Estimate on Expectation for Influence Maximization in Social Networks*

Yao Zhang, Qing Gu**, Jun Zheng, and Daoxu Chen

State Key Lab. for Novel Software and Technology,
Nanjing University, Nanjing 210093, China
zhangyao.cs@gmail.com, junzheng@smail.nju.edu.cn,
{guq,cdx}@nju.edu.cn

Abstract. Finding the most influential nodes is an important issue in social network analysis. To tackle this issue, Kempe et al. proposed the natural greedy strategy, which, although provides a good approximation, suffers from high computation cost on estimating the influence function even if adopting an efficient optimization. In this paper, we propose a simple yet effective evaluation, the *expectation*, to estimate the influence function. We formulate the expectation of the influence function and its marginal gain first, then give bounds to the expectation of marginal gains. Based on the approximation to the expectation, we put forward a new greedy algorithm called Greedy Estimate-Expectation (GEE), whose advantage over the previous algorithm is to estimate marginal gains via expectation rather than running Monte-Carlo simulation. Experimental results demonstrate that our algorithm can effectively reduce the running time while maintaining the influence spread.

1 Introduction

Information diffusion is one of the most important issues in social network analysis. A problem in this field is to find a k-nodes subset S that nodes in S can influence the largest number of nodes in the whole network. This problem, referred as *influence maximization problem*, can be applied to many areas such as product marketing and application promotion in online communities.

Domingos and Richardson [1][2] first investigated the influence propagation in the area of viral marketing [3][4]. Then, Kempe et al. [5] formulated the influence maximization problem. They proposed a natural greedy algorithm to solve the influence maximization problem, which provided a $(1 - 1/e)$-approximation. However, their greedy algorithm was quite time-consuming to evaluate the influence spread, as it needed to run random process for a large amount of times to guarantee an accurate estimate on the influence spread.

* This work is supported in part by the National High-Tech Research and Development Plan of China (863) under Grant No. 2006AA01Z177, the 973 Program of China under Grant No. 2009CB320705, the NSFC Project under Grant No. 60873027 and Jiangsu Provincial NSF Project under Grant No. BK2006115.

** Corresponding author.

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 99–106, 2010.

Recent researches have focused on solving this drawback, and several improvements have been proposed. Leskovec et al. put forward an optimization called Cost-Effective Lazy Forward (CELF) [6]; Kimura and Saito posed a bond-percolation based improvement [7] and a model called SPM (Shortest Path Model) [8]; Chen et al. [9] studied the influence maximization from two complementary directions: one was to improve the simple greedy algorithm, and the other was to design new efficient heuristics.

In this paper, we propose a novel evaluation, the *expectation*, to the influence spread, whose advantage is that it avoids running Monte-Carlo simulation. We formulate the expectation of influence function and its marginal gain, and give bounds to the expectation of marginal gains in theory. Then, we show that a good estimate on the expectation can be obtained by graph-based algorithms, and furthermore, a pruning technique is proposed for estimating the expectation.

Based on the expectation, a new greedy algorithm, referred as *GEE (Greedy Estimate-Expectation)*, is put forward for the influence maximization problem. Experimental results demonstrate that GEE is well-performed in the influence spread and running time for both independent cascade (IC) model and weighted cascade (WC) model compared to the simple greedy algorithm with CELF optimization (10-140 times faster in running time and only at most 2.4% lower in influence spread). And moreover, the running time would be even faster if we apply CELF optimization to our GEE algorithm.

The main contributions of this paper can be concluded as follows: first, we provide a novel evaluation, the *expectation*, to estimate the influence function, which, to the best of our knowledge, is the first time that using expectation to circumvent a large amount of computation on running random process; second, we give a theoretical explanation to the effectiveness of SPM (Shortest Path Model) [8]; third, we put forward the first expectation-based greedy algorithm and demonstrate its effectiveness on real-life networks.

2 Background

Influence Maximization Problem. We define $\sigma(S)$ as the number of nodes that are influenced by k-nodes set S, then the influence maximization problem is formulated as *finding a subset* $\hat{S}$ *in* V*, where* $|\hat{S}| = k$*, to maximize* $\sigma(\hat{S})$. The computation of $\sigma(S)$ is based on information diffusion models.

Information Diffusion Models. We discuss two information diffusion models: independent cascade model and weighted cascade model [5]. In both of them, node v is influenced by its neighbor u with a probability $p_{u,v}$. In IC model, $p_{u,v}$ is an independent parameter, and in WC model, $p_{u,v}$ is assigned to $1/d_v$.

The information diffusion process for two models is described below [5]. First, the initiate set S is given. We call nodes in S active nodes, while nodes in $V \backslash S$ inactive. Nodes can transform from active state to inactive state, but can not switch verse vice. When node u first becomes active at step t, it provides only a single chance to activate each currently inactive neighbor v with probability $p_{u,v}$. If u succeeds, v will become active at step $t+1$, and u can not activate

v any more after step t. If v has multiple active neighbors at step t, neighbors' activations are sequenced in an arbitrary order. The diffusion process stops when there are no more activities in the network.

General Greedy Algorithm. Kempe et al. proposed a simple greedy algorithm to approximate the solution [5], which starts with an empty set $S = \emptyset$, and iteratively, selects a node u for set S to maximize the marginal gain $\delta_s(u) = \sigma(S \cup \{u\}) - \sigma(S)$, then the algorithm stops until $|S| = k$.

$\sigma(S)$ is computed by simulating the random process for R times (R could be very large in order to guarantee efficiency). Leskovec et al. [6] proposed a CELF optimization, which can get the same result but is much faster than Kempe et al.'s algorithm, for its great reduction of computing $\delta_s(u)$. But it still costs for hours on large-scale networks.

3 Proposed Method

In this section, we use *expectation* to estimate the influence function $\sigma(S)$ and the marginal gain $\delta_s(u)$, and give an approximation to the expectation of $\delta_s(u)$. Then we propose an algorithm called *Greedy Estimate-Expectation(GEE)* for the influence maximization problem.

3.1 Estimate on Expectation

We denote $p(S, v)$ as the propagation probability that v is influenced by Set S. Suppose the probabilities that other nodes influence node v are independent, according to information diffusion models described above, we have $p(S, v) = 1 - \prod_{\forall u \in S}(1 - p(u, v))$.

The expectation of $\sigma(S)$, formulated as $E(\sigma(S))$, is:

$$E(\sigma(S)) = \sum_{\forall v \in V} p(S, v) * |\{v\}| = \sum_{\forall v \in V} p(S, v) \tag{1}$$

According to $\delta_s(u) = \sigma(S \cup \{u\}) - \sigma(S)$ and the above equations, we have $E(\delta_s(u)) = \sum_{\forall v \in V}(1 - p(S, v)) * p(u, v)$. Suppose $R(u, G)$ is the set of nodes which are reachable from u, so:

$$E(\delta_s(u)) = \sum_{v \in R(u,G)} (1 - p(S, v)) * p(u, v) \tag{2}$$

We denote $p_{pathi}(u, v)$ as the propagation probability from u to v through path i. Let $\hat{p}(u, v) = max\{p_{pathi}(u, v) | v \in R(u, G)\}$, and $\lambda_u = max\{\lambda_{u,v} | v \in R(u, G)\}$, where $\lambda_{u,v}$ is the number of paths from u to v.

Theorem 1. *For IC model and WC model, if $p(S, v)$ is given, then:*

$$\sum_{v \in R(u,G)} (1 - p(S, v)) * \hat{p}(u, v) \leq E(\delta_s(u))$$

$$\leq min\{1,\ \lambda_u \sum_{v \in R(u,G)} (1 - p(S, v)) * \hat{p}(u, v)\}.$$

We define $k_{u,v}$ as the distance(shortest path) from u to v, and then, for IC model with uniform propagation probability p, we have $\hat{p}(u,v) = p^{k_{u,v}}$.

If we denote t_i as the number of path whose length is i from node u to node v, then in IC model, $p(u,v) \leq \sum_{i=k_{u,v}}^{l} t_i p^i$, where l is the maximum length of paths from u to v.

Theorem 2. *For IC model with uniform propagation probability p, if $p(S,v)$ is given, and $t = max(t_{k_{u,v}}, t_{k_{u,v}+1}, ..., t_l)$, then:*

$$E(\delta_s(u)) \leq t * \frac{p^{n-1}-1}{p-1} \sum_{v \in R(u,G)} (1 - p(S,v)) p^{k_{u,v}}.$$

Theorem 1 gives bounds to the expectation of $\delta_s(u)$ for both IC model and WC model. Theorem 2 provides an upper bound that is closer to the expectation of $\delta_s(u)$ for IC model. In a sparse graph with a small value of p, $t\frac{p^{n-1}-1}{p-1}$ is close to 1, which means $E(\delta_s(u))$ can be estimate by $\sum_{v \in R(u,G)} (1 - p(S,v)) p^{k_{u,v}}$ effectively. It is an amazing result that theoretically interprets why Shortest Path Model (SPM: the model where each node is activated only through the shortest paths) [8] works well.

Theorem 1 and Theorem 2 show that the expectation of $\delta_s(u)$ can be estimate by computing $\hat{p}(u,v)$. Suppose $\hat{E}(\sigma(S))$ is the estimate of $E(\sigma(S))$ through estimating $p(u,v)$ by $\hat{p}(u,v)$, then $\hat{E}(\delta_s(u)) = \sum_{v \in R(u,G)} (1 - p(S,v)) \hat{p}(u,v)$.

According to the equation that $p(S \cup \{u\}) = p(S,v) + (1 - p(S,v))p(u,v)$, $p(S \cup \{u\}, v)$ can be estimated by previous $p(S,v)$ in a greedy approximate algorithm. The value of $p(u,v)$ can be effectively approximated by $\hat{p}(u,v)$. For IC model, we are able to obtain $\hat{p}(u,v)$ by $k_{u,v}$ through *Breadth-First Search (BFS)*, which takes $O(n(n+m))$ time. For WC model, the algorithm to get $\hat{p}(u,v)$ is resemble to shortest-path algorithm in a weighted graph, such as the *Dijkstra Algorithm*. Using *Fibonacci heap*, the running time is $O(n(n \log n + m))$.

We denote $d_{u,max}$ as the maximum degree of node $v \in R(u,G)$. Let $z = p * (d_{u,max} - 1)$.

Theorem 3. *For IC model with uniform propagation p, if $z < 1$, $\forall \varepsilon > 0$, $\exists K = \lfloor \log_z (\varepsilon * \frac{1-z}{d_u p} + z^n) \rfloor$, for all node $v \in R(u,G)$ and $k_{u,v} \leq K$, then*

$$\hat{E}(\delta_s(u)) - \sum_{\substack{v \in R(u,G) \\ k_{u,v} \leq K}} (1 - p(S,v)) p^{k_{u,v}} < \varepsilon.$$

Theorem 3 suggests that under the condition of $z < 1$, $\hat{E}(\delta_s(u))$ in IC model can be approximated effectively in the case that $k_{u,v} \leq K$, which means *BFS* for computing $\hat{p}(u,v)$ can be terminated within K layers! We call this *BFS pruned BFS*. For those networks whose K are not large, *pruned BFS* makes progress in running time compared to original *BFS* that takes $O(n(n+m))$ time.

3.2 GEE Algorithm

We put forward an algorithm called *Greedy Estimate-Expectation(GEE)* for influence maximization problem. There are two phases in *GEE*: calculating $\hat{p}(u,v)$ and greedily obtaining the set S_k.

Algorithm 1. Greedy Estimate-Expectation Algorithm

Phase One

If *MODEL=IC*, then Get $\hat{p}(u,v)$ for all (u,v) by *full BFS* or *pruned BFS*
If *MODEL=WC*, then Get $\hat{p}(u,v)$ for all (u,v) by *Fibonacci heap* or *pruned BFS*

Phase Two

Input $\hat{p}(u,v)$ for all (u,v)
Initialize $S := \emptyset$, and $p(S,v) := 0$ Foreach $v \in V$
for $i := 1$ to k **do**
  Foreach $u \in V \backslash S$, $\hat{E}(\delta_s(u)) := \sum_{v=0}^{|V|} \hat{p}(u,v)(1 - p(S,v))$, if $\hat{p}(u,v)! = 0$
  Select a node u with maximum $\hat{E}(\delta_s(u))$
  $S := S \cup \{u\}$
  $p(S,u) := 1$
  Foreach $v \in V \backslash S$, update $p(S,v) := p(S,v) + \hat{p}(u,v)(1 - p(S,v))$
end for
return S

The running time of the first phase depends on its implementation. As mentioned above, $\hat{p}(u,v)$ for all nodes (u,v) can be obtained by running *BFS* or *pruned BFS* with $O(n(n+m))$ time or $O(n\frac{\bar{d}^K-1}{\bar{d}-1})$ time, and by using *Fibonacci heap* with $O(n(n\log n + m))$ time. We also take *pruned BFS* as a complement for WC model, and it performances well in our experiments.

The second phase of *GEE* is to get a node u for S_k once at a time with the maximum $\hat{E}(\delta_s(u))$. Instead of running random process for sufficient times, *Phrase Two* takes $O(kn^2)$ in running time, and if we only consider K layers in *BFS*, the average running time could be $O(kn\frac{\bar{d}^K-1}{\bar{d}-1})$. Moreover, we can also use the CELF optimization to accelerate *Phrase Two*.

$\hat{E}(\sigma(S))$, the estimate function of $E(\sigma(S))$, is a submodular function, which means *GEE* algorithm provides a $(1-1/e)$-approximation according to the property of submodular function [10].

4 Experiments

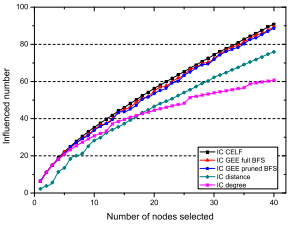
We employ two collaboration networks from paper-lists in sections of the e-print arXiv. The first network is from the "General Relativity and Quantum Cosmology" (*Gr-Qc*) section with 5242 nodes and 28980 edges[1]. The second network is from the "High Energy Physics - Theory" (*Hep*) section with 15233 nodes and 58891 edges [9]. All experiments are implemented on a PC with Intel 2.20GHz Pentium Dual E2200 processor and 4GB memory.

Table 1 lists algorithms used in our experiments. *Degree* and *Distance* are simple heuristics used in [5]. Note that in *GEE with pruned BFS*, if initializing $\varepsilon = 10^{-8}$ and using $\bar{d}$, then we have $K = 6$ in *Gr-Qc* graph and $K = 5$ in *Hep* graph. To make experiment results convincing, we simulate the random process for $R = 20000$

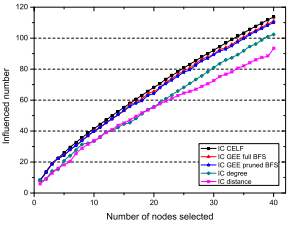
[1] http://snap.stanford.edu/data/ca-GrQc.html

Table 1. Algorithms used in experiments

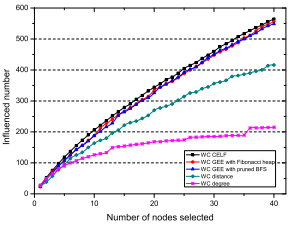
Algorithm	Applied Model	Description
Degree	IC, WC	Degree heuristic
Distance	IC, WC	Distance heuristic
Greedy with CELF	IC, WC	CELF optimization($R = 20000$)
GEE with pruned BFS	IC, WC	Algorithm 1 ($\varepsilon = 10^{-8}$)
GEE with full BFS	IC	Algorithm 1
GEE with Fibonacci heap	WC	Algorithm 1



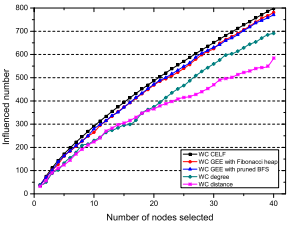
(a) IC Model for Gr-Qc Graph



(b) IC Model for Hep Graph



(c) WC Model for Gr-Qc Graph



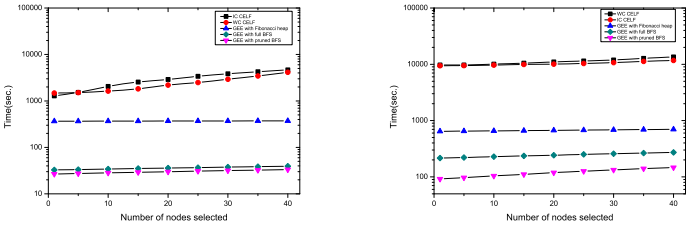
(d) WC Model for Hep Graph

Fig. 1. Influence spread. (a) IC model for Gr-Qc graph. (b) IC model for Hep graph. (c) WC model for Gr-Qc graph. (d) WC model for Hep graph.

times (the same as times in [9]), and then, take the average of the influence spread numbers as the influence spread results for each algorithm.

In IC model, we mainly discuss experiments with an uniform $p = 0.01$. We also consider $p = 0.02$. We do not report its result as its trends on influence performance and running time are similar to the situation that $p = 0.01$.

Influence spread. In all figures, we discuss about percentages of influence spread for the case of $k = 40$. Figure 1(a)(b) demonstrate the influence spread results of different algorithms with probability $p = 0.01$ in IC model. Our *GEE with pruned BFS* and *GEE with full BFS*, performs quite well. They are only about 1.5% and 2.4% lower than the *Greedy with CELF* in the *Gr-Qc* graph and *Hep* graph respectively. And the differences between two *GEE* algorithms are less than 1%, which means our pruning technique works well as we expect.



(a) Gr-Qc Graph (b) Hep Graph

Fig. 2. Running time(sec.). (a) Gr-Qc graph. (b) Hep graph.

Figure 1(c)(d) demonstrate the influence spread results of different algorithms in WC model. The performance of our *GEE with pruned BFS* and *GEE with Fibonacci heap* are quite close to *Greedy with CELF*. *GEE with Fibonacci heap* is only 1.4% and 2.3% lower than the *Greedy with CELF* in the *Gr-Qc* graph and *Hep* graph respectively. It is surprising that *GEE with pruned BFS* with much faster running time, outperforms *GEE with Fibonacci heap* for some value of k when $k < 40$.

Running time. Figure 2 demonstrates the running time of various algorithms. We do not list running times of *Degree* and *Distance* for their poor performances on the influence spread. Our *GEE*s run orders of magnitude faster compared to *Greedy with CELF*. Specifically, when $k = 40$, *GEE with Fibonacci heap* is 11 times and 19 times faster than *Greedy with CELF* in the *Gr-Qc* and *Hep* graph respectively, and *GEE with full BFS* is 119 times and 43 times faster than *Greedy with CELF* in the *Gr-Qc* and *Hep* graph respectively. *GEE with pruned BFS*, for its pruning technique, impressively saves the running time for about 80 to 140 times in the *Gr-Qc* and *Hep* graph!

5 Discussion

Our *GEE*s show very impressive experiment results for both IC model and WC model: they further improve the running time (range from 10 times to 140 times faster when 40 nodes is selected), while almost match the influence spread of *Greedy with CELF* (only 1.4% to 2.4% lower).

There are other well-performed algorithms for the influence maximization problem: shortest path model(SPM) [8], bond-percolation based algorithm(BP) [7], and degree discount heuristic [9]. We do not list these experimental results for the following reasons: degree discount heuristic, although almost matches the influence spread in much faster time compared to the simple greedy algorithm, is only limited to the IC model; BP is similar to the improvement of the greedy algorithm discussed in [9], which costs as much time as the CELF optimization; SPM can only apply to IC model, and moreover, it can get the similar result as our *GEE with full BFS* in slower time due to the implementation of SPM.

6 Conclusion

In this paper, we propose a new evaluation, the *expectation*, to estimate the influence function and its marginal gain for the influence maximization problem. We give bounds to the expectation of $\delta_s(u)$, and further, theoretically interpret the effectiveness of Shortest Path Model(SPM) [8]. Then we put forward an expectation based algorithm called Greedy Estimate-Expectation(GEE). Using two collaboration networks, we experimentally demonstrate that our GEE algorithm impressively shortens the running time while maintaining the influence results that obtained by the simple greedy algorithm with the CELF optimization.

Future research will try to use the expectation to estimate the influence function on other diffusion models. Another direction is to explore the internal structures of networks to improve the influence spread.

References

1. Domingos, P., Richardson, M.: Mining the network value of customers. In: KDD, pp. 57–66 (2001)
2. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: KDD, pp. 61–70 (2002)
3. Goldenberg, J., Libai, B., Muller, E.: Talk of the networks: a complex systems look at the underlying process of word-of-mouth. Marketing Letters 12 (2001)
4. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. TWEB 1(1) (2007)
5. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD, pp. 137–146 (2003)
6. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.S.: Cost-effective outbreak detection in networks. In: KDD, pp. 420–429 (2007)
7. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: AAAI, pp. 1371–1376 (2007)
8. Kimura, M., Saito, K.: Tractable models for information diffusion in social networks. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 259–271. Springer, Heidelberg (2006)
9. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: KDD, pp. 199–208 (2009)
10. Nemhauser, G., Wolsey, L., Fisher, M.: An analysis of the approximations for maximizing submodular set functions. Mathematical Programming 14, 265–294 (1978)