An Upper Bound based Greedy Algorithm for Mining Top-k Influential Nodes in Social Networks

Chuan Zhou, Peng Zhang, Jing Guo, and Li Guo
Institute of Information Engineering, Chinese Academy of Sciences
Beijing, 100093, China
{zhouchuan,zhangpeng,guojing,guoli}@iie.ac.cn

ABSTRACT

Influence maximization [4] is NP-hard under the Linear Threshold (LT) model, where a line of greedy algorithms have been proposed. The simple greedy algorithm [4] guarantees accuracy rate of $1 - 1/e$ to the optimal solution; the advanced greedy algorithm, e.g., the CELF algorithm [6], runs 700 times faster by exploiting the submodular property of the spread function. However, both models lack efficiency due to heavy Monte-Carlo simulations during estimating the spread function. To this end, in this paper we derive an upper bound for the spread function under the LT model. Furthermore, we propose an efficient **UBLF** algorithm by incorporating the bound into CELF. Experimental results demonstrate that UBLF, compared with CELF, reduces about 98.9% Monte-Carlo simulations and achieves at least 5 times speed-raising when the size of seed set is small.

Categories and Subject Descriptors H.2.8 [Database Management]: Database Applications - Data Mining
General Terms Theory, Algorithms, Performance
Keywords Influence Maximization, CELF, Upper Bound.

1. INTRODUCTION

Influence maximization is defined as finding a small subset of nodes that maximizes *spread of influence* in social networks based on a given stochastic influence propagation model. Popular stochastic influence propagation models include the *Independent Cascade* (IC) model and the *Linear Threshold* (LT) model [4]. However, influence maximization under both models is NP-hard.

Kempe et al. [4] observed that the spread function is monotone and submodular, and proposed a simple greedy algorithm which repeatedly chooses the seed node with the maximal marginal gain. The simple greedy algorithm can approximate the optimal solution with a factor of $(1-1/e-\epsilon)$ for any $\epsilon > 0$.

However, the simple greedy algorithm is computational inefficiency, and two types of solutions have been proposed. First, many heuristic algorithms, such as DegreeDiscount [1] and ShortestPath [5], were proposed with orders of magnitude faster, but without theoretical accuracy guarantee, which may incur unboundedly bad results in some practical applications. Second, some sophisticated greedy algorithms were proposed with fewer Monte-Carlo estimations of the spread function. A representative work by Leskovec et al. [6] exploited the submodular property of the spread function and proposed a Cost-Effective Lazy Forward (CELF) algorithm, which improves the running time by up to 700 times. Moreover, Goyal et al. [3] proposed an extension of CELF, i.e., the CELF++ algorithm, which can further reduce the number of spread estimations by $35\% - 55\%$.

Although CELF and CELF++ significantly improve the simple greedy algorithm, their time costs are still very heavy on a large network [1]. In particular, they are relatively inefficient at the initial step, because they need to estimate the initial upper bound of spread using Monte-Carlo for each node in the network, leading to N times of Monte-Carlo calls (N is the network size). When N is very large, CELF and CELF++ are inapplicable. This limitation raises a fundamental question that *can we derive an upper bound of spreads which can be used to prune unnecessary spread estimations (Monte-Carlo calls) in the CELF algorithm*?

To answer the question, in this paper we derive an upper bound of spread under the LT model [1]. Based on the bound, we propose a new greedy-based algorithm *Upper Bound based Lazy Forward* (UBLF for short). Experiments show that UBLF is at least 5 times faster than CELF.

2. THE UPPER BOUND FOR $\sigma_L(S)$

In this part we derive the upper bound for the spread $\sigma_L(S)$ under the LT model. Note that the exact computation of $\sigma_L(S)$ is #P-hard [2]. As disucssed in the work [4], the LT model is tantamount to the reachability in a live-edge graph. Let $\mathcal{P}$ be the set of all simple paths with the starting node in $S \subseteq V$, then we have

$$\sigma_L(S) = \sum_{\pi \in \mathcal{P}} \prod_{e \in \pi} w(e). \quad (1)$$

Eq. (1) was originally given in the work [2]. Based on the equation, we derive a new Theorem 1 as follows,

THEOREM 1. *The upper bound for spread $\sigma_L(S)$ is*

$$\sigma_L(S) \leq \sum_{t=0}^{N-|S|} \Pi_0^S \cdot W^t \cdot \mathbf{1} \quad (2)$$

where $W = (w_{ij})$ is the weight matrix.

Proof: For $t = 0, \ldots, N-|S|$, let B_t be the set of all simple paths with length t in $\mathcal{P}$, and C_t be the set of all paths with length t and starting node in S. If a path belonging to C_t, the nodes are allowed to reappear. Then, we have

$$\sigma_L(S) = \sum_{t=0}^{N-|S|} \sum_{\pi \in B_t} \prod_{e \in \pi} w(e)$$

[1]For the upper bound of spread under the IC model, refer to the work [7].

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577336 .

Table 1: Number of Monte-Carlo simulations at the first 10 iterations.

Datasets	Algorihms	1	2	3	4	5	6	7	8	9	10	Sum
ca-GrQc	CELF	**5,242**	1	1	1	2	1	1	2	2	1	**5,254**
	UBLF	**43**	1	1	1	2	3	1	1	2	1	**56**
Wiki-vote	CELF	**7,115**	1	1	1	1	2	2	2	3	2	**7,130**
	UBLF	**58**	1	1	2	2	3	3	2	2	1	**75**

$$\leq \sum_{t=0}^{N-|S|} \sum_{\pi \in C_t} \prod_{e \in \pi} w(e) = \sum_{t=0}^{N-|S|} \Pi_0^S \cdot W^t \cdot \mathbf{1}$$

where the first '=' is derived from both Eq. (1) and the definition of B_t, the first '$\leq$' is due to $B_t \subseteq C_t$ and the second '=' is obtained from the graph theory. □

Furthermore, if the weight matrix W satisfies the condition $\max_v \sum_u w(u,v) < 1$, the upper bound of $\sigma_L(S)$ can be relaxed to

$$\sigma_L(S) \leq \Pi_0^S \cdot (E - W)^{-1} \cdot \mathbf{1}, \quad (3)$$

where E is a unit matrix and $(E-W)^{-1}$ is the inverse of the matrix $(E-W)$. By doing so, the upper bound in Eq. (3) is computationally tractable.

3. THE UBLF ALGORITHM

The key idea behind CELF is that the marginal gain of a node in the current iteration cannot be more than that in previous iterations. However, CELF demands N spread estimations to establish the initial bounds of marginal increments. In contrast, UBLF uses the upper bound given in Theorem 1 to rank all nodes in the initialization step, which eventually reduces the total number of spread estimations. We summarize UBLF in Algorithm 1.

Algorithm 1: UBLF under the LT model

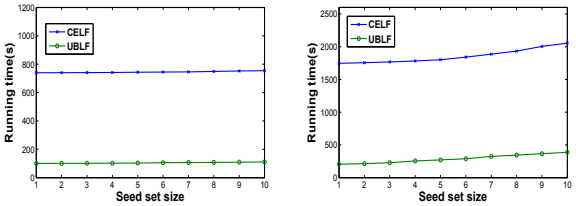
```
01: Input: weight matrix W and budget k
02: Output: the most influential node set S
03: initial S ← ∅ and δ ← bounds in Eq. (2) or Eq. (3)
04: for i = 1 to k do
05:    set I(v) ← 0 for v ∈ V\S
06:    while TRUE do
07:       u ← arg max_{v∈V\S} δ_v
08:       if I(u) = 0
09:          δ_u ← MC(S ∪ {u}) − MC(S);  I(u) ← 1
10:       end if
11:       if δ_u ≥ max_{v∈V\(S∪{u})} δ_v
12:          S ← S ∪ {u};  break
13:       end if
14:    end while
15: end for
16: output S
```

In Algorithm 1, the column vector $\delta = \{\delta_u\}$ denotes the upper bounds of marginal increments under the current seed set S, *i.e.*, $\delta_u \geq \sigma_I(S \cup \{u\}) - \sigma_I(S)$. Before searching for the first node (*i.e.*, $S = \emptyset$), we estimate the upper bound for each node by using Eq. (2) or Eq. (3). Then, the algorithm proceeds the same as CELF.

4. EXPERIMENTS

We conduct experiments on two real-world data sets, ca-GrQc and Wiki-vote [2], to evaluate the **UBLF** algorithm. We assign edge weights by following $w(u,v) = 1/(d_{in}(v)+1)$, where $d_{in}(v)$ is the in-degree of node v. We only compare UBLF with CELF, as the performance of CELF++ is almost the same as CELF [3]. In all the experiments, we run 10,000 times of Monte-Carlo simulations to estimate the spread.

[2]For details, visit *http://snap.stanford.edu/data/*.



(a) ca-GrQc (b) Wiki-vote

Figure 1: Runtime *w.r.t.* seed size k.

From Table 1, we can observe that the number of Monte-Carlo calls in UBLF is significantly reduced compared to that in CELF, especially in the first iteration. From the last column of Table 1, the total number of Monte-Carlo calls at the first 10 iterations of UBLF is reduced by 98.93% and 98.95% on the two data sets. From Fig. 2, we can observe that UBLF is at least **5** times faster than CELF.

5. CONCLUSIONS

In this paper we derived an upper bound for the spread function in the social network influence maximization problem. Based on the bound, we further proposed a new Upper Bound based Lazy Forward algorithm (**UBLF** in short). Compared with CELF, UBLF can significantly reduce the number of Monte-Carlo calls, e.g., over **98.9% reduction of Monte-Carlo calls** in our experiments. The experimental results also verified that UBLF is at lease **5 times** faster than CELF when the size of seed set k is small.

Acknowledgements. This work was supported by the NSFC (No. 61370025) and the Strategic Leading Science and Technology Projects of CAS (No.XDA06030200).

6. REFERENCES

[1] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD 2009*.

[2] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM 2010*.

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

[4] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD 2003*.

[5] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. In *PKDD 2006*.

[6] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD 2007*.

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