Physica A 420 (2015) 124–133

Contents lists available at ScienceDirect

Physica A

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

SMG: Fast scalable greedy algorithm for influence maximization in social networks

Mehdi Heidari *, Masoud Asadpour, Hesham Faili

School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran

HIGHLIGHTS

- SMG changes the way Monte-Carlo simulation is calculated in greedy approach.
- SMG makes huge improvement in the times that nodes of graph should be traversed.
- SMG prevents recreation of graph instances needed for Monte-Carlo simulation.
- SMG brings scalability to any greedy method which uses Monte-Carlo simulation (sub-modular NP-Compelete problems like influence maximization).
- We show improvement of SMG in both time complexity and experiments.

ARTICLE INFO

Article history:
Received 8 September 2014
Available online 7 November 2014

Keywords:
Social networks
Influence maximization
Scalable greedy
State machine Monte-Carlo
Propagation model
Viral marketing

ABSTRACT

Influence maximization is the problem of finding k most influential nodes in a social network. Many works have been done in two different categories, greedy approaches and heuristic approaches. The greedy approaches have better influence spread, but lower scalability on large networks. The heuristic approaches are scalable and fast but not for all type of networks. Improving the scalability of greedy approach is still an open and hot issue. In this work we present a fast greedy algorithm called State Machine Greedy that improves the existing algorithms by reducing calculations in two parts: (1) counting the traversing nodes in estimate propagation procedure, (2) Monte-Carlo graph construction in simulation of diffusion. The results show that our method makes a huge improvement in the speed over the existing greedy approaches.

1. Introduction

The goal of the influence maximization problem is to find k nodes in a social network which, activating them, results in activation of the maximum number of individuals in the network. This problem has many applications especially in marketing, advertisement and elections. For example, in viral marketing, the goal is to maximize the number of individuals influenced via face to face advertisements of a particular product, in this case, k free samples of the product is given to k individuals hoping that they try it and upon satisfaction start recommending the product to their friends. If the product has rather better quality compared to its competitors it is expected that the recommendation circulate among friends and friends of friends, etc., and those people get convinced to buy it. The number of seed set, k, is limited to the budget considered for advertisement.

To solve the influence maximization problem in a network, we should have the influence cascade model, i.e. the way recommendations or influences are propagated from the receivers of free samples to their friends, and from friends to friends of friends, etc. The work by Kempe, Kleinberg and Tardos [1] provided the first formulation of influence maximization as an

* Corresponding author.
E-mail addresses: heidari_mehdi@ut.ac.ir (M. Heidari), asadpour@ut.ac.ir (M. Asadpour), hfaili@ut.ac.ir (H. Faili).

http://dx.doi.org/10.1016/j.physa.2014.10.088
0378-4371/

optimization problem. Their work is based on two basic influence cascade models: the Independent Cascade (IC) model and the Linear Threshold (LT) model.

In both models, a social network is modeled as a weighted directed graph $G = (V; E)$, where the vertices of V represent individuals and edges in E represent relationships, and the orientations of the edges indicate the direction of influence. In the IC model each edge has an activation probability and influence is propagated by activated nodes independently; these nodes activate their inactive neighbors based on the activation probabilities of their edge. In the LT model, in addition to the weights on edges, each vertex has a threshold too, and a vertex will be activated if the sum of weights that it receives from its active neighbors exceeds its threshold.

2. Previous works

Domingos and Richardson [2] studied influence propagation as an algorithmic problem to optimize marketing decisions. They studied movie customers network and proposed a probabilistic model. They modeled customers network as a graph and used a Markov random field to calculate influence propagation among them. So they triggered algorithmic study of influence maximization.

Kemp et al. [1] in 2003 formulated the influence maximization as an optimization problem. They proved NP-hardness and sub-modularity of influence maximization under two presented models in their work. They used the greedy hill climbing algorithm as an approximate solution to this problem. The greedy approximation had already been proved that is $(1 - 1/e)$ approximation on sub-modular functions [3]. Their algorithm adds k nodes to answer vector S in k steps of greedy choice. In each step of seed selection the algorithm calls a propagation estimator function (F) for each node to calculate their marginal gain by Monte-Carlo simulation. After the marginal gain estimation, the node with maximum marginal gain is chosen and added to seed set S. In practice, the algorithm is too slow and is not scalable to large networks.

Leskovec et al. [4] proposed a lazy forward version of the greedy algorithm. Their algorithm try to reduce the number of marginal gain calculations when it is not needed. Adding the first node to seed set needs calling propagation estimator function for all n nodes of graph. But in the second seed-selection step, since marginal gains of the previous step is available and also since, according to sub-modularity, the marginal gain of a particular node does not increase in the next step, the next seed element could approximately be found without re-calculating marginal gain for the $n - 1$ remaining nodes. So, to improve the speed, this method targets the reduction in the number of calls to propagation estimator function F. Of course, there is no guarantee how much it could be reduced.

Chen et al. [5] improved the speed of greedy algorithm on Independent Cascade (IC) model. Since in the IC model each pair of vertices only coexists in a small portion of the graphs, there is no need to increase the number of simulations to compensate the correlation effect, so an improvement can be gained by reducing the number of simulations. Although this considerably improves the running time, but it only works for the IC model and not in general.

Chen [6] changed the way simulation is calculated in the greedy algorithm by using a threshold parameter to gain a local tree around the node which is being evaluated. So simulation could be performed locally. As the value of the threshold reduces, the local simulation result gets closer to global simulation. So this method is only an approximation of the greedy algorithm but not as exact as them.

Goyal [7] presented an algorithm called CELF++. This algorithm tries to reduce the recalculation of propagation estimator function by using a data structure which stores next step marginal gain for a particular node. But it is highly probable that the stored marginal gain has not been calculated based on the last selected seed. So, in this case the stored marginal gain does not help and should be calculated again in the next step. CELF++ could not considerably reduce recalculations of propagation. On the methodology section we describe how our algorithm reduces these recalculations.

There are many works that have targeted scalability by using heuristic methods. But, there is no approximation guarantee available for them. Although they are not as dependable as the greedy methods however, they are very fast. Most of them use graph centrality measures and assume those measures are proportional to the amount of influence [5,8–11]. Some heuristic methods work wiser and consider a discount to handle sub-modularity too [5,8]. The discount could be the probability of a node being activated by the seed set [6,8] or another measure which has a direct relation with that probability [5]. Out work is focused on the greedy approach and the heuristic approaches are not related to our work so we do not go into detail in this paper.

In the next section we present our method that improves the speed of greedy algorithms from 2 aspects: (1) it prevents recalculations of the propagation estimator function. Note that this aspect was targeted by CELF++ but it could not efficiently prevent recalculations and just had a small improvement. (2) It minimizes the amount of Monte-Carlo graph constructions. It should be mentioned that our method could be incorporated into all previous greedy methods.

3. Our proposed method

3.1. The general greedy algorithm

Algorithm 1 shows the general greedy algorithm. The outer loop iterates k steps. In each iteration the most influential node is found and added to seed set S. Finding the most influential node, needs the inner loop which iterates over candidate