How to Network in Online Social Networks

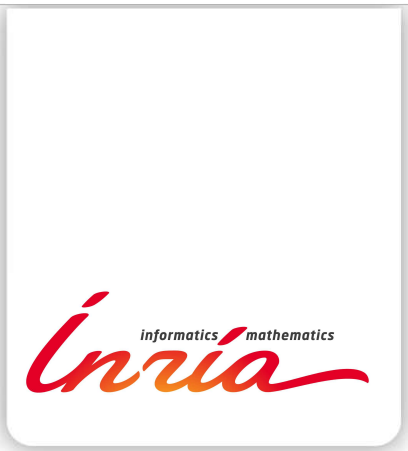
Giovanni Neglia, Xiuhui Ye, Maksym Gabielkov, Arnaud Legout

How to Network in Online Social Networks

Giovanni Neglia, Xiuhui Ye, Maksym Gabielkov, Arnaud Legout

RESEARCH REPORT
N° 8423
December 2013
Project-Teams MAESTRO, DIANA

ISSN 0249-6399 ISRN INRIA/RR--8423--FR+ENG



How to Network in Online Social Networks

Giovanni Neglia*, Xiuhui Ye†, Maksym Gabielkov‡, Arnaud Legout‡

Project-Teams MAESTRO, DIANA

Research Report n° 8423 — December 2013 — 13 pages

Abstract:
In this paper, we consider how to maximize users' influence in Online Social Networks (OSNs). More specifically, we study how social relationships impact influence in both directed OSNs (such as Twitter or Google+) and undirected ones (such as Facebook). Our problem introduces some new twists in comparison to the classic influence maximization problem originally defined in [5], where K influential individuals have to be selected. First, even if the user follows or proposes its friendship to the most influential individuals, there is no guarantee that they will follow back or accept the friendship request, i.e. they may not *reciprocate*. Second, following or proposing friendship is a quite cheap operation in OSNs so that the user can easily change dynamically its set of connections. A third difference in comparison to the classic formulation is that we quantify the influence not only by the number of individuals who actively replicate the information but also who can see the information. We show that, despite these three differences, greedy algorithms have the same theoretical guarantees than in the standard influence maximization problem, i.e. they reach a $(1-1/e)$ approximation ratio. These greedy algorithms require the knowledge of the whole topology and are computationally expensive because of the inherent cost of evaluating the effect of a cascade. We show by simulations on the complete Twitter graph that much more practical heuristics are almost as effective. For example, exploiting simply the knowledge of degree and reciprocation probability of each node i (respectively d_i and r_i), the strategy that selects the nodes with the largest product $r_i d_i$ performs at most 2% worse than the above mentioned greedy algorithm. Moreover, the even simpler random selection strategy requires only to know the set of users and achieves similar performance when the information replication probability of the cascade process is as large as 1%.

Key-words: Twitter, social network, influence

* Inria Sophia-Antipolis Méditerrané, MAESTRO project-team.
† Politecnico di Torino, Dipartimento di Elettronica e Telecomunicazioni.
‡ Inria Sophia-Antipolis Méditerrané, DIANA project-team.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Propagation de l'information dans les réseaux sociaux

Résumé : Dans ce papier, on évalue comment maximiser l'influence des utilisateurs dans les réseaux sociaux. En particulier, on étudie comment les liens sociaux modifient l'influence dans les réseaux sociaux dirigés (comme Twitter) et non dirigés (comme Facebook). Cette étude introduit trois différences par rapport au problème classique de maximisation introduit dans [5], où K utilisateurs influents sont sélectionnés. Premièrement, si un utilisateur suit ou propose une relation d'amitié à l'utilisateur le plus influent, il n'y a aucune garantie que l'utilisateur le plus influent accepte la relation, c'est-à-dire qu'il y ait réciprocité. Deuxièmement, créer une relation de suivi ou d'amitié est peu coûteux dans les réseaux sociaux, par conséquent, les utilisateurs peuvent facilement changer leurs relations. Troisièmement, on quantifie l'influence non seulement par le nombre d'utilisateurs qui peuvent répliquer une information, mais aussi par le nombre d'utilisateurs qui reçoivent cette information.

On montre qu'en dépit de ces trois différences, les algorithmes gloutons conduisent aux mêmes résultats théoriques que dans le problème standard de maximisation de l'influence, c'est-à-dire qu'ils atteignent un facteur d'approximation $(1-1/e)$. Ces algorithmes gloutons ont besoin de la connaissance de toute la topologie et sont coûteux à cause de l'évaluation de l'effet d'une cascade.

On montre avec des simulations sur le graphe social complet de Twitter que des heuristiques simples ont une efficacité proche d'un algorithme glouton. Par exemple, en connaissant simplement le degré d_i et la probabilité de réciprocité r_i de chaque nœud i, la stratégie qui choisit les nœuds avec le plus grand produit $r_i d_i$ est seulement au plus 2% moins efficace qu'un algorithme glouton. De plus, la stratégie qui consiste à simplement choisir les nœuds au hasard obtient des performances similaires quand la probabilité de réplication de l'information pour le processus de cascade est de 1%.

Mots-clés : Twitter, réseau social, influence

1 Introduction

The general problem to select individuals taking into account their *network value*, i.e. their contribution to information propagation in the network, was first introduced in [2], which motivated Kempe *et al.* to define a general optimization framework in [5, 6]. Because our analysis extends the foundational work from Kempe *et al.*, we start by summarizing their contribution. They modeled the propagation of information in a social network (e.g. a tweet) using two different discrete-time models. The first one is the *order independent cascade model*. Nodes with/without the information are respectively called *active/inactive*. When one node, say it u, becomes active at time t, it has one chance to influence (infect) all its non-active neighbors, who may then become active at time $t+1$. From time $t+1$ on, node u is still active but no more *contagious*. The contagion attempts from new active nodes at $t+1$ are arbitrarily ordered. The probability of success needs to be specified in order to completely describe the model. A quite general case is when u's success probability to infect v depends on the set S of v's neighbors that already attempted to influence v. We denote such probability $p_v(u, S)$. In a *decreasing cascade model* this probability is non-increasing in S, that is $p_v(u, S) \geq p_v(u, T)$ whenever $S \subseteq T$. This corresponds to the fact that the more nodes have already tried in vain to infect v, the less likely v is to be influenced by other attempts. Starting from an initial set of active nodes A, the process will stop in at most $n-1$ steps. The main performance metric of interest is the final set $\phi(A)$ of active nodes or better its expected size $\mathrm{E}[|\phi(A)|] \triangleq \sigma(A)$. The second model is the *general threshold model*. In this case each node has a monotone activation function $f_v : 2^V \rightarrow [0, 1]$, and a threshold θ_v chosen independently and uniformly at random from the interval $(0, 1]$. A node v becomes active at time $t+1$ if $f_v(S) \geq \theta_v$, where S is the set of active nodes at time t. Interestingly [6] shows that the two models are equivalent, in the sense that for any activation functions $f_v(.)$, there exist corresponding activation success probabilities $p_v(.)$ such that the distribution over final active sets $\varphi(A)$ is the same under both models.

The optimization problem introduced in [5] is to choose the initial set A under the constraint that $|A| \leq K$ so that the expected size of the active nodes' final set is maximized. The authors show that the problem is NP-hard, but that a natural greedy heuristic reaches a $(1 - 1/e)$ approximation factor for the decreasing cascade model (and for the corresponding general threshold model). The greedy heuristic simply incrementally increases the set A starting from an empty set and adding at each time the node v_i that maximizes the marginal gain $\sigma(A \cup \{v\}) - \sigma(A)$. If at each step the selected node is a $1 - \epsilon$ approximation of the best node, then the greedy algorithm achieves a $(1 - 1/e - \epsilon')$ approximation factor, where ϵ' depends on ϵ polynomially. The key for proving this result is to show that $\sigma(A)$ is a non-negative, monotone, submodular function on sets,[1] then the conclusion about greedy algorithm's approximation ratio follows from known results on such functions [8, 7].

An implicit assumption of the influence maximization problem as defined by Kempe *et al.* is that, once the influential individuals have been identified, they can be *recruited* in order to spread the information of interest. Recruitment is costly (in terms of money or social investment) and then the available budget limits the number of individuals to be selected to K. In this paper we consider a user of an Online Social Network (OSN) who can only exploit its networking strategy inside the OSN itself in order to maximize its influence, i.e. it can select which other users to follow (in directed OSNs such as Twitter or Google+) or to propose its friendship to (in undirected OSNs such as Facebook), but it cannot recruit them outside the OSN. Due to this different point of view, our problem introduces some new twists. First, even if the user follows or proposes its friendship to the most influential individuals, there is no guarantee that they

[1] A set function $f(.)$ is submodular if $f(S \cup \{z\}) - f(S) \geq f(T \cup \{z\} - f(T)$ whenever $S \subseteq T$ and it is monotone if $f(S \cup \{z\}) \geq f(S)$ for each S and z.

will follow it back or accept the friendship request, i.e., they will *reciprocate*. Second, following or proposing friendship are quite cheap operations in an OSN so that more aggressive dynamic strategies are feasible, even if there may be some intrinsic constraints imposed by the OSN. For example Twitter puts a cap on the maximum number of users each account can follow at a given time instant, but the user may dynamically change them, for example replacing those who do not reciprocate with different ones. A third difference in comparison to the classic formulation is that we quantify the influence not only by the number of individuals who actively replicate the information but also who can see the information. Our main theoretical contribution is to show that, despite the differences highlighted, greedy algorithms still guarantee the same approximation ratios in all the different variants of the problem we considered.

We note that in the classic influence maximization problem greedy algorithms require to know the topology of the social network as well as all the functions $p_v(u, S)$ for every node v. The same is true in our problem. This information may hardly be available to a user. Moreover greedy algorithms are computationally expensive, because of the inherent cost of evaluating the expected size of $\phi(A)$, that can only be estimated by Monte Carlo simulations of the cascade process on the (large) social network graph. The second main contribution of this paper is to show that, from the practical point of view, the user needs much less information and computational resources. We reach this conclusion comparing by simulations the greedy algorithms to much simpler heuristics on the complete Twitter social graph as crawled in 2012 [3]. This graph has 505 million nodes and 23 billion connections. We show that, if only the degree and reciprocation probability of each node i (respectively d_i and r_i) are known, the simple strategy to select the nodes with the largest product $r_i d_i$ performs at most 2.5% worse than the above-described greedy algorithm. Moreover, randomly selecting its neighbors achieves similar performance when the replication probability of the cascade process is as large as 1% and only requires to know the set of users. Similar results have been observed in [4]. That paper introduces the *effective density*, which is the product of the average degree and the replication probability, and shows empirically across many different, but relatively small, networks (with at most 20 thousands nodes) that selecting the highest degree nodes performs well in networks with low effective density, while selecting random nodes is as good as any other strategy in networks with high effective density. The greedy strategy appears to be advantageous only for intermediate densities when the network exhibits a modular structure. Our results can be read in the same way if we extend the effective density to include also the reciprocation probability. Our paper then confirms the empirical findings in [4] on a much larger network. Moreover, by varying the replication probability of the cascade process, we have explored different effective densities and we have never observed that greedy algorithms offer a significant advantage in Twitter graph. This is another indirect proof that OSNs' structure is not modular, but closer to random graphs.

2 Problem formulation and analysis

We move now to extend the results in [5, 6] to actual OSNs. For the sake of simplicity we refer to Twitter in our description of the problem, but the same issues hold in most of the other social networks. Twitter is one of the largest social networks with more than 500 million registered accounts, it allows its users to send short messages called *tweets*. However, it differs from other social networks, such as Facebook and Google+, because it uses exclusively directed edges (arcs) among accounts. Twitter has no notion of bidirectional friendship, but it allows users to *follow* other users, i.e. to subscribe for their messages. Following does not require any approval from the user being followed. If Alice follows Bob, then Alice is called a *follower* of Bob and Bob a *following* of Alice. Twitter users can *retweet* received tweets that means forwarding the tweet

to their followers. In this paper we use the notation (V, E) to refer to the Twitter social graph, where V is the set of Twitter users and E is the set of directed edges. We orient the arcs in such a way that they show the tweet propagation direction, e.g. if A follows B the arc is directed from B to A because A receives tweets from B.

We evaluate the influence of a user u in Twitter through two related metrics: the average number of users who retweet u's tweets and the average number of users who read u's tweets. By default the tweets of a user are visible to all its followers (even if they do not read them immediately). If we consider the number of retweets, there is a natural one-to-one mapping of the order independent cascade model in our setting, so for the moment we focus on this metric. Let a node be active if it has read the tweet and decided to retweet it. Then the original influence maximization problem can be rephrased as follows: how to choose K nodes that should initially tweet the message in order to maximize the expected number of retweets. In this case $p_v(u, S)$ is the probability that node v reads and decides to retweet the message tweeted or retweeted from u, given that the nodes in S have already tweeted or retweeted it. In the original problem formulated by Kempe *et al.* it is not specified how the K initial users should be infected, i.e., in our language, how should they be convinced to tweet the message. In this paper we focus on a specific user u_0 that is trying to maximize its influence and cannot reach other Twitter users through some external communication network. Then, u_0 can only rely on a smart "networking strategy": it can carefully select a given set of users to follow and hope that these users will follow it back and will eventually retweet its tweets. The strategic choice of u_0 is then the selection of its set of followings in order to maximize its influence. We consider for the moment that u_0 makes this choice once and for all at its registration. We observe that Twitter puts a cap to the maximum number of initial followings that is $K = 2000$ (this limit is increased when the user gets more then 2000 followers). More formally, let B denote the set of u_0's followings and let $\varphi(B)$ be the set of nodes that retweets a tweet originally emitted from u_0. We can write $\varphi(B) = \sum_{v \in V} X_v$, where X_v is a Bernoulli random variable, that is equal to 1 iff node v is active at the end of the cascade. Our problem can be formally stated as follows:

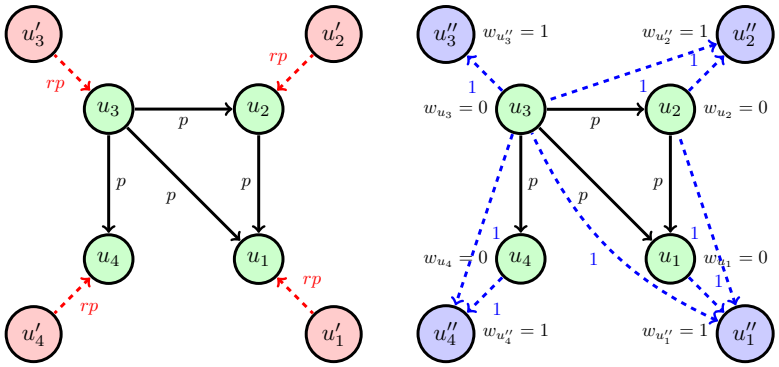
$$\begin{array}{ll} \underset{B}{\text{Argmax}} & \mathrm{E}\left[|\varphi(B)|\right] \\ \text{subject to} & |B| \leq K. \end{array} \tag{1}$$

In the same spirit of [5, 6] we assume to know 1) the probability $r(u)$ that a given user u would reciprocate u_0 if u_0 follows u and 2) the probability $p_v(u, S)$ that node v reads and decides to retweet the message tweeted or retweeted from u, given that the nodes in S have already tweeted or retweeted it. The knowledge of $p_v(u, S)$ is also required for $u = u_0$ or $u_0 \in S$.

The greedy algorithm for Problem 1 corresponds to the following behavior: user u_0 selects K followings one after the other, maximizing at each step the marginal increment of the function $\mathrm{E}\left[|\varphi(.)|\right]$. Our first theoretical result is the following.

Proposition 1. *The greedy algorithm is a* $(1 - 1/e)$ *approximation algorithm for Problem 1.*

Proof. Let (V, E) be the social network's graph without node u_0. Consider a new graph $(\hat{V}, \hat{E})$, where for each node u in V we add a new node u' and a link oriented from u' to u . Let V' be the set of these newly added nodes and $h : V \to V'$ the function such that $h(u) = u'$. On this graph we define success probability functions as follows: $\hat{p}_v(t, S) = p_v(t, S)$ for $t \neq v'$ and $v' \notin S$, $\hat{p}_v(v', S) = r(v)p_v(u_0, S)$ and $\hat{p}_v(t, S) = r(v)p_v(t, S)$ if $v' \in S$. Fig. 1 (a) illustrates the graph transformation. We consider now the order independent cascade model on the graph $(\hat{V}, \hat{E})$. u_0's choice of the set of its followings $B \subseteq V$ corresponds to the choice of the set $A = h(B)$ of initial active nodes in V'. Moreover, the probabilities have been defined in such a way that it is possible to couple the two processes so that $\phi(h(B)) = \varphi(B) + K$, where adding K corresponds



(a) reciprocation (b) readers

Figure 1: Graph transformations. The original nodes are green. The added nodes are red/blue, added arcs are dashed. We have specified the new success probability functions for the simple case when $p_v(.)$ and $r(.)$ are constant and respectively equal to p and r.

to the fact that the initial set of active nodes is counted by $\phi(.)$ ($A \subseteq \phi(A)$). It follows that $\sigma(h(B)) = \mathrm{E}[\varphi(B)] + K$ and Problem 1 is equivalent to solve the influence maximization problem on $(\hat{V}, \hat{E})$ with the additional constraint that the nodes can only be selected in V'. This does not change the property of the function $\sigma(.)$, that is non negative, monotone and sub-modular, then the results in [5, 6] still hold. In particular the greedy algorithm is a $(1 - 1/e)$ approximation algorithm for the influence maximization problem defined on $(\hat{V}, \hat{E})$ and then for Problem 1.[2] □

We now consider a variation of problem 1 where node u_0 is not required to select all the K followings at once, but it can apply more complex dynamic strategies. For example node u_0 can stop following nodes that do not reciprocate by a given time T and start following new users. In this way u_0 can follow during a given time window more than K users (but at most K at the same time) and reach in general a larger number of followers (the number can approach K if there are at least K nodes in the network willing to reciprocate u_0). This improvement in comparison to the original problem is obtained at the price of a longer time required to select the best followings. The best possible result achievable by u_0 is obtained if we assume u_0 to know *a priori* which nodes would reciprocate. For each node v, let R_v be the Bernoulli random variable indicating if node v reciprocates node u_0 by time T after u_0 starts following v. Clearly it holds $r(v) = \mathrm{E}[R_v]$. We introduce then the following ideal optimization problem:

$$\begin{aligned} &\underset{B}{\text{Argmax}} && \mathrm{E}[|\varphi(B)|] \\ &\text{subject to} && |B| \leq K \text{ and } R_v = 1 \ \ \forall v \in B. \end{aligned} \tag{2}$$

The greedy algorithm for this problem orderly selects K reciprocating nodes. At each step the reciprocating node that maximizes the marginal improvement of $\varphi(B)$ is added to the current set and the success probability functions are left

Proposition 2. *The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for problem 2.*

[2] Also the results for the case when the greedy algorithm selects at each step a $(1 - \epsilon)$ approximation of the best node can be extended to our case, but for the sake of conciseness we only refer to the simpler case.

Proof. The proof is analogous to that of Proposition 1. In this case an additional node u' is added only for each reciprocating node u, i.e. for each $u \in V$ such that $R_u = 1$ and the probabilities can be updated as follows: $\hat{p}_v(t, S) = p_v(t, S)$ for $t \neq v'$ and $v' \notin S$, $\hat{p}_v(v', S) = p_v(u_0, S)$ and $\hat{p}_v(t, S) = p_v(t, S)$ if $v' \in S$, where only the nodes v' such that $R_v = 1$ need to be considered. □

While Problem 2 requires to know a priori which users are willing to reciprocate u_0, the greedy algorithm can be implemented online without such knowledge. This practical greedy algorithm operates in steps, where each step has a duration at most equal to T time units. At each step the user follows the node v that brings the largest marginal increase in comparison to the already selected nodes assuming that v reciprocates. If node v reciprocates by time T, node u_0 maintains user v in its list of followings, otherwise it removes it. The algorithm stops when K users reciprocate or when there are no more users to select in the network. It is easy to check that the practical greedy algorithm selects exactly the same users that the greedy algorithm with a priori knowledge of the reciprocating nodes would, but it requires in general a longer time to execute. The reasoning above leads us to conclude that:

Proposition 3. *The greedy algorithm for Problem 2 can be implemented without a priori knowledge of which users reciprocate, and its expected number of retweets is at least* $(1-1/e)$ *of the value obtained by any online algorithm where each node can be selected at most once and reciprocation delays of at most* T *time units are tolerated.*

Finally, we move to consider the case where user's influence is quantified through the number of users who read its tweet (or better that have the tweet in their stream). This problem can be mapped to a variant of the previous case (where we consider the number of retweets) introducing opportune nodes' weights. We need to change the original graph (V, E) as follows. For each user $u \in V$, we introduce a new node u'' and the directed arcs (u, u'') and (v, u''), for each node v such that $(v, u) \in E$ (see Fig. 1 (b)). We denote V'' and E'' respectively the set of new nodes and arcs and $(\tilde{V}, \tilde{E})$ the new graph. By doubling each node, we can separately account for the two roles of a user as retweeter and as reader. Going back to the cascade model terminology, at a given time step if node u is active, the corresponding user has retweeted the tweet, and if node u'' is active the corresponding user has read the tweet. In order to correctly model the process, we introduce activation success probabilities as follows: $\tilde{p}_v(t, S) = p_v(t, S)$ for $v \in V$ and $\tilde{p}_{v''}(t, S) = 1$ for $v'' \in V''$. We also introduce nodes' weights $w_v = 0$ for $v \in V$ $w_{v''} = 1$ for $v'' \in V''$. Let X_v be the Bernoulli random variable that indicates if node v is active when the cascade terminates. The number of users that see the tweet is given by $\psi(B) = \sum_{v \in \tilde{V}} w_v X_v$ where $B \subseteq V$ is the set of followings selected by node u_0. Two different problems can then be defined depending if the set B has to be selected at the begin or can be changed dynamically, similarly to what is done above. The only difference is the fact that the weighted objective function $\mathrm{E}[\psi(B)]$ is considered instead of the unweighted one $\mathrm{E}[\varphi(B)]$. Obviously the function $\mathrm{E}[\psi(B)]$ is non-negative and non-decreasing, we can also prove that

Proposition 4. *The function* $E[\psi(B)]$ *is submodular.*

Proof. We adapt some results in [6] relative to the size of the different sets to the case where we consider a weighted sum of the set elements. We need to prove that $\psi(B_1 \cup \{z\}) - \psi(B_1) \geq \psi(B_2 \cup \{z\}) - \psi(B_2)$ for any z whenever $B_1 \subseteq B_2$. Let $C = \varphi(B)$ be the (random) set of nodes active at the end of the cascade starting from the nodes in B. Imagine now to start a new cascade process on the graph activating node z, but taking into account the fact that all the nodes in C have already tried to infect their neighbors. This new cascade is called the *residual cascade process* and has success probabilities $p_v^{(C)}(u, S) \triangleq p_v(u, S \cup C)$. We denote this new stochastic

process as $\mathcal{S}_C(z)$ and the additional nodes in $V \setminus C$ made active by it as $\varphi_C(z)$. In Theorem 3 of [6], it is proven that $\varphi_C(z)$ is distributed as $\varphi(B \cup \{z\}) - \varphi(B)$. Then it holds:

$$\mathrm{E}\left[\psi(B \cup \{z\}) - \psi(B)\right] = \mathrm{E}\left[\sum_{v \in \varphi_{\varphi(B)}(z)} w_v\right] \tag{3}$$

Consider $C_1 \subseteq C_2$, and the corresponding residual processes $\mathcal{S}_{C_1}(z)$ and $\mathcal{S}_{C_2}(z)$. If we couple the equivalent general threshold models by selecting the same threshold at each node, it can be shown (see Lemma 3 in [6]) that pathwise $\varphi_{C_1}(z) \supseteq \varphi_{C_2}(z)$. It follows that $\sum_{v \in \varphi_{C_1}(z)} w_v \geq \sum_{v \in \varphi_{C_2}(z)} w_v$ and then

$$\mathrm{E}\left[\sum_{v \in \varphi_{C_1}(z)} w_v\right] \geq \mathrm{E}\left[\sum_{v \in \varphi_{C_2}(z)} w_v\right] \text{ whenever } C_1 \subseteq C_2. \tag{4}$$

Let us now consider two cascade processes whose initial activation sets are respectively B_1 and B_2 with $B_1 \subseteq B_2$, if we couple them as above, we can similarly show that $\phi(B_1) \subseteq \phi(B_2)$, then

$$P(\phi(B_1) = C_1, \phi(B_2) = C_2) = 0 \text{ whenever } C_1 \not\subseteq C_2. \tag{5}$$

We can now wrap-up our intermediate results. Let $B_1 \subseteq B_2$, then

$$\begin{aligned}
\mathrm{E}\left[\varphi(B_1 \cup \{z\}) - \varphi(B_1)\right] &= \mathrm{E}\left[\sum_{v \in \varphi_{\varphi(B_1)}(z)} w_v\right] \\
&= \sum_{C_1} \mathrm{E}\left[\sum_{v \in \varphi_{C_1}(z)} w_v\right] P(\varphi(B_1) = C_1) \\
&= \sum_{C_1} \sum_{C_2 \supseteq C_1} \mathrm{E}\left[\sum_{v \in \varphi_{C_1}(z)} w_v\right] P(\varphi(B_1) = C_1, \varphi(B_2) = C_2) \\
&\geq \sum_{C_1} \sum_{C_2 \supseteq C_1} \mathrm{E}\left[\sum_{v \in \varphi_{C_2}(z)} w_v\right] P(\varphi(B_1) = C_1, \varphi(B_2) = C_2) \\
&= \sum_{C_2} \mathrm{E}\left[\sum_{v \in \varphi_{C_2}(z)} w_v\right] P(\varphi(B_2) = C_2) \\
&= \mathrm{E}\left[\varphi(B_1 \cup \{z\}) - \varphi(B_1)\right],
\end{aligned}$$

where we have used Eqs. (3), (4) and (5). □

From the general results for non-negative, non-decreasing submodular functions it follows that if selecting all its followings at the begin according to the greedy algorithm that incrementally maximizes $\mathrm{E}\left[\psi(.)\right]$ guarantees a $(1 - 1/e)$ approximation ratio. The same result holds in the dynamic case. Due to lack of space we do not define formally the two problems and the corresponding propositions, but we summarize our conclusions as follows:

Proposition 5. *The greedy algorithms for the static and dynamic versions of our problem reach a $(1 - 1/e)$ approximation ratio also when the objective function is $E[\psi(.)]$, the expected number of users who see the tweet.*

3 Experiments on Twitter

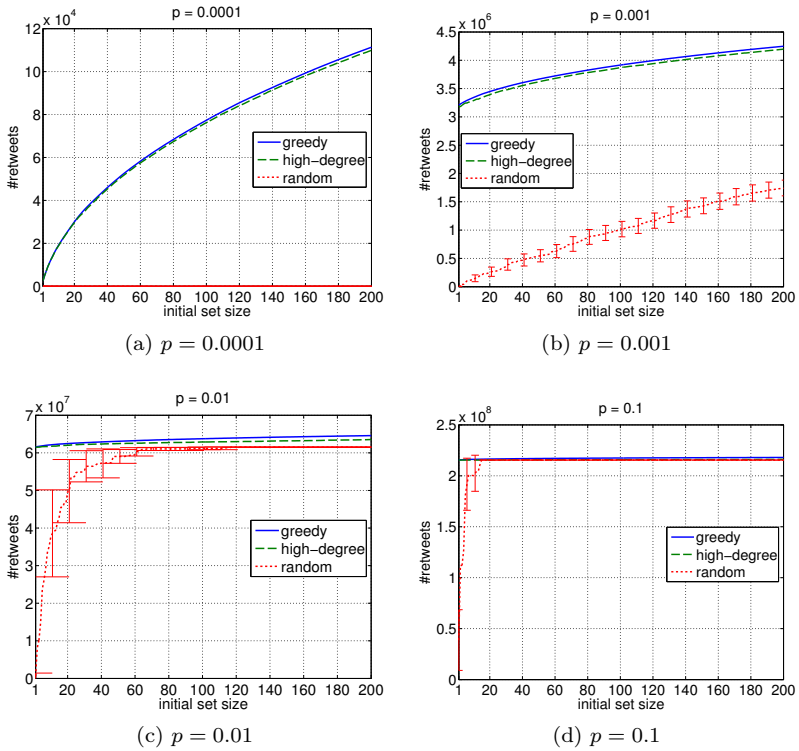
In this section we present the experimental part of our study. In Section 3.1, we describe the dataset and the methodology that we used to perform our simulations. In Section 3.2 we discuss the results.

3.1 Methodology

For our experiments, we considered the simple case when $p_v(u, S) = p$ is constant and evaluated different selection strategies on the complete dataset of Twitter as crawled in July 2012 [3]. This dataset is a social graph with 505 million nodes and 23 billion arcs and requires roughly 417GB of storage in the form of edgelist. A naive implementation of the experiment would require to load the graph into memory and then use Montecarlo simulations of the retweet process in order to estimate the objective functions with high accuracy. The followers of the initial node u_0 would retweet with probability 1, then their followers will retweet with probability p and so on till no new node is retweeting. However we could not afford loading this amount of data in memory with our actual resources. We have overcome this problem by generating a set of pruned graphs. A *pruned* graph is obtained from the original one by sampling each edge with probability p (and with probability $1 - p$ the edge is removed from the graph). Computing the set of reachable nodes from u_0 on a pruned graph is equivalent to counting the number of retweeting nodes in a specific sample of the retweet random process, but memory requirement is reduced by a factor p (usually $p << 1$) at the expense of storage increase, because we need to work on multiple pruned graphs (see discussion below) in order to reach the required accuracy.

Despite this expedient our software was still hitting memory constraints when computing reachability for the larger values of p we considered. To work around this problem we took the following approach consisting of two steps. First we computed the Strongly Connected Components (SCCs) of the graph. Second, we constructed a Directed Acyclic Graph (DAG) by abstracting each SCC as a single node and replace multiple arcs between the nodes with a single arc. Provided that p is quite large, we will observe big SCCs in the pruned graph, thus we can achieve a big reduction in size using our approach. Then we compute the reachability on the obtained DAG and deduce the reachability of the original pruned graph by taking into account the number of nodes in each SCC and the fact that the nodes belonging to same SCC have the same reachability. This approach decreases the computation time, as well as memory and storage requirements (because of the more compact DAG representation).

Another challenging aspect of the simulation analysis is to determine how many pruned graph samples we need in order to achieve a given precision for the estimates of the expected number of retweets. To this purpose we have used two different models that we can only describe shortly here because of space constraints.The first model approximates the cascade process with a branching process, where the probability to have a follower with k followers is $kq_k/\langle k\rangle$, where q_k is the distribution of the number of followers in the graph and $\langle k\rangle$ is the average number of followers (see for example [1, Chapter 8] for a justification of such expression). This model requires that different active nodes have different followers and is good only for small values of p. In particular it may be accurate only when the branching process dies out with probability 1, because otherwise the model predicts that the expected number of active nodes is infinite while this number is obviously limited by the total number $N(= |V|)$ of nodes in the graph. The branching process extinguishes with probability 1 if $p\sum_k kq_k/\langle k\rangle < 1$, i.e. $p < 2 \times 10^{-4}$ in the considered Twitter graph. The second model addresses the case for $p > 2 \times 10^{-4}$. In this case branching process' theory predicts that the process can still extinguish with a probability p_{ext} that is a decreasing function of p. The intuition behind our second model is to couple the branching



(a) $p = 0.0001$ (b) $p = 0.001$

(c) $p = 0.01$ (d) $p = 0.1$

Figure 2: Performance of greedy, high-degree and random approach on the Twitter social graph for different values of retweet probability p. The reciprocation probability is considered to be $r = 1$.

process and the actual cascade model and assume that the cascade will reach almost all the N nodes when the branching process does not extinguish and a negligible number of nodes when it does. In particular, given that we are interested in providing upper-bounds for the variability of the process, we simply consider that the number of active nodes is equal to a random variable that is equal to N with probability $1 - p_{ext}$ and to 0 with probability p_{ext}. Some further refinements of the model lead to the conclusion that the number of samples needed to achieve a reasonable prevision is below 100 for all the values of p we considered, i.e. $p = 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$. This result is quite surprising, given the high variability of the degree distribution $\{q_k\}$ (that is power law) and the even higher variability of the skewed distribution $kq_k/\langle k \rangle$.

With the approach described above we have been able to perform our experiments on the *real* social graph with hundreds of millions of nodes and tens of billions of arcs.

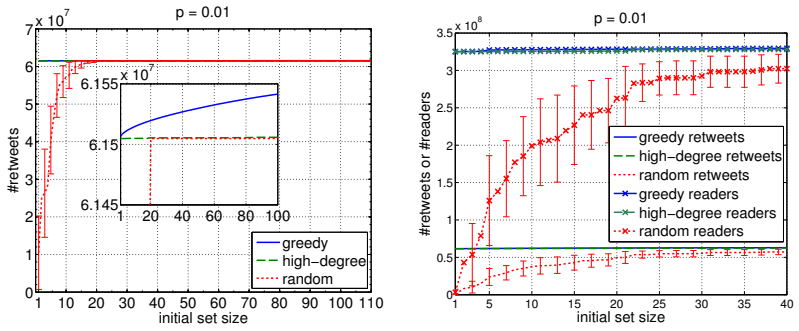
3.2 Results

We considered three selection strategies for the user u_0 to select its followers: i) the **greedy** strategy described above; ii) the **high-degree** strategy which consists in picking the nodes orderly according to their number of followers (from the largest number to the smallest); iii) the **random** strategy where followers are selected uniformly at random from the whole set of users. We have performed simulations for four values of the retweet probability: $p = 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$. The performance of the random strategy strongly depend on the choice of the random set of followers,

so we show 95% confidence intervals in the plots below.

In Fig. 2, we can see how these approaches perform in different situations when the reciprocation probability is equal to 1, i.e. u_0 is followed back from every node it follows. Moreover we consider that u_0's followers retweet u_0's tweets with probability 1, while all the others users retweet with probability p. The figure shows the expected number of retweets versus the initial number K of followers u_0 can choose. The average number of followers in the original graph is $\langle k \rangle \approx 45$, but the effective density, as defined in [4], is $\langle k \rangle p \approx 4 * 10^{-3}$ for $p = 10^{-4}$ (in Fig. 2 (a)). Then in this case most of the nodes are not retweeted by any follower, but due to the skewness of the distribution q_k (the number of followers can be as high as 24,635,412), there are some hubs in the social network that have an expected number of retweeters significantly larger than 0. The cascade processes from different followers of u_0 do not overlap much (each pruned graph is almost a forest of small-depth trees with a multitude of singletons), so that the high-degree strategy performs almost as well as the greedy algorithm. Due to this structure, the expected number of retweeters significantly increases as the number of u_0's followers keeps increasing. The random strategy performs very poorly, and basically, even selecting 200 followers, all of them are singletons with a very high probability. When $p = 10^{-3}$ (Fig. 2 (b)), the cascade originated from the node with the largest degree is already able to reach about $3 * 10^6$ users (roughly 1% of the whole social network) and both the greedy and the high-degree strategy select this node as first. The other followers selected from u_0 using these two strategies bring a much less significant improvement: even adding 199 more followers the expected number of retweeters increases by only 33%. The random strategy starts paying off because there are much less singletons in the pruned graph. Further increasing p to 10^{-2}, the contribution of the first follower is even larger and the contribution of the others even more marginal, as it is shown in the plot in Fig. 2 (c). In fact, a non-negligible strongly connected component appears in most of the pruned graphs and a careful choice of the first follower allows u_0 to have roughly one tenth of the nodes retweeting its tweets (this follower is not necessarily in the largest strongly connected component, but can reach it). The other 199 followers provide roughly 5% more retweeters. We observe that the effective density is about 0.4, then more than half of the nodes have 0 out-degree/in-degree in the pruned graphs. The greedy strategy and the high-degree one exhibit here the most significant difference in our set of experiments, but this is limited to less than 2% Interestingly, while most of the nodes do not provide any additional retweet, the probability to randomly pick a node in the largest strongly connected component is now quite high, so that the random strategy provides a small number of retweeters until a good follower in such component (or able to reach it) is selected and then the number of retweeters jumps to a value comparable to that of the other two strategies. These jumps cause the high variability revealed by the large confidence intervals in the figure. Moreover the figure shows how the good follower is very likely to be selected among the first 10-20 nodes. The same reasoning allows to explain also the curves in Fig. 2 (d) for $p = 0.1$. In this case greedy and high-degree are almost indistinguishable and random has almost the same performance for $K \geq 20$.

In the simulation described above we considered that every user u_0 chooses to follow will reciprocate ($r = 1$). We decided to perform some simulations considering for each user a reciprocation probability determined by the formula $r = \min\{\frac{\#\text{followings}}{\#\text{followers}+100}, 1\}$, where $\#followings$ and $\#followers$ are respectively the number of followings and followers for the given user. The rationale behind is that a user with a lot of followers and a few followings is not likely to reciprocate u_0. In any case we do not claim that this formula has any particular value, apart from allowing us to simply test the effect of heterogeneous reciprocation probabilities. The results for $p = 0.01$ are qualitatively unchanged as shown in Fig. 3 (a). We have also compared the different algorithms in terms of the expected number of users who can read the tweet. A result is presented in Fig. 3 (b). The number of readers is obviously much bigger than the number of retweeters,



(a) Reciprocation probability (b) Number of readers

Figure 3: Extensions of the experiments by taking into account the probability that users will follow back (a), and by looking at the number of users who received the tweet instead of the number of users who retweeted it (b).

but there is no significant difference in the relative performance of the three algorithms.

4 Conclusions

In this paper we have considered a user of a social network who tries to maximize its influence through a careful networking strategy. We have shown how greedy algorithms guarantee a good $1 - 1/e$ approximation ratio, but much simpler strategies like selecting users with the largest number of followers or even selecting random users may practically reach the same performance on real online social networks.

This research is partially supported by Alcatel Lucent Bell Labs in the framework of the ADR Network Science. The authors would like to thank Alonso Silva (Alcatel Lucent Bell Labs), Paolo Giaccone (Politecnico di Torino) and Damien Saucez (Inria) for the helpful discussions.

References

[1] A. Barrat, M. Barthélemy, and A. Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, 2008.

[2] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 57–66, New York, NY, USA, 2001. ACM.

[3] Maksym Gabielkov and Arnaud Legout. The Complete Picture of the Twitter Social Graph. In *ACM CoNEXT 2012 Student Workshop*, Nice, France, December 2012.

[4] Habiba and Tanya Y. Berger-Wolf. Working for influence: network density and influential individuals. In *Proceedings of the IEEE ICDM 2011 Workshop on Data Mining in Networks (DaMNet 2011)*, 2011.

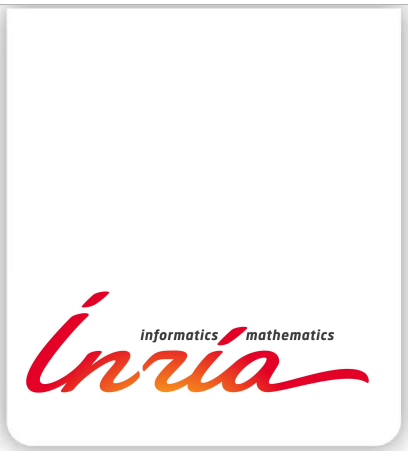
[5] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on*

Knowledge discovery and data mining, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[6] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Proceedings of the 32nd international conference on Automata, Languages and Programming*, ICALP'05, pages 1127–1138, Berlin, Heidelberg, 2005. Springer-Verlag.

[7] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA, 1988.

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



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399