MATI: An Efficient Algorithm for Influence Maximization in Social Networks

Maria-Evgenia G. Rossi[1], Bowen Shi[1], Nikolaos Tziortziotis[1],
Fragkiskos D. Malliaros[2], Christos Giatsidis[1], and Michalis Vazirgiannis[1]

[1] École Polytechnique, France
[2] Center for Visual Computing, CentraleSupélec and Inria, France and UC San Diego, USA
E-mail: maria.rossi@polytechnique.edu

1 Introduction

Influence maximization (IM) has attracted a lot of attention due to its numerous applications, including diffusion of social movements, the spread of news, viral marketing and outbreak of diseases. The problem can formally be described as follows: given a *social network* where the relations among users are revealed, a *diffusion model* that simulates how information propagates through the network and a parameter k, the goal is to locate those k users that maximize the spread of influence. Kempe et al. [3] formulated the problem in the aforementioned manner while adopting two diffusion models borrowed from mathematical sociology: the *Linear Threshold* (LT) and the *Independent Cascade* (IC) model. According to both, at any discrete time step a user can be either active or inactive and the information propagates until no more users can be activated.

In this study, we propose MATI, an efficient IM algorithm under both the LT and IC models. By taking advantage of the possible paths that are created in each node's neighborhood, we have designed an algorithm that succeeds in locating the users that can maximize the influence in a social network while also being scalable for large datasets. In order to limit the computation of the possible paths and the respective probabilities of them being "active", we use a pruning threshold θ that reduces the running time but also the accuracy of the influence computation. Extensive experiments show that MATI has competitive performance when compared with the baseline methods both in terms of influence and computation time. Due to space limitations we present only the respective methodology and results for the MATI algorithm under the LT model.

2 MATrix Influence (MATI) algorithm

A social network is typically modeled as a directed graph $G = (V, E)$, consisting of $|V|$ users represented as nodes and $|E|$ edges reflecting the relationship between users. We assume that $\mathscr{T}(u) = \{\tau_1, \tau_2, \ldots, \tau_M\}$ represents the set of all possible paths that exist in the graph starting from node u and leading to "leaf" nodes. Each path τ_i consists of a sequence of nodes: $\tau_i = \{n_{i1}, n_{i2}, \ldots, n_{iN}\}$. Let $p^{\tau}_{\ell,\ell+1}$, $1 \leq \ell \leq N-1$, represent the influence weight (probability) between two successive nodes in path τ. Then $\mathscr{F}(\tau_i) = \{f_{i1}, f_{i2}, \ldots, f_{iN}\}$ represents the probability path for every path τ_i starting from node u to be *a*ctive. Each f_{ij} is equal to $\prod_{\ell=1}^{j-1} p^{\tau_i}_{\ell,\ell+1}$ if $j \geq 1$, and 1 otherwise.

Algorithm 1 MATILT

```
1: Input: G = (V,E), k                          ▷ k: budget (number of seed nodes)
2: Initialize: S = ∅
3: 𝒜, Ω = CALCSTATSLT(G)
4: Q = CALCINF(𝒜, V)                            ▷ Q: CELF queue [4]
5: for i = 1 to k do
6:     s, σ(s) = Q.top()
7:     S = S ∪ s
8:     U = V \ S
9:     for each u ∈ U do
10:        σ(u) = Q(u)
11:        for each v ∈ S do
12:            σ(u) −= Ω(v,u)
13:            σ(u) −= Ω(u,v)
14:        end for
15:        Q.add((u, σ(u)))
16:    end for
17: end for
18: return S
```

The *forward cumulative influence* $\Omega(u,v)$ corresponds to the influence of node u to v and to the nodes that can be found right after v in the paths $\mathscr{T}(u)$ of node u.

Goyal et al. [2] showed that the spread of a set S of nodes is the sum of the spread of each individual node $u \in S$ on the subgraphs induced by the set $V-S+u$:

$$\sigma(S) = \sum_{u \in S} \sigma^{V-S+u}(u), \tag{1}$$

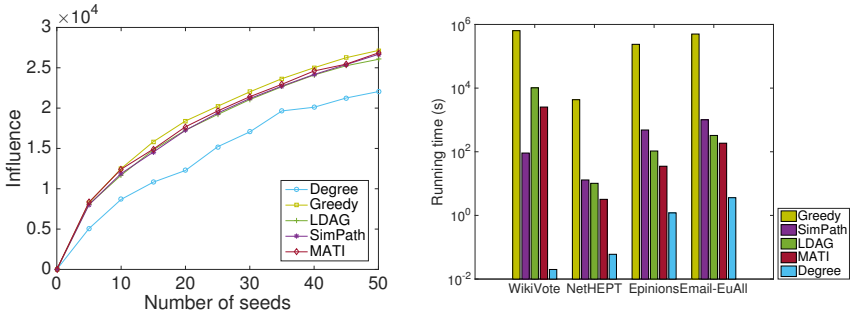
where $\sigma^{V-S+u}(u)$ denotes the total influence of u in the subgraph induced by $V-S+u$. Similar to [2], we write $V-S$ to denote the difference of sets V and S, $V \setminus S$, and $V-S+u$ to denote $((V \setminus S) \cup \{u\})$.

Theorem 1 constitutes the core of the MATI algorithm under the LT model. Actually, it is used for the calculation of the influence gain after the addition of a node x to a set of nodes S.

Theorem 1 *Under the LT model, to calculate the influence after adding a node x to a set of nodes S, one has to subtract from the sum of the individual spread of S and x the forward cumulative influence Ω of all the nodes that belong to set S which contain node x in paths connecting the latter to nodes in set S. That is,*

$$\sigma(S+x) = \sigma(S) + \sigma(x) - \sum_{y \in S} \Omega(x,y) - \sum_{y \in S} \Omega(y,x).$$

Alg. 1 shows the complete structure of MATI algorithm under the LT model. CALCSTATSLT computes $\mathscr{A}$ (i.e., $\mathscr{A}(S,u)$ is the probability the single node u to be activated (influenced) by S) and Ω, and CALCINF returns the influence of all nodes $v \in V$.



(a) (b)

Fig. 1: (a) Influence spread in number of nodes under the LT model for the EPINIONS dataset. (b) Comparison of running times in seconds.

3 Empirical Analysis

We have conducted experiments in real-world datasets in order to evaluate the performance of the MATI algorithm and compare it to state-of-the-art influence maximization algorithms on the quality of results and efficiency. We have used four publicly available graph datasets[3]: NETHEPT, WIKIVOTE, EPINIONS and EMAIL-EUALL and compared the respective results with those of four baseline algorithms: i) *Degree* which considers high-degree nodes as influential [3], ii) *Greedy* for which following the literature [3], we run $10,000$ Monte Carlo (MC) simulations to estimate the spread of any seed set, iii) *LDAG* algorithm using locality properties as proposed in [1] and iv) *SimPath* algorithm proposed in [2]. The threshold θ for the MATI algorithm is set to 0.0001.

The quality of the seed sets obtained by different algorithms is evaluated based on the expected spread measured in number of nodes (Fig. 1a). Due to space limitations we only present the respective results for the EPINIONS dataset. The seed sets obtained via MATI are quite competitive in quality compared to those of the Greedy, LDAG and SimPath algorithms. For all four datasets, the influence loss for up to 50 seeds is less than 2%. Figure 1b reports the execution time required by various algorithms for the LT model. In all cases, MATI is faster than the Greedy and LDAG algorithms. In all datasets except WikiVote, MATI also performs slightly better that SimPath.

References

1. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: ICDM (2010)
2. Goyal, A., Lu, W., Lakshmanan, L.V.S.: Simpath: An efficient algorithm for influence maximization under the linear threshold model. In: ICDM (2011)
3. Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. In: KDD (2003)
4. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: KDD (2007)

[3] https://snap.stanford.edu/data/index.html