Information Sciences 316 (2015) 107–122

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Deprecation based greedy strategy for target set selection in large scale social networks

Suman Kundu *, Sankar K. Pal

Center for Soft Computing Research, Indian Statistical Institute, 203 B.T. Road, Kolkata 700108, India

ARTICLE INFO

Article history:
Received 5 July 2014
Received in revised form 2 April 2015
Accepted 11 April 2015
Available online 17 April 2015

Keywords:
Top-*k* nodes selection
Social network
Greedy deprecation strategy
Influence maximization
Big data
Target set selection

ABSTRACT

The problem of target set selection for large scale social networks is addressed in the paper. We describe a novel deprecation based greedy strategy to be applied over a pre-ordered (as obtained with any heuristic influence function) set of nodes. The proposed algorithm runs in iteration and has two stages, (i) Estimation: where the performance of each node is evaluated and (ii) Marking: where the nodes to be deprecated in later iterations are marked. We have theoretically proved that for any monotonic and sub-modular influence function, the algorithm correctly identifies the nodes to be deprecated. For any finite set of input nodes it is shown that the algorithm can meet the ending criteria. The worst case performance of the algorithm, both in terms of time and performance, is also analyzed. Experimental results on seven un-weighted as well as weighted social networks show that the proposed strategy improves the ranking of the input seeds in terms of the total number of nodes influenced.

1. Introduction

Information diffusion over social networks in the form of "word-of-mouth" is studied in different fields of research including epidemiology [9,34], sociology [2,39] and economics [15,16]. More recently, scholars of computer science got interested in the field due to the emergence of online social networks, like, Twitter, Facebook and YouTube, and their extreme popularity. Different research issues have been addressed in this direction [11,18,20,21,38,41]. One of the important problems within the area of said research is *target set selection*.

A variant of the problem of target set selection is to select *k*-top influential nodes such that they maximize the influence on the network. There are other variants in the literature such as those in [3,29] which we will not cover in this study. Solutions of the target set selection problem have endless applications. For example, they are useful in viral marketing through online social networks [10,27], in identifying top stories in news network, in finding the highest influencing blogs in the blogger network [26], in providing personalized recommendation [17,40], in determining the impact of an article from the scientists' citation network, and in spreading social awareness through social media.

Diffusion of information, in a nutshell, is the process by which an innovation or idea is spread over the networks by means of communication among the social entities [36]. It is the newness of the information that drives the cascade over the networks. One of the simplest models of the diffusion process available for the computer science researchers is *independent cascade model* [15,20]. The model runs in discrete steps. In each step, an active or influenced node tries to activate one of its

* Corresponding author.
E-mail addresses: suman@sumankundu.info (S. Kundu), sankar@isical.ac.in (S.K. Pal).

http://dx.doi.org/10.1016/j.ins.2015.04.024
0020-0255/

inactive neighbors with a probability p, called propagation probability or diffusion speed. Irrespective of its success, the same node will never get a chance to activate the same neighbor. The process, however, is highly stochastic, and Kempe et al. [20] showed that the optimization problem is NP hard. They also provided a Greedy Hill Climbing algorithm, which gives $(1-\frac{1}{e}-\epsilon)$ approximation to the optimal solution. However, the algorithm is time consuming, especially for large scale networks. For example, it takes days to compute on a network of size 30 K nodes [6]. Various improvements of the greedy algorithm in terms of computation time are described in [3,12,24]. On the other hand, several heuristic algorithms [3,4,6] are developed which run faster, but they provide sub-optimal results.

This paper addresses the aforesaid problem within the context of information diffusion on large scale social networks. We describe a *deprecation based greedy strategy* (DGS) for target set selection and apply it over a list of nodes which are pre-ordered based on some fast heuristic influence score. We theoretically prove that the method correctly identifies the nodes to be deprecated as well as provides a guaranteed solution to the target set selection problem when the influence function is monotonic and sub-modular. The convergence of the proposed algorithm is proved analytically. It is shown experimentally, with seven real life large scale social network data sets (both weighted and un-weighted) that applying DGS over a heuristic algorithm produces better solution for the target set selection problem.

The paper is organized as follows: Section 2 describes the preliminary concepts of networks related to this study. Problem statement and related investigations are briefly explained in Section 3. Section 4 illustrates the Deprecation based Greedy Strategy (DGS), and its proof of correctness, convergence and optimization guarantee. Experiments and results are reported in Section 5.

2. Preliminaries

We describe in this section some notations and definitions related to social networks.

2.1. Social networks

A social network represents a social structure made up of individuals or organizations and their relations (e.g., friendship, co-authorship of scientific papers, co-appearance in a movie, and following-followers). Social networks are described using a graph $G(V,E)$ where V is the set of nodes representing the individuals or organizations and E is the set of edges representing the social ties.

2.2. Information diffusion

Information diffusion process and the effect of "word-of-mouth" in social networks is well studied in sociology [36]. During the diffusion process there exist two sets of nodes, namely, active and inactive nodes. The active nodes are those who have already adapted the behavior, i.e., have the information, while the inactive nodes are those who do not have. One of the fundamental processes of information diffusion available in the literature is cascade model. Goldenberg et al. [15,16] inspected cascade model in the marketing perspective. In this model, a node u is influenced by its neighbor v with a probability $\lambda_{u,v}$, called propagation probability.

2.2.1. Independent cascade (IC) model

The IC model of [15] is the simplest form of the cascade model of diffusion and runs in discrete time. Initially, a few nodes are activated. At each successive step, an active node tries to activate one of its inactive neighbors. The node, however, gets only one chance to activate that particular node irrespective of its success. The process terminates when no further activation is possible. Edge $e(u,v) \in E$ is assigned a non-negative propagation probability $\lambda_{u,v}$ which indicates the probability at which node u is activated by v.

2.3. Centrality

In a social network, centrality of a node provides a measure of its relative importance in the network. This is considered to be an important structural attribute [14] of the network. Two ways of measuring it are as follows:

- *Degree centrality:* Degree centrality of a node v is defined in terms of the numbers of incident edges as [30],

$$C_D(v) = \sum_{i=1}^{n} e(u_i, v) \tag{1}$$

where

$$e(u_i, v) = 1, \text{ if the nodes } u_i \text{ and } v \text{ are connected, i.e., an edge exists between them, and} = 0, \text{ otherwise.}$$

- *Diffusion degree centrality:* The diffusion degree of a node v is defined as [33],

$$C_{DD}(v) = \sum_{u \in \Gamma(v)} \left\{ \lambda_{u,v} + \left(\lambda_{u,v} \times \sum_{i \in \Gamma(u)} \lambda_{i,u} \right) \right\} \tag{2}$$

where $\Gamma(v)$ denotes the neighbor set of v, and $\lambda_{u,v}$ denotes the propagation probability by which node v influences node u. One may note that Eq. (1) is based on the structural property of the network, whereas, Eq. (2) incorporates the property of information diffusion along with the structural information. Thus, it is closely coupled with the independent cascade model of diffusion.

3. Problem statement and related work

Consider an influence function $\sigma : 2^V \rightarrow \mathbb{N}$ for a social network $G(V, E)$. Given a set of initial active nodes $S \in 2^V$, $\sigma(S)$ returns the expected number of active nodes at the end of the information diffusion. In top-k nodes selection problem, we are interested to find the k number of influential nodes, which in turn produce the maximum influence in the network after diffusion of the information. So, this is a maximization problem defined as follows:

$$\begin{aligned} &\underset{S}{\text{maximize}} && \sigma(S) \\ &\text{subject to } |S| = && k, k > 0. \end{aligned}$$

Let us now describe some of the related studies conducted so far. Domingos and Richardson are the pioneers in examine the algorithmic aspect of the problem and provided a probabilistic model based on Markov random field [11,35]. Later, Kempe et al. [20] and Kleinberg [22] studied it as a discrete optimization problem and showed that the optimization problem is NP hard. They provided a general greedy hill climbing algorithm (shown in Algorithm 1) to approximate the initial active set, and proved that the influence function $\sigma(\cdot)$ is sub-modular and monotonic in nature for independent cascade model of information diffusion. Furthermore, they derived the provable approximation guarantees for the algorithm. However, their greedy algorithm is time consuming, especially for large scale social networks. Leskovec et al. [24] investigated the top-k nodes problem focusing mainly on (i) contaminant detection for water distribution network and (ii) finding important stories in a blog network. They proposed "Cost Effective Lazy Forward (CELF)" for optimization which is as fast as 700 times than the greedy algorithm [20]. On the other hand, Chen et al. in [5] designed a new heuristic algorithm, namely, the degree discounted algorithm which provides much better results than the classical degree and centrality based heuristic algorithms.

Algorithm 1. General Greedy Algorithm

input : A Social Network $G(V, E)$ and k
output : Set $S \in 2^V$ having cardinality k

initialization: $S := \emptyset$
while $|S| \neq k$ **do**
  $v^* \leftarrow \underset{v \in V \setminus S}{\arg\max}\ \hat{\sigma}(S \cup \{v\})$;
  $S \leftarrow S \cup \{v^*\}$;
end

Even-Dar and Shapira [13] studied the problem in the context of a probabilistic voter model [7,19]. It was shown as a special case that the high degree heuristic, which appears to be the most natural solution of the problem, provides the optimal solution. There are other approaches to deal with the top-k nodes problem using such as co-operative game [29], tree properties [1] and set based coding algorithm [20].

4. Deprecation based Greedy Strategy (DGS)

In this section, we describe a new deprecation based greedy strategy, algorithm and the proof of its correctness. We also address the convergence of the algorithm and its approximation guarantees.

4.1. Strategy

In the greedy algorithm, mentioned in Section 3, in each iteration, the marginal contribution towards information diffusion is evaluated for every node with respect to the seeds, already selected. The node with maximum marginal contribution is added to the set of existing seeds. Thus, for selecting k nodes, the marginal contributions of all the nodes towards information diffusion are evaluated k times. Since the information diffusion process is stochastic, the influence is usually estimated by simulation over sufficiently large (about 10,000 times) number of executions. This, in turn, requires a huge computation time.

Centrality based heuristic algorithms, on the other hand, chooses k number of nodes by their centrality scores, and the algorithms are fast. The assumption here is that, a correlation exists between the centrality scores and influence. However, it may not be always true. The correlation depends upon the network, relations and the information diffusion process. During the execution of the algorithm, it may happen that a node is classified as 'higher influencing' based on the centrality, but it actually has lower influence in the network.

Thus it appears that a judicial integration of the merits of greedy and heuristic algorithms may lead to a system which is efficient in terms of both computational time and performance. Accordingly, we have developed a deprecation strategy, stated as follows:

Unlike the greedy approach where a node is added when its marginal contribution is maximum, in deprecation based greedy strategy, we remove the least performing node. To speed up this process we deprecate the multiple lower influencing nodes at a time. For this, we take the advantage of the centrality measures and pre-order the nodes accordingly. The strategy of identifying a node to be deprecated is based on the realization that if there exist k number of successors having higher influence, then the concerned node is designated as wrongly positioned, i.e., a position higher than its actual. We mark it as a candidate for removal, i.e., deprecate it. Similarly, we identify all those nodes whose marginal influence is lower than those of at least k of its successors, and mark all of them as deprecated in a single iteration. Subsequent iterations are executed only over the resulting reduced list (ordered set of nodes).

4.2. Algorithm

For a given social network $G(V,E)$ let us define an order $\succcurlyeq$ over V such that,

$$u \succcurlyeq v \iff \tilde{\sigma}(\{u\}) \geqslant \tilde{\sigma}(\{v\}).$$

Here, the function $\tilde{\sigma} : 2^V \to \mathbb{N}$ is a heuristic estimation of the influence function $\sigma(S)$. As stated in Section 3, an influence function $\sigma(\cdot)$ returns the number of nodes influenced by the input set of seeds. In our experiment, we calculated $\tilde{\sigma}(\cdot)$ using different well known heuristic algorithms (e.g., high degree heuristic, diffusion degree heuristic and degree discount heuristic).

Now let us define an ordered m-tuple on V^m as $\tau = (v_1, v_2, \ldots, v_m)$ where $v_i \succcurlyeq v_{i+1}, \forall i = 1, 2, \ldots, m-1; m \leqslant n$, and there is no repetition in τ, i.e.,

$$v_i = v_j \Rightarrow i = j. \tag{3}$$

The influence function $\sigma(\cdot)$ for IC model of information diffusion is *sub-modular* and *monotonic* in nature [20–22]. The monotonicity of the function $\sigma : 2^V \to \mathbb{N}$ conforms to the fact that when the number of active nodes is higher, influence is also higher i.e.,

$$\sigma(S \cup \{v\}) \geqslant \sigma(S) \quad \forall v \in V \quad \text{and} \quad \forall S \in 2^V. \tag{4}$$

The sub-modularity condition corresponds to the fact that

$$\begin{aligned} &\sigma(S \cup \{v\}) - \sigma(S) \geqslant \sigma(T \cup \{v\}) - \sigma(T) \\ &\forall S \subseteq T \subseteq V \quad \text{and} \quad \forall v \in V; S, T \in 2^V \end{aligned} \tag{5}$$

i.e., the influencing effect of a node on the network decreases as more and more nodes get activated.

The greedy deprecation strategy runs iteratively and has two stages, namely (1) Estimation and (2) Marking. Input for the algorithm consists of the tuple τ of size m, as described above, and k where $n \geqslant m > k$. Here, τ represents the sequence of top m nodes based on the order $\succcurlyeq$.

1. *Estimation:* In this stage, we first select a node v_i from the given sequence τ, and include it as a seed in S with the same sequence as it appeared in τ. Then we calculate

   $$\tau_{\sigma(\cdot)} = (\hat{\sigma}(S_1), \hat{\sigma}(S_2), \ldots, \hat{\sigma}(S_m))$$

   where

   $$S_i = \{v_1, v_2, \ldots, v_i\}.$$

   $\hat{\sigma}(S)$ is the estimated value of influence caused by the seed set S. This is usually obtained through simulation. The marginal contribution of node v_i to the cascade, when all $v_j, \forall j < i$ are already in the seed set, is computed as

   $$\hat{\Sigma}_i = \hat{\sigma}(S_i) - \hat{\sigma}(S_{i-1}).$$

2. *Marking:* The output of the aforesaid estimation stage, $\tau_{\sigma(\cdot)}$, is considered as the input of the marking stage. Here, we mark one or more nodes for deprecation based on their marginal contribution $\hat{\Sigma}_i$ as follows.
   Let us define a relation R on V as

$$R = \{(v_i, v_j) : v_i, v_j \in V; \hat{\Sigma}_i < \hat{\Sigma}_j; j > i\}.$$

The relation R can be interpreted as a mapping which maps a node v_i to its better performing (in terms of marginal contribution to the cascade process) successors $v_j,\ j > i$. R is anti-reflexive, anti-symmetric and transitive. These properties have been proved in Section 4.3.

Let a function $g : V \rightarrow 2^V$ be defined as

$$g(v_i) = \{v_j : (v_i, v_j) \in R\}. \tag{6}$$

$g(v_i)$ denotes the set of all the successors of node v_i which perform better than itself. Therefore, the set of nodes to be deprecated in order to select the top-k nodes is

$$D^* = \{v_i : |g(v_i)| \geqslant k\}.$$

The subsequent iterations are performed on the reduced tuple, as obtained by removing $|D^*|$ components from the original tuple τ. The iteration stops when $|D^*| = 0$, and the first k components of the reduced tuple τ are returned as the resulting target set (seeds). The block diagram of the algorithm is provided in Fig. 1.

4.3. Properties of relation R

1. *Anti-Reflexive:* R is anti-reflexive, i.e., $(v_i, v_i) \notin R$.

Proof. As per the interpretation, $(v_i, v_j) \in R$ when node v_j is a successor of node v_i and $\hat{\Sigma}_i < \hat{\Sigma}_j$. A node v_i can not be its own successor (Eq. (3)). Hence, $(v_i, v_i) \notin R$. □

2. *Anti-Symmetric:* R is anti-symmetric, i.e., if $(v_i, v_j) \in R$ then $(v_j, v_i) \notin R$.

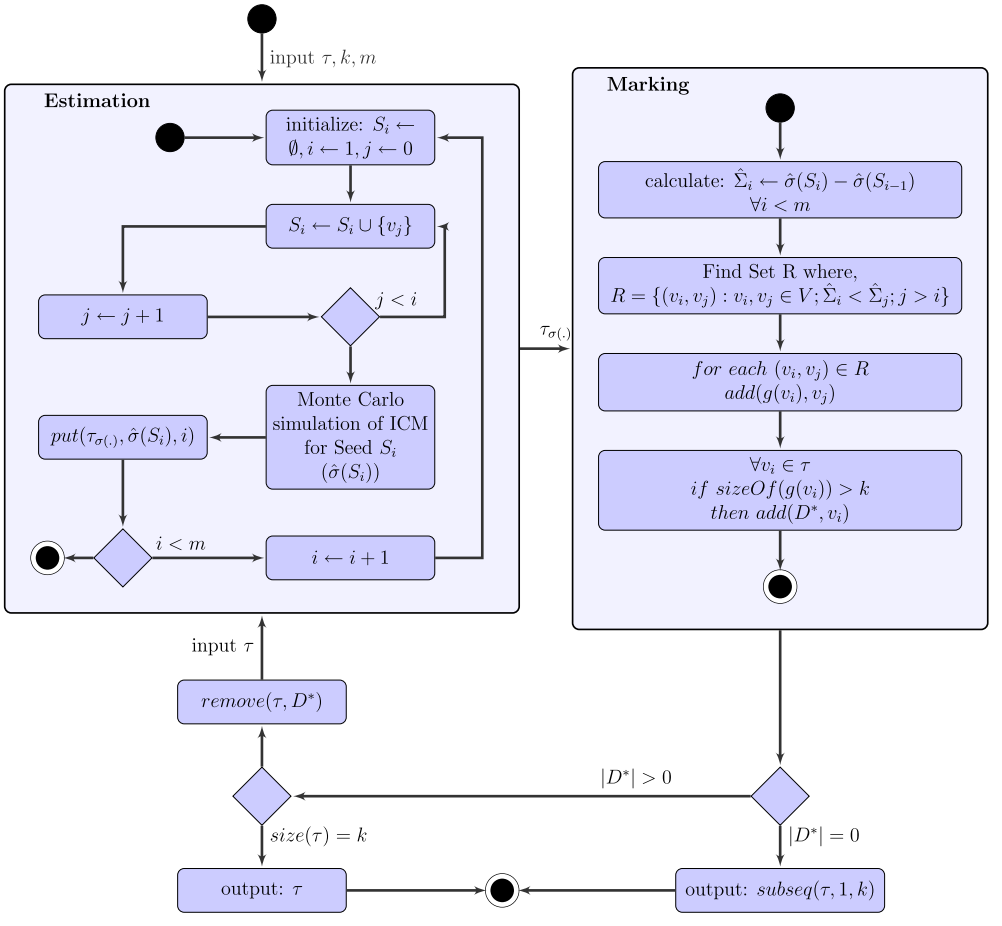


Fig. 1. Block diagram of DGS algorithm.

Proof. Let us consider that $(v_i, v_j), (v_j, v_i) \in R$. This implies, v_j is a successor to v_i and v_i is also a successor to v_j. This is not possible due to the fact that a successor node v_j can not be a predecessor node to the same node v_i (Eq. (3)). Hence, when $(v_i, v_j) \in R$ then $(v_j, v_i) \notin R$ and vice-verse (by contradiction). □

3. *Transitive:* R is transitive, i.e., when $(v_i, v_j) \in R$ and $(v_j, v_k) \in R$ then $(v_i, v_k) \in R$.

Proof. Let us consider, $(v_i, v_j) \in R$. Therefore, v_j is a successor to v_i and $\hat{\Sigma}_i < \hat{\Sigma}_j$. Also consider, $(v_j, v_k) \in R$. Then, v_k is a successor to v_j and $\hat{\Sigma}_j < \hat{\Sigma}_k$. As the relation $<$ on $\mathbb{N}$ is transitive, we have $\hat{\Sigma}_i < \hat{\Sigma}_k$. Similarly, we can argue that the successor relation is also transitive, because if a node v is a successor to u and w is a successor to v, then w is also a successor to u. Therefore, v_k is a successor to v_i.

That means, both v_k is a successor to v_i and $\hat{\Sigma}_i < \hat{\Sigma}_k$ imply $(v_i, v_k) \in R$. Hence, the relation R is transitive. □

4.4. Proof of correctness

Lemma 4.1. *For a sub-modular influence function, removing a node u from S will no way decrease the marginal contribution of any node* $v \in S \setminus \{u\}$.

Proof. Let us consider the above statement as false, i.e, on removing a node u there exists at least one node $v \in S \setminus \{u\}$ such that

$$\sigma(Q) - \sigma(Q \setminus \{v\}) < \sigma(S) - \sigma(S \setminus \{v\}) \tag{7}$$

where $Q = S \setminus \{u\}$.

But, as per the sub-modular property of $\sigma(\cdot)$ explained in Eq. (5), the following inequality

$$\sigma(Q) - \sigma(Q \setminus \{v\}) \geqslant \sigma(S) - \sigma(S \setminus \{v\}) \text{ (as } Q \subset S). \tag{8}$$

is true. Eqs. (7) and (8) contradict each other. Therefore, the assumption on the existence of node v for which marginal influence decreases, is wrong for a sub-modular influence function. Hence, for a sub-modular function, removing a node u from S will never reduce the marginal contribution of a node $v \in S \setminus \{u\}$ (proved). □

Lemma 4.2. *For any node v having* $|g(v)| \geqslant k$ *there exist at least k number of nodes producing higher marginal influence in the network.*

Proof. As per the definition of function $g(\cdot)$ (Eq. (6)), we can say that if a node u is an element of $g(v)$ then node u has more marginal gain than node v. When the cardinality of $g(v)$ is equal to or more than k, then it is obvious that the node v has k or more nodes which have higher marginal gain than that of node v (proved). □

Lemma 4.3. *For any two nodes* u, v *in* D^*, *if* $(u, v) \in R$ *then there must be at least k nodes not in* D^* *having higher marginal influence than those of u and v.*

Proof. A node $v \in D^*$ satisfies $|g(v)| \geqslant k$ (by definition). Therefore, the node v has k or more nodes having higher influence in the network (from Lemma 4.2). The same applies to node u. Thus we only have to prove that at least k such elements are not members of D^*.

Now given $(u, v) \in R$ implies $v \in g(u)$. Let us consider the base case, i.e., $|g(u)| = k, |g(v)| = k$ and assume that $g(v) \cap D^* = \emptyset$. It is given that $v \in D^*$ and $v \in g(u)$. Therefore $g(u) \cap D^* = \{v\}$. That is, node u has only $k - 1$ nodes with higher marginal influence and they are not elements of D^*. Now, let $w \in g(v)$, i.e., $(v, w) \in R$. Thus, $(u, w) \in R$ (due to transitivity property of R). Therefore, any node $w \in g(v)$ is also a member of $g(u)$ i.e., $g(v) \subset g(u)$. Clearly, $|g(u)| > k$ (as $|g(v) \cup \{v\}| > k$). So, node u has at least k nodes which have higher marginal gain and are not member of D^* (As $g(v) \cap D^* = \emptyset$).

For the sake of argument, consider also $w \in D^*$, then v has only $k - 1$ nodes satisfying the condition. The same is also true for u. But, as $|g(w)| \geqslant k$ (by definition), w has k such nodes. With the same argument stated above one can show that $g(w) \subset g(v)$. Thus, v surely has at least k nodes those are not elements of D^*, and the same also applies to node u. Thus, by the law of induction, we can now conclude that even if $g(\cdot) \cap D^* \neq \emptyset$ for some arbitrary nodes (here u or v) there exists at least one successor (here $w \in D^*$ i.e., $|g(w)| \geqslant k$) for which $g(\cdot) \cap D^* = \emptyset$.

Hence, for any pair of nodes $u, v \in D^*$, if $(u, v) \in R$ then there exist at least k nodes, not members of D^*, producing higher marginal influence in the network (proved). □

Lemma 4.4. *For all nodes* $v \in D^*$ *there exist at least k number of nodes not in* D^* *producing higher marginal influence in the network.*

Proof. This lemma can be easily inferred from the aforesaid lemmas. A node $v \in D^*$ satisfies $|g(v)| \geqslant k$ (by definition). Therefore, the node v has at least k number of nodes having higher influence in the network (from Lemma 4.2).

Now there are two possibilities for a node $w \in g(v)$, either $w \notin D^*$ or $w \in D^*$.

- When $w \in g(v)\ \&\ w \notin D^*$, we do not have to prove anything further. It concludes the statement of the lemma.
- When $w \in g(v)\ \&\ w \in D^*$ i.e., $v, w \in D^*$ and $(v, w) \in R$ then also there exist at least k elements having higher marginal influence than both of $v,\ w$, but not members of D^* (from Lemma 4.3).

Hence, for all $v \in D^*$ there exist at least k number of nodes, not belonging to D^*, that would produce higher marginal influence in the network (proved). □

Theorem 4.1. *Given a sub-modular influence function $\sigma(\cdot)$, relation R and function $g(\cdot)$ defined over V, it is safe to remove all the nodes $v \in D^*$ and not consider them as members of the target set without affecting the performance.*

Proof. In the problem of target set selection, we are asked to select k number of seeds. For any node $v \in D^*$, there exist at least k number of nodes which are not member of D^*, and provide better marginal gain than by node v (Lemma 4.4). That is $S \setminus D^*$ will still contain k or more elements producing higher influence. Again, removing any node from the seed will never reduce the marginal performance of the remaining nodes (Lemma 4.1). Thus, the removal of all nodes $v \in D^*$ will not produce a less optimal solution. Hence, it is safe to remove these nodes from further consideration (proved). □

4.5. Convergence

Theorem 4.2. *Given a social network $G(V, E)$ with finite number of nodes, $|D^*|$ converges to 0.*

Proof. Let us consider the node $v \in V$ as a member of D^*. Therefore, the node v will be removed and would not further be considered in the next iterations. So, in each iteration, the number of considerable nodes (say, $V';\ V' \subseteq V$) gets reduced.

Since V is a finite set, V' is also a finite set. Now to be a member of D^*, the node v must have $|g(v)| \geqslant k$ for any constant $k > 0$. Thus the maximum possible cardinality of D^* is $(|V'| - k)$. Clearly, as nodes are removed from $V',\ |V'|$ converges to k. If we consider that $|D^*|$ never gets a value 0 until $|V'|$ converges to k then it is clear that after $|V'|$ converges, any further iteration would not lead to add any node to D^* as none can satisfy the condition $|g(v)| \geqslant k$. Hence, we can infer that $|D^*|$ converges to 0. □

4.6. Worst case performance analysis

In this section we present the performance analysis of DGS (deprecation based greedy strategy), both in term of the worst case time requirement and worst case influence guarantee.

Theorem 4.3. *Given a social network $G(V, E)$, a heuristic function $\tilde{\sigma} : 2^V \to \mathbb{N}$ and an order $u \succcurlyeq v \iff \tilde{\sigma}(\{u\}) \geqslant \tilde{\sigma}(\{v\})$, defined over V, the set of seeds S of cardinality k selected by DGS ensures at least the same influence as of selecting the set of seeds of cardinality k based on the order $\succcurlyeq$.*

Proof. In the process of greedy deprecation strategy, we input the ordered m-tuple $\tau = (v_1, v_2, \ldots, v_m)$ defined over V^m where $v_i \succcurlyeq v_{i+1},\ \forall i = 1, 2, \ldots, m;\ m \leqslant n$. We then find D^* and remove them from τ. The next iteration runs over the reduced tuple τ. Finally, after convergence, we select the top-k nodes from the reduced τ. Now, in the worst case, the algorithm might run for only one iteration, i.e., the stopping criterion is satisfied just after the first iteration. In that case, we may not find any node for deprecation. In this scenario, the final output will be the top-k nodes selected from the ordered m-tuple τ which is basically the same set of nodes selected based on $\succcurlyeq$. Hence, in the worst case, DGS produces the result, exactly the same as obtained using $\succcurlyeq$. □

Theorem 4.4. *Given a social network $G(V, E)$ and an m-tuple $\tau,\ (m \leqslant |V|)$, the worst case time requirement of the DGS algorithm is $O((m + k)^2)$.*

Proof. Worst case scenario in DGS occurs when the following two conditions are satisfied.

1. the stopping criterion does not meet even when only k number of nodes remain, and
2. for each iteration, only one node is marked as deprecated.

Thus, in such case the total number of iterations required is $(m-k)$ and in each iteration, all the remaining nodes are evaluated for their marginal contribution. Total number of evaluation conducted here is $m+(m-1)+(m-2)+\ldots+(m-k+1)+(m-k)$ i.e.,

$$m(m-k+1)-\frac{(m-k+1)(m-k)}{2}=\frac{(m-k+1)\times(m+k)}{2}=\frac{(m+k)^2-(2k-1)(m+k)}{2}$$

Assuming an evaluation be performed in 1 operation, the worst case time complexity of DGS is in the order of $(m+k)^2$ as $k<m$, i.e., the time complexity is $O((m+k)^2)$. □

5. Experiments and results

Experiments have been conducted over seven real life social networks to validate the aforesaid mathematical findings. We have considered Independent Cascade Model of information diffusion and applied DGS over the list of nodes pre-ordered by high degree heuristic (HDH), diffusion degree heuristic (DiDH) and degree discount heuristic (DDH). Performance of DGS is evaluated in two ways. Firstly we measure the improvement made by DGS over the seeds selected by the corresponding heuristic methods. Then we compare the overall influence with other target set selection algorithms. Following sub sections mention the detailed characteristics of the data sets, the configurations of DGS, discussion on comparing methods and the results on seven data sets.

5.1. Data sets used

The five un-weighted and two weighted social networks are used in the experiment. The un-weighted networks are friendship networks of Twitter [8], Slashdot [25] and Pokec [37], European Email Communication Network (EUEmail) [23], and ego network of Google+ (GPlus) [28]. Two weighted networks used are friendship network of an online community of University of California [32] (OCLinks) and US airport network [31] (USAirport). Properties of these data sets are listed in Table 1.

5.2. Implementation of DGS

5.2.1. Heuristic influence function

As the DGS runs on an ordered tuple which is based on a heuristic influence score, an influence function is required to order the nodes. Any heuristic influence function would work for this purpose. In the experiments, the following heuristic influence functions are used,

- *High degree:* Here, node's influence score is the in-degree score of the node.
- *Diffusion degree:* In this function, node's influence score is calculated based on the diffusion degree score proposed in [33].
- *Degree discount:* The function calculates the discounted degree score as per [5].

5.2.2. Estimation of $\sigma(\cdot)$

There are no deterministic way to estimate the value of $\sigma(\cdot)$. In our experiments $\sigma(\cdot)$ is estimated by Monte Carlo simulation. Monte Carlo simulation is described in Section 5.3.

Table 1
Features of data sets.

Property	Twitter	Slashdot	EUEmail	Pokec	GPlus	OCLinks	US Airport
ML Nodes	455818	82168	265214	1632803	107614	1899	1574
Edges	822487	948464	420045	30622564	13673453	20296	28236
Nodes in Largest WCC[a]	455818	82168	224832	1632803	107614	1893	1572
Edges in Largest WCC	822487	948464	395270	30622564	13673453	20292	28235
Nodes in Largest SCC[b]	2208	71307	34203	1304537	69501	1893	1402
Edges in Largest SCC	10401	912381	151930	29183655	9168660	20292	28032
Avg. Clustering Coefficient	0.0175	0.0617	0.3093	0.1094	0.4901	0.1094	0.5042
Number of Triangles	57769	602592	267313	32557458	1073677742	14319	245172
Fraction of Closed Triangles	0.0002781	0.02411	0.004106	0.01611	0.6552	0.01969	0.1721
Diameter	7	12	13	11	6	8	8
90-Percentile Effective Diameter	4	4.7	4.5	5.3	3	3.7	3.8

[a] Weakly Connected Component.
[b] Strongly Connected Component.

5.2.3. Input size

One may note that the best performance in terms of influence is obtained when $m = |V|$. However, this is time consuming. In order to reduce the execution time we consider a value of m, say $m = 10k$, less than $|V|$ for the data set used.

5.3. Performance evaluation

Performance of the selected set of seeds is analyzed based on the total nodes influenced by them. The total number of nodes influenced by a set of seeds is estimated using stochastic simulation. We have used Monte Carlo process for the same. We executed the simulation for 10,000 times and reported the average value as result. Each run of a Monte Carlo simulation of independent cascade model of information diffusion works as follows. It takes the seed set (i.e., the initial set of active nodes) and the network as input. At any time t the simulator selects an active node randomly and tries to activate one of its incident edges associated with an inactive node. The activation occurs with a probability referred as the propagation probability. If succeeds, the neighboring node is added to the set of the existing active nodes. Otherwise, it flags the edge as visited. Once visited, the same link is not considered further by the same node. The simulation stops when no further activation occurs, and the cardinality of the set of the active nodes is returned as the number of total nodes influenced.

5.4. Comparing methods

We analyze the performance of DGS in the following two ways:

A. We first execute the greedy deprecation strategy on the ordered tuple based on the three heuristic influence functions discussed before. We then compare the performance of the k seeds selected after applying DGS on these functions with the seeds selected by the corresponding heuristic algorithm. That is, we select seeds using DGS on high degree influence function (DGS on HDH) and compare them with those selected by the high degree heuristics (HDH). Similarly compare the diffusion degree heuristic (DiDH) with DGS on diffusion degree influence function (DGS on DiDH), and the degree discount heuristic (DDH) with DGS on degree discount influence function (DGS on DDH).
B. We compare the overall performance of DGS with the other target set selection algorithms. For this purpose we run our experiment for different values of k. We compare our results with heuristics methods (HDH, DiDH and DDH) as well as the recent Prefix excluding Maximum Influence Arborescence Model (PMIA). We have implemented PMIA algorithm as described in [4] with the best possible value of the threshold parameter (θ) as noted in the original paper.

5.4.1. Reference point

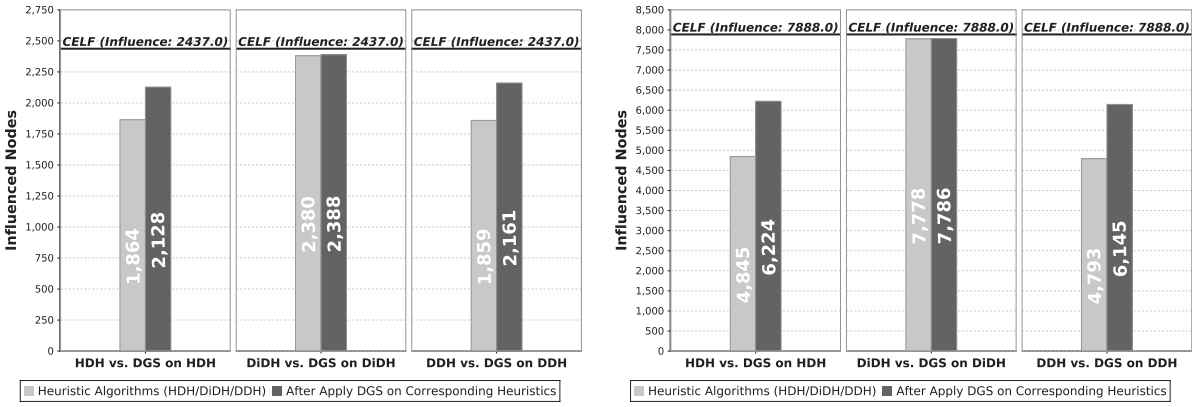
We use CELF [24], a well known fast greedy algorithm which indicates at least $\frac{1}{2}(1 - 1/e)$ optimization, as a reference method. That means, it is desirable that an algorithm produces results closer to that of CELF. In the experiment, k seeds obtained by CELF are simulated using Monte Carlo simulation to get the approximate possible influence by those k seeds, and this value is plotted as the reference line in the results. A point to note here is that the input size of CELF is also restricted to $10k$ as in DGS in order to have fair comparison between CELF and DGS.

5.5. Results

5.5.1. Twitter network

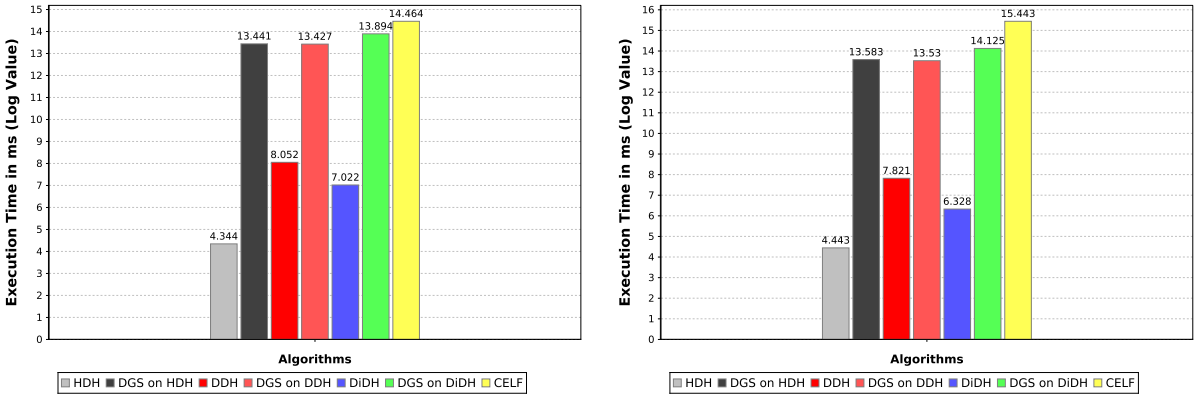
Fig. 2 shows bar charts to demonstrate the effect of DGS on different heuristic influence functions for Twitter data. The Twitter data is a directed social network where a node represents a user, and a link represents the following-followers relationship between users. We have experimented with different values of the propagation probability. Here, we show the results for two such different values for illustration. Each bar is marked with the actual number of nodes influenced by $50(= k)$ seeds. A reference line showing the number of nodes influenced by 50 seeds, selected with CELF, is placed on every chart. The charts clearly show that applying DGS on HDH and DDH increases the total number of nodes influenced by 50 seeds. With 0.05 propagation probability, DGS provides 14% and 16% gain when applying over HDH and DDH respectively and for propagation probability 0.1, the rise is 28.5% and 28.2%. For DiDH, applying DGS also produces better results but the difference is not significantly high. A point may be noted here is that the amount of influence cause by an algorithm would never decrease due to the application of DGS on it (as proved in Theorem 4.3).

The execution time of different algorithms for Twitter network is plotted in Fig. 3. The value shown here for each bar is the log value with base e of the actual execution time. As expected, the execution time of the DGS on HDH, DGS on DiDH and DGS on DDH is higher compared to that of the corresponding heuristic method. However, it is seen that the DGS algorithm runs much quicker than CELF. In Fig. 3(a) we see that the log value of execution time for DGS on DiDH is 13.894 which corresponds to the value of 1,081,520 ms, whereas log value 14.464 of CELF corresponds to 1,913,309 ms. So, CELF takes almost 76.91% more time as compared to the slowest version of DGS algorithm. If we compare the number of times a node's performance is evaluated in CELF with the DGS, the difference (e.g., 7, 8 or 9 in DGS compare to 50 in CELF) is evident. In case of CELF, the algorithm iterates k times; on the other hand, the proposed algorithm can identify the weak seeds in fewer numbers of iterations. A chart showing the number of iterations required by the different algorithms is shown in Fig. 4. The



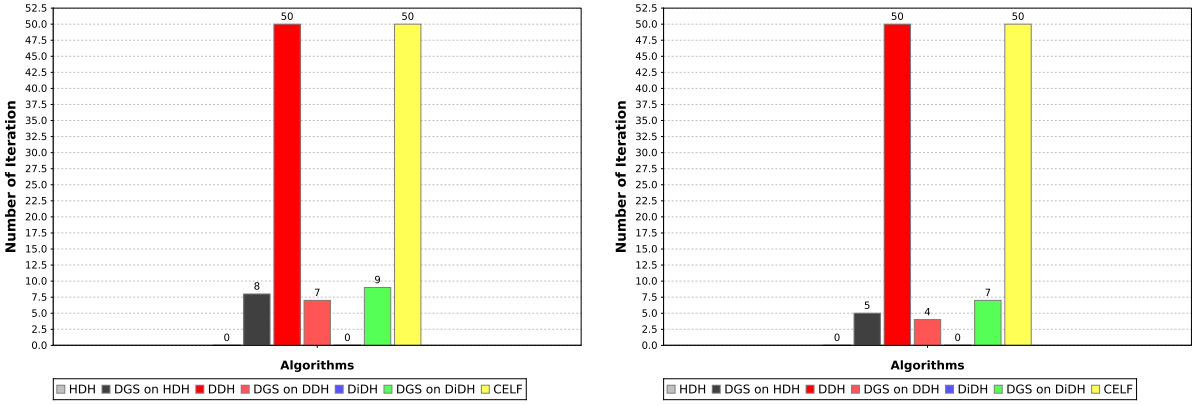
(a) Propagation Probability 0.05 & $k = 50$ (b) Propagation Probability 0.1 & $k = 50$

Fig. 2. Effect of DGS over total number of nodes influenced for different heuristic influence functions on Twitter network.



(a) Propagation Probability 0.05 & $k = 50$ (b) Propagation Probability 0.1 & $k = 50$

Fig. 3. Bar chart showing the execution time (in log with base e) of different algorithms on Twitter network.



(a) Propagation Probability 0.05 & $k = 50$ (b) Propagation Probability 0.1 & $k = 50$

Fig. 4. Number of iteration require for different algorithms of Twitter network.

proposed DGS converges after a very small number (< 10) of iterations. HDH and DiDH do not require evaluating a node's performance, so the number of iterations is 0. For DDH, the number of iterations is equal to $k(= 50)$. However, in each iteration it does not evaluate a node's performance; it just recalculates the degree of a node with the discounted value based on the previous iteration. So, in terms of execution time it is superior to CELF and DGS.

Experiment has also been conducted with different values of k. Two such results are shown in Fig. 5, as example. The graph clearly shows that for all values of k, DGS provides better quality seeds in terms of the number of nodes influenced. The superiority becomes more prominent after $k = 10$.

5.5.2. EUEmail network

EUEmail network data was generated using email communication data from a large European research institution. Each node in the network corresponds to an email id and directed link is created between two nodes i and j, if i sends a message to j. We have experimented with this network in the same way as we have done with Twitter data. Results are shown in Fig. 6 which shows an improvement made by our algorithm for all the heuristics. For example, there are 7.1%, 6.9% and 4.63% improvement when we apply DGS over HDH, DiDH and DDH respectively.

It is observed that DGS when applied over different heuristics takes very less computational time compared to CELF, however, the seeds selected by DGS and CELF produce almost similar influence in the network. Comparison of the execution time is shown in Fig. 6(b). It is clear from the diagram that the time taken even by the slowest DGS (e.g., 431,021 ms, i.e., log value 12.974) is much less than the time taken by CELF (28,456,333 ms, i.e., log value 17.16).

Similar to Twitter data we compare the overall performance for different values of k and plotted the variation of the number of nodes influenced vs k in Fig. 7. The graph shows that for all values of k, DGS provides better quality seeds than the other comparing methods.

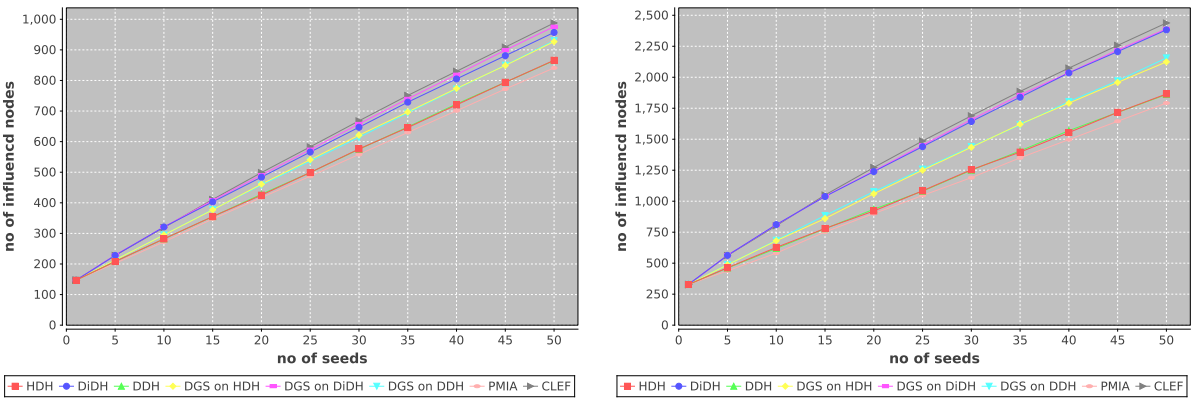
5.5.3. Slashdot network

The network of Slashdot is a friendship network of the technology related news website Slashdot. Each node is a user here, and the connection between them indicates whether a user is a friend to other or not. Fig. 8 shows the results with the propagation probability 0.01. We selected 50 top seeds using different comparing methods and find the total influence using Monte Carlo simulation in the network. Unlike Twitter and EUEmail networks, Slashdot is seen to have only a small improvement after applying DGS over the heuristics. This is true for even with other values of k. Fig. 9 shows a plot which depicts the total influence of selected nodes vs k. Execution time-wise it shows similar patterns as we found in the cases of Twitter and EUEmail.

5.5.4. OCLinks and USAirport networks

OCLinks and USAirport are weighted networks, i.e., each edge is associated with a weight. The weight denotes differently for different data sets. We, in our experiments, normalized these weights and considered them as propagation probabilities. The OCLinks data originated from an online community for students at the University of California, Irvine. The data set contains 1899 users who sent or received at least one message. Weights are given to each edge denoting the number of messages sent from the source to the destination user of the edge. The network of USAirport is downloaded from the Bureau of Transportation Statistics (BTS) Transtats site (Table T-100; id 292) with the following filters: Geography = all; Year = 2010; Months = all; and columns: Passengers, Origin, Dest. The weights here correspond to the number of seats available on the scheduled flights.

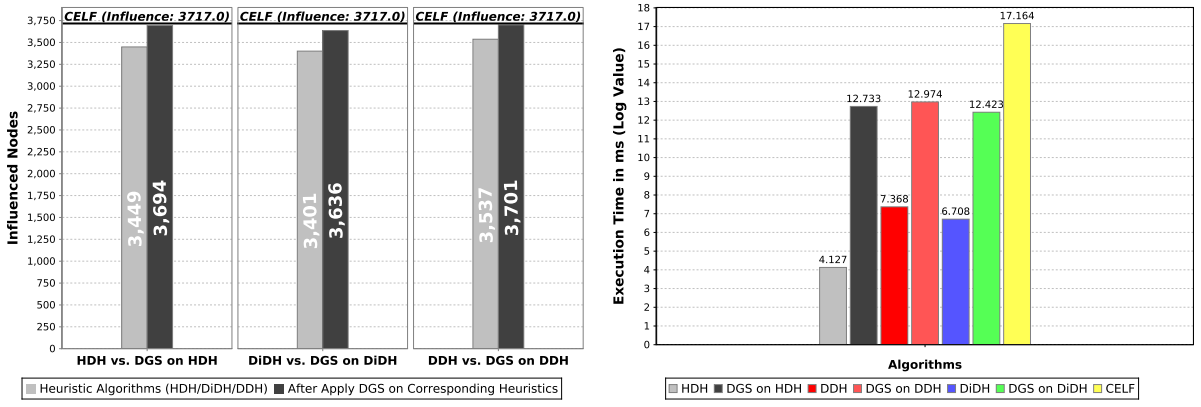
Results for OCLinks data are shown in Figs. 10 and 11. Fig. 10(a) shows the improvement upon applying DGS over all the three heuristics. For all the cases we see that the resulting influence after applying DGS is at per with the benchmark CELF. The same is evident in Fig. 11 for other values of $k > 35$. Further, the proposed DGS takes significantly lower time than CELF (e.g., 37.978 s against 394.993 s) producing almost the same influence.



(a) Propagation Probability 0.025

(b) Propagation Probability 0.05

Fig. 5. Number of influenced nodes vs number of seeds for different algorithms on Twitter network.



(a) (b)

Fig. 6. Effect of DGS in terms of (a) total influence (b) execution time of different algorithms on EUEmail network. Propagation probability is set to 0.05 and $k = 50$.

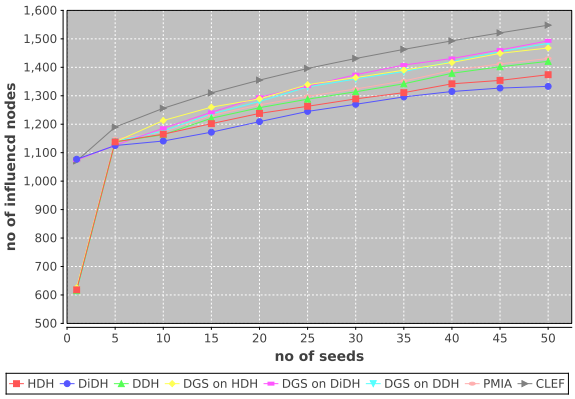
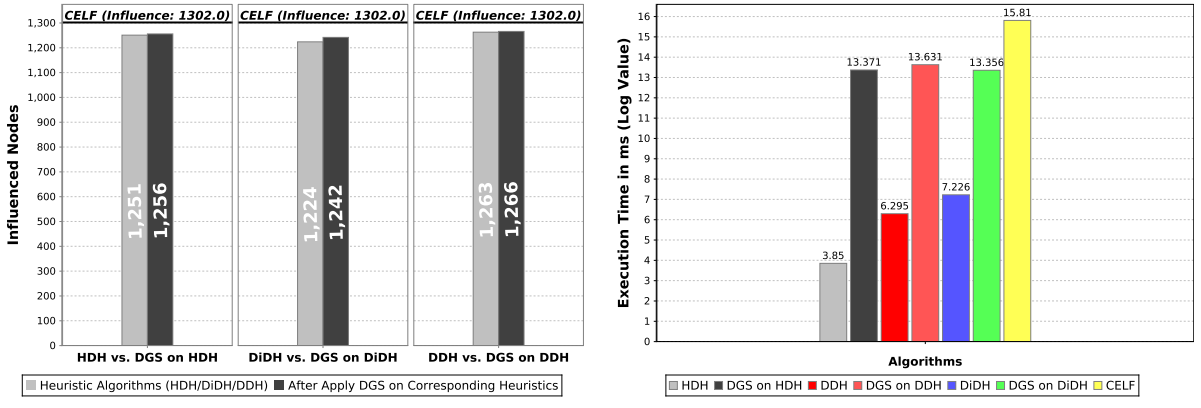


Fig. 7. Number of nodes influenced vs number of seeds for different algorithms on EUEmail network with propagation probability 0.025.



(a) (b)

Fig. 8. Effect of DGS in terms of (a) total influence (b) execution time of different algorithms on Slashdot network. Propagation probability is set to 0.01 and $k = 50$.

On the other hand, for USAirport network applying DGS produces the same influence as of CELF (Fig. 12(a)), whereas taking much less time 69.519 s (log value 11.149) compare to CELF's 743.282 s (log value 13.519) (Fig. 12(b)). Fig. 13 shows the plots comparing the DGS with other algorithms for different values of k up to 50. Here we observe an interesting phenomenon. For HDH and DiDH the influence is almost same for all the values of k. This is due to the fact that these two

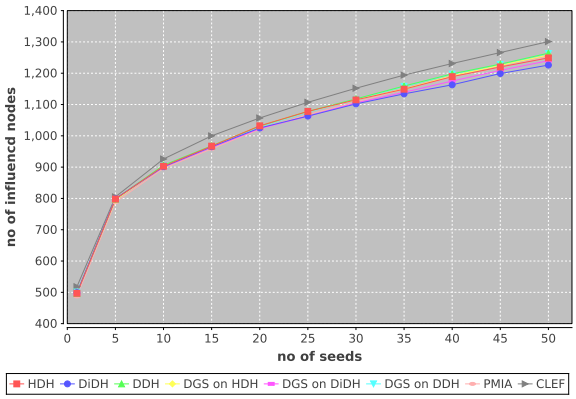
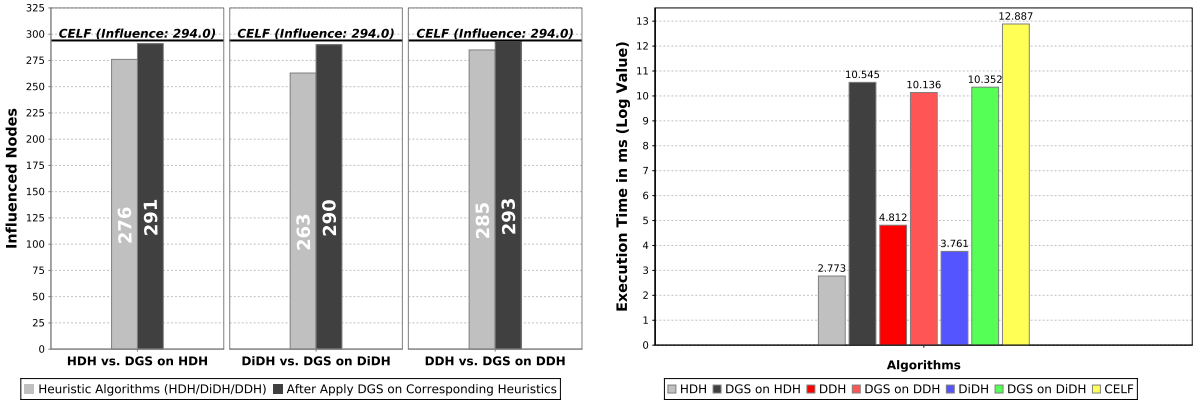


Fig. 9. Number of nodes influenced vs number of seeds for different algorithms on Slashdot network with propagation probability 0.01.



(a) (b)

Fig. 10. Effect of DGS in terms of (a) total influence (b) execution time of different algorithms on OCLinks network ($k = 50$).

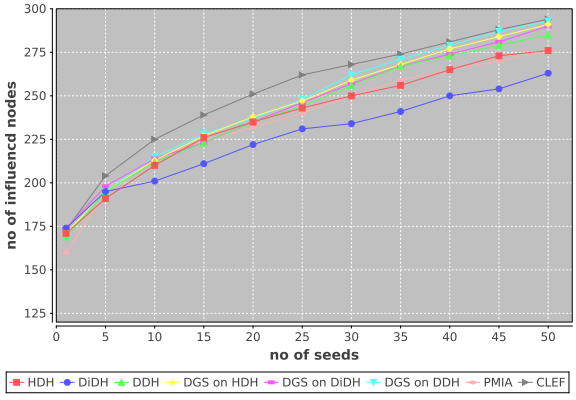
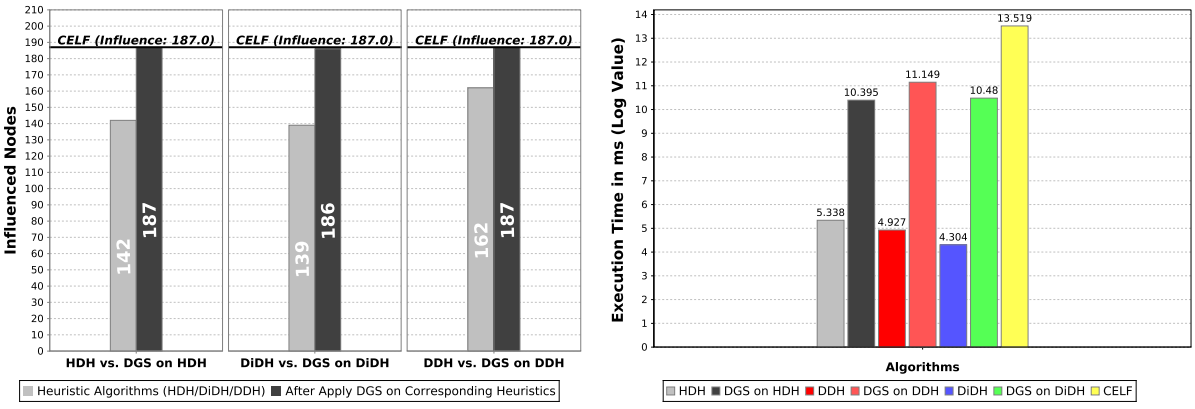


Fig. 11. Plot showing number of nodes influenced vs number of seeds of OCLinks network.

approaches select all the nodes from the same neighborhood which lead to the same influence. DDH and PMIA show better results after $k > 15$, but noway close to that of CELF. The proposed strategy, on the other hand, produces superior results by selecting seeds which have the same influence as of CELF in the network for all the values of k.



(a) (b)

Fig. 12. Effect of DGS in terms of (a) total influence (b) execution time of different algorithms on USAirport network ($k = 50$).

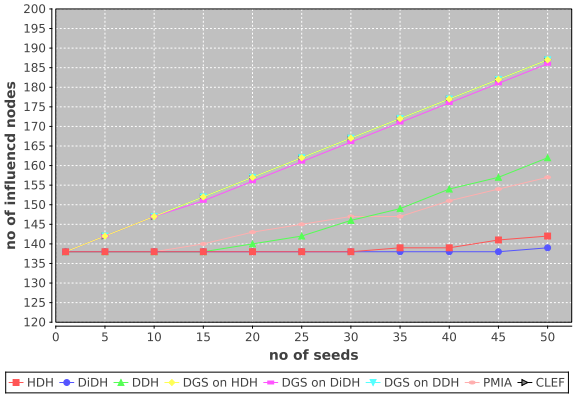
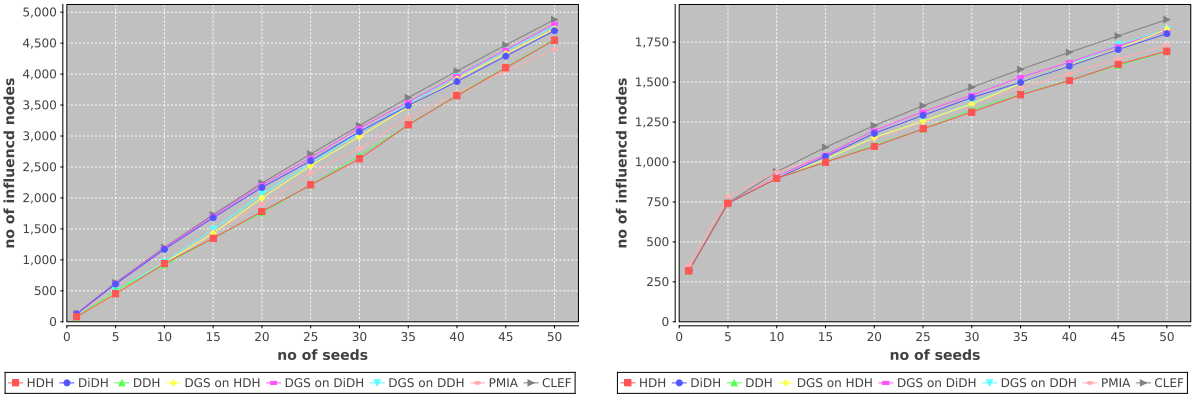


Fig. 13. Plot showing number of nodes influenced vs number of seeds of USAirport network.



(a) Gplus (b) Pokec

Fig. 14. Plots showing influenced nodes vs k results of different algorithms. k varies from 1 to 50 (propagation probability 0.01).

5.5.5. GPlus and Pokec

The data set of Google+ (Gplus) is an ego network extracted from the 'circles' of Google + social network. On the other hand, Pokec is a most popular online social network of Slovakia. Pokec network is a friendship network where each node is a user. Results of these GPlus and Pokec corroborate to the observations as found for Twitter or EUEmail, i.e., DGS could correctly extract the better quality seeds. For reference we have included some sample results in Fig. 14.

6. Discussions and conclusion

In this paper, we proposed a new strategy, called *deprecation based greedy strategy* (DGS) and the corresponding algorithm for finding top-k influential nodes of social networks from a list of nodes pre-ordered through the help of heuristic measures. The algorithm searches for the lower contributory nodes in the list and marks them as deprecated. These deprecated nodes are then removed from any further consideration. We show analytically that when the influence function is monotonic and sub-modular, the algorithm correctly identifies the deprecated nodes, i.e., removal of these nodes does not affect the best results. We also proved theoretically that for a finite number of nodes, the algorithm converges in finite time. We calculated the possible approximation guarantees for the proposed algorithm and analyzed its worst case time requirement.

Experiments have been conducted to verify the theoretical findings over five large scale social network data and two weighted network data. We used three heuristic influence functions for pre-ordering the nodes. We computed the total influence of the selected seeds by these heuristic influence functions, and of the seeds obtained after applying DGS over these pre-ordered sets of nodes. Comparing the total influence of the selected seeds before and after applying the DGS shows that applying DGS improves the results for all the data sets, except Slashdot network where the performance is similar. These signify that applying DGS does no way produce inferior quality of seeds, as proved theoretically in Theorem 4.3.

It may further be noted that the scope of such improvement after applying DGS is likely to be more (or less) if the heuristic method, in question, has lower (or higher) performance. This is what is reflected in the results. For example, in Twitter data sets, the original gap (difference of results of DDH and the benchmark CELF) in performance was 31%. When we apply the proposed DGS, the gap is reduced to 12%, i.e., an improvement of the gap-reduction by about 60%. On the other hand, for EUEmail, the original gap was only 7.7% and this is further reduced by DGS to 0.7%, i.e, improvement in the reduction of the gap is more than 90%.

Execution time-wise the proposed DGS takes much less time than CELF algorithm while providing closer (for Twitter, EUEmail and GPlus) or similar (for OCLinks and USAirport) influence. The algorithm also converges in less than 10 iterations in all the test cases we experimented with.

Five out of seven network data sets, we experimented with, are large scale data, i.e., they are voluminous showing Big data characteristics. With these networks we achieved performance comparable to that of the benchmark CELF, while reducing the execution time by more than 50% for GPlus and Pokec, 76% for Twitter, 88% for Slashdot, and 98% for EUEmail. Thus, the proposed strategy (DGS) is applicable to large scale networks, and has strong promise in dealing with some of the Big data issues.

Acknowledgments

The authors acknowledge the Department of Science and Technology, Govt. of India for funding the Center for Soft Computing Research at Indian Statistical Institute. S.K. Pal acknowledges the J.C. Bose National Fellowship and INAE Chair Professorship. The support provided by Prof. Debesh K. Das of Department of Computer Science and Engineering, Jadavpur University is greatly acknowledged.

References

[1] O. Ben-Zwi, D. Hermelin, D. Lokshtanov, I. Newman, An exact almost optimal algorithm for target set selection in social networks, in: Proc. of the 10th ACM Conference on Electronic Commerce, ACM Press, Stanford, CA, 2009, pp. 355–362.
[2] E. Berger, Dynamic monopolies of constant size, J. Comb. Theory, Ser. B 83 (2) (2001) 191–200.
[3] N. Chen, On the approximability of influence in social networks, SIAM J. Discrete Math. 23 (3) (2009) 1400–1415.
[4] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 1029–1038.
[5] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, Paris, 2009, pp. 199–208.
[6] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 88–97.
[7] P. Clifford, A. Sudbury, A model for spatial conflict, Biometrika 60 (3) (1973) 581–588.
[8] M. De Choudhury, H. Sundaram, A. John, D. Seligmann, A. Kelliher, "Birds of a Feather": Does User Homophily Impact Information Diffusion in Social Media? 2010, pp. 1–31. Arxiv preprint arXiv:1006.1702.
[9] Z. Dezso, A.-L. Barabasi, Halting viruses in scale-free networks, Phys. Rev. E 65 (5) (2002) 1–4.
[10] P. Domingos, Mining social networks for viral marketing, IEEE Intell. Syst. 20 (1) (2005) 80–82.
[11] P. Domingos, M. Richardson, Mining the network value of customers, in: Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Francisco, CA, 2001, pp. 57–66.
[12] P.A. Estevez, P. Vera, K. Saito, Selecting the most influential nodes in social networks, in: Proc. of 2007 International Joint Conference on Neural Networks, IEEE, 2007, pp. 2397–2402.
[13] E. Even-Dar, A. Shapira, A note on maximizing the spread of influence in social networks, Internet Network Econ. 111 (4) (2007) 184–187.
[14] L. Freeman, Centrality in social networks conceptual clarification, Soc. Networks 1 (3) (1979) 215–239.
[15] J. Goldenberg, B. Libai, E. Muller, Talk of the network: a complex systems look at the underlying process of word-of-mouth, Market. Lett. 12 (3) (2001) 211–223.
[16] J. Goldenberg, B. Libai, E. Muller, Using complex systems analysis to advance marketing theory development: modeling heterogeneity effects on new product growth through stochastic cellular automata, Acad. Market. Sci. Rev. 9 (3) (2001) 1–18.
[17] J. Guo, P. Zhang, C. Zhou, Y. Cao, L. Guo, Personalized influence maximization on social networks, in: Proc. of the 22Nd ACM International Conference on Information & Knowledge Management, ACM, New York, San Francisco, California, USA, 2013, pp. 199–208.
[18] J. Ha, S.-W. Kim, C. Faloutsos, S. Park, An analysis on information diffusion through BlogCast in a blogosphere, Inform. Sci. 290 (2015) 45–62.

[19] R.A. Holley, T.M. Liggett, Ergodic theorems for weakly interacting infinite systems and the voter model, Ann. Probab. 3 (4) (1975) 643–663.
[20] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, NY, 2003, p. 137.
[21] D. Kempe, J. Kleinberg, E. Tardos, Influential nodes in a diffusion model for social networks, Autom. Lang. Program. 3580 (2005) 1127–1138.
[22] J. Kleinberg, Cascading behavior in networks: algorithmic and economic issues, in: N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani (Eds.), Algorithmic Game Theory, Cambridge University Press, 2007, pp. 613–632.
[23] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: densification and shrinking diameters, ACM Trans. Knowl. Discovery Data 1 (1) (2007) 2.
[24] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, San Jose, 2007, pp. 420–429.
[25] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney, Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters, Internet Math. 6 (1) (2009) 29–123.
[26] Y.-M. Li, C.-Y. Lai, C.-W. Chen, Discovering influencers for marketing in the blogosphere, Inform. Sci. 181 (23) (2011) 5143–5157.
[27] S. Liu, C. Jiang, Z. Lin, Y. Ding, R. Duan, Z. Xu, Identifying effective influencers based on trust for electronic word-of-mouth marketing: a domain-aware approach, Inform. Sci. 306 (2015) 34–52.
[28] J.J. McAuley, J. Leskovec, Learning to discover social circles in ego networks, in: P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 25, Lake Tahoe, Nevada, 2012, pp. 548—-556.
[29] R. Narayanam, Y. Narahari, A Shapley value-based approach to discover influential nodes in social networks, IEEE Trans. Autom. Sci. Eng. 8 (1) (2011) 130–147.
[30] J. Nieminen, On the centrality in a graph, Scand. J. Psychol. 15 (4) (1974) 332–336.
[31] T. Opsahl, Why Anchorage is Not (that) Important: Binary Ties and Sample Selection, 2011. <http://wp.me/poFcY-Vw>.
[32] T. Opsahl, P. Panzarasa, Clustering in weighted networks, Soc. Networks 31 (2) (2009) 155–163.
[33] S.K. Pal, S. Kundu, C.A. Murthy, Centrality measures, upper bound, and influence maximization in large scale directed social networks, Fundam. Inform. 130 (3) (2014) 317–342.
[34] R. Pastor-Satorras, A. Vespignani, Epidemics and immunization in scale-free networks, in: S. Bornholdt, H.G. Schuster (Eds.), Handbook of Graphs and Networks: From the Genome to the Internet, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, 2002. chap. 5.
[35] M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, in: Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, Edmonton, Alberta, 2002, pp. 61–70.
[36] E.M. Rogers, Diffusion of Innovations, fifth ed., Free Press, 2003.
[37] L. Takac, M. Zabovsky, Data analysis in public social networks, in: Proc. of International Scientific Conference & International Workshop Present Day Trends of Innovations, No. May, Lomza, Poland, 2012, pp. 1–6.
[38] C. Wang, L. Deng, G. Zhou, M. Jiang, A global optimization algorithm for target set selection problems, Inform. Sci. 267 (2014) 101–118.
[39] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (6684) (1998) 440–442.
[40] Z. Yu, C. Wang, J. Bu, X. Wang, Y. Wu, C. Chen, Friend recommendation with content spread enhancement in social networks, Inform. Sci. 309 (2015) 102–118.
[41] T. Zhu, B. Wang, B. Wu, C. Zhu, Maximizing the spread of influence ranking in social networks, Inform. Sci. 278 (2014) 535–544.