Optimizing a static greedy algorithm for influence maximization

Leonardo Capone, Nicola Di Mauro, Floriana Esposito
Università degli Studi di Bari "Aldo Moro"
Via Orabona, 4, 70125 Bari
leonardocapone89@gmail.com, nicola.dimauro@uniba.it, floriana.esposito@uniba.it

One of the main problem in social networks and viral marketing is that of finding a set of nodes maximizing the spread of influence. Corresponding algorithms solving this problem are required to have both guaranteed accuracy and high scalability. Greedy algorithms are able to find accurate solutions but fail in efficiency. This paper presents a modification of an existing greedy algorithm to solve the influence maximization problem by integrating a memoization technique. Experimental results with a first prototypical implementation on real-world social networks proved the validity of the proposed technique.

***Keyworkds**: influence maximization, memoization, social networks*

1. Introduction

Nowadays people are connected by heterogeneous social relationships in large-scale online social networks that provide a platform for information dissemination and marketing. The success of viral marketing is rooted in the interpersonal influence empirically studied in [Richardson et al., 2002; Huang et al., 2012]. With the advent of social networks the information spreads in the form of “word-of-mouth” communications, and it is noticeable to observe how much they affect our daily life style.

Influence maximization is a fundamental problem for viral marketing and it has been originally formulated as an optimization problem in [Kempe et al., 2003]. It consists in finding a set of seed nodes which maximize the *influence spread* in a social network computed as the expected number of nodes influenced by the seed nodes. In [Kempe et al., 2003] has been proved the NP-completeness of the influence maximization problem and it has been provided a greedy approximation algorithm that yields an influence spread solution that is no less than a given bound of the optimal value.

Given a seed set of nodes there is no exact algorithm that gives the corresponding influence spread. Usually it is approximated using a large number of Monte Carlo simulations. However, this reduces the scalability of the general greedy algorithm proposed in [Kempe et al., 2003] since it requires too many Monte Carlo simulations. To overcome this problem one can reduces the times of influence spread estimations [Leskovec et al., 2007, Cheng et al., 2012], or proposes various heuristics to use more efficient methods for influence spread estimation [Chen et al., 2010]. The hot interest in scalable and accurate methods to solve the influence maximization problem is confirmed by a lot of recent works [Chen et al., 2009; Goyal et al., 2011; Leskovec et al., 2007; Jiang et al., 2011; Kimura et al., 2010].

Two widely used information diffusion models to solve the *influence maximization problem* [Kempe et al., 2003] are the *independent cascade* (IC) [Kempe et al., 2003] and the *linear threshold* (LT) [Watts, 2002] models. The IC model is sender-centered and each active node independently influences its inactive neighbors with given diffusion probabilities, while the LT model is receiver-centered and a node is influenced by its active neighbors if their total weight exceeds the threshold for the node.

In this paper, we provide an optimization of an existing greedy algorithm [Cheng et al., 2012] with its corresponding evaluation adopting the IC model on two real world social networks.

2. Preliminaries

In order to mathematically model the information diffusion in a social network, we firstly recall the IC and LT models according to [Kempe et al., 2003].

Let $G = (V, E)$ be a directed network (graph) where V is a set of nodes and $E \subset V \times V$ is a set of directed links. For each node $v \in V$, let $F(v)$ be the set of all the nodes that have links from v (its *children nodes*), i.e., $F(v) = \{u \in V; (v, u) \in E\}$, and $B(v)$ be the set of all the nodes that have links to v (its *parent nodes*), i.e., $B(v) = \{u \in V; (u, v) \in E\}$. We say a node is *active* if it has been influenced with the information. In the IC and LT models the diffusion processes unfold in discrete time steps, and it is assumed that nodes can switch their state only from inactive to active.

2.1 Independent Cascade Model

In the IC model it is necessary do define in advance a probability $p_{u,v}$, called *propagation probability*, for each directed link. Given an initial set A of active nodes the diffusion process works as follow. If a node u is active at step t, it has a chance to activate each currently inactive node $v \in F(u)$ with probability $p_{u,v}$. Whether or not u succeeds in activating v, it cannot make any further attempts to activate v in subsequent time steps and the overall process terminates if no more activations are possible.

2.2 Linear Threshold Model

In the LT model, for any node $v \in V$, it is necessary to specify a weight $w_{u,v} > 0$ for each of its parent node u, such that their sum is lesser or equal to 1.

When an initial set A of active nodes is given, and a threshold θ_v for each node v is set to be uniformly distributed in the interval [0,1], the diffusion process works as follows. An inactive node v at time step t is influenced by each active parent node u according to the weight $w_{u,v}$. If the total weight from active parents is greater than θ_v, then v will be active at time step t+1. The overall process terminates if no more activations are possible.

Given an initial active set A, let $\sigma(A)$ denote the expected number of active nodes at the end of the random process in the IC or in the LT model. We call $\sigma(A)$ the *influence degree* of the set A.

2.3 Influence Maximization Problem

The influence maximization problem is defined as follows. Given a positive integer k, find a set A of k nodes such that, for each set B of k nodes, $\sigma(A) \geq \sigma(B)$.

A greedy algorithm that approximately solves this problem has been proposed in [Kempe et al., 2003], and it is sketched in the following.

```
1. Set $A \leftarrow Ø$
2. **for** i=1 to k **do**
3.    choose a node $v_i \in F$ maximizing $\sigma(A \cup \{v\})$
4.    set $A \leftarrow A \cup \{v\}$
5. **endfor**
```

In [Kempe et al., 2003], this algorithm has been proved to obtain an approximate solution whose value is at least (1 – 1/e) $\sigma(A^*)$, where A^* is the optimal solution. This factor has been obtained by proving that the influence function $\sigma()$ is *submodular* and using a result obtained in [Nemhauser et al., 1978]. Formally, a submodular function satisfies:

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T),$$

for all elements v and all pairs of sets $S \subseteq T$.

2.4 Approximate Influence Degree

However, how to exactly computing σ(A) is not known and it is approximated using a large amount of Monte Carlo simulations [Kempe et al., 2003], thus degrading the efficiency of the algorithm.

In particular, the Monte Carlo approach used to approximate the influence degree $\sigma(A)$ in the IC model works as follows. Let A_t be the set of nodes that are activated in the time step t, and $A_0 = S$. For any link $(u,v) \in E$ such that $u \in A_i$ and v is not yet activated, then v is activated by u in the time step t + 1 with the propagation probability $p_{u,v}$. This process is repeated until A_{i+1} is empty.

The random process to estimate the influence degree in the LT model is quite similar to that used for the IC model but taking into account that the probability of u activating v is usually not the same as the probability of v activating u, thus requiring a slightly modification.

3.The new optimized algorithm

A recent interesting approach to overcome the problem to use a large amount of Monte Carlo simulations has been proposed in [Cheng et al., 2012]. The authors proved that the submodularity is not guaranteed in existing implementations of greedy algorithm, caused by the independence among Monte Carlo simulations executed in different iterations of the greedy algorithm. They proposed a static greedy algorithm to strictly guarantee the submodularity property, by reusing the results of Monte Carlo simulations during the whole process of greedy algorithm. The results is to dramatically reduce the random simulations thus effectively improving the scalability of the greedy approach.

In particular the authors of [Cheng et al., 2012] introduced the concept of *snapshot* obtained a priori according to the characteristic of the IC model. A snapshot is a graph G' obtained from the original graph G, where an edge (u, v) is removed with probability $1\text{-}p_{u,v}$. Then, for each snapshot G', the influence spread of a set of nodes S is the number of nodes reachable from S. Hence, the influence degree $\sigma(S)$ can be obtained by averaging over many snapshots.

The process of the corresponding static greedy algorithm is the following:

4.randomly sampling R snapshots from the underlying social network G;
5.start from an empty seed set S, then iteratively add one node a time into S such that the node provides the largest marginal gain of $\sigma(S)$, which is estimated on the R snapshots.

In particular, the static greedy algorithm, named Static, is formalized as follows, where $R(S)$ is a function returning the nodes reachable from the nodes in S:

```
initialize $S = \emptyset$
for i = 1 to $R$ do
    generate $G_i'$ by removing each edge $(u, v)$ from $G$ with probability $1\text{-}p_{u,v}$
endfor
for $i$ = 1 to $k$ do
    set $sv = 0$ for all $v \in V \setminus S$
    for $j$ = 1 to $R$ do
        for all $v \in V \setminus S$ do
            $sv$ += $|R(S \cup \{v\})|$
        endfor
    endfor
    $S = S \cup \{\text{argmax}_{v \in V \setminus S}\{ sv / R \}\}$
endfor
output $S$
```

In this paper, we tried to improve this static greedy algorithm by adopting a memoization approach. In particular, at time step t, we have to compute $R(S \cup \{v\})$ for each node $v \in V \setminus S$. However we can notice that the nodes reachable from the set $\{S \cup \{v\}\}$ corresponds to the nodes reachable from S plus the nodes reachable from v and not already reached from S. More formally, let S be equal to the nodes $\{s_1, s_2, \ldots, s_n\}$, then:

$$\begin{aligned} |R(S \text{ U } \{v\})| = \quad & | R(S) \text{ U } R(\{v\})| = \\ & | R(\{s_1, s_2, \ldots, s_n\}) \text{ U } R(\{v\})| = \\ & | R(\{s_1\}) \text{ U } R(\{s_2\}) \ldots \text{ U } \ldots R(\{s_n\}) \text{ U } R(\{v\})|. \end{aligned}$$

Hence, at each time step *t*, instead of computing $R(S \text{ U } \{v\})$ we can use its decomposition and take advantage of the previous computations. Indeed, for each snapshot, before to start the iterative process, we can compute the sets $R(\{v\})$ for each node of the snapshot. Then in the iterative process we can exploit this sets and the basic union operations to compute the influence spread. The static greedy algorithm exploiting memoization, named StaticM, becomes:

```
initialize $S = \emptyset$, $Q = \emptyset$
for i = 1 to $R$ do
    generate $G_i'$ by removing each edge $(u, v)$ from $G$ with probability $1\text{-}p_{u,v}$
endfor
for $i$ = 1 to $k$ do
    set $sv = 0$ for all $v \in V \setminus S$
    for $j$ = 1 to $R$ do
        for all $v \in V \setminus S$ do
            $sv \mathrel{+}= | Q \text{ U } R(\{v\})|$
        endfor
    endfor
    $S = S \text{ U } \{\text{argmax}_{v \in V \setminus S}\{ sv / R \}\}$
    $Q = Q \text{ U } R(\{v\})$
endfor
output $S$
```

The set *Q* in the algorithm represents the set of nodes reachable from the best chosen *k* nodes at the previous time step. It is enlarged, at each iteration, with the nodes $R(\{v\})$ for that node *v* such that the cardinality of $R(S \text{ U } \{v\})$ is maximized.

4. Experimental evaluation

In this section we use two real social network datasets, such as Epinions and Slashdot datasets available at http://http://snap.stanford.edu/data/, in order to evaluate the efficacy of our proposed approach improving the static greedy algorithm.

The Epinions dataset [Richardson et al., 2003] concerns a *who-trust-whom* online social network of a a general consumer review site Epinions.com. Members of the site can decide whether to “trust” each other. All the trust relationships interact and form the Web of Trust which is then combined with review ratings to determine which reviews are shown to the user. The corresponding graph consists of 75879 nodes and 508837 edges.

Slashdot is a technology-related news website know for its specific user community. The website features user-submitted and editor-evaluated current primarily technology oriented news. In 2002 Slashdot introduced the Slashdot Zoo feature which allows users to tag each other as friends or foes. The Slashdot dataset [Leskovec et al., 2009] network contains friend/foe links between the users of Slashdot. The network was obtained in February 2009 and consists of 82168 nodes and 948464 edges.

For both the datasets we used the IC model as a diffusion model and we set the propagation probability p to 0.5.

We compared the new StaticM algorithm against the Static and Random algorithm. The Random algorithm simply selects *k* seed nodes among the possible and computes its corresponding influence spread.

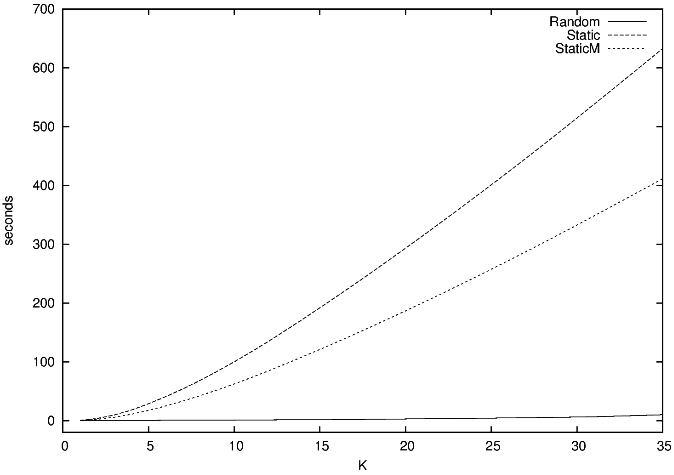

Fig.1 – Time in seconds on the Epinions dataset for Random, Static and StaticM.

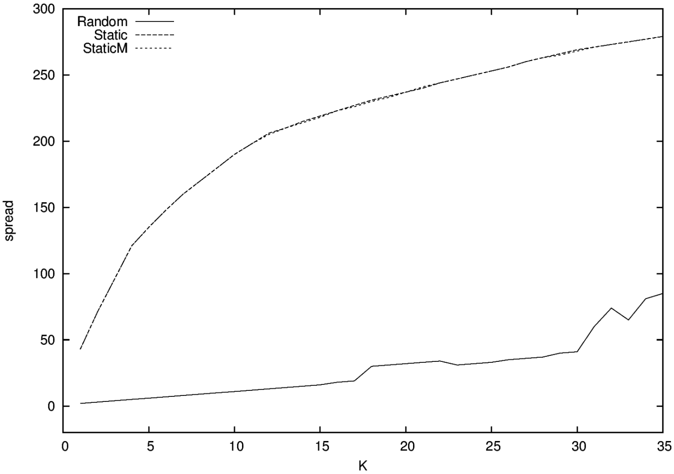

Fig.2 – Influence spread on the Epinions dataset for Random, Static and StaticM.

As we can see from Figure 1 StaticM requires a time lesser than that required by Static to compute the influence spread on the Epinions dataset. Random has the best performances since it just randomly picks nodes from the available ones. Figure 2 plots the influence spread for k ranging from 1 to 35 obtained by the algorithms. Both Static and StaticM reach, for each k, the same value and both obviously find solutions better than those obtained with Random. Similar results are obtained on the Slashdot dataset as reported in Figure 3 and Figure 4.

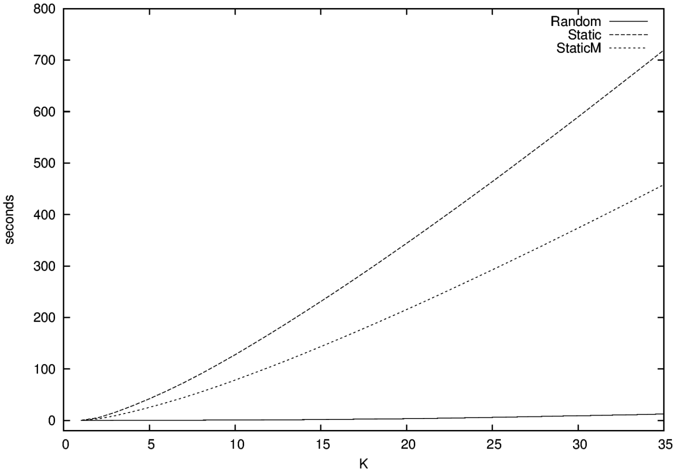

Fig.3 – Time in seconds on the Slashdot dataset for Random, Static and StaticM.

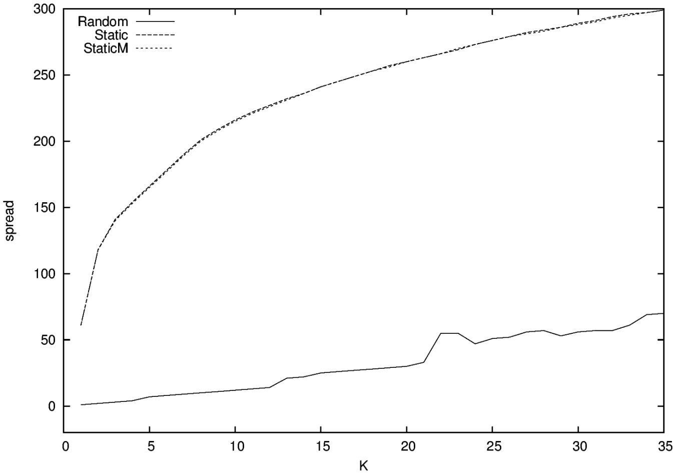

Fig.4 – Influence spread on the Slashdot dataset for Random, Static and StaticM.

5. Conclusions

One of the main problem in social networks and viral marketing is that of finding a set of nodes maximizing the spread of influence. Algorithms for solving this problem are required to have both guaranteed accuracy and high scalability. The proposed greedy algorithms are able to find accurate solutions but fail in efficiency. In this paper we have proposed a modification of a static greedy algorithm by integrating a memoization technique. Experimental results on the Epinions and Slashdot real world datsets exploited with a prototypical implementation proved the validity of the proposed technique.

Acknowledgements

This work fulfills the research objectives of the PON02_00563_3489339 project "PUGLIA@SERVICE - L'Ingegneria dei Servizi Internet-Based per lo sviluppo strutturale di un territorio "intelligente" funded by the Italian Ministry of University and Research (MIUR).

Bibliography

[Chen et al., 2009] Chen W., Wang Y., Yang S., Efficient influence maximization in social networks, in Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2009, 199-207.

[Chen et al., 2010] Chen W., Wang C., Wang Y., Scalable influence maximization for prevalent viral marketing in large-scale social networks, in Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2010, 1029-1038.

[Cheng et al., 2012] Cheng S., Shen H., Huang J., Zhang G., Cheng X., Static greedy: solving the apparent scalability-accuracy dilemma in influence maximization. CoRR abs/1212.4779, 2012.

[Goyal et al., 2011] Goyal A., Lu W., Lakshmanan L.V.S., CELF++: optimizing the greedy algorithm for influence maximization in social networks, in Proceedings of the 20th International Conference on World Wide Web, 2011, 47-48.

[Huang et al., 2012] Huang J., Cheng X.Q., Shen H.W., Zhou T. and Jin X., Exploring social influence via posterior effect of word-of-mouth recommendations, in Proceedings of the 5th ACM International Conference on Web Search and Data Mining, 2012, 573-582.

[Jiang et al., 2011] Jiang Q., Song G., Cong G., Wang Y., Si W., Xie K., Simulated annealing based influence maximization in social networks, in Proceedings of the 25th AAAI Conference on Artificial Intelligence, 2011, 127-132.

[Kempe et al., 2003] Kempe D., Kleinberg J., Tardos E., Maximizing the spread of influence through a social network, in proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, 2003, 137-146.

[Kimura et al., 2010] Kimura M., Saito K., Nakano R., Motoda H., Extracting influential nodes on a social network for information diffusion, Data Mining and Knowledge Discovery, 20(1), 2010, 70-97.

[Leskovec et al., 2007] Leskovec J., Krause A., Guestrin C., Faloutsos C., VanBriesen J., Glance N.S., Cost-effective outbreak detection in networks, in Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2007, 420-429.

[Leskovec et al., 2009] Leskovec J., Lang K., Dasgupta A., Mahoney M., Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters, Internet Mathematics 6(1), 2009, 29-123.

[Nemhauser et al., 1978] Nemhauser G., Wolsey L., Fisher M., An analysis of the approximations for maximizing submodular set functions. Mathematical Programming, 14, 1978, 265–294.

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

[Richardson et al., 2003] Richardson M., Agrawal R., Domingos P., Trust Management for the Semantic Web, The Semantic Web - ISWC 2003, 2003, 351-368.

[Watts, 2002] Watts D.J., A simple model of global cascades on random networks, PNAS 99, 2002, 5766–5771.