social networks, biological networks, citation networks and communication networks. In this section, our goal is to show that power-law structures can bring extra benefits to the randomized framework by investigating the patterns of cascading diffusion processes on power-law networks. Our result reveals that power-law networks can easily cause *collisions* between diffusion processes and a simple random walk sampler when the network structure satisfies some mild conditions, i.e., *the network has high degree moments*. Therefore, the result indicates that the randomized framework in conjunction with a random walk sampler can detect information cascades effectively in such networks.

6.1. Diffusion models

Our analysis is focused on three diffusing models for they generalize a large variety of diffusion processes in real world. We briefly describe them here.

- *Random Walk (RW) model.* Petition letter delivering [47] and drug smuggling [48] are examples of diffusions that the total amount of the diffusible objects does not change over time (e.g., there is always one

letter or one package of drug on the network while diffusing). We model this kind of diffusion, called *conservative diffusion* [49], by a simple random walk. That is, a diffusible object starts from an initial infected node, and recursively, it infects a randomly chosen neighbour and moves to that node at the next step.

- *Independent Cascade (IC) model.* The IC model is widely used to model information cascades in social networks [17,43], such as tweets retweeting and videos sharing in OSNs. In the IC model, each cascade starts from an initial seed. When node u first becomes active at step t, it is given only one chance to activate its neighbours $Nb(u)$ with success probability p_{uv} where $v \in Nb(u)$. If u succeeds in activating v, then v will become active at step $t+1$; but whether u succeeds, it cannot make any further attempts to activate its neighbours.
- *Susceptible-Infective (SI) model.* The SI model is widely used in epidemiology [50, Chapter 9]. In the network scenario, we consider a variation of the common SI model. In this model, at each time step, an edge $(u, v) \in E$, that connects an infective node u and a susceptible node v, is chosen uniformly at random; then, node v becomes infected at this time step. This process repeats until all nodes are infected. The difference between the SI model and the IC model is that, an infected node in the SI model can keep on infecting its susceptible neighbours.

6.2. Observations from diffusion models

To motivate our further analysis, we first simulate these models on the HEPTH citation network (refer to Table 2). The fractions of infected nodes by diffusions are controlled: for the RW and SI model, we stop the diffusions after having infected 1%, 5% and 10% nodes; for the IC model, we set $p_{uv} = 0.01, 0.012$ and 0.015 on each edge. We depict the degree distribution of the original graph, and compare it with the degrees of the uninfected nodes after diffusions in Fig. 5. We observe that when more nodes are infected, the tails of CCDF curves drop more quickly than the heads. This indicates that large degree nodes are easier to be infected than small degree nodes, which is consistent with our claims in Section 4. Therefore, if a sampler can sample large degree nodes with higher probability, it has higher chance to discover a cascading diffusion. Random walk (RW) is such a sampler that prefers to visit large degree nodes in networks. In the following discussion, we theoretically show why a RW sampler can effectively discover diffusions in power-law networks, and one condition that the network should satisfy.

Table 2
Summary statistics of networks.

Network	HEPTH	Enron	Slashdot	Gnutella
Nodes	27,400	33,696	77,360	62,586
Edges	352,040	180,811	507,833	147,892

6.3. Using A RW sampler to probe diffusions

Let I denote the set of infected nodes, and $I_d \subseteq I$ is the set of infected nodes of degree d. Let R denote the set of nodes visited by a RW sampler, and $R_d \subseteq R$ is the set of visited nodes of degree d by RW. Let i, i_d, r and r_d denote their cardinalities respectively. Then the probability that a RW sampler fails to discover the diffusion is

$$P(R \cap I = \emptyset) = \prod_d P(R_d \cap I_d = \emptyset). \tag{9}$$

Furthermore, remember that n_d is the number of nodes of degree d in the graph. We obtain

$$P(R_d \cap I_d = \emptyset) = \begin{cases} 0 & r_d + i_d > n_d, \\ \dfrac{\binom{n_d - i_d}{r_d}}{\binom{n_d}{r_d}} & \text{otherwise.} \end{cases}$$

Substituting it to Eq. (9), we have

$$P(R \cap I = \emptyset) \leqslant \prod_d \frac{\binom{n_d - i_d}{r_d}}{\binom{n_d}{r_d}} \leqslant \prod_d \left(1 - \frac{i_d}{n_d}\right)^{r_d}. \tag{10}$$

A RW sampler visits a node of degree d with probability $d\theta_d/\langle d\rangle$ in G, where θ_d is the fraction of nodes of degree d in the graph, and $\langle d\rangle$ denotes the average degree of nodes in G. For an l-length random walk, it contains $ld\theta_d/\langle d\rangle$ samples of degree d. Note that these samples may have duplicates because a node may be visited more than one time by the random walker. Nevertheless, it can be proven that the number of duplicated nodes can be ignored when l is small [51], i.e., $l \approx r$. Therefore, $r_d \approx ld\theta_d/\langle d\rangle$, and Eq. (10) becomes

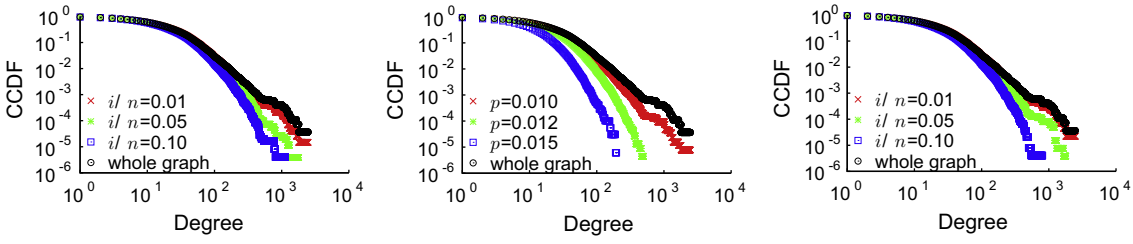
$$P(R \cap I = \emptyset) \leqslant \prod_d \left(1 - \frac{i_d}{n_d}\right)^{ld\theta_d/\langle d\rangle}.$$

If we want to guarantee that the above probability is smaller than ϵ, we have

$$l \geqslant \frac{\log \epsilon}{\sum_d \frac{d\theta_d}{\langle d\rangle} \log\left(1 - \frac{i_d}{n_d}\right)} = \frac{\log \epsilon}{\sum_d \frac{d\theta_d}{\langle d\rangle} \log\left(1 - \frac{iq_d}{n\theta_d}\right)} \triangleq l_{\text{low}}, \tag{11}$$

where $q_d = i_d/i$, is the fraction of infected nodes of degree d.

Consequently, l_{low} is the minimum number of steps a walker should walk. If l_{low} is small, we can conclude that the RW sampler is effective to discover the cascading diffusions. Note that several factors will impact l_{low}: (a) the degree distribution of G, i.e., $\{\theta_d\}_{d>0}$; (b) the fraction of infected nodes i/n; and (c) $\{q_d\}_{d>0}$. Among them, we are most interested in (c). $q_d = i_d/i$ describes the fraction of infected nodes of degree d, and it is closely related to the nature of a diffusion model. We refer $\{q_d\}_{d>0}$ as the *diffusion profile* of a diffusion model. In order to understand the value of l_{low}, we need to calculate the diffusion profiles for different models.



(a) RW (b) IC (c) SI

Fig. 5. An example on the HEPTH network. Degree distributions of the original graph, and uninfected nodes in the graph under different diffusion models (averaged over 50 runs).

6.3.1. Diffusion profile for the RW model

The RW model has a very simple diffusion profile, i.e., $q_d^{(\mathrm{RW})} = d\theta_d/\langle d\rangle$. Using the Taylor series $\log(1-x) = -\sum_{j=1}^{\infty} x^j/j$, for $0 \leqslant x < 1$, Eq. (11) becomes

$$l_{\mathrm{low}}^{(\mathrm{RW})} = \frac{-\log\epsilon}{\sum_{j=1}^{\infty}\frac{1}{j}\left(\frac{i}{n}\right)^j \sum_d \frac{d\theta_d}{\langle d\rangle}\left(\frac{q_d^{(\mathrm{RW})}}{\theta_d}\right)^j} = \frac{-\log\epsilon}{\sum_{j=1}^{\infty}\frac{1}{j}\left(\frac{i}{n}\right)^j \frac{\langle d^{j+1}\rangle}{\langle d\rangle^{j+1}}}.$$

For power-law networks, we know that $\langle d^j\rangle$ converges only when scaling exponent $\gamma \geqslant j+1$. Empirically, it is known that $2 \leqslant \gamma < 3$ in real-world networks [52,53]. This indicates that $\langle d^{j+1}\rangle$, $j \geqslant 1$, will increase drastically as the maximum degree of the network increases, and causes $l_{\mathrm{low}}^{(\mathrm{RW})}$ to be small in real-world networks.

6.3.2. Diffusion profile for the IC model

Next we provide a mean-field analysis for $\{q_d^{(\mathrm{IC})}\}_d$ to facilitate numerical calculation on a given network since there is no closed form solution for $q_d^{(\mathrm{IC})}$ and $l_{\mathrm{low}}^{(\mathrm{IC})}$ in the case of IC model.

For the IC model, uninfected nodes can only be infected by active nodes (i.e., nodes become infected at last step). Let $p_d(t)$ be the fraction of active nodes of degree d at step t. By definition, we have $p_d(t) = [i_d(t) - i_d(t-1)]/n_d$. Under the configuration model [54], the probability, that a node of degree d is connected to a node of degree h, is $p_{dh} = dh/(2m)$. Then a node of degree d will have $n_h p_{dh}$ neighbours of degree h on average. (It is easy to verify that $\sum_h n_h p_{dh} = d$.) Since a node of degree h got infected at step t has probability $p_h(t)$, a node of degree d will have on average $n_h p_{dh} p_h(t)$ active neighbours of degree h at step t. Thus, a node of degree d becomes infected at step $t+1$ with probability $1 - \prod_h (1-p)^{n_h p_{dh} p_h(t)}$ (assume $p_{uv} = p, \forall u, v$). Finally, we obtain the probability that a randomly chosen node of degree d becomes active at step $t+1$ by

$$p_d(t+1) = \left(1 - \frac{i_d(t)}{n_d}\right)\left(1 - \prod_h (1-p)^{n_h p_{dh} p_h(t)}\right), \tag{12}$$

where the first item $1 - i_d(t)/n_d$ is the probability that a randomly chosen node of degree d is not infected at previous steps. Thus far, we obtain the total infected nodes till step $t+1$ by

$$i_d(t+1) = i_d(t) + n_d p_d(t+1). \tag{13}$$

The fraction of nodes of degree d in the infected nodes at step t, i.e., the diffusion profile, can be readily obtained by

$$q_d^{(\mathrm{IC})}(t) = \frac{i_d(t)}{\sum_d i_d(t)}. \tag{14}$$

Combining Eqs. (12)–(14), we can numerically calculate $q_d^{(\mathrm{IC})}$ for a given graph with initial condition $p_d(t)|_{t=0}$ given, and $l_{\mathrm{low}}^{(\mathrm{IC})}$ is obtained by Eq. (11).

6.3.3. Diffusion profile for the SI model

In the SI model, because an infectious node can keep on infecting the susceptible neighbours at every time step, then the probability, that a node of degree d is infected at step t, is $p_d(t) = i_d(t)/n_d$. At time step $t+1$, an edge $(u, v) \in E$ s.t. $u \in I(t)$ and $v \in V \setminus I(t)$ is chosen uniformly at random. According to the inspection paradox [55], the probability that node v has degree d is

$$\rho_d(t+1) = \frac{d\theta_d(1 - p_d(t))}{\sum_h h\theta_h(1 - p_h(t))}. \tag{15}$$

Then, the fraction of infected nodes of degree d after step $t+1$ is

$$p_d(t+1) = p_d(t) + \frac{\rho_d(t+1)}{n_d}. \tag{16}$$

Finally, the diffusion profile of SI model is

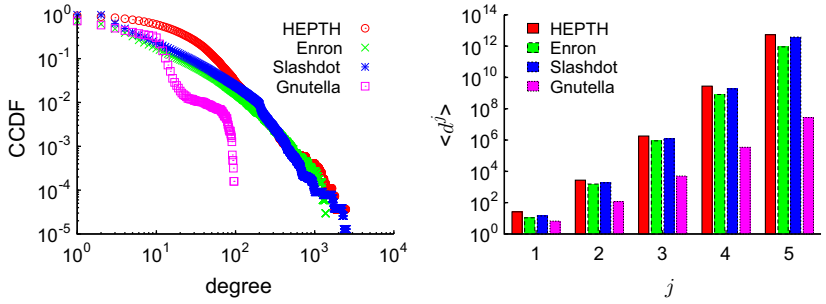
$$q_d^{(\mathrm{SI})}(t) = \frac{n_d p_d(t)}{\sum_h n_h p_h(t)}. \tag{17}$$

Combining Eqs. (15)–(17), we can numerically calculate $q_d^{(\mathrm{SI})}$ for a given graph with given initial condition $p_d(t)|_{t=0}$, and $l_{\mathrm{low}}^{(\mathrm{SI})}$ will be obtained by Eq. (11) thereafter.

6.4. Numerical solutions for the minimum number of steps

We consider different networks such as citation network HEPTH, communication network Enron, social network Slashdot, and P2P technology network Gnutella, and we numerically calculate l_{low} on these networks. A brief summary of these networks is given in Table 2. All the four networks reveal power-law degree distributions (Fig. 6(a)), but with different moment distributions (Fig. 6(b)). Note that Gnutella has smaller moments than the others.

For each network and diffusion model, Fig. 7 depicts l_{low} with respect to the fraction of infected nodes i/n under three different ϵ's. We can observe that when more nodes become infected (i.e., i/n increases), RW sampler is easier to discover the diffusion process (i.e., l_{low} drops quickly).



(a) Degree distribution. (b) j-th order moments in degree.

Fig. 6. Degree and moment distributions.

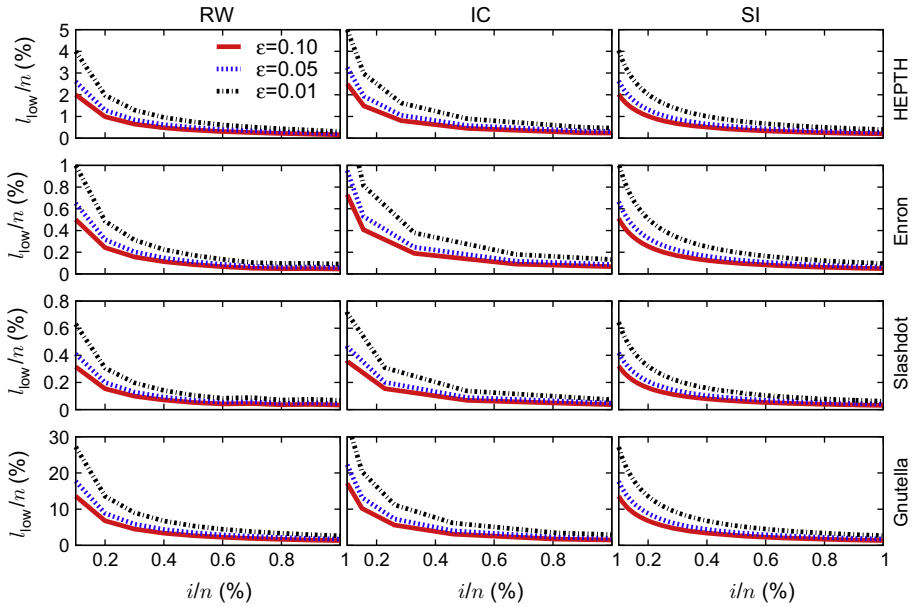


Fig. 7. Numerical solutions for l_{low} with different diffusion models. We use initial condition $p_1(0) = 1/n_1$ and $p_d(0) = 0$ for $d > 1$ in both IC and SI model. For the IC model, we set $p_{uv} = 0.02$ in the first three graphs and $p_{uv} = 0.1$ in Gnutella. Note that l_{low}/n is relatively large on Gnutella network.

Comparing the Gnutella network with the other networks, we find that the sampler needs relatively longer steps to discover the diffusion process because the Gnutella network has smaller moments than the others (as shown in Fig. 6(b)).

In summary, we demonstrate that the RW sampler can easily discover cascading diffusion processes in power-law networks with large degree moments, and actually, this condition is easily satisfied in real-world OSNs due to the divergences of $\langle d^j \rangle, j \geqslant 2$ when $2 \leqslant \gamma < 3$. Therefore, the randomized framework in conjunction with a RW sampler can detect information cascades effectively in real-world OSNs.

7. Experiments

In this section, we conduct experiments to verify the previous analysis. The goal is to demonstrate the efficiency of the randomized method and the trade-off between solution quality and computational efficiency. We first introduce our evaluation methods, and then analyze the experimental results based on the two ground-truth datasets which we have introduced in Section 3.

7.1. Evaluation methods

7.1.1. Solution quality evaluation method

Because it is difficult to obtain an optimal solution *OPT*, we compare our solution quality with the solution obtained by GA on complete ground-truth data. Since our method only uses incomplete data, its solution quality is usually worse than that of GA. Thus, we mainly evaluate how close our method can approximate GA, e.g., within 90% or 95% to GA.

7.1.2. Computational efficiency evaluation method

We use the number of times of calculating reward gains to measure the time complexity of a method. The naive GA

requires $O(nB)$ calculations, and hence we can evaluate the speed-up of a method A in terms of the reduction of reward gain calculations with respect to naive GA, by

$$\text{Speed-up}(A) = \frac{\#\text{of calculations of reward gain by naive GA}}{\#\text{of calculations of reward gain by method } A}.$$

We will compare RG with two state-of-the-art methods AG and CELF, which are two most efficient implementations of GA. Note that AG and CELF are actually the same, so we only compare RG with AG.

7.2. Experiments on Sina Weibo and Twitter

7.2.1. Evaluating RG under the worst-case situation

In the first experiment, we implement RG as described in Algorithm 1. In each round of RG, we use a smaller sample size $n_X(K/n, 0.9)$ at each round that guarantees about 59% of coverage on *OPT*, and a larger sample size $n_X(K/(2n), 0.9)$ at each round that guarantees about 78% of coverage on *OPT*, respectively. We compare the solution quality and computational efficiency, and these results are shown in Fig. 8.

In Fig. 8(a) and (c), because AG uses the complete data, it achieves the highest reward. RGs in fact also perform very well, which are within 95% to AG, and if we increase the sample size with guaranteed cover ratio changing from 59% to 78%, the performance of RG also improves significantly.

The main appeal of the randomized framework is that it is more computationally efficient than the other state-of-the-art methods. In Fig. 8(b) and (d), we compare the speed-up of RG and AG with naive GA. Firstly, we observe that RGs are faster than AG. Secondly, it is much faster than naive GA on both datasets. If we increase the sample size used in each round (and cover ratio increases accordingly), RG becomes slower. Therefore, cover ratio can be used to trade off between the solution quality and computational efficiency in RG.

In fact, the superiority of RG's efficiency is not obvious from Fig. 8(b) and (d); i.e., RG is only about two times faster than AG. This is because we do not implement RG in an efficient manner. RG can also be implemented more efficiently by exploiting the submodularity of reward function as AG and CELF do. So what we observe here is actually the worst-case performance of RG, and it is still more efficient than state-of-the-art methods. We will see RG's true power of efficiency in the following experiments.

7.2.2. A more efficient implementation of RG and using node's attributes

In here, we explore a more efficient implementation of RG. In particular, we mimic the AG and CELF in having a lazy update in reward gains. Moreover, instead of sampling in each round, we conduct the sampling procedure at the very beginning and choose final nodes from enough samples. We call such a form of sampling as batch sampling, and it is easy to prove that this modification will improve

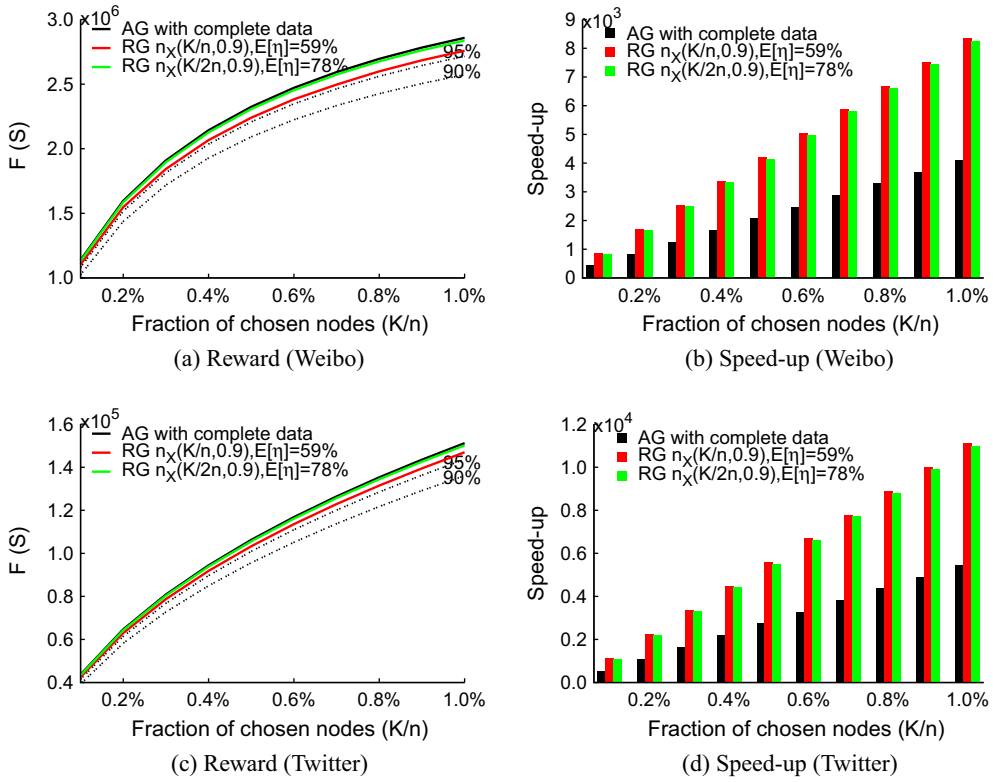
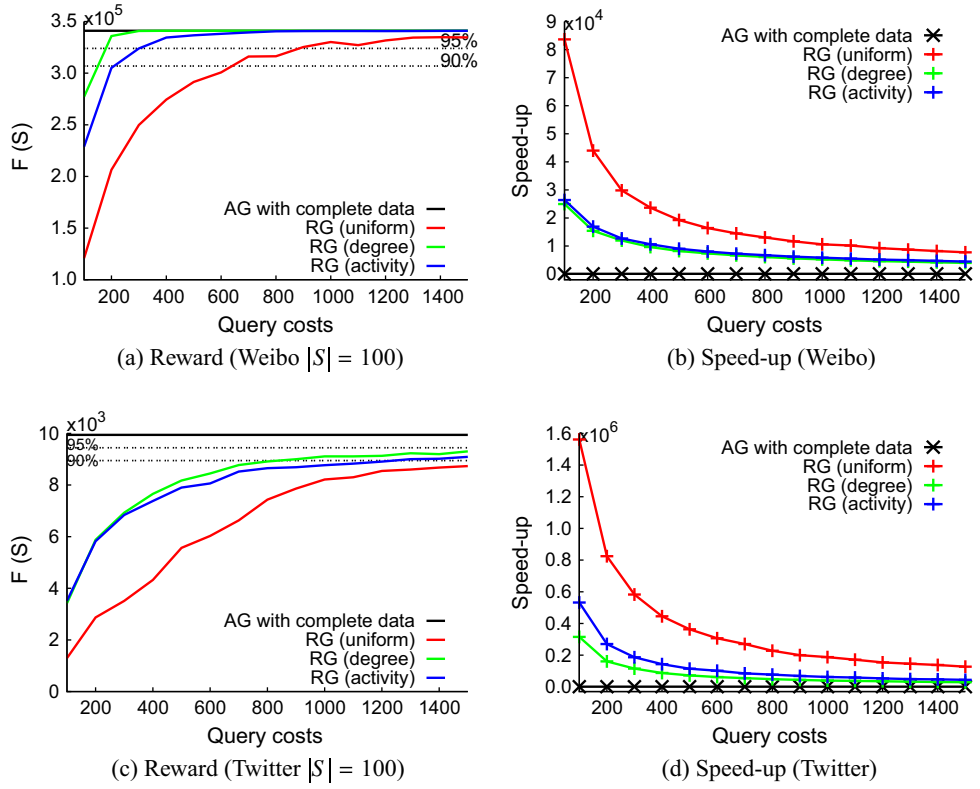


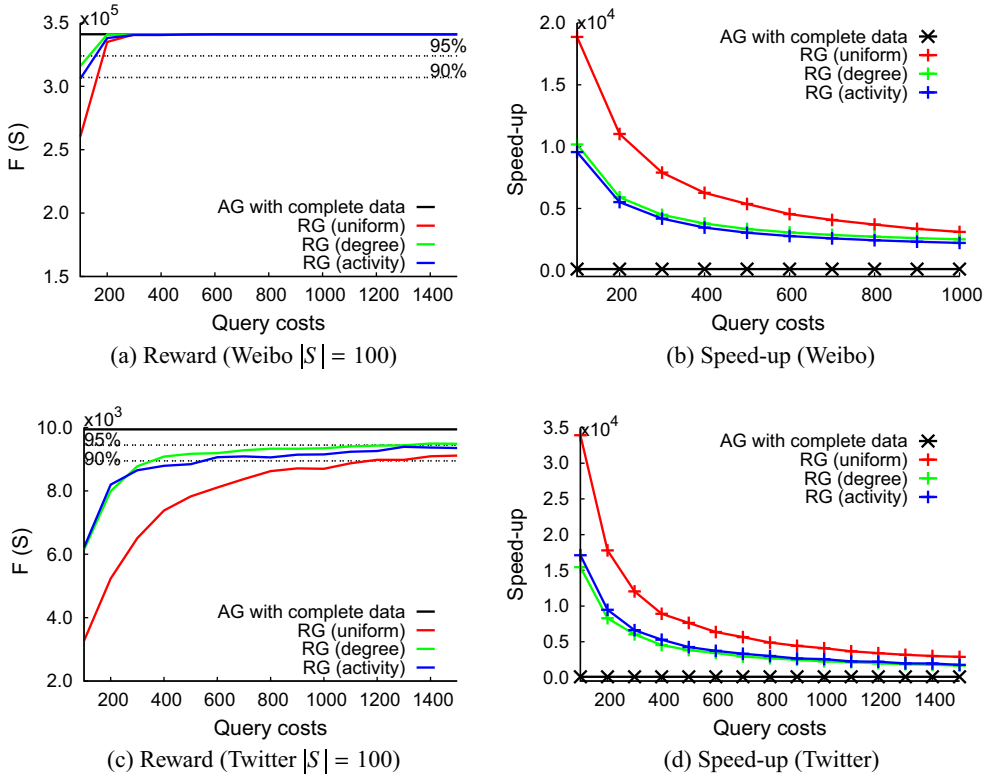
Fig. 8. Performance of RG with respect to different sample size (averaged over 50 runs).



(a) Reward (Weibo $|S| = 100$) (b) Speed-up (Weibo)

(c) Reward (Twitter $|S| = 100$) (d) Speed-up (Twitter)

Fig. 9. Improving sample quality by using node's attributes ($B = 100$, averaged over 50 runs).



(a) Reward (Weibo $|S| = 100$) (b) Speed-up (Weibo)

(c) Reward (Twitter $|S| = 100$) (d) Speed-up (Twitter)

Fig. 10. Improving sample quality by exploring structure of networks ($B = 100$, averaged over 50 runs).

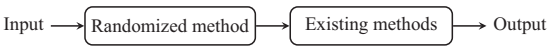


Fig. 11. The proposed randomized method supplements existing methods.

the solution quality lower bound because the probability of including an optimal node increases in each round.

In this experiment, we fix $B = 100$ and evaluate the performance of RG with respect to query costs, i.e., sample size. We consider three sampling strategies: (1) select a node uniformly at random; (2) select a node with probability proportional to its degree; and (3) select the node with probability proportional to its activity (#posts). Fig. 9 shows the results.

In Fig. 9(a) and (c), we can see that the solution quality increases as sample size increases, and sampling by degree is the best strategy follows by sampling by activity, and uniform sampling is the worst. From the speed-up curves in Fig. 9(b) and (d), we can see that uniform node sampling is the most efficient among the three strategies. All these strategies are in general much more efficient than that of AG, e.g., with 1000 query costs, RG is about 40 to 187 times faster than AG.

7.2.3. Improving sample quality by random walk

We now evaluate the benefits of exploiting network structures by using RW samplers. We set $B = 100$ and conduct three RW samplers: (1) select a neighbour uniformly at random; (2) select a neighbour with probability proportional to its degree; and (3) select a neighbour with probability proportional to its activity. Fig. 10 shows the results.

Both the reward curves in Fig. 10(a) and (c) speed-up curves in Fig. 10(b) and (d) shows similar results as in previous experiment: RW biased by degree generates the best solution but at a higher computational cost, then follows by activity, and uniform RW is the worst but the most efficient. If we compare the reward curves in Fig. 9(a) and (c) with Fig. 10(a) and (c), respectively, we can find that the RW can obtain better reward than node sampling, which coincides with our analysis in Section 6.

8. Summary and discussion

We summarize the paper in this section.

As the follow model is adopted by many OSNs, which subset of followees one should follow is a practical problem for OSN users. In this work, we study the followee selection problem for the purpose of maximizing information coverage and minimizing time delay. In other words, we want to detect important information in an OSN as much as possible and as soon as possible by following a few followees. However, this problem is NP-complete, and we lack the complete OSN data. To solve these challenges, we design a randomized method that guarantees both solution quality and computational efficiency.

The solution quality is guaranteed by exploiting the non-decreasing submodular property of the reward function as stated in Theorem 1. The importance of Theorem 1 lies not only in its statements on the existence of a solution quality lower bound of RG, but also in that it describes the method to obtain better solutions, i.e., by increasing the probability of including optimal followees in samples.

The computational efficiency lies in the fact that RG can leverage the birthday paradox to reduce the search space significantly. Although we lose solution quality lightly, a major speedup of computational efficiency can be achieved. We also find that the power-law structure of real-world networks can facilitate a random walk to detect information cascades efficiently.

It is interesting to see how RG behaves when $K/n \to 0$ (or $B/n \to 0$). From Fig. 3, we observe that when $K/n \to 0$, the number of samples required in each round increases drastically. Because a large sample size hurts RG's computational efficiency, RG is *not* suitable for very small budget B. For example, to select $B = 10^3$ users from an OSN having $n = 10^8$ users, or $B/n = 10^{-5}$, we need to generate as many as 10^5 samples in each round of RG.

Thus, we hasten to give an explanation. The aim of introducing RG is not to replace but supplement existing methods. The main contribution of RG is in significantly reducing the search space, and it assists existing methods to obtain the final solution from a reduced search space. In practice, we should use the randomized method to choose a small fraction of OSN users as *candidates* and then use existing methods to choose final information sources from candidates (because they guarantee better solution quality), as illustrated in Fig. 11. In the previous example, we can use RG to choose 0.1% of the population, or 10^5 users, as candidates, and then use existing methods to choose 10^3 users from these candidates. A conservative estimation of the computational efficiency reveals that this approach is at least 10^3 faster than applying GA over the entire OSN directly, without considering the expensive query cost of GA.

One limitation of this work is that we assume every user has the same cost of being followed, and thus we only consider the optimization problem with a cardinality constraint, i.e., $|S| \leqslant B$. In fact, different users usually have different costs of being followed. For example, if a user posts too many tweets, we need to spend much more time or attention to read them if we choose to follow that user, i.e., the cost is large. Therefore, a more general problem is to consider that each user u has a cost c_u, and the cardinality constraint is generalized to a knapsack constraint, i.e., $\sum_{u \in S} c_u \leqslant B$. How to design a randomized method for such a constraint offers an opportunity for future work.

Acknowledgements

The authors thank the editors and reviewers for their constructive comments and suggestions that greatly improved the quality of this paper. This work was supported in part by the National Natural Science Foundation (61103240, 61103241, 61221063, 91118005, 61221063, U1301254), 863 High Tech Development Plan (2012AA011003), 111 International Collaboration Program, of China, and the Application Foundation Research Program of SuZhou (SYG201311).

Table A.3
Summary of some frequently used notations.

Sign	Description
G	An undirected graph
V, E, C	Sets of nodes, edges and cascades
n, m	Number of nodes/edges in G
t_{uc}	The time user u joins cascade c
t_c	Start time of cascade c
$size(c)$	Size of cascade c, $size(c) = \|\{u : t_{uc} < \infty\}\|$
S, S_k	Sets of selected nodes (after round k)
X, X_k	Sets of node samples (in round k)
OPT	An optimal node set
F	A non-decreasing submodular function, $F(\emptyset) = 0$
$\delta_s(S)$	Reward gain, i.e., $\delta_s(S) = F(S \cup \{s\}) - F(S)$
B	Budget, i.e., $\|S\| < B$
K	Number of nodes in OPT, $K \leqslant B$
η	Cover ratio
α	A subset contains α fraction items of a set
γ	The scaling exponent of power-law distribution
p	Confidence probability
$n_X(\alpha, p)$	Number of samples to guarantee a cover ratio
d_u	Degree of node u
n_d	Number of nodes of degree d in G
I, I_d	Sets of infected nodes (of degree d) by diffusions
R, R_d	Sets of sampled nodes (of degree d) by random walk
i, i_d, r, r_d	Cardinalities of sets I, I_d, R, and R_d
GA	Greedy algorithm
AG	Accelerated greedy [33]
CELF	Cost-effective lazy forward approach [18]
RG	Randomized greedy
RW	Random walk

Appendix A. Notations and abbreviations

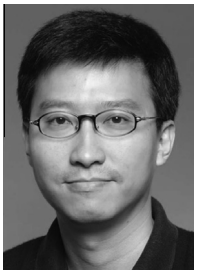
See Table A.3.

References

[1] H. Kwak, C. Lee, H. Park, S. Moon, What is Twitter, a social network or a news media? in: WWW'10: Proceedings of the 19th International World Wide Web Conference, 2010.
[2] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes Twitter users: real-time event detection by social sensors, in: WWW'10: Proceedings of the 19th International World Wide Web Conference, 2010.
[3] A. Tsotsis, First credible reports of bin Laden's death spread like wildfire on Twitter. <http://techcrunch.com/2011/05/01/news-of-osama-bin-ladens-death-spreads-like-wildfire-on-twitter> (May 2011).
[4] 2011 England riots. <http://en.wikipedia.org/wiki/2011_England_riots> (March 2014).
[5] M. Efron, Information search and retrieval in microblogs, J. Am. Soc. Inf. Sci. Technol. 62 (6) (2011) 996–1008.
[6] M. Efron, Improving retrieval of short texts through document expansion, in: SIGIR'12: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2012.
[7] R. McCreadie, C. Macdonald, Relevance in microblogs: Enhancing tweet retrieval using hyperlinked documents, in: OAIR'13: Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, 2013.
[8] N. Kanhabua, W. Nejdl, Understanding the diversity of tweets in the time of outbreaks, in: the 1st International Web Observatory Workshop at WWW, 2013.
[9] J. Zhao, J.C. Lui, D. Towsley, X. Guan, Y. Zhou, Empirical analysis of the evolution of follower network: a case study on Douban, in: NetSciCom'11: The 3rd IEEE International Workshop on Network Science for Communication Networks, 2011.
[10] Followee. <http://www.oxforddictionaries.com/definition/english/followee> (June 2014).
[11] L. Weng, A. Flammini, A. Vespignani, F. Menczer, Competition among memes in a world with limited attention, Sci. Rep. 2 (335) (2012) 1–8.
[12] Twitter follow limit. <http://support.twitter.com/articles/66885-why-can-t-i-follow-people> (March 2014).
[13] Sina Weibo follow limit. <http://help.weibo.com/faq/q/77> (March 2014).
[14] S. Khuller, A. Moss, J.S. Naor, The budgeted maximum coverage problem, Inf. Process. Lett. 70 (1999) 39–45.
[15] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack Problems, Springer, 2004.
[16] P. Domingos, M. Richardson, Mining the network value of customers, in: KDD'01: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001.
[17] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: KDD'03: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.
[18] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: KDD'07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007.
[19] A. Krause, C. Guestrin, Submodularity and its applications in optimized information gathering, J. ACM Trans. Intell. Syst. Technol. 2 (4) (2011) 1–20.
[20] M. Mondal, B. Viswanath, P. Druschel, K.P. Gummadi, A. Clement, A. Mislove, A. Post, Defending against large-scale crawls in online social networks, in: CoNEXT'12: Proceedings of the 8th International Conference on Emerging Networking EXperiments and Technologies, 2012.
[21] Twitter API limits. <https://dev.twitter.com/docs/rate-limiting> (March 2014).
[22] Sina Weibo API limits. <http://open.weibo.com/wiki/Rate-limiting> (March 2014).
[23] B. Cui, S. Moskal, H. Du, S.J. Yang, Who shall we follow in Twitter for cyber vulnerability? in: SBP'13: The 6th International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, 2013.
[24] L. Lovász, Random walks on graphs: a survey, Comb. Paul Erdös Eighty 2 (1993) 353–397.
[25] D. Bloom, A birthday problem, Am. Math. Mon. 80 (1973) 1141–1142.
[26] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, R. Zadeh, WTF: the who to follow service at Twitter, in: WWW'13: Proceedings of the 22ed International World Wide Web Conference, 2013.
[27] D. Liben-Nowell, J. Kleinberg, The link prediction problem for social networks, in: CIKM'03: Proceedings of the 12th ACM International Conference on Information and Knowledge Management, 2003.
[28] L. Backstrom, J. Leskovec, Supervised random walks: predicting and recommending links in social networks, in: WSDM'11: Proceedings of the 4th International ACM Conference on Web Search and Data Mining, 2011.
[29] D. Yin, L. Hong, X. Xiong, B.D. Davison, Link formation analysis in microblogs, in: SIGIR'11: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2011.
[30] J. Cheng, D. Romero, B. Meeder, J. Kleinberg, Predicting reciprocity in social networks, in: SocialCom'11: Proceedings of the 4th IEEE International Conference on Social Computing, 2011.
[31] A. Krause, Optimizing Sensing Theory and Applications, Ph.D. thesis, School of Computer Science Carnegie Mellon University, 2008.
[32] W.E. Hart, R. Murray, Review of sensor placement strategies for contamination warning systems in drinking water distribution systems, J. Water Resour. Plan. Manage. 136 (2010) 611–619.
[33] M. Minoux, Accelerated greedy algorithms for maximizing submodular set functions, Optim. Tech. 7 (1978) 234–243.
[34] N.A. Christakis, J.H. Fowler, Social network sensors for early detection of contagious outbreaks, PLoS ONE 5 (9) (2010) 1–8.
[35] L. Sun, K.W. Axhausen, D.-H. Lee, M. Cebrian, Efficient detection of contagious outbreaks in massive metropolitan encounter networks, Sci. Rep. 4 (5099) (2014) 1–6.
[36] M. Garcia-Herranz, E.M. Egido, M. Cebrian, N.A. Christakis, J.H. Fowler, Using friends as sensors to detect global-scale contagious outbreaks, PLoS ONE 9 (4) (2014) 1–7.
[37] S.L. Feld, Why your friends have more friends than you do, Am. J. Soc. 96 (6) (1991) 1464–1477.
[38] J. Ugander, B. Karrer, L. Backstrom, C. Marlow, The anatomy of the Facebook social graph, in: ICWSM'11: Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, 2011.

[39] Y.-C. Ho, Q. Zhao, Q. Jia, Ordinal Optimization: Soft Optimization for Hard Problems, Springer, 2007.
[40] Y.-C. Ho, R.S. Sreenivas, P. Vakili, Ordinal optimization of DEDS, Discrete Event Dyn. Syst.: Theory Appl. 2 (1992) 61–88.
[41] X. Guan, Y.-C. Ho, F. Lai, An ordinary optimization based bidding strategy for electric power suppliers in the daily energy market, IEEE Trans. Power Syst. 16 (4) (2001) 788–797.
[42] T.W.E. Lau, Y.-C. Ho, Universal alignment probabilities and subset selection for ordinal optimization, J. Optim. Theory Appl. 93 (3) (1997) 455–489.
[43] S. Bikhchandani, D. Hirshleifer, I. Welch, A theory of fads, fashion, custom, and cultural change as informational cascades, J. Polit. Econ. 100 (5) (1992) 992–1026.
[44] G. Nemhauser, L. Wolsey, M. Fisher, An analysis of approximations for maximizing submodular set functions – I, Math. Programming 14 (1978) 265–294.
[45] 476 million Twitter tweets, 2011. <http://snap.stanford.edu/data/twitter7.html>.
[46] L.C. Freeman, Centrality in social networks: conceptual clarification, Soc. Networks 1 (1978) 215–239.
[47] D. Liben-Nowell, J. Kleinberg, Tracing information flow on a global scale using internet chain-letter data, Proc. Natl. Acad. Sci. USA 105 (12) (2008) 4633–4638.
[48] M. Dell, Trafficking networks and the Mexican drug war, in: Proceedings of the 2011 NEUDC Conference, 2011.
[49] R. Ghosh, K. Lerman, T. Surachawala, K. Voevodski, S. Teng, Non-conservative diffusion and its application to social network analysis, in: CoRR 1102.4639, 2011.
[50] A. Barrat, M. Barthelemy, A. Vespignani, Dynamical Processes on Complex Networks, Cambridge University Press, 2008.
[51] B. Ribeiro, P. Basu, D. Towsley, Multiple random walks to uncover short paths in power law networks, in: NetSciCom'12: The 4th IEEE International Workshop on Network Science for Communication Networks, 2012.
[52] H. Jeong, S.P. Mason, A.-L. Barabasi, Z.N. Oltvai, Lethality and centrality in protein networks, Nature 411 (6833) (2001) 41–42.
[53] S. Lehmann, B. Lautrup, A.D. Jackson, Citation networks in high energy physics, Phys. Rev. E 68 (2003) 026113.
[54] M. Molloy, B. Reed, A critical point for random graphs with a given degree sequence, Random Struct. Algor. 6 (1995) 161–179.
[55] S.M. Ross, The inspection paradox, Probab. Eng. Inf. Sci. 17 (2003) 47–51.

Junzhou Zhao received the B.S. degree in automatic control from Xi'an Jiaotong University, Xi'an, China, in 2008. He is currently a Ph.D. candidate with the Systems Engineering Institute and MOE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University under the supervision of Prof. Xiaohong Guan. His research interests include online social network measurement and modelling.



John C.S. Lui received the Ph.D. degree in computer science from UCLA. He is currently a professor in the Department of Computer Science and Engineering at The Chinese University of Hong Kong. His current research interests include communication networks, networksystem security (e.g., cloud security, mobile security, etc.), network economics, network sciences (e.g., online social networks, information spreading, etc.), cloud computing, large-scale distributed systems and performance evaluation theory. He serves in the editorial board of IEEEACM Transactions on Networking, IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, Journal of Performance Evaluation and International Journal of Network Security. He was the chairman of the CSE Department from 2005 to 2011. He received various departmental teaching awards and the CUHK Vice-Chancellors Exemplary Teaching Award. He is also a corecipient of the IFIP WG 7.3 Performance 2005 and IEEEIFIP NOMS 2006 Best Student Paper Awards. He is an elected member of the IFIP WG 7.3, fellow of the ACM, fellow of the IEEE, and croucher senior research fellow. His personal interests include films and general reading.

Don Towsley holds a B.A. in Physics (1971) and a Ph.D. in Computer Science (1975) from University of Texas. From 1976 to 1985 he was a member of the faculty of the Department of Electrical and Computer Engineering at the University of Massachusetts, Amherst. He is currently a Distinguished Professor at the University of Massachusetts in the Department of Computer Science. He has held visiting positions at IBM T.J. Watson Research Center, Yorktown Heights, NY; Laboratoire MASI, Paris, France; INRIA, Sophia-Antipolis, France; AT&T Labs-Research, Florham Park, NJ; and Microsoft Research Lab, Cambridge, UK. His research interests include networks and performance evaluation. He currently serves as Editor-in-Chief of IEEE/ACM Transactions on Networking and on the editorial boards of Journal of the ACM, and IEEE Journal on Selected Areas in Communications, and has previously served on numerous other editorial boards. He was Program Co-chair of the joint ACM SIGMETRICS and PERFORMANCE 92 conference and the Performance 2002 conference. He is a member of ACM and ORSA, and Chair of IFIP Working Group 7.3. He has received the 2007 IEEE Koji Kobayashi Award, the 2007 ACM SIGMETRICS Achievement Award, the 1998 IEEE Communications Society William Bennett Best Paper Award, and numerous best conference/workshop paper awards. Last, he has been elected Fellow of both the ACM and IEEE.

Xiaohong Guan received the B.S. and M.S. degrees in automatic control from Tsinghua University, Beijing, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, US, in 1993. From 1993 to 1995, he was a consulting engineer at PG&E. From 1985 to 1988, he was with the Systems Engineering Institute, Xi'an Jiaotong University, Xi'an, China. From January 1999 to February 2000, he was with the Division of Engineering and Applied Science, Harvard University, Cambridge, MA. Since 1995, he has been with the Systems Engineering Institute, Xi'an Jiaotong University, and was appointed Cheung Kong Professor of Systems Engineering in 1999, and dean of the School of Electronic and Information Engineering in 2008. Since 2001 he has been the director of the Center for Intelligent and Networked Systems, Tsinghua University, and served as head of the Department of Automation, 2003–2008. He is an Editor of IEEE Transactions on Power Systems and an Associate Editor of Automata. His research interests include allocation and scheduling of complex networked resources, network security, and sensor networks. He has been elected Fellow of IEEE.