Alshahrani *et al. J Big Data (2018) 5:28*
https://doi.org/10.1186/s40537-018-0137-4

Journal of Big Data

RESEARCH Open Access

Selection of top-K influential users based on radius-neighborhood degree, multi-hops distance and selection threshold

Mohammed Alshahrani[1], Fuxi Zhu[1*], Lin Zheng[2], Soufiana Mekouar[3] and Sheng Huang[1]

*Correspondence:
fxzhu@whu.edu.cn
[1] Computer School, Wuhan University, Wuhan, China
Full list of author information is available at the end of the article

Abstract

Influence maximization in the social network becomes increasingly important due to its various benefit and application in diverse areas. In this paper, we propose DERND D-hops that adapt the radius-neighborhood degree to a directed graph which is an improvement of our previous algorithm RND d-hops. Then, we propose UERND D-hops algorithm for the undirected graph which is based on radius-neighborhood degree metric for selection of top-K influential users by improving the selection process of our previous algorithm RND d-hops. We set up in the two algorithms a selection threshold value that depends on structural properties of each graph data and thus improves significantly the selection process of seed set, and use a multi-hops distance to select most influential users with a distinct range of influence. We then, determine a multi-hops distance in which each consecutive seed set should be chosen. Thus, we measure the influence spread of selected seed set performed by our algorithms and existing approaches on two diffusion models. We, therefore, propose an analysis of time complexity of the proposed algorithms and show its worst time complexity. Experimental results on large scale data of our proposed algorithms demonstrate its performance against existing algorithms in term of influence spread within a less time compared with our previous algorithm RND d-hops thanks to a selection threshold value.

Keywords: Influence maximization, Multi-hops distance, Selection threshold, Top-K influential users, DERND D-hops, UERND D-hops

Introduction

The last decade has seen a broad research focus on studying and analysis of social network phenomena especially the influence maximization that received a significant interest and attention due to its various application in various situations such as advertisement, rumor control, spread of epidemic, understanding the collective behavior of users in online systems by observing users behaviors across product and contents. A lot of work has been made followed up the work of Domingos [1] and Kemp et al. [2] that present a greedy algorithm which provide the highest influence coverage while suffer from scalability issue, which pushed a lot of researchers to investigate in the improvement of time complexity of the original greedy algorithm [3–5]. But despite the huge work dedicated to reduce the runtime complexity, the majority if not all based greedy takes a lot of time to complete the selection of seed set. This motivates to develop

methods based heuristics and as stated by Kemp [2] and Chen et al. [6] that degree gives acceptable influence spread in a low time. Selection of most influential users has become a vital task in the field of social network analysis. Identification of such influential users permits summarizing the underlying interactions of the network by observing how such set of selected nodes may influence a lot of users and how it can incite the spread of certain cascade behavior over the network, which helps in profoundly understanding and discovering interesting and favorable properties shared by important users in the network.

Various algorithms have been developed to tackle the problem of influence maximization. Most of the algorithms are based on either heuristics that detect most influential nodes according to the score value of introduced metrics or algorithms [7, 8], or greedy algorithm. However, the major issue with most algorithms based heuristics is that it did not take into account separating seed set from each other and little works have been addressed this problem. By this way, how we can identify efficiently the identification of such seed set in a way the promoted product, will be maximized?

So, in this paper, we try to combine designing an efficient selection method based on degree over radius hops and separating the selected seed by a number of hops. Another challenge that faces methods based heuristics is that those algorithms may perform better on some graph while providing a little performance on other graphs, we dealt with this issue by determining a selection threshold and multi-hops distance value that depend on each graph data in order to keep good influence coverage while maintaining acceptable time complexity. And that the methods based greedy algorithm suffer from high time complexity and as known, social networks have an increasing scale and are sparse in nature; which makes its applicability impractical.

In this paper, we propose two extended algorithms from our previous work [9]. The proposed algorithms come with the improvement of the seed set selection process by designing a selection threshold value for each graph data for the purpose of preventing the choice of the node with low influence that have a low power in other users to adopt the promoted behavior. The selection threshold value is based on structural properties that vary with the graph. As well as, we tried to fix neighborhood hops at radius minus 1, this that when we compute the neighborhood of radius is as computing the neighborhood of entire graph and this is applied for each candidate seed to be selected. This helps to select the node that has an influence on a different range of users. Thus we extended our algorithms to a directed graph and then test the selected seed set on independent cascade and linear threshold models. The aim is to identify the most influential users that are close to touching a large number of users within the real social networks. At a glance, the proposed algorithms identify the most influential users in the network that when measuring the influence of diffusion models showed its performance against most well-known approaches in the literature.

The proposed idea comes from observation of real-world phenomena that users get touched or influenced if they are geographically close to each other and that the selected nodes have a good reputation, the reputation here refers in the directed graph as how many indegree nodes have compared with outdegree (i.e effect of benefit versus effort) and that nodes with the highest score have the highest influence on other users to adopt the information. While the reputation of a node in an undirected graph refers to how

many degrees each node possess, and as know the nodes with the highest degree gain more followers that request the acceptance and that the product promoted by these users have a high probability to be influential and incite others to adopt more and more information. To sum up, our main contributions can be listed as follow:

- The improvement of the selected seed efficiency presented in [9] by introducing a threshold value that avoids the selection of negligible nodes.
- The extension of RND d-hops algorithm [9] to a directed graph that takes into account the indegree and outdegree of nodes as ratio benefit and effort and then ranks nodes according to the introduced score value.
- The limitation of the consecutive seed set choice by radius minus 1 to get better results in term of influence coverage.
- The examination of the two proposed algorithms on Independent cascade model and linear threshold model.
- The proposed algorithms outperform the state of the art algorithms in term of influence coverage especially for the undirected graph on graphs with different densities.

The rest of the paper is organized as follows. In "Related work" section, we provide a comprehensible literature review of work made in the selection of top-K influential users. In "Preliminaries" section, we provide a preliminary and in "System model of selection of top-K influential users" section we present the system model of our problem of selection of top-K influential users and we propose two algorithms, one for directed graph and the second for undirected graph and we explain our contribution compared with our previous work presented in [9]. Thereafter, we provide the complexity analysis of Directed Extended Radius-Neighborhood D-hops DERND D-hops and Undirected Extended Radius-Neighborhood D-hops UERND D-hops algorithms. "Results and discussion" section is devoted to extensive experimental results on the directed and undirected graph on two diffusion models compared with the state of the art algorithms including PageRank, Degree discount heuristic, BCT, TIM+ and others to test the effectiveness of the proposed algorithm in term of influence coverage and running time complexity. In "Conclusion" section we conclude our paper by outlining some limitation and pointing out future research directions.

Related work

In this section, we attempt to provide a comprehensible overview of related work on a selection of top-K influential users, by providing a literature review that tackled the select of seed set based on multi-hops distance and then we exposed works that addressed the problem of influence maximization from neighborhood perspective and other centrality measures. The reason for selection of such works comes from the design of our extension of [9] that uses both multi-hops distance and neighborhood constraints to select top-K influential users. We note that the purpose is to highlight the general ideas of each related paper and the difference between the existing approaches and our proposed approach.

Wang et al. [10] proposed an efficient algorithm for distance-aware influence maximization problem, that takes into account the distance in which a query was promoted.

As consequence, the top-K influential users are different depending on the query promotion locations. For this purpose, they developed two approaches maximum influence arborescence-Distance aware (MIA-DA) that uses the maximum influence arborescence model (MIA) to compute the influence spread and reverse influence sampling-Distance aware (RIS-DA) based on information of pre-sampled query locations. They also tried to improve the RIS-DA approach that returns a good approximation solution for any query. In the same direction as [10], Wang et al. [11] proposed a priority-based algorithm which seeks to find users based on their influence level, by using the distance-aware to select top-K users that are more close to the promotion locations. Thereafter, they calculated the influence coverage based on MIA model. Then, they proposed a novel index that was used as bounds in pruning strategies. They adopted as well various pruning strategies that exclude low ranked nodes from evaluation to accelerate the search of most influential users. In the same direction as [10, 11], Nguyen et al. [12] propose a heuristic algorithm that takes into account propagation probabilities of nodes in the network. Then they estimate the optimal number of hops between neighbors for nodes selections. This work is the closest work to ours in term of considering the optimal number of hops between neighbors to boost the influence spread. Liu et al. [13] propose a neighborhood centrality measure that takes into account node neighbors and neighbors of neighbors that was called in their paper neighborhood centrality. The neighborhood centrality relies on immediate neighbors and 2-hops neighbors and as argued by authors that increasing the neighborhood hops decrease the influence spread. Similarly, Bae et al. [14] propose a new coreness centrality measure using the k-shell indices that were considered more performant than a degree and other centrality measures of its neighbors to compute the influence coverage of a node in a network using the k-shell indices of its neighbors. However, their approach is limited to test on the scale-free network, which limits its applicability and its efficiency on the large-scale real-world network. Likewise, Ruan et al. [15] present an improved coreness measure by decreasing the impact of densely local connections and taking into consideration the effect of the connections between nodes on the nodes' spreading capability.

Zhang et al. [16] proposed a VoteRank algorithm to select a set of decentralized spreaders that did not fall within the same influence range with the highest spread capability. Their approach relies on all nodes voting in a spreader in each step, and the voting capability of neighbors of chosen spreader will be decreased in next step. The susceptible, infected, and resistant (SIR) model is used as diffusion model to calculate the influence coverage.

Zhang et al. [17] proposed two greedy algorithms namely greedy and greedy++ based 2- players coordination game (CG). They incorporate their CG into the general diffusion process that is considered as voter model and linear threshold model. They prove that the objective function of the CG is monotone and submodular and then they accelerate the computation of spread coverage by using two heuristics Lazy forward and Static-Greedy. Their greedy ++ algorithm is faster by three orders of magnitude. But still, need improvement in term of running time. Analysis of equilibria is missing and it would be interesting to study the existence of pure and mixed Nash equilibrium in which users have no incentive to deviate by choosing other strategies (i.e users are all satisfied by the current profit).

L et al. [18] present a review the state of the art of the influence maximization and outline concepts and metrics used to identify vital nodes. Then, they conducted extensive empirical analysis over existing approaches to real-world datasets, and point out future research directions.

Radicchi et al. [19] present a method that investigates the detection of superblockers that seeks to minimize the spread of influence and superspreaders that when selecting the influence coverage is maximized. And then mention that recently is was argued that the identification of superblockers and superspreaders are equivalent. They conduct extensive analysis over a big real-world network in the purpose to identify if there is a similarity between superblockers and of superspreaders. They found that the two problems are not equivalent and that superblockers did not act optimally as superspreaders. Namtirtha et al. [20] conduct a study of the analysis of k-shell most influential node and found that even core node in k-shell decomposition method is not the most influential spreader and that lower core node may be the most influential. As to deal with this issue, they propose an indexing method using nodes k-shell, degree, contact distance and neighbors influence potential to increase the spread of influence under SIR model.

Alshahrani et al. [21] proposed a new algorithm PrKatz for selection of top-K influential users based on Katz centrality and the propagation probability threshold that permits to compute the influence over all the paths and select the one that maximizes the influence. The algorithm PrKatz relies on the use of a combination of Katz centrality and propagation probability threshold tested over each edge for each user in the network. Then top-K influential users are extracted in a decreasing order following the new formulated Katz centrality. Their algorithm outperforms the state of the art algorithms in term of influence coverage.

Alshahrani et al. [9] proposed two novel algorithms namely "RND d-hops" and "CPRND d-hops" considering the results of [6]. These proposed algorithms are for selecting Top-K propagators based on degree centrality heuristic and distance that separate the selection of each seed node from another within the network. This study considers the good performance of degree centrality measure in terms of influence achieved and low runtime complexity. Furthermore, the in-depth investigation has been performed on the usefulness of this metric.

Due to the good performance in term of influence coverage and running time of algorithm "RND d-hops" [9]. We extended the results of RND d-hops algorithm in term of controlling the identification of seed set based on a precomputed selection threshold value depending on the structure of each data and by selecting each consecutive seed set far away r−1 hops to overcome the selection of nodes in the same range of influence coverage. We extended also, our analysis to the directed graph under the Independent cascade model and linear threshold model. As well as, we discussed complexity analysis of each proposed algorithm.

Preliminaries

Independent cascade model (IC)

The independent cascade models start with a seed set S that have been touched by the information in either directed or undirected graph, or the influence spread in a discrete time step. If a node gets infected at time *t*, it will try to infect each inactivates neighbors

at time $t+1$. This procedure continues till no further activation is possible. If a node u has a set of neighbors $x_1, x_2, \ldots, x_k$, so the activation of k neighbors of u are performed independently with probability $p_{(u,x_1)}, p_{(u,x_2)}, \ldots, p_{(u,x_k)}$. We note that in case of the directed graph the infection is performed following edge orientation.

Linear threshold model (LT)

The linear threshold model can be described as stated by Kempe et al. [2], that each edge holds a weight $b_{(i,j)}$ that denote the influence of node i on node j and that each node has a threshold of infection in which the node may accept the information if the sum of weight with neighbors is less than the node threshold of activation the node j. Generally, in the linear threshold model, a node is infected depending on two parameters which are the set of infected neighbors at a given time step and a node threshold that was assigned randomly which remains constant.

System model of selection of top-K influential users

We formulate the problem of influence maximization on a graph G = (V, E) as follows: We seek to select top-K users (i.e seed set) from the graph G, $S \in V$, so as the influence spread will be maximized. This problem is studied in various research papers [9, 21] and has been proven to be categorized in NP-Hard problem. We study in this paper, the influence maximization problem for directed and undirected graph under two cascade models namely Independent cascade model and linear threshold model. Figure 1, shows our system model and how the selection of seed set is performed generally for directed and undirected graphs. The details of selection process of each seed set is given in algorithm 1 and algorithm 2. We assume that each selected seed should be separated from next seed by a certain multi-hops distance $D = r - 1$ and all its immediate neighbors should be excluded. The first supposition which states that the next seed should be separated by a certain multi-hops distance is due that the selection of nodes with

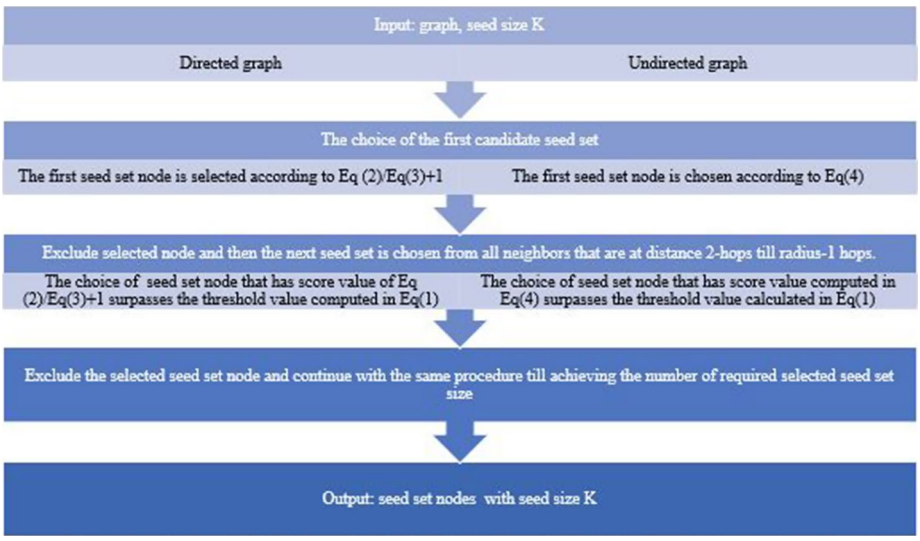

Fig. 1 The procedure of selection of seed set

different coverage of influence will have an influence on different users from previously selected seed set. This can be observed in real-world applications, taking, for instance, a small company that distributed a free sample of a product to users who are separated by certain multi-hops distance will have a higher influence on different users and may influence distinct users from a different background than targeting seed set that are close to each other. The second assumption of excluding all neighbors of selected seed comes from that users always have great influence on immediate neighbors who are more likely to adopt the diffused information, since as the multi-hops distance increase between seed and other nodes, the rate of adoption is more likely to decrease. For this, we excluded all neighbors that may have the same influence on other users as the selected seed of the graph. We assume that each network has selection threshold value defined formally in Eq. (1) that should take into account when selecting seed set. This selection threshold value depends on diameter value of the graph that determines the minimal score for each candidate node should have to be selected as seed. The adding of this selection threshold is motivated by the observation that sometimes the selection of seed set without constraint may results in the choice of low ranked nodes that will decrease the influence. So, setting a certain threshold value would certainly increase the influence as proved by our experiments. The formula of our threshold value th_I is given as follow:

$$th_I = \frac{d * r}{2} - r \tag{1}$$

where r is the graph radius and d represents the graph diameter. Here the diameter and radius represent the node number that a selected node seed should have. So, according to the equation, a node would be selected if at least it is linked half number of radius*diameter minus the initial and farthest nodes from candidate node.

Directed graph algorithm for top-K influential users DERND D-hops

In this subsection, we present an algorithm for top-K influential users selection on a directed graph. The algorithm is based mainly on indegree and outdegree of each node over radius distance. As we carried various experiment, we noticed that the ratio of counting from indegree on outdegree of the actual node to its radius. This permits to value the indegree of a node while discount the effort of the node that incarnates in the linking to other nodes. We know that the node is more important if it has a lot indegree while if the node has as well a lot outdegree, this will accurately identify important node since some nodes link to others as a feedback of following. The ratio will quantify how the node is important depending on the users that link to it and the effort that makes node by following other nodes. Formally, we write the indegree of node v over a radius hops denoted by $Ind_r(v)$ as follows:

$$Ind_r(v) = \sum_{u \in neig_r^I(v)} A(u, v) \tag{2}$$

where A represent the adjacency matrix.

$neig_r^I(v)$: represent all indegree neighbors of node node v from its immediate indegree neighbors till graph radius r.

In the same manner, we express the outdegree of node v over a radius hops denoted by $Outd_r(v)$ as follows:

$$Outd_r(v) = \sum_{u \in neig_r^O(v)} A(v,u) \tag{3}$$

where, $neig_r^O(v)$: represent all outdegree neighbors of node node v from its immediate outdegree neighbors till graph radius r.

Algorithm 1 Top-K influential users Selection DERND D-hops.

Input: Directed graph G, Integer K
Output:set of influential users S of size K

$S \leftarrow \emptyset$
$data\text{-}score_r \leftarrow data(node, rdeg_r(v) = (\frac{Ind_r(v)}{1+Outd_r(v)})$
$size = K$
$Seed \leftarrow node\text{-}order_{Decr}(data\text{-}score_r, rdeg_r(v), 1)$
$S = S \cup Seed$
$sed = S[length(S)]$
$Seedk \leftarrow \emptyset$
$data\text{-}score_r \leftarrow Exclude(sed, data\text{-}score_r)$
$While(length(S) < size)$
$sed = S[length(S)]$
$neigd_s = neighborhood(G, d_s = r-1, sed, "indegree")$
$neigd_I = neighborhood(G, d_I, sed, "indegree")$
$Selected\text{-}neigborhood = setdiff(neigd_s, neigd_I)$
If(length $(Selected\text{-}neigborhood) == 0$)
$sed \leftarrow node\text{-}order_{Decr}(data\text{-}score_r, rdeg_r(v), 1)$
End If
$neigd_s = neighborhood(G, d_s = r-1, sed, "indegree")$
$neigd_I = neighborhood(G, d_I, sed, "indegree")$
$Selected\text{-}neigborhood = setdiff(neigd_s, neigd_I)$
$candidate\text{-}seed \leftarrow match(data\text{-}score_r, Selected\text{-}neigborhood)$
$Seedk \leftarrow getnode.Data\text{-}score_r(max(degree(Data\text{-}score_r(V(G) == candidate\text{-}seed)))))$
If$(get.rdeg_r.data\text{-}score_r(match(V(G) == seedk) > th_I)$
$selected\text{-}seeds \leftarrow seedk$
$selected\text{-}seed \leftarrow selected\text{-}seeds[length(selected\text{-}seeds)]$
$S = S \cup selected\text{-}seed$
$data\text{-}score_r \leftarrow Exclude(selected\text{-}seed, data\text{-}score_r)$
End If
Else
$sed \leftarrow node.order_{Decr}(data\text{-}score_r, rdeg_r(v), 1)$
$data\text{-}score_r \leftarrow Exclude(sed, data\text{-}score_r)$
End Else
End While
Return S

The algorithm DERN D-hops called Directed Extended Radius-Neighborhood D-hops algorithm, is an extension of our work [9] to reduce the time complexity compared to our previous version [9] and improving the influence spread against the state of the art algorithms based heuristics.

In this extended version of our algorithm, we try to filter the selection of influential users based on a selection threshold value that depend on structural properties of each graph and by selecting each two consecutive seed set by a distance D. The distance D is defined by taking all nodes neighbors of selected seed S_d from d_{I+1} hops till D hops. This permits to have a certain quality in term of seed selection, since the algorithm require that a node cannot be selected if it is under a certain selection threshold value and separate consecutive seed by a number of hops to avoid the selection of closest nodes in the same region of influence.

The algorithm starts by initializing the seed set size S to an empty set (Line 1). Then it computes for each vertex in graph G, the corresponding ratio of indegree over outdegree+1, the added one is to avoid the null in the denominator. This ratio is the effort performed by the node versus the benefit received (Line 2). After assigning to each node its corresponding ratio that characterizes node importance in a directed graph, we set the size of the free sample that we are ready to offer for free of charge, this size depends on the available budget and how much we could offer free of charge in a way that we maximize the profit. Next, we select the first seed in the queue by sorting the obtained data in line 1 in a decreasing order, which results in the selection of node with the highest score ratio (Lines 3–4).

Then, the selected seed, that represent node that will be selected to initiate the cascade process, *Seed* is added to seed set S (Line 5) and the last selected seed from seed set is stored in *sed*. Thereafter *Seedk* is initialized with empty set that will contains all nodes that have a certain score value that should surpass the selection threshold value and then exclude selected seed *sed* from graph data $data - score_r$ (Lines 6–8). Thereafter, the algorithm proceeds by checking if the size of S is different from fixed size K and assign to *sed*, that represent the basic parameter that store node that was selected and added to S in order to take it as input to get ego network, the last selected seed set S.

Next, $neigd_s$ that represent all indegree nodes of selected seed *sed* from its immediate indegree neighborhood till farthest nodes by d_s hops. In other words, its like creating an ego network for seed *sed* through its immediate indegree neighborhood till nodes connected to that seed by indegree across d_s hops. The same thing is applied for $neigd_I$. Then *Selected-neighborhood*, that represent all nodes that are candidate to be selected as seed set S, is selected from the last selected seed denoted by taking the difference of neighborhood of seed that is farthest D-hops away minus node neighborhood of seed *sed* that is farthest one hops away based on indegree centrality (Lines 9–13).

In the case of selected seed *sed* has no more than immediate neighbors, the algorithm pick another seed set *sed* from graph data and proceeds as previously to take the difference of neighborhood selected neighborhood that are farthest D hops minus all immediate neighbors and select candidate seed that may be selected as seed set by matching $Selected_neighborhood$ with graph nodes to select node with the highest score ratio (Lines 14–20). Then, the algorithm select as seed set all nodes *SeedK* that has the highest score value and select one selected seed that surpasses the selection threshold value, then exclude the selected seed from graph data (Lines 21–27). If no condition is not successful to select the seed set, select from graph data the seed set *sed* with the highest score value and then exclude it from graph data and finally return the seed set S (Lines 28–33). The algorithm run once for each $K = 10, 20, 30, 40, 50$ from (Lines 3–8) and run till getting K seed set from Lines 9–33. This could be justified that the algorithm at each tenth of seed size K, it needs to select the most central node with highest score value and then proceeds by executing the remaining of procedure based on multi-hops distance, radius-neighborhood degree and selection threshold value.

Top-K influential users selection UERND D-hops algorithm for undirected graph

In this algorithm, we use the neighborhood radius metric from [9], due to its efficiency in term of identifying most important nodes in term of influencing other nodes to adopt

behavior, product and then increase significantly the influence spread within the network. This metric relies mainly on counting the immediate degree of node till the graph radius. The radius neighborhood degree of a node v can be written as introduced in [9] by:

$$deg_r^U(v) = \sum_{u \in neig^r_{r_h=1}(v)} deg(u) \tag{4}$$

The metric start by identifying the neighbors of each node u from $r_h = 1$ hops that represent immediate neighbors of node u and increment at each step by 1 till the graph radius denoted by r and then the degree of each node is computed by counting all immediate neighbors of node till the graph radius. The notation $neig^r_{r_h=1}(v)$, represents neighbors of node v counted from immediate neighborhood $r_h = 1$ till radius of the graph r. The notation $deg(u)$ represent the degree of node u.

Algorithm 2 Top-K influential users selection UERND D-hops algorithm.

Input: Undirected graph G, Integer K
Output: set of influential users S of size K

1: $data\text{-}score_r \leftarrow data(V(g), deg_r^U(v))$
2: $S \leftarrow \emptyset$
3: $Size = K$
4: $Seed \leftarrow node.order_{Decr}(data\text{-}score_r, deg_r^U(v), 1)$
5: $S \leftarrow S \cup Seed$
6: $S_d \leftarrow S[length(S)]$
7: $Seedk \leftarrow \emptyset$
8: $data\text{-}score_r \leftarrow Exclude(s_d, data\text{-}score_r)$
9: $while(length(S) < size)$
10: $S_d \leftarrow S[length(S)]$
11: $neigd_s = neighborhood(G, D, s_d)$
12: $neigd_I = neighborhood(G, d_I, s_d)$
13: $Selected - neigborhood = setdiff(neig_(d_s), neig_(d_I))$
14: $If(length(Selected - neigborhood) = 0)$
15: $s_d \leftarrow node - order - Decr(data_score_r, deg_r^U(v), 1)$
16: $data - (score_r) \leftarrow Exclude(s_d, data - (score_r))$
17: $EndIf$
18: $neigd_s = neighborhood(G, D, s_d)$
19: $neigd_I = neighborhood(G, d_I, s_d)$
20: $Selected\text{-}neigborhood = setdiff(neigd_s, neigd_I)$
21: $For(i = 1 : length(Selected\text{-}neigborhood))$
22: $Seedk \leftarrow getnode.Data(deg_r)(max(degree(data\text{-}score_r(V(G) == Selected\text{-}neigborhood(i)))$
23: End For
24: $If(get.Deg_r^U(v)(match(V(G) == Seedk) > th_I)$
25: $Sed \leftarrow sed[length(sed)]$
26: $S \leftarrow S \cup sed$
27: $data - score_r \leftarrow Exclude(sed, data\text{-}score_r)$
28: End If
29: $Else$
30: $s_d \leftarrow node\text{-}order_{Decr}(data\text{-}score_r, deg_r^U(v), 1)$
31: $data\text{-}score_r \leftarrow Exclude(s_d, data\text{-}score_r)$
32: End Else
33: End While
34: Return S

The main idea of algorithm 2 is straightforward and is the same to some extent as algorithm 1 for a directed graph. So, the procedure of selection of top-K influential users will be the same, in the difference that here we will consider both indegree and outdegree as in undirected graph there is a mutual relationship between users. Here the algorithm employs the radius-neighborhood degree introduced in

[9] and improves the efficiency of the algorithm by determining a selection threshold for each graph data and by controlling the seed selection from 2 to $r-1$ hops from actually selected seed set. This has two objectives, a larger multi-hops distance from actually chosen seed set permits to have a large choice of seed set and setting a selection threshold that permits to avoid the selection negligible nodes that have a marginal influence spread. In the next section, we provide a complexity analysis of two introduced algorithms for the directed and undirected graph to test at which extent will perform when we are dealing with a large scale graph.

It very important and critical to evaluate the performance of algorithm which shows its superiority in term of achieving some goals and benefit in a minimum time. Foremost one of the alternative way is testing algorithms with the same input and observe which algorithm provide the best performance in term of benefit and running time. However, its most likely that some algorithms will perform better than other algorithms for certain input of data. We tried to tackle this problem in the current work to adapt our algorithm on different datasets with different densities and properties over-directed and undirected graph. The performance analysis covers normally the time and space complexity. In this paper we will cover only the analysis in term of time complexity, in another term, we will try to perform an asymptotic analysis and try to find the worst and average case of time complexity of the two proposed algorithm. We will proceeds firstly with analysis of undirected graph algorithm for top-K influential users selection. So, as first analysis we start computing the time complexity required by an algorithm to complete the calculation of seed set S. As depicted in algorithm 1 below uses the radius-neighborhood degree introduced in our paper [9], which relies on computing the neighborhood of each vertex from immediate neighbors to radius hops. So, for each vertex u, it computes $neig_r(u) = \{set.neighborhood_r(u) |$ detect all paths of length r between u and neighbors till radius $set.neighborhood_r(u)$ and then count for vertex the length of $neig_r(u)\}$. And that, for each vertex, the degree is computed from immediate neighbors of candidate vertex to radius graph. So, it requires $(1+2+\cdots+n)r$, where r is the hops numbers between candidate vertex and the graph radius. Next, for each vertex a length of its neighborhood from its immediate to r farthest neighbors, which need nL. Then, the runtime complexity of first line 1 of algorithm 1, can be computed as: $r(1+2+\cdots+n)+nL = r(1+2+\cdots+n)+nL = r(n(n+1))/2+nL = O(rn^2+nL)$ So, the line 1 of our algorithm required in overall a time complexity of $O(n^2r+nL)$, where n is the number of a vertex in graph G and is L is the length of each neighborhood vertex list. The most time complexity comes from line 1 of the algorithm, as it increases with graph size. The while loop has a runtime complexity $klog(K)$, since the two loop is nested and that the inner for loop runs independently of the outer node while number of an iteration loop. Thus, the time complexity of 9–23, is the time required for inner loop for that takes k multiplied by outer loop while that takes $log(K)$. Therefore the time complexity for Lines (9–33) is $O(klogK)$, where k is the length of each results neighborhood of candidate seed and K is the size of seed set required as input in our algorithm. In overall, the time complexity of our algorithm 1(2) is $O(rn^2+nL+klogK)$.

Results and discussion

To examine the effectiveness and efficiency of our proposed algorithms on directed and undirected graphs under two well-known diffusion models namely IC model and LT model, in this section, we report experiments on large scale graph with different size and densities. We evaluate our approach on four datasets two directed and two undirected under IC model and LT model. As well as, we compared our algorithms against the state of the art approaches including simple heuristics to method based approximation algorithms. The dataset used in our experiments and corresponding information are summarized in Table 1. All experiments are performed on a Linux server machine with an Intel Xeon 2.50GHZ 16 cores, and 120G memory.

We compared as well as our algorithms to the state of the art approaches under the IC model and LT model including: simple heuristics including degree, Page Rank and sophisticated heuristic such as degree discount heuristic, BCT, TIM+ and IMM:

- Degree: The degree centrality that selects Top-K propagators with the highest degree centrality. This baseline heuristic was used for comparison purpose in various research work such as [2] and [6].
- Page Rank: It is used by Google search engine. Page Rank proceeds by counting the number and quality of links of a node to all other nodes in such a way to determine how important is the node. Each node depends on the PageRank of all other nodes [22].
- Degree discount heuristic: It was introduced in [6] selects seed set based on the degree centrality score and discount the edge that bond with the next selected seed from the nodes degree computation.

Approximation algorithms

- BCT algorithm: It was proposed in [23] to find the most cost-effective seed users who can influence the most relevant users to the advertisement.
- TIM+ algorithm: It was proposed in [24] to improve the scalability of time complexity while providing low coverage spread of influence.
- IMM algorithm: It was proposed in [25] and provides the same worst-case guarantees as existing approaches but tries to improve its efficiency. This enables IMM to support a larger class of diffusion models than existing algorithms such TIM, TIM+ [24]. IMM compute the influence spread on large graph in less time than TIM and TIM+, since it try to reduce the time complexity by excluding unnecessary computation routine in Reverse-Reachable set. In spite of good results in term of speeding up

Table 1 Datasets information

Datasets	Nodes	Edges	Type	Diameter
Nethept	15,233	58,891	Undirected	14
Netphy	37,154	196,591	Undirected	12
Email-EuAll	265,214	420,045	Directed	14
Munmun-twitter-social	465,017	834,797	Directed	8

the running time, the IMM algorithm suffers from difficulty of estimation of maximum influence and that taking all possible seeds set guarantee only one seed set as output.

To compare our proposed algorithms against existing algorithms related to our proposed approach listed above [6, 22–25], we run all algorithms under the IC model with $p_{u,v} = 0.1$ and LT model with a uniform probability distribution and we set the seed set size K from 10 to 50. The results of our algorithms to the result of discrete influence maximization. For $d_s - d_I = r - 1$, which portrays the multi-hops distance in which each consecutively selected seed separated with next seed by radius minus the graph center and by excluding only immediate selected seed neighbors. The selection threshold value of each graph depends on graph diameter as depicted in formula (1). We performed experiments on directed and undirected graphs under two diffusion models namely IC model and LT model.

Figure 2 and Table 2 report experiments of two undirected graph namely Nethept and Netphy data under the IC model. We can notice clearly through observing experiments that the followed methodology by our algorithm outperforms all existing algorithms in term of influence coverage. This ascertains that our algorithm is performant in term of touching a large number of users under the IC model. While our algorithm still needs an improvement in term of LT model as it requires that a certain amount of neighbors should be affected to permit to infect the current node as shown in Figs. 3, 4.

Tables 3, 4, 5 and 6 depict the running time of our algorithm on undirected graph nethept and netphy under the IC model and LT model. As one can notice that the running time of an algorithm is reduced compared with results of RND d-hops [9] and that despite that our algorithm consume time than existing one but provide a higher influence spread.

Figures 5, 6 shows influence coverage on twitter data under the IC model and LT model respectively. We can notice that our algorithm outperform existing approaches in term of influence spread under the IC while has a low influence spread on LT

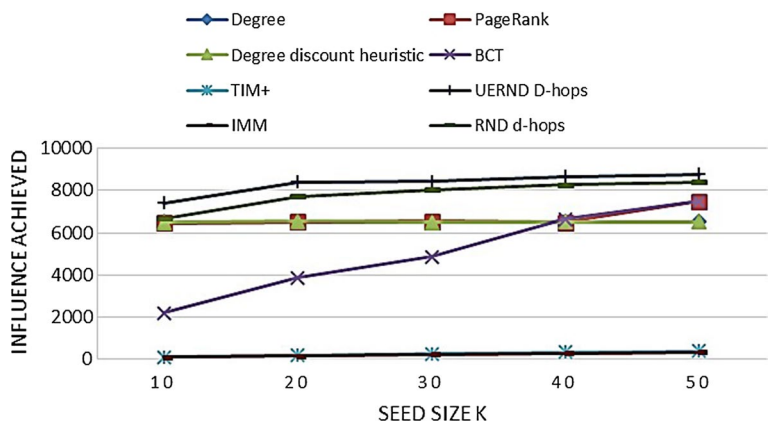

Fig. 2 Influence coverage of our algorithm UERND D-hops compared with state of the art algorithms under the IC model on Nethept data. Degree PageRank Degree discount heuristic BCT TIM+ UERND D-hops IMM RND d-hops

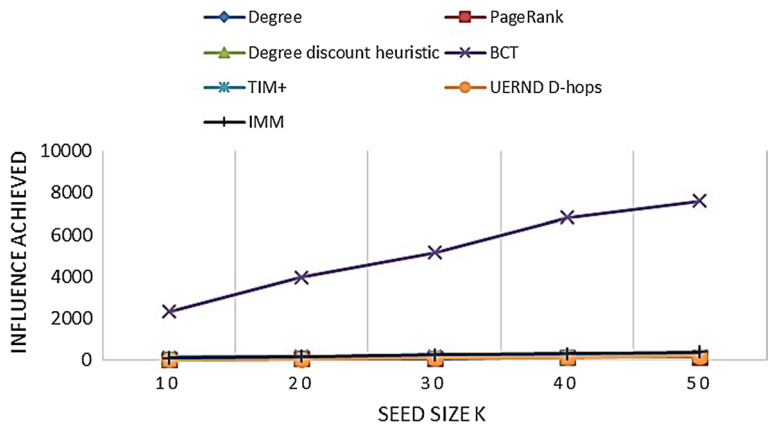

Fig. 3 Influence spread under LT model for Nethept data. Degree PageRank Degree discount heuristic BCT TIM+ UERND D-hops IMM

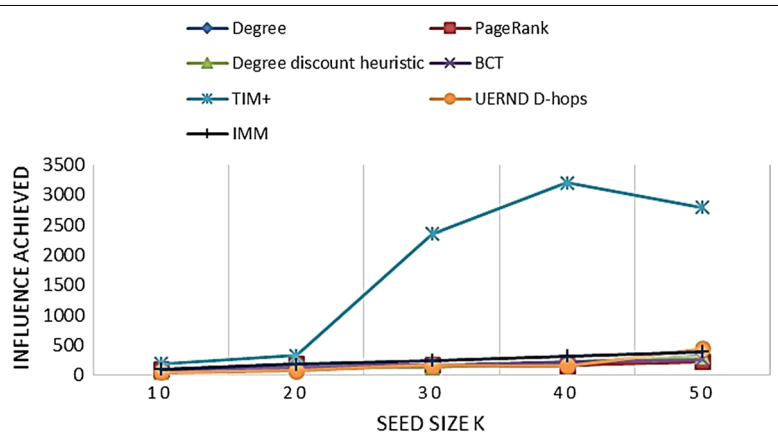

Fig. 4 Influence coverage under LT model on netphy data. Degree PageRank Degree discount heuristic BCT TIM+ UERND D-hops IMM

Table 2 The influence spread of our algorithm under the IC model on Netphy data

Seed size K	RND d-hops	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	UERND D-hops
10	19,658	19,626	19,612	19,603	71	104	91	19,692
20	19,667	19,604	19,538	19,589	122	185	164	19,704
30	19,741	19,583	19,574	19,631	164	260	227	19,828
40	19,768	19,593	19,641	19,752	206	332	289	19,877
50	19,796	19,576	19,549	19,937	249	398	351	20,049

Table 3 The running time of our algorithm under the IC model on Nethept data for K = 50

Algorithm	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	RND d-hops	UERND D-hops
Time (s)	3.17	4.45	4.21	1.4E−2	5.02	1.216	232.15	53.19

Table 4 Runtime of algorithms on Nethept data for K = 50 under the LT model

Algorithm	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	UERND D-hops
Time (s)	0.67	0.59	0.78	8.993E−3	2.03	1.09	31

Table 5 Running time of our algorithm compared with existing work on netphy data for K = 50 under the IC model

Algorithm	RND d-hops	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	UERND D-hops
Time (s)	3881.37	52.21	74.68	51.96	0.67	16.48	2.47	230.82

Table 6 Runtime of algorithms on netphy data for K = 50 under the LT model

Algorithm	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	UERND D-hops
Time (s)	2.67	2.7	4.67	0.60	2.08	2.15	141.15

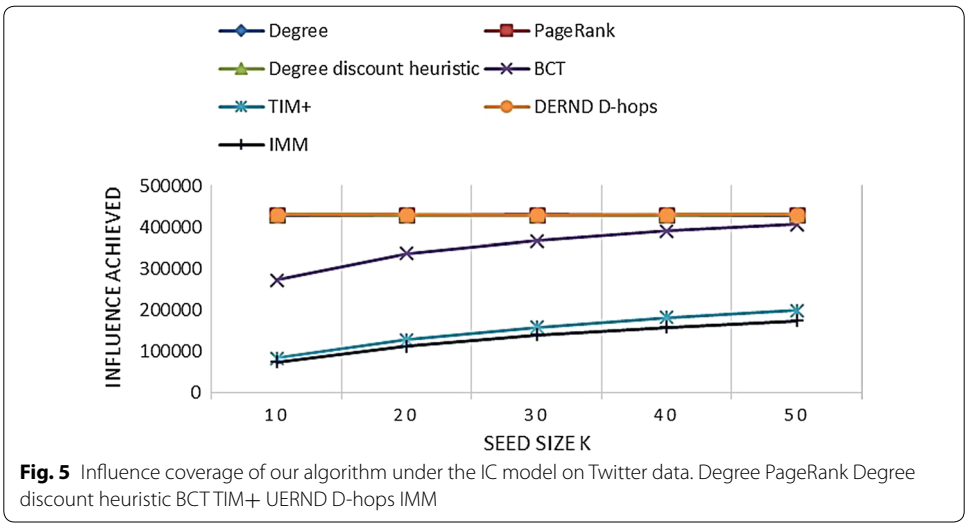

Fig. 5 Influence coverage of our algorithm under the IC model on Twitter data. Degree PageRank Degree discount heuristic BCT TIM+ UERND D-hops IMM

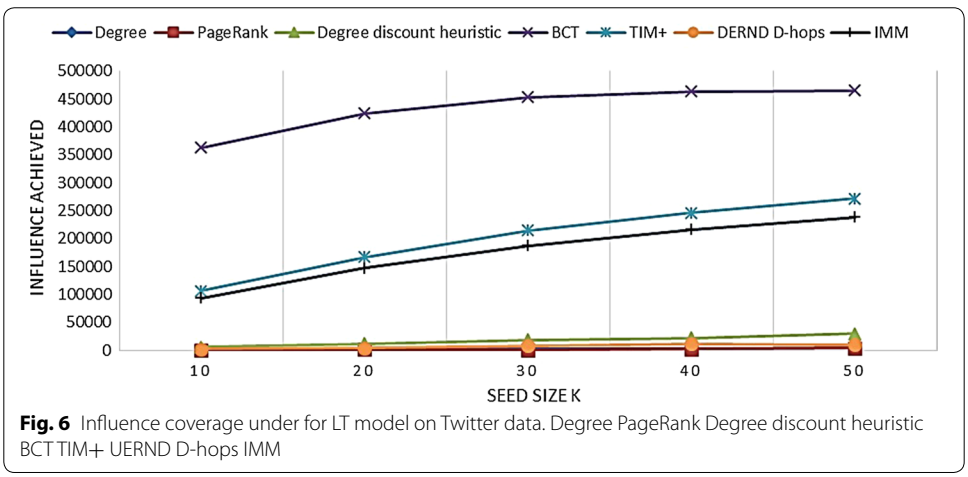

Fig. 6 Influence coverage under for LT model on Twitter data. Degree PageRank Degree discount heuristic BCT TIM+ UERND D-hops IMM

model. This could be justified that our seed selection method suit more the IC model since our approach tries to select users that are not close to each other and that the LT need a fraction of neighbors to be infected to infect the current user. This results are the same as for Eu-Email data as depicted in Figs. 7, 8. So, according to experiments results our approach gives good performance under the IC model and is not good for LT model.

Tables 7, 8, 9 and 10 represent the running time of our approaches on the directed graph and LT model and IC. We can observe clearly that the LT model for our approach is faster than other approaches such as degree discount heuristics and that even the running time under IC for our algorithm still acceptable to be applied on large scale graph.

To sum up, according to experiments on large scale graph with different topology and size, we can notice that our approach perform very well on undirected graph under the IC model, which is illustrated in Fig. 1 and Table 2. This can be justified by score value of neighborhood computed from active node till its radius, this helps to identify the node with most direct and indirect relationship. This permits to quantify in an efficient manner the potential users to be targeted and that adding the constraint of a selection threshold value for each node score favorize node with a certain efficiency and that separate each consecutively selected node by a certain number of hops enables targeting nodes that have an influence on users from a distinct region of influence. This computation of neighborhood number of score value for each node makes our approach a little bit time consuming but still perform well on large scale graph and therefore adequate for the real-world application. And as another observation for results of our approach under the LT model for undirected and directed setting, the one can observe that our approach gives lower results in term of influence

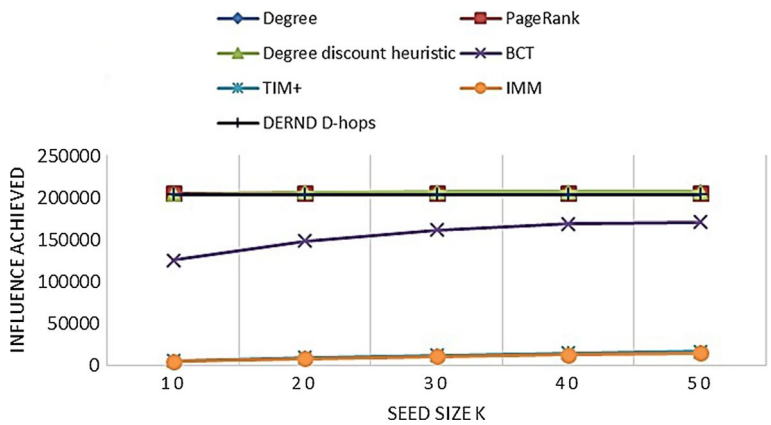

Fig. 7 Influence spread on Eu-Email data under the IC model. Degree PageRank Degree discount heuristic BCT TIM+ UERND D-hops IMM

Table 7 Runtime of algorithms on Twitter data under for K = 50 under the IC model

Algorithm	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	DERND D-hops
Time (s)	15,891.53	17,985.24	11,470.08	3.3E−2	2.63	0.41	44,980.57

Table 8 Runtime of algorithms on Twitter data for K = 50 under the LT model

Algorithm	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	DERND D-hops
Time (s)	21.57	21.04	54.41	3.40E−2	2.06	0.27	38,144.08

Table 9 Runtime of algorithms on Eu-Email data for K = 50 under the IC model

Algorithm	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	DERND D-hops
Time (s)	2383.71	1743.48	4197.73	0.4	13.7	1.75	19,127.62

Table 10 Runtime of algorithms on Eu-Email data for K = 50 under the LT model

Algorithm	Degree	PageRank	Degree discount heuristic	BCT	TIM+	IMM	DERND D-hops
Time (s)	280.39	251.29	11.19	2.4E−2	5.15	0.25	16,051.73

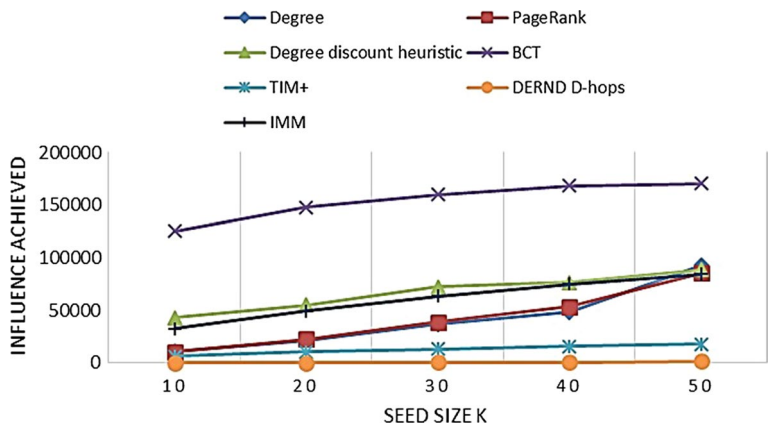

Fig. 8 Influence spread under the LT model of Eu-Email data. Degree PageRank Degree discount heuristic BCT TIM+ UERND D-hops IMM

spread and as discussed before that separating the selection of seed set by a certain hops reduces the influence of LT model, since this model needs a number of users to be activated in order that it can activate new nodes in its immediate neighborhood, this could be improved further by controlling the multi-hops distance in which we separate the selected nodes. However, the approach still doing well in term of time complexity and improved significantly compared to our previous approaches [9, 21].

In directed setting, we can notice clearly from our experiment on two large-scale data, that it performs well in term of influence coverage and surpass well-known algorithms from state of the art literature and that this is thanks to our followed methodology by providing reward of each gained indegree node and discount each outdegree node +1, to prevent the zero in the denominator. This methodology is very efficient in a way that sometimes users gain followers only as a feedback of following back, so this measure will measure how much effort made by each node versus how much followers it receives and that despite sometimes proposed score may give a little bit lower results in term of

influence but it will perform better on real-world situation. Since as known always people follow known people who have a lot of connections and that known people rarely follow back modest people(i.e nodes that have few connections), so this metric will certainly work very well in real-world application. Consequently, as our approach computes for each node the corresponding indegree and outdegree over radius hops, this will be more time consuming but as stated before, the results will be accurate and near to real-world situations.

Conclusion

In this paper, we proposed new algorithms for maximizing the influence spread by the selection of top-K influential users within the network. Specifically, we presented two efficient algorithms for directed and undirected graph namely "DERND D-hops", "UERND D-hops" respectively under the IC model and LT model. The proposed algorithms try to improve the results of the undirected graph of our previous algorithm RND d-hops [9] and extend this algorithm to the directed graph. As observed through experiments and as argued by previous works that methods based heuristics may perform better on some graph while not be good on other graphs. For this purpose we dealt with this issue by introducing structural characteristics for each graph data, through a selection threshold value that permits to improve the selection of seed set in both directed and undirected graph and by using a predefined multi-hops distance for the selection of consecutive seed set depending on graph radius. This permitted the selection in a moderately large region to pick the most suitable nodes as the seed set. We computed the worst case time complexity and we demonstrate that our proposed algorithms is better than the state of the art approach especially for undirected graph in term of influence coverage and we reduced the time complexity compared with [9]. As a future extension, an investigation and more in-depth study should be performed regarding the multi-hops distance that separates all seed set in each data and not relies only on consecutive seed set. As well as, the IC model and LT model should be generalized to a more accurate and performant model that take into account not just propagation probability threshold but as well the type of shared content and that in real-world scenario companies may promote different products and that the extent of adoption of such product requires the study of users behavior and which product is more likely to be interested by a specific user. So, such study is very interesting besides including the structural properties of each data to boost the influence spread in a real-world scenario.

Abbreviations

RND d-hops: Radius-neighborhood degree over d-hops; DERN D-hops: Directed Extended Radius-Neighborhood D-hops; UERND D-hops: Undirected Extended Radius-Neighborhood D-hops; MIA: maximum influence arborescence; RIS: reverse influence sampling; DA: Distance aware; SIR: susceptible, infected, and resistant model; CG: coordination game; IC: independent cascade model; LT: linear threshold model.

Authors' contributions

MA has written algorithm 1 and algorithm 2. FZ has conducted the analysis part of the complexity and revised the paper. LZ have conducted an analytical and critical analysis of the paper and written the introduction. SM have made experimental results. SH has written related work and conclusion. All authors read and approved the final manuscript.

Author details

[1] Computer School, Wuhan University, Wuhan, China. [2] Department of Computer Science, Hong Kong Baptist University, Kowloon, Hong Kong, China. [3] Faculty of sciences, Mohammed V University, Rabat, Morocco.

Acknowledgements
This research is supported by two grants from the National Natural Science Foundation of China with Project Nos. 61272277 and 91746206.
Authors thank Alibaba Cloud Co., Ltd for the technical support.

Competing interests
The authors declare that they have no competing interests.

Availability of data and materials
All datasets used are open source and available online. Nethept and Netphy datasets used are available and can be accessed directly at http://research.microsoft.com/enus/people/weic/graphdata.zip. Email-EuAll can be downloaded from https://snap.stanford.edu/data/email-EuAll.html and munmun-twitter-social can be accessed and downloaded from http://konect.uni-koblenz.de/networks/munmun_twitter_social.

Consent for publication
Not applicable.

Ethics approval and consent to participate
Not applicable.

Funding
This research is supported by two grants from the National Natural Science Foundation of China with Project Nos. 61272277 and 91746206.

Publisher's Note

Received: 9 May 2018 Accepted: 14 August 2018
Published online: 27 August 2018

References

1. Richardson M, Domingos P. Mining knowledge-sharing sites for viral marketing. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. New York: ACM; 2002. p. 61–70.
2. Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining. New York: ACM; 2003. p. 137–146.
3. Lv J, Guo J, Ren H. Efficient greedy algorithms for influence maximization in social networks. JIPS. 2014;10(3):471–82.
4. Zhou C, Zhang P, Zang W, Guo L. On the upper bounds of spread for greedy algorithms in social network influence maximization. IEEE Trans Knowl Data Eng. 2015;27(10):2770–83.
5. Mirzasoleiman B, Badanidiyuru A, Karbasi A, Vondrák J, Krause A. Lazier than lazy greedy. In: AAAI. Austin: AAAI Press; 2015. p. 1812–1818.
6. Chen W, Wang Y, Yang S. Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. New York: ACM; 2009. p. 199–208.
7. Rao A, Spasojevic N, Li Z, DSouza T. Iout score: measuring influence across multiple social networks. In: 2015 IEEE international conference on Big Data (Big Data). Piscataway: IEEE; 2015. p. 2282–2289.
8. Wang Y, Vasilakos AV, Jin Q, Ma J. PPRank: economically selecting initial users for influence maximization in social networks. IEEE Syst J. 2015;11(4):2279–90.
9. Alshahrani M, Zhu F, Bamiah M, Mekouar S, Huang S. Efficient methods to select top-k propagators based on distance and radius neighbor. In: To appear in the proceedings of international conference on Big Data and computing. 28–30 April 2018; Shenzhen, China. New York: ACM; 2018. p. 78–85.
10. Wang X, Zhang Y, Zhang W, Lin X. Efficient distance-aware influence maximization in geo-social networks. IEEE Trans Knowl Data Eng. 2017;29(3):599–612.
11. Wang X, Zhang Y, Zhang W, Lin X. Distance-aware influence maximization in geo-social network. In: ICDE. Piscataway: IEEE; 2016. p. 1–12.
12. Nguyen D-L, Nguyen T-H, Do T-H, Yoo M. Probability-based multi-hop diffusion method for influence maximization in social networks. Wirel Pers Commun. 2017;93(4):903–16.
13. Liu Y, Tang M, Zhou T, Do Y. Identify influential spreaders in complex networks, the role of neighborhood. Phys A Stat Mech Appl. 2016;452:289–98.
14. Bae J, Kim S. Identifying and ranking influential spreaders in complex networks by neighborhood coreness. Phys A Stat Mech Appl. 2014;395:549–59.
15. Ruan Y-R, Lao S-Y, Xiao Y-D, Wang J-D, Bai L. Identifying influence of nodes in complex networks with coreness centrality: decreasing the impact of densely local connection. Chin Phys Lett. 2016;33(2):028901.
16. Zhang J-X, Chen D-B, Dong Q, Zhao Z-D. Identifying a set of influential spreaders in complex networks. Sci Rep. 2016;6:27823.
17. Zhang Y, Zhang Y. Top-k influential nodes in social networks: a game perspective. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. New York: ACM; p. 1029–1032.

18. Lü L, Chen D, Ren X-L, Zhang Q-M, Zhang Y-C, Zhou T. Vital nodes identification in complex networks. Phys Rep. 2016;650:1–63.
19. Radicchi F, Castellano C. Fundamental difference between superblockers and superspreaders in networks. Phys Rev E. 2017;95(1):012318.
20. Namtirtha A, Dutta A, Dutta B. Identifying influential spreaders in complex networks based on kshell hybrid method. Phys A Stat Mech Appl. 2018;499:310–24.
21. Alshahrani M, Zhu F, Sameh A, Mekouar S, Huang S. Top-k influential users selection based on combined katz centrality and propagation probability. In: To appear in 3rd IEEE international conference on cloud computing and Big Data Analysis. 20–22 April 2018; Chengdu, China. Piscataway: IEEE. 2018.
22. Wills RS. Google's pagerank. Math Intell. 2006;28(4):6–11.
23. Nguyen HT, Dinh TN, Thai MT. Cost-aware targeted viral marketing in billion-scale networks. In: INFOCOM 2016-the 35th annual IEEE international conference on computer communications. Piscataway: IEEE; 2016. p. 1–9.
24. Tang Y, Xiao X, Shi Y. Influence maximization: near-optimal time complexity meets practical efficiency. In: Proceedings of the 2014 ACM SIGMOD international conference on management of data. New York: ACM; 2014. p. 75–86.
25. Tang Y, Shi Y, Xiao X. Influence maximization in near-linear time: a martingale approach. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data. New York: ACM; 2015. p. 1539–1554.