Influence Maximization in Social Networks Based on Non-backtracking Random Walk

Jingzhi Pan, Fei Jiang*, Jin Xu
School of Electronics Engineering and Computer Science
Peking University
Beijing, China
{panjingzhi, fei.jiang1989, jxu}@pku.edu.cn

***Abstract*—Influence maximization is an optimization problem that aims at finding a small set of nodes as seed nodes to maximize the spread of influence in a social network. In this paper, we introduce a new technique called Non-backtracking Random Walk into the influence maximization problem. An optimized algorithm, called NBRW, is devised through sampling nodes when applying restricted non-backtracking walk in network communities. Node's traversing number gives a good estimate of its influence. Our algorithm can be executed in linear time, which beats other influence maximization algorithms. We give extensive experiments under various diffusion models against other state-of-the-art node-selection algorithms. It demonstrates that in terms of influence range and convergent speed our algorithm gives a superior performance under most diffusion models, especially for linear threshold model.**

Keywords-influence maximization; random walk; non-backtracking; diffusion models; social networks

I. INTRODUCTION

Influence maximization problem is to find a small number of individuals strategically in a social network to adopt a new product or innovation to trigger maximum further adoptions. In literature [1], Kempe, Kleinberg and Tardos define this problem as following discrete optimization problem. A social network is modeled as a graph $G = (V, E)$ with vertices (nodes) in V representing individuals and edges in E representing the relationship between two individuals. Influence is propagated in the network under a diffusion model. Influence maximization problem aims to find k nodes (seed nodes) in the network such that under the specific diffusion model the expected number of nodes influenced by the k seed nodes is as large as possible, where k is a parameter.

After Kempe, many studies aim to improve the performance and efficiency of algorithms for solving influence maximization problem, including improving diffusion models, adjusting the algorithm adaptive to specific diffusion model, refining algorithms to reduce computational costs, designing algorithms for large-scale networks and searching new ideas to select seed nodes. From 2003 to date, quite a number of achievements have been made, such as the shortest-path based diffusion models (SPM and SP1M), the cost-effective lazy forward-selection (CELF), the maximum influence arborescence (MIA) model and prefix excluding MIA (PMIA) model, the degree discount method and so on.

In this paper, we introduce a new technique called Non-backtracking Random Walk into the influence maximization problem, devise a novel algorithm by sampling nodes when non-backtracking walking in the network, and estimate the influence of nodes by their traversing number. The algorithm prevents the excessive accumulation of high-degree vertices' influence and finds the crucial nodes with lower degree quickly. We leverage non-backtracking walk in different network communities so as to extend diffusion scope as large as possible. The time complexity of our NBRW algorithm is *O(n)*, which is very fast among all works so far. By comparative experiments, it is shown that our algorithm performs excellent in terms of influence range and convergent speed in all kinds of datasets under the linear threshold model, and gives satisfactory results in ego networks under the weighted cascade model.

The rest of the paper is organized as follows. In section 2, we review recent works related to influence maximization. In section 3, we introduce some preliminaries, including the diffusion models, the concept of random walk and the non-backtracking operator. In section 4, we present our algorithm in detail. In section 5, the effectiveness of our algorithm is demonstrated by extensive experiments. At last, we conclude this paper.

II. RELATED WORK

Since influence maximization problem is introduced into social networks by Domingos and Richardson [2, 3] in 2001, there is an increasing number of researchers beginning to study this problem. To solve influence maximization problem, researchers have proposed lots of algorithms. These algorithms can be mainly classified into two categories: greedy algorithms and heuristic algorithms.

A. Greedy Algorithms

Domingos and Richardson are the first to study influence maximization in social networks, but their method is probabilistic. Later, Kempe et al. [1] systematically define this problem as a discrete optimization problem and prove that the optimization problem of finding the most influential nodes is NP-hard. Also, they present a greedy approximation algorithm applicable to all three models, and by using an analysis framework based on submodular functions they prove that the

978-1-5090-1192-6/16 $31.00 © 2016 IEEE
DOI 10.1109/DSC.2016.114

optimal solution for influence maximization can be efficiently approximated to within a factor of $(1 - 1/e - \varepsilon)$.

In 2006, Kimura and Saito [4] propose two natural models for information diffusion in a social network, called SPM and SP1M, which are special cases of ICM based on shortest paths. Their experiments show that the SPM and SP1M give good approximation to the ICM as finding sets of influential nodes but more efficiently.

In 2007, Leskovec et al. [5] represent an optimized algorithm of greedy strategy named Cost-Effective Lazy Forward algorithm. CELF adopts the submodularity property of influence maximization to greatly reduce the number of computations in influence spread of vertices which achieves near-optimal placements, and provides a novel theoretical result for non-constant node cost functions, while being 700 times faster than a simple greedy algorithm. 4 years later, Goyal [6] propose CELF++ algorithm, which improves the efficiency of CELF by 35-55%.

In 2007, Estevez, Vera and Saito [7] propose a Set Covering Greedy (SCG) algorithm dealing with the problem of overlapping neighborhoods, using concepts taken from set covering algorithms. SCG finds the set of nodes whose influence is maximum over the network without overlapping neighborhoods.

In 2009, Chen et al. [8] study efficient influence maximization from two complementary directions, propose an improvement of the original greedy algorithm to further reduce its running time, and propose a new degree discount heuristics algorithm that improves influence spread.

In 2010, Chen et al. [9] put forward MIA model and its extension PMIA model as well as algorithms for these models. These two models are based on maximum influence paths and utilize local arborescence structures to avoid global computations. Wang, Cong and Song [10] propose a new algorithm called Community based Greedy algorithm for mining top-k influential nodes, this method is divided into two parts: firstly, detecting communities in a social network with information diffusion, and then selecting influential nodes from the detected communities.

In 2011, Goyal, Lu and Lakshmanan [11] study influence maximization under the linear threshold model, first they propose SIMPATH algorithm based on CELF under the LT model, and then provide two optimizations to reduce the number of spread estimation calls made by simpath called Vertex Cover Optimization and Look Ahead Optimization.

In 2015, Heidari, Asadpour and Faili [12] put forward a greedy method based on the idea of state machine (State Machine Greedy, SMG) by reducing repeated computations to improve the computational efficiency of the greedy method. SMG saves the last state of simulations at the end of each step of greedy choice and uses it to efficiently prevent recalculations in estimation of marginal gain for candidate nodes in selection of next seed.

B. Heuristic Algorithms

In addition to greedy strategies, heuristic algorithms are also used to study influence maximization such as the previously mentioned Degree Discount method.

In 2011, Jiang et al. [13] propose an approach based on Simulated Annealing (SA) for influence maximization problem, they present two heuristic methods to accelerate the convergence process of SA, called SAEDV and SASH, and a new method of computing influence to speed up the proposed algorithm. Lü et al. [14] devise LeaderRank, utilizing the leadership topology and identifying influential users. LeaderRank refines PageRank to make it adaptive to social networks and uses ground node regulate probability flows, making LeaderRank parameter-free.

In 2012, Jung, Heo and Chen [15] provide an algorithm IRIE that integrates the advantages of influence ranking (IR) and influence estimation (IE) methods for influence maximization in both the independent cascade (IC) model and its extension IC-N that incorporates negative opinion propagations. The basic idea of IRIE is similar to greedy methods, selecting one node each round, until k nodes are selected. Hou, Yao and Liao [16] propose a ranking measure from a structural view for identifying influential nodes, present the concept of all-around distance to quantify the influence of nodes through synthetically combining degree, betweenness and k-shell ranking measures. Zeng and Zhang [17] propose a mixed degree decomposition (MDD) procedure considering both the residual degree and the exhausted degree, which outperforms k-shell and degree methods in ranking spreaders.

In 2013, Liu, Ren and Guo [18] present an improved method to generate the ranking list to evaluate node's spreading influence, by taking into account the shortest distance between a target node and the node set with the highest k-core value. Chen et al. [19] propose a local ranking algorithm named ClusterRank, which takes into account not only the number of neighbors and the neighbors' influence, but also the clustering coefficient.

In 2015, Tang, Shi and Xiao [20] provide a martingale approach to solve influence maximization problem, the proposed algorithm named IMM consists of two phases, sampling and node selection, which runs in near-linear time and achieves high efficiency.

III. PRELIMINARIES

In this section, we will introduce some preliminaries including diffusion models, random walk and the concept of non-backtracking.

A. Diffusion Models

The influence maximization problem is to maximize the scope of information diffusion. So we need to know how information is propagated in a social network, that's why diffusion models are proposed. There are three most basic and widely studied diffusion models, that is, the linear threshold model (LTM), the independent cascade model (ICM) and the weighted cascade model (WCM). We employ two of them in this paper. In these models, each node is either active or inactive. When one node is active, it means that the corresponding individual adopts the information or the innovation. On the contrary, an inactive node stands for an individual that does not adopt the information or the innovation. These three models are all progressive models in

which nodes are influenced by their neighbors to switch from being inactive to being active, but do not switch in the other direction. Suppose that the diffusion process starts from a given active node set, we say seed node set.

Linear threshold model. In this model [1], a node v is influenced by each neighbor w according to a weight $b_{v,w}$ bound to an edge such that $\sum_{w\ neighbor\ of\ v} b_{v,w} \leq 1$. Any node v becomes active for which the total weight of its active neighbors is at least $\theta_v : \sum_{w\ active\ neighbor\ of\ v} b_{v,w} \geq \theta_v$. The threshold θ_v is chosen uniformly by node v at random from the interval [0, 1]; this represents the weighted fraction of v's neighbors that must become active in order for v to become active.

For the lack of threshold knowledge and for simplicity, we set the threshold of all nodes to be the same in the experiments.

Independent cascade model. This model is a probabilistic model, proposed by Goldenberg, Libai and Muller [21, 22]. In this model, a node try to activate its currently inactive neighbors only when it first becomes active. Suppose node v is a newly-activated node in step t. For each inactive neighbor w, v can activate w with a probability $p_{v,w}$. If it succeeds, then w will become active in step $t+1$. However, whether v succeeds or not, it cannot try to activate w in later steps. If w has multiple newly-activated neighbors in step t, the sequence its neighbors try to activate it does not matter.

Weighted cascade model. This model [1] can be viewed as a special case of the Independent Cascade Model. In this model, the probability that active node v activates its inactive neighbor w relates to the degree of w. Suppose the degree of w is $d(w)$, then the probability $p_{v,w} = \frac{1}{d(w)}$.

B. Random Walk

A random walk is a mathematical formalization of a path that consists of a succession of random steps [23].

In computer science, the Markov chain describing the sequence of nodes visited by a random walker is called a random walk. A random variable $s(t)$ contains the current state of the Markov chain at time step t: if the random walker is in state i at time t, then $s(t) = i$. The random walk is defined with the following single-step transition probabilities of jumping from any state or node $i = s(t)$ to an adjacent node $j = s(t+1)$: $P(s(t+1) = j | s(t) = i) = a_{ij}/a_{i.} = p_{ij}$, where $a_{i.} = \sum_{j=1}^{n} a_{ij}$, the elements a_{ij} of the symmetric adjacency matrix A of the graph are defined as usual as: $a_{ij} = 1$ if node i is connected to node j, and $a_{ij} = 0$ otherwise.

C. Non-backtracking Operator

A non-backtracking walk on a graph is a directed path such that no edge is the inverse of its preceding edge, which means a walk starting at v cannot turn around and return to it immediately. The non-backtracking operator in our algorithm has many benefits. Firstly, this mechanism prevents the excessive accumulation of high-degree vertices' influence. Secondly, it finds the crucial nodes such as cut-vertex with lower degree precisely and quickly. Thirdly, the leaf nodes in the network will not be selected into the seed node set, because their neighbors can activate them and under the action of non-backtracking operator, they have nowhere to go. Moreover, the non-backtracking matrix can be used to count non-backtracking walk paths of a given length.

IV. ALGORITHMS

In this section, we describe our algorithm NBRW in detail. Firstly, we introduce mathematical theory of our algorithm, and then we explain the actual implementation of NBRW, followed by pseudo code. Finally, we analyze the time complexity of this algorithm.

To redeem the performance of spectral algorithms in sparse networks, Florent et al. [24] define a non-backtracking matrix B on the directed edges of the graph. Specifically,

$$B_{(u\to v),(w\to x)} = \begin{cases} 1 & if\ v = w\ and\ u \neq x \\ 0 & otherwise \end{cases}$$

To apply this concept in the undirected graph, each undirected edge is replaced by two directed edges, as seen in Fig. 1. B is a $2m \times 2m$ matrix for an undirected graph, the definition of B guarantees the path is non-backtracking for $b_{(u\to v),(v\to u)} = 0$.

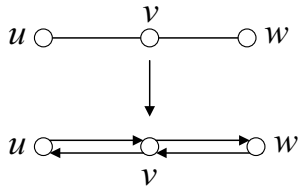


Fig. 1. An illustrating graph

For example, in Fig. 2, there are 4 nodes and 3 undirected edges in the graph. Replace each undirected edge with two directed edge($u \to v, v \to w, v \to x, v \to u, w \to v, x \to v$), so B is a 6×6 matrix, only $b_{(u\to v),(v\to w)} = 1$, $b_{(u\to v),(v\to x)} = 1$, $b_{(w\to v),(v\to u)} = 1$, $b_{(x\to v),(v\to u)} = 1$, $b_{(w\to v),(v\to x)} = 1$, $b_{(x\to v),(v\to w)} = 1$, the other elements are zero. Therefore, the non-backtracking matrix B of the graph in Fig. 2 is:

$$B = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (1)$$

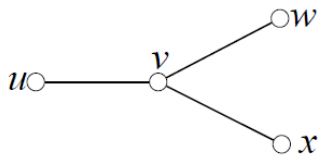


Fig. 2. An example graph

Now, we define B^+ as the normalization of the non-backtracking matrix B. Specifically,

$$B^{+}_{(u\to v),(w\to x)} = \begin{cases} \dfrac{1}{d(v)-1} & if\ v = w\ and\ u \neq x \\ 0 & otherwise \end{cases}$$

in which, *d(v)* represents the degree of node v.

A non-backtracking random walk on a graph is a walk that cannot return to its preceding visiting node immediately. As shown in Fig. 3, suppose that at time $t-1$ a random walker just traverses from node x and stops at node u. Then, at time t, the non-backtracking walker cannot be allowed returning to node x immediately. The next visiting node should be either node v or node w.

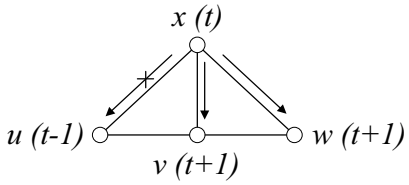


Fig. 3. An illustrating graph

With matrix B^{+}, the non-backtracking random walk can be formalized as follows. If the random walker is at node x from node u at time t, then the probability it reaches node v at time $t+1$ is $P(x \to v, t+1) = B^{+} \times P(u \to x, t)$. With starting at a concerned node, after sufficient steps of walks, the probabilities at which a non-backtracking random walk can reach at corresponding nodes will converge to a stationary state. These probabilities will provide us crucial information for estimating nodes' influence. To calculate the probability, we define the matrix B^{*} as seen below,

$$B^{*} = \begin{pmatrix} 0 & I - D^{-1} \\ -I & D^{-1}A \end{pmatrix}$$

B^{*} is a $2n \times 2n$ matrix, which allows us to work with a $2n$-dimensional matrix rather than a $2m$-dimensional one and significantly reduces the computational complexity of the algorithm.

Summing over incoming and outgoing edges also can reduce the complexity of computation. Given a $2m$-dimensional vector g, define g^{in} and g^{out} as the n-dimensional vectors

$$\mathrm{g}^{in}_{u} = \sum_{v\in N(u)} \mathrm{g}_{v\to u} \quad \text{and} \quad \mathrm{g}^{out}_{u} = \sum_{v\in N(u)} \mathrm{g}_{u\to v}$$

If we apply B^{+} to g, each edge $w \to v$ with $w \neq u$ contributes $\frac{1}{d(u)}$ times to the incoming edge $v \to u$. Similarly, each incoming edge $v \to u$ contributes $\frac{d(u)-1}{d(u)}$ times to the outgoing edges of u. As a result, we have

$$(B^{+}\mathrm{g})^{in}_{u} = \frac{1}{d(u)} \sum_{v\in N(u)} (\mathrm{g}^{in}_{v} - \mathrm{g}^{out}_{u})$$

and

$$(B^{+}\mathrm{g})^{out}_{u} = \frac{d(u)-1}{d(u)} \mathrm{g}^{in}_{u}$$

Succinctly,

$$\begin{pmatrix} (B^{+}\mathrm{g})^{out} \\ (B^{+}\mathrm{g})^{in} \end{pmatrix} = B^{*} \begin{pmatrix} \mathrm{g}^{out} \\ \mathrm{g}^{in} \end{pmatrix}$$

Finally, at the stationary state, g^{in} equals to a probability distribution in which each item represents its probability of a non-backtracking random walker will be in the corresponding node at present. For the sake of efficiency, we estimate the influence of nodes by the traversing number of the nodes when stimulating a non-backtracking random walk in the social networks instead of calculating the probability mathematically.

In order to gain more effectiveness and refrain the walk from sinking into a dilemma, we apply a divide and conquer strategy: we first separate the whole network into communities and then non-backtracking random walk is restricted in network communities. This strategy is motivated by two observations: (1) the probability of influential nodes to be selected in each community is not equal, but inclined to the dense communities with more nodes. Restricting non-backtracking random walk in network communities adjusts the probability which extends the diffusion range intuitively, (2) if walk in the whole network, once the walk traverses out of the community, it is hard to walk back to this community.

Next, we will introduce our algorithm. Firstly, as a preprocessing, partition the network by using existing community detection algorithm. Next, sampling nodes as the start points of non-backtracking random walk, and then a non-backtracking random walker traverses from the start points in the community the start point belonged to. Finally, ranking the nodes according to the traversing number. In addition, we just sample 100 nodes as start points of non-backtracking random walk. Moreover, the end condition of our algorithm is that the traversing number of each node is at least two. The whole framework called as NBRW is described in algorithm 1.

Algorithm 1 The overall framework of NBRW

Input: Graph G = (V, E) and parameter k
Output: k nodes selected

```
1.  Coms ← partition the graph G into communities;
2.  StartNodes ← random.choice(V,100);
3.  For all startnode in StartNodes do
4.      ComCurrent
            ← Community which the startnode in;
5.      NodesWithAccNum
            ← RandomWalkNB(startnode, ComCurrent);
6.      AllNodesWithAccNum.append(
            NodesWithAccNum);
7.  EndFor
8.  AllNodesSorted
        ← DescSort(AllNodesWithAccNum);
9.  Pick the first k nodes from AllNodesSorted
```

Given a start node, a random walker wanders to its neighbors with equal probability. Notice that it is not allowed walking to the node it comes from and it can only walk around its own community. The algorithm for non-backtracking random walk in a community for a given start node is described in algorithm 2.

Algorithm 2 The algorithm for non-backtracking random walk in a community

Input: start node and community C = (Vc, Ec)
Output: nodes with accessed number in Vc

```
1.  prenode ← -1; NodesWithAccNum ← 0;
2.  NodesWithAccNum[startnode]
        ← NodesWithAccNum[startnode] + 1;
3.  While TRUE
4.      /* If all nodes have been accessed two times, exit
           the loop */
5.      neighbors ← startnode.neighbors()
6.      Do While nextnode = prenode
7.          nextnode
                ← random.choice(neighbors);
8.      EndWhile
9.      NodesWithAccNum[nextnode]
            ← NodesWithAccNum[nextnode] + 1;
10.     prenode ← startnode;
11.     startnode ← nextnode;
12. EndWhile
```

To analyze the time complexity of this algorithm, suppose the average degree of the social network is $\langle k \rangle$, we need to make $O(\langle k \rangle)$ visits from any start node to their neighbors, so, to make sure each node is accessed twice at least, $O(\langle k \rangle n)$ visits is needed. In this way, the time complexity of NBRW is $O(\langle k \rangle n)$, $O(n)$ in a simple manner since that $\langle k \rangle$ is usually a constant around 10 for general real networks.

V. Experiments And Results

In order to verify the effectiveness of the algorithm, we make experiments in various kinds of real social networks to check the practical diffusion results of the algorithm and compare with other node-selection algorithms to analyze the relative advantages and disadvantages. Then we analyze the reasons these methods behave so and summarize the conditions in which they should be used. We make the experiments using three datasets under two diffusion models introduced in the third part. The detail information of the datasets and the algorithms for comparison will be introduced next. This section is divided into two parts. In the first part, we introduce the datasets and the algorithms for comparison in the experiments; in the second one, we analyze the experiment results from different aspects, and summarize merits and defects.

A. Datasets and Comparative Algorithms

Datasets. To verify the effectiveness of the algorithm, we make experiments in three real social network datasets: Facebook, HepTh and CondMat.

- Facebook is collected from survey participants using Facebook app. The dataset includes node features (profiles), circles, and ego networks (the ego network is a network consist of a focal node "ego" and the nodes to whom ego is directly connected to, called "alters", plus the ties, if any, among the alters). The data includes a total of 4039 nodes and 88234 edges.
- HepTh is from the e-print repository arXiv and covers scientific collaborations between authors papers submitted to High Energy Physics - Theory category. If an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. If the paper is co-authored by k authors this generates a completely connected (sub)graph on k nodes. The data includes a total of 9877 nodes and 25998 edges.
- CondMat is also from the e-print repository arXiv and covers scientific collaborations between authors papers submitted to Condense Matter category. Similarly, if an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. If the paper is co-authored by k authors this generates a completely connected (sub)graph on k nodes. The data includes a total of 23133 nodes and 93497 edges.

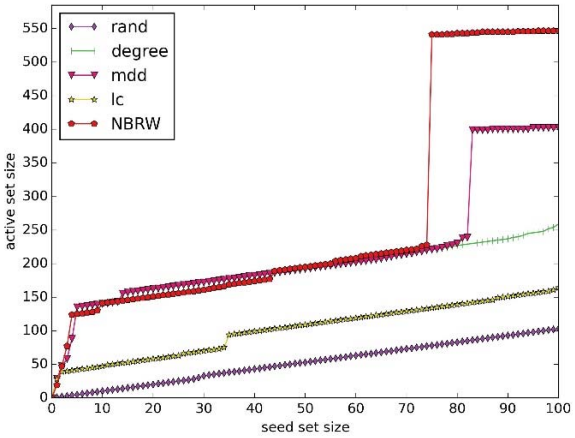
Comparative Algorithms. To evaluate our algorithm, we compare our algorithm with other node-selection algorithms: Random, Degree, Local Centrality, MDD.

- Random is a straightforward algorithm that just selects each of the k nodes completely randomly.
- Degree selects the top-k nodes with maximum degree as seed nodes.
- The mixed degree decomposition algorithm (MDD) [17] considers both the residual degree and the exhausted degree in the process of getting k-core. For node i, we respectively denote the residual degree (number of links connecting to the remaining nodes) and the exhausted degree (number of links connecting to the removed nodes) as $k_i^{(r)}$ and $k_i^{(e)}$. In each step of MDD procedure, the nodes are removed according the mixed degree $k_i^{(m)} = k^{(r)} + \lambda * k_i^{(e)}$. The tunable parameter λ in this algorithm is set to 0.7 in the experiments.
- Local centrality (LC) [25] is a heuristic algorithm that considers both the nearest and the next nearest neighbors. The local centrality $C_L(v)$ of node v is defined as $Q(u) = \sum_{w \in \Gamma(u)} N(w)$, $C_L(v) = \sum_{u \in \Gamma(v)} Q(u)$, Where $\Gamma(u)$ is the set of the nearest neighbors of node u and $N(w)$ is the number of the nearest and the next nearest neighbors of node w. This method selects nodes by local centrality measure.

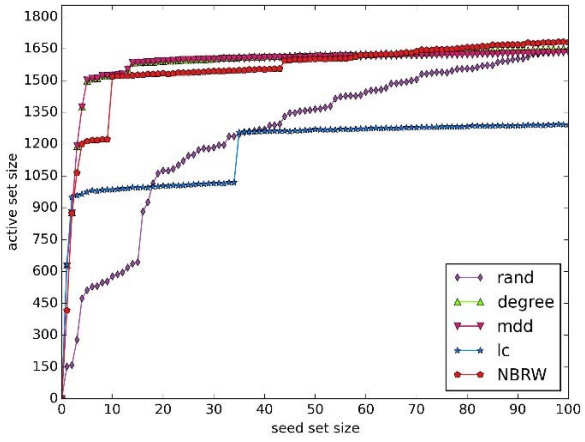
B. Experiment Results

Select nodes by using the five methods above as seed nodes, simulate the information propagation process under the linear threshold model and the weighted cascade model, then observe the diffusion scope under the influence of seed nodes. To ensure the high confidence level, 10000 repeated simulations is carried out in the seed node set, and the average of the 10000 results is taken as the final result of the propagation process.

Fig. 4 shows the diffusion scope through different seed node sets under the linear threshold model and the weighted cascade model respectively in Facebook. The x-axis represents the number of seed nodes selected as initial active nodes, denoted as "seed set size". The y-axis stands for the number of active nodes in all at the end of the diffusion process, denoted as "active set size". (a) is the result under the linear threshold model when the threshold is 0.5, (b) is the result under the weighted cascade model.



(a) Linear Threshold Model, $\theta = 0.5$



(b) Weighted Cascade Model

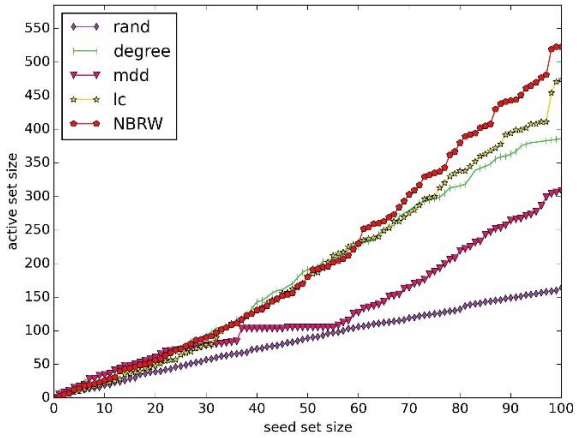
Fig. 4. Diffusion results in Facebook

As it is shown, in each picture, the active set size increases with the increase of seed set size, or just stays the same. When it stays the same, which means the new join seed node has been activated by preceding nodes. So even if it is not selected as seed node, it can still be active at last. Thus adding it into the seed set cannot extend the diffusion scope. This is an obvious case especially in the Degree method. In most of the approaches, Random selection method performs worst. This is because it does not use any property of the network while others all take advantage of the attributes of the network more or less.

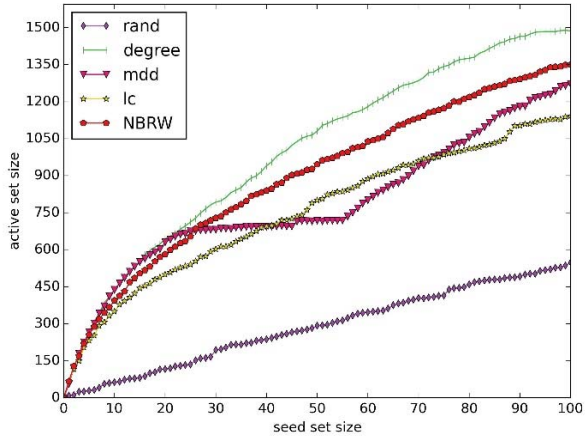
As you can see, there is a sudden rise in the curve of NBRW and MDD under the linear threshold model. This is mainly due to the existence of threshold in the model. In linear threshold model, for each inactive node, it can be activated by its active neighbors only when the total influence of them exceeds the threshold, so there may exist a period of accumulation. With more active nodes joining, the total influence increases slowly but is still smaller than the threshold. Once the number of active nodes is large enough to exceed the threshold, quite a number of nodes meet the condition being activated.

Under the linear threshold model, compared to other algorithms, the sudden rise of the NBRW algorithm appears much earlier, and the final number of the active nodes of our algorithm is 36% more than MDD algorithm, and 111% more than Degree method. Under the weighted cascade model, compared with LC, the number of active nodes of NBRW is always greater, and the final number of the active nodes of our algorithm is the largest among all of the algorithms.

Fig. 5 shows the diffusion scope through different seed node sets under the linear threshold model and the weighted cascade model respectively in HepTh. Similarly, the x-axis represents the number of seed nodes selected as initial active nodes, denoted as "seed set size". The y-axis stands for the number of active nodes in all at the end of the diffusion process, denoted as "active set size". (a) is the result under the linear threshold model when the threshold is 0.5, (b) is the result under the weighted cascade model.

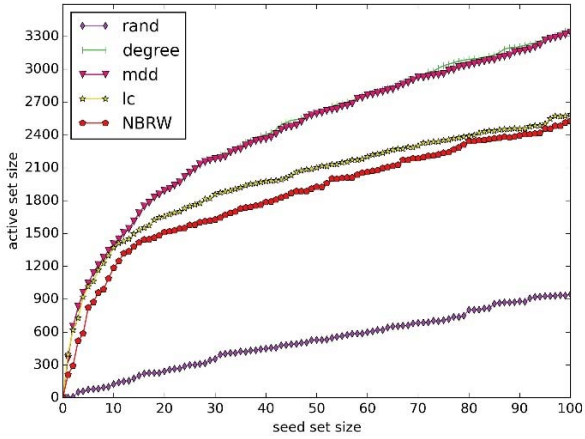


(a) Linear Threshold Model, $\theta = 0.5$



(b) Weighted Cascade Model

Fig. 5. Diffusion results in HepTh

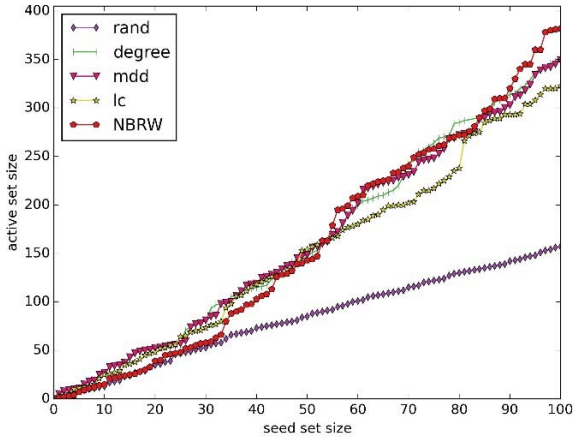


(b) Weighted Cascade Model

Fig. 6. Diffusion results in CondMat

Under the linear threshold model, the final number of the active nodes of NBRW is the largest among all of the algorithms, 10% more than LC, 35% more than Degree, and 70% more than MDD. Under the weighted cascade model, the performance of NBRW is not as good as Degree, but it performs better than other algorithms, and each node we select increases the number of active nodes.

Fig. 6 shows the diffusion scope through different seed node sets under the linear threshold model and the weighted cascade model respectively in CondMat. The same as the first two figure, the x-axis represents the number of seed nodes selected as initial active nodes, denoted as "seed set size". The y-axis stands for the number of active nodes in all at the end of the diffusion process, denoted as "active set size". (a) is the result under the linear threshold model when the threshold is 0.5, (b) is the result under the weighted cascade model.



(a) Linear Threshold Model, $\theta = 0.5$

Under the linear threshold model, the final number of the active nodes of NBRW is the largest among all of the algorithms. Under the weighted cascade model, the performance of NBRW is not as good as other algorithms. This is because we select the nodes whose degree may be not high but which is crucial to influence other inactive nodes, when other algorithms rank nodes taking into account the degree. So, when more crucial nodes is added into the seed node set, with the total influence of active nodes piles up, quantitative cause a qualitative change, the final number of active nodes becomes the largest under the linear threshold model.

From the results of the experiments, it is adequate to demonstrate that NBRW algorithm is effective in dealing with influence maximization problem. Under the linear threshold model, our algorithm performs best in all of the datasets, despite which kind and which scale the datasets is. Moreover, our algorithm does well in the ego networks. As an aside, the different diffusion models is mutual exclusive more or less as it is implied in this work, because of the different influence mechanism.

VI. Discussion And Conclusion

In this paper, we introduce a new technique called Non-backtracking Random Walk into the influence maximization problem, devise an efficient algorithm running in polynomial time by sampling nodes when non-backtracking walking in the network, and estimate the influence of nodes by traversing number. It is shown that the algorithm we propose achieves a largest range of information diffusion under the linear threshold model, and also performs well in an ego network under the weighted cascade model. That demonstrates that the seed nodes we select is very crucial in activating their inactive neighbors. Based on our results, we believe that NBRW provides truly effective measures in solving influence maximization problem with satisfactory influence spread and very fast running time.

In the future, we would like to study how to select start points when traversing in a biased manner instead of randomly and study the theory of non-backtracking random walk with restart to provide theoretical guarantee for influence maximization problem.

ACKNOWLEDGMENT

Supported by the State Key Development Program of Basic Research of China (973) under Grant No. 2013cb329606, the National Natural Science Foundation of China under Grant Nos. 61372191 and 61572492.

REFERENCES

[1] Kempe D, Kleinberg J, Tardos É, "Maximizing the spread of influence through a social network," Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 137-146.

[2] Domingos P, Richardson M, "Mining the network value of customers," Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2001, pp. 57-66.

[3] Richardson M, Domingos P, "Mining knowledge-sharing sites for viral marketing," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 61-70.

[4] Kimura M, Saito K, "Tractable models for information diffusion in social networks," European Conference on Knowledge Discovery in Databases: Pkdd, Springer-Verlag, 2006, pp. 259-271.

[5] Leskovec J, Krause A, Guestrin C, et al. "Cost-effective outbreak detection in networks," Kdd 07 Acm Sigkdd International Conference on Knowledge Discovery & Data, 2007, pp. 420-429.

[6] Goyal A, Lu W, Lakshmanan L V S, "CELF++: optimizing the greedy algorithm for influence maximization in social networks," International World Wide Web Conference, ACM, 2011, pp. 47-48.

[7] Estevez P A, Vera P, Saito K, "Selecting the Most Influential Nodes in Social Networks," International Joint Conference on Neural Networks, IEEE, 2007, pp. 2397-2402.

[8] Chen W, Wang Y, Yang S, "Efficient influence maximization in social networks," Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 199-208.

[9] Chen W, Wang C, Wang Y, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010, pp. 1029-1038.

[10] Wang Y, Cong G, Song G, et al. "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010, pp. 1039-1048.

[11] Goyal A, Lu W, Lakshmanan L V S, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," Data Mining (ICDM), 2011 IEEE 11th International Conference on, IEEE, 2011, pp. 211-220.

[12] Heidari M, Asadpour M, Faili H, "SMG: Fast scalable greedy algorithm for influence maximization in social networks," Physica A: Statistical Mechanics and its Applications, 2015, pp. 124-133.

[13] Jiang Q, Song G, Cong G, et al. "Simulated Annealing Based Influence Maximization in Social Networks," AAAI, 2011, pp. 127-132.

[14] Lü L, Zhang Y C, Yeung C H, et al. "Leaders in Social Networks, the Delicious Case," Plos One, vol. 6, 2011, pp. e21202.

[15] Jung K, Heo W, Chen W, "Irie: Scalable and robust influence maximization in social networks," Data Mining (ICDM), 2012 IEEE 12th International Conference on. IEEE, 2012, pp. 918-923.

[16] Hou B, Yao Y, Liao D, "Identifying all-around nodes for spreading dynamics in complex networks," Physica A Statistical Mechanics & Its Applications, vol. 391, 2012, pp. 4012-4017.

[17] Zeng A, Zhang C J, "Ranking spreaders by decomposing complex networks," Physics Letters A, vol. 377, 2012, pp. 1031-1035.

[18] Liu J G, Ren Z M, Guo Q, "Ranking the spreading influence in complex networks," Physica A Statistical Mechanics & Its Applications, vol. 392, 2013, pp. 4154–4159.

[19] Chen D B, Gao H, Lü L, et al. "Identifying influential nodes in large-scale directed networks: the role of clustering," Plos One, vol. 8, 2013, pp. e77455.

[20] Tang Y, Shi Y, Xiao X, "Influence maximization in near-linear time: a martingale approach," Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM, 2015, pp. 1539-1554.

[21] Goldenberg J, Libai B, Muller E, "Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth," Marketing Letters, vol. 12, 2001, pp. 211-223.

[22] Goldenberg J, Libai B, Muller E, "Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata," Academy of Marketing Science Review, vol. 2001, 2001, pp. 1.

[23] Pearson K, "The Problem of The Random Walk," Nature, vol.72, 1905, pp. 342.

[24] Florent K, Cristopher M, Elchanan M, et al. "Spectral redemption in clustering sparse networks," Proceedings of the National Academy of Sciences of the United States of America, vol. 110, 2013, pp. 20935-20940.

[25] Chen D, Lü L, Shang M S, et al. "Identifying influential nodes in complex networks," Physica a: Statistical mechanics and its applications, vol. 391, 2012, pp. 1777-1787.