Sampling Based Influence Maximization on Linear Threshold Model

Su Jia and Ling Chen

1 Department of Computer Science, Yangzhou University, Yangzhou, China
E-mail: jiasu.forever@qq.com

Abstract. A sampling based influence maximization on linear threshold (LT) model method is presented. The method samples the routes in the possible worlds in the social networks, and uses Chernoff bound to estimate the number of samples so that the error can be constrained within a given bound. Then the active possibilities of the routes in the possible worlds are calculated, and are used to compute the influence spread of each node in the network. Our experimental results show that our method can effectively select appropriate seed nodes set that spreads larger influence than other similar methods.

1. Introduction
With the rapid development of Internet, people use social networks more frequently. In social networks, users can influence others’ opinions, actions, emotion through their own influence propagation. The prevalence of social networks has brought up some new applications, for example: public opinion control, viral marketing etc. Hence, how to measure and forecast one user’s or more users’ influence on others in social networks is a challenging problem. The problem of influence maximization is defined as: given a network under a propagation model, we should find a set of starting nodes that can eventually influence the largest number of nodes in the network. The starting nodes set is called seed set, influence maximization will find the seed set with the largest propagation. Recently, with the prosperity of many social networks, such as Facebook, Twitter, Sina and Renren, analysis of social networks has become a hot point of research. Especially the research on influence maximization in social networks has great significance in advertisement release, viral marketing[1], and message propagation.

Over the past decade, many new propagate models and influence maximization methods have been proposed. Those models and methods adopted theories and technologies in the areas of mathematics, computer science and data mining.

Kempe et al.[1] use a discrete optimization model to describe the problem of influence maximization. A given network is presented by a directed weighted graph, where nodes represent the users, and the weights on directed edges represent the influence between users. An integer k (k≤|V|) is given as the size of the seed set, it is also called the budget. We need to find k users as seeds, and if they are activated and propagated under a diffusion model, the influence will get the largest spread.

In LT model [1], every node will choose a value from [0, 1] independently and uniformly as its active threshold. For any inactive node, when total weight of its active neighbors exceed its threshold, it will turn to be active. In both models, diffusion process will end when no more inactive nodes can be activated. At this point, the number of nodes that can be activated during the propagation process can be used to measure the influence of the seed set.
Domingos and Richardson[2], [3] first proposed and studied the influence maximization problem. They treat influence maximization as a discrete optimization problem, and propose a greedy algorithm. However, due to the huge time costed by the greedy algorithm, it cannot be applied to social networks of large size. Leskovec et al. proposed CELF [2] along with its improved version algorithm CELF++ [3] and its improvement CELF++ which delay to calculate marginal gain of influence spread in order to reduce the processing time. Chen et al. [4] proposed a nodes’ degree based method named DegreeDiscount which is an improvement of the Degree algorithm. Algorithm PMIA [5] constructs a tree structure for each node, and the influence spread and refresh is restricted within such tree region so as to reduce calculating time. However, it requires more memory space. To reduce the computation time and the memory requirement, we present an efficient method for influence maximization on linear threshold model. The method first uses Chernoff bound to sample the routes in the possible world in the social network. After sampling, it calculates the active possibility of each route in the possible world. Finally, those possibilities will be used to calculate influence indicator of every node in social networks. Our experimental results show that the algorithm can effectively select appropriate seed nodes set to maximize the influence of propagation.

2. Concept and Definitions

2.1. Linear Threshold Model

A social network can be represented by a directed graph \( G = \langle V, E, P \rangle \), where V is the vertex set, E is the directed edge set, \( p_{ij} \) is possibility matrix. Let \( (v_j, v_j) \in E \) be a directed edge in G, element \( p_{ij} \) is the possibility for influence spreading from \( v_j \) to \( v_j \). At any time, each node in the network stays in either active or inactive state. An active node will stay in active state throughout the whole process. An inactive node can be activated by its active neighbors. The more active neighbors it has, the larger probability it can be activated. As one of the basic propagation models of social networks, linear threshold model [1] mainly focuses on the threshold behavior in the process of influence propagation, namely, the cumulative effect of influence during the propagation. In this model, when an attempt by an active node to activate its inactive neighbor fails, its influence will be accumulated during the active process. Such accumulated influence will have contribution to the possibility that other active nodes activate this non-activated node. Given a graph \( G=(V,E) \), denote neighbors of node \( v \) in \( G \) as \( N(v) \). Let \( S \) be the seed nodes set, \( \theta_v \in [0,1] \) be the threshold of node \( v \). The propagation process under linear threshold [1] is as follows. For a node \( v \) in \( G \), denote the set of its active neighbors as \( S(v) \). At any time, if \( \sum_{u \in S(v)} p_{uv} \geq \theta_v \), then node \( v \) changes its state to active.

After \( v \) being activated, it will attempt to activate its inactive neighbors. The influence of an inactive node will increase over time. The above procedure will be repeated until no more nodes can be activated.

2.2. Uncertain graph and possible world

We consider the original activation probability on an edge of the graph as the existence probability of the edge, then the graph can be treated as an uncertain graph.

Definition 1 (uncertain graph):

An uncertain graph can be represented by a triple \( G = \langle V, E, P \rangle \), where \( P: E \rightarrow (0,1) \) is the edge existence function, \( V \) and \( E \) represent the sets of vertexes and edges respectively.
The possibility \( p \in (0,1) \) on each edge represents the existence possibility of this edge, if \( p = 1 \), then this edge definitely exists. If the probabilities of all edges in the graph are equal to 1, the graph is a traditional deterministic graph.

Definition 2 (Possible World):

For an uncertain graph \( G = (V, E, P) \), if a certain graph \( G' = (V, E') \) satisfies \( E' \in E \), then \( G' \) is one possible world that belongs to \( G \).

Obviously, an uncertain graph \( G = (V, E, P) \) contains \( 2^{|E|} \) possible worlds. We denote the set of all possible worlds of \( G \) as \( \mathbb{W}(G) \).

Let \( G' = (V, E') \) be a possible world of \( G \), then the existence possibility of \( G' \) is:

\[
\Pr(G') = \prod_{e \in E'} p(e) \prod_{e \in E - E'} [1 - p(e)] \tag{1}
\]

It can easily be seen that the sum of existence possibilities of all possible worlds of \( G \) is 1, that is:

\[
\sum_{G' \in \mathbb{W}(G)} \Pr(G') = 1 \tag{2}
\]

3. The Activation Possibility of Route

Kempe et al. [1] had proved that LT model is equivalent to the “Live-edge” model (LE) [1].

In LE model, every node chooses one in-edge as “live-edge” based on the probability of in-edge connecting with the node.

Once an edge is selected, it can be called a ‘live’ edge, while the other edges not selected are called ‘dead’.

After such edge selection for all the nodes, we can obtain a possible world of \( G \). Let \( \sigma(S) \) the number of nodes that can be activated by the seed set \( S \) in the LT model. Then \( \sigma(S) \) is equal to the mathematical expectation of the number of vertices that can be reached in all possible worlds.

Denote \( G' \) as a possible world of \( G \), \( \sigma_{G'}(S) \) is the number of reachable nodes starting from \( S \) in Graph \( G' \), then we have

\[
\sigma(S) = \sum_{G' \in \mathbb{W}(G)} \Pr(G') \sigma_{G'}(S) \tag{3}
\]

Then \( \sigma_{G'}(S) \) can be calculated by the following formula:

\[
\sigma_{G'}(S) = \sum_{v \in F} I(S, v, G') \tag{4}
\]

In (4)

\[
I(S, v, G') = \begin{cases} 1 & \text{exit route from } S \text{ to } v \\ 0 & \text{otherwise} \end{cases}
\]

Combine (3) and (4) we can get
$$\sigma(S) = \sum_{G \in \Psi(G)} \sigma_G(S)$$

$$= \sum_{G \in \Psi(G)} \sum_{v \in V} I(S, v, G') \Pr(G')$$

$$= \sum_{G \in \Psi(G)} \sum_{u \in U} \sum_{v \in V} I(S, v, G') \Pr(G')$$

$$= \sum_{u \in S} \sum_{v \in V} I(S, v, G') \Pr(G')$$

$$= \sum_{u \in S} \sum_{v \in V} \Pr(u, v)$$  \hspace{1cm} (5)$$

Here

$$\Pr(u, v) = \sum_{v \in V} \Pr(u, v)$$

is the activate possibility of the existence of the route between u and v.

Now we define:

$$\Pr(u) = \sum_{v \in V} \Pr(u, v)$$  \hspace{1cm} (6)$$

then

$$\sigma(S) = \sum_{u \in S} \Pr(u)$$  \hspace{1cm} (7)$$

From (7) we can see that calculating $$\sigma(S)$$ needs to compute $$\Pr(u)$$ for every node u in S, namely, we should calculate the possibilities of all the routes starting from u.

4. ROUTE SAMPLING

The use of sections to divide the text of the paper is optional and left as a decision for the author. From (6) we can see that formula (7) needs to calculate $$\Pr(u, v)$$ for every vertex v, where $$\Pr(u, v)$$ is defined as the existence possibility of the route between u and v, that is:

$$\Pr(u, v) = \sum_{G \in \Psi(G)} I(u, v, G')$$  \hspace{1cm} (8)$$

It can be seen from formula (8) that computing $$\Pr(u, v)$$ needs to calculate $$I(S, v, G')$$ in all possible worlds of G. That means to detect whether there is a path from node u to v in every possible world. It needs to enumerate all the possible worlds of G, which requires huge amount of computation time. To reduce the computational cost, we use a sampling method to randomly select some possible worlds from $$W(G)$$, Denote the set of selected possible worlds are $$U(G)$$. We use $$U(G)$$ as a replacement of $$W(G)$$, and use

$$\frac{1}{|U(G)|} \sum_{G \in U(G)} I(u, v, G')$$

as an unbiased estimation of
To restrict the error of the estimation with a given bound $\varepsilon$, we use Chernoff bound to estimate the size of sampling set $U(G)$.

**Theorem 1 (Chernoff Bound):**

Let $X_1, X_2, \ldots, X_r$ be independent and identically distributed random variables, they have the same exception $\mu = E[X_i]$, if $r > \frac{3}{\varepsilon^2} \ln \frac{2}{\delta}$, then

$$\Pr\left(\left|\frac{1}{r} \sum_i X_i - \mu\right| \geq \varepsilon \mu\right) \leq \delta$$

Let $I_1, I_2, \ldots, I_r$ be the values of $f(u, v, G')$ of the possible worlds $G_1, G_2, \ldots, G_r$ in $U(G)$ and $U(G) = r$, then

$$\mu = \frac{1}{|U(G)|} \sum_{G' \in U(G)} f(u, v, G')$$

is the exception for $\frac{1}{r} \sum_i X_i$. For a given probability $\delta$, if we set

$$|U(G)| \geq \frac{3}{\varepsilon^2} \ln \frac{2}{\delta}$$

Then the relative error of estimation will be less than the given error bound $\varepsilon$.

Since the exception $\mu = E[X_i]$ is required in (9), we set a threshold $\rho$ such that the route $(u, v)$ with $\Pr(u, v)$ less than $\rho$ can be ignored. Then we can use $\rho$ to replace $\mu$ in (9), and set

$$r = \frac{3}{\varepsilon^2 \rho} \ln \frac{2}{\delta}$$

as the size of sampling set.

5. **Framework of The Algorithm and Complexity Analysis**

According to the above analysis, we propose a sampling based influence maximization algorithm named Sampling_IM. First, the algorithm calculates the sampling size $r$ using (10), and constructs $r$ possible worlds. Then it calculates the activate routes in each possible world. Based on the activate routes obtained, the algorithm can compute the route’s activating possibility $\Pr(u)$ for each node $u$.

Finally, we select $p$ nodes with the largest $\Pr(u)$ values as the seeds set $S$. Given an error bound $\varepsilon$, possibility $\delta$, route exception threshold $\rho$, the framework of our algorithm Sampling_IM is described as follows:

**Algorithm 1 Sampling_IM**

**Input:** $G = (V, E, P)$
\[ p : \text{size of seed set}; \]
\[ \varepsilon : \text{error bound}; \]
\[ \delta : \text{possibility (confidence)}; \]
\[ \rho : \text{routes exception threshold}; \]

**Output:** \( S : \) seed set;

1: use (10) to calculate sampling size \( r \):

/*construct \( r \) possible worlds*/

2: for \( k = 1 \) to \( r \) do

\( \text{construct_PW}(G, P, G); \)

end for \( k \);

3: for all node pair \((u, v)\) in \( G \) do

\( \Pr(u, v) = 0; \)

end for;

/*calculate the number of nodes that can be activated by each node \( u \) of the graph*/

4: for \( i = 1 \) to \( r \) do

\( \text{path_counting}(G); \)

end for \( i \);

5: for every node \( u \in V \) do

\( \Pr(u) = \sum_{r \in P, u \neq v} \Pr(u, v); \)

end for \( u \);

6: get \( p \) nodes with larger values of \( \Pr(u) \) to formulate seed set \( S \);

7: output seed set \( S \)

In algorithm Sampling_IM, step 1 calculates the sampling size \( r \) using (10), given parameters \( \varepsilon \), \( \delta \) and \( \rho \). Step 2 constructs \( r \) possible worlds of graph \( G \). Steps 3 and 4 calculate the number of nodes that can be activated by each node \( u \) in the graph. Step 6 selects \( p \) nodes with the larger \( \Pr(u) \) values than that of the others to constitute the seeds set \( S \).

In the algorithm, step 1 uses (10) to calculate \( r \), the complexity is \( O(1) \). Step 2 constructs \( r \) possible worlds of graph \( G \), the complexity is \( O(r \times m) \), where \( m = |E| \) is the number of edges in \( G \). The complexity of Step 3 is \( O(n^2) \).

Step 4 calculates the number of nodes that can be activated by each node \( u \) in \( G \), its complexity is \( O(r \times n^2) \). Step 5 accumulates the \( \Pr(u) \) for all nodes in \( G \), it costs \( O(n \times p) \) time. Step 6 selects \( p \) nodes to constitute the seeds set \( S \), its time complexity is \( O(n \times p) \). Step 7 outputs seed set \( S \) in \( O(p) \) time. In summary, the complexity of algorithm Sampling_IM is \( O(n^2) \).

6. **Experiment and Analysis**

In order to verify the accuracy of the proposed algorithm, we conduct experiments on four datasets DBLP, Epinions, LiveJournal and NetHPET that are benchmarks in the literature of influence maximization[7]. Then we analyze the test results.
The algorithm is coded by Matlab and run on a PC with 2.4GHz Intel I5 6300HQ, 16G memory under Microsoft Windows 10.

6.1. Datasets and contrast algorithm
We test our algorithm on three true world datasets as shown in Table I.

| Name | DBLP | Epinions | NetHPET |
|------|------|----------|---------|
| Node | 650k | 13k      | 15k     |
| edge | 2M   | 84k      | 62k     |

We compare our algorithm with PageRank [6] and DegreeDiscountIC [4]. PageRank is a commonly used algorithm for ranking web pages. In LT model, the transition probability on edge \((u, v)\) is interpreted as influence of node \(v\) on \(u\). After the nodes are ranked by PageRank, \(k\) nodes with the highest ranking values are selected as the seed set. DegreeDiscountIC is used for IC model. In the method, all edge propagation probability is set as 0.01. The basic idea of this algorithm is that it always activates the inactive nodes with the largest degree. However, after some neighbors of node \(v\) are selected as seeds, the degree of \(v\) should be decreased by a discount. Then the algorithm activates \(v\) according to its discounted degree. In LT model, this method can be used as a general heuristic method [4].

6.2. Propagation model and settings
Linear threshold model (LT) is a classic propagate model, we widely used WC method [1] to set up parameters on the edges.

The propagation possibility on edge \((u, v)\) in graph \(G\) is set as:

\[
b_{u,v} = \frac{A(u,v)}{\text{N}(v)}
\]

Here, \(A(u,v)\) is the number of directed edges from node \(u\) to node \(v\), and \(\text{N}(v)\) is the indegree of node \(v\):

\[
\text{N}(v) = \sum_{u \in \text{V}^n(v)} A(u,v)
\]

The activate threshold is generated randomly and uniformly between 0 and 1.

6.3. Experiment result and analysis
Tables should have only horizontal rules and no vertical ones. Generally, only three rules should be used: one at the top of the table, one at the bottom, and one to separate the entries from the column headings. Table rules should be 0.5 points wide.

6.3.1. Result on DBLP. In the experiments on DBLP, we set \(\epsilon = \frac{1}{100000}\), \(\rho = 6\), \(\delta = 0.99\)

By (10) we get \(r = 3516000\) which means that we should sample 3516000 possible worlds in DBLP in our experiment. We compare the influence spreads of different algorithms, and the result is shown in Figure 1.
Figure 1: Comparison of the influence spreads of different algorithms on DBLP

From Figure 1, we can see that on DBLP datasets, the seed sets of different sizes selected by algorithm Sampling_IM can achieve the largest propagation range compared with algorithm PageRank and DegreeDiscountIC.

6.3.2. Result on Epinions. In the experiments on Epinions, we set $\varepsilon = \frac{1}{1000}$, $\rho = 6$, $\delta = 0.99$ By (10) we get $r = 351600$, and we sampled 351600 possible worlds in Epinions in our experiment. We compare the influence spreads of different algorithms, and the result is shown in Figure 2.

Figure 2: Comparison of the influence spreads of different algorithms on Epinions

From Figure 2, we can see that on Epinions datasets, the seed sets of different sizes selected by algorithm Sampling_IM can achieve the largest propagation range compared with DegreeDiscountIC. Compared with PageRank, Sampling_IM can achieve larger propagation range in most situations. And with expansion of the seed set size, Sampling_IM can achieve larger propagation range increment than PageRank.

6.3.3. Results on NetHPET. If In the experiments on NetHPET, we set $\varepsilon = \frac{1}{1000}$, $\rho = 6$, $\delta = 0.99$ Using (10), we get $r = 351600$, therefore we sampled 351600 possible worlds in the experiment on DBLP. Comparison of the influence spreads of different algorithms is shown in Figure 3.
Figure 3: Comparison of the influence spreads of different algorithm on NetHPET

From Figure 3, we can see that on NetHPET datasets, the seed sets of different sizes selected by algorithm Sampling_IM can achieve almost the same propagation range compared with algorithm PageRank and DegreeDiscountIC. In some cases, the spread of Sampling_IM is larger than the PageRank algorithm.

The experimental results above show that the proposed algorithm Sampling_IM can select seed set with larger propagation influence compared with the other two algorithms. Therefore, the algorithm Sampling_IM outperforms DegreeDiscountIC and PageRank in terms of influence spread. Furthermore, since Sampling_IM uses the sampling method instead of enumerating all the possible worlds in the network, it can greatly reduce the computation time.

7. Conclusions

In social networks, how to select one or more users that will make maximum influence on other users is a critical problem. In this paper, we present a sampling based algorithm for influence maximization on linear threshold model. The method uses Chernoff bound to estimate the size of the sampling possible worlds so that the error can be constrained within a given bound. Then the active possibilities of the routes in the possible worlds are calculated, and are used to compute the influence spread of each node in the network. The nodes with the largest influence spread are selected to form the seed set. Our experimental results show that our method can effectively select appropriate seed nodes set that can spread larger influence than other similar methods, and we use sampling technique to improve the efficiency.

8. References

[1] D.Kemp, J.M. Kleinberg, E. Tardos 2003 Maximizing the spread in influence through a social network, in KDD
[2] J. Leskovec et al. In KDD 2007 Cost-effective outbreak detection in networks.
[3] A. Goyal, W. Lu, and L. V. S. Lakshmanan, 2011 “CELF++: Optimizing the greedy algorithm for influence maximization in social networks,” in WWW (Companion Volume).
[4] W. Chen, Y. Wang, and S. Yang, in KDD 2009 “Efficient influence maximization in social networks.”
[5] Chen W, Wang C, Wang Y. in KDD 2010 “Scalable influence maximization for prevalent viral marketing in large-scale social networks.”
[6] Pavel Berkhin. 2005, 2(1):73-120 A Survey on PageRank Computing[J]. Internet Mathematics.
[7] Kim J, Kim S K, Yu H. 2013 Scalable and parallelizable processing of influence maximization for large-scale social networks. In ICDE, pages 266-277.

Acknowledgments

This research was supported in part by the Chinese National Natural Science Foundation under grant No. 61379066, 61070047, 61379064, 61472344, 61402395, 61602202, Natural Science Foundation of Jiangsu Province under contracts BK20130452, BK2012672, BK2012128, BK20140492 and Six talent peaks project in Jiangsu Province(Grant No. 2011-DZXX-032).