Threshold Estimation Models for Influence Maximization in Social Network

Ashis Talukder, Md. Golam Rabiul Alam, Anupam Kumar Bairagi, Sarder Fakrul Abedin,

Hoang T. Nguyen, and Choong Seon Hong

Department of Computer Engineering, Kyung Hee University.

Email: {ashis, robi, anupam, saab0015, nguyenth, cshong}@khu.ac.kr

Abstract

The Influence Maximization (IM) problem determines a small set of influential users that maximizes the influence spread in the network where influence is measured by the number of active nodes. Among the two classical models, in the Linear Threshold (LT) a node is activated if the total influence of all the active in-neighbors is no less than a given threshold and thus threshold selection is important. In this research we observe that the threshold depends on the application of IM and the influence weight. Then we propose different threshold models based on influence weight. Our models are linear and the simulation on real dataset shows that they have fast running time.

1. INTRODUCTION

The Influence Maximization (IM) problem has got great attention in recent years and the pioneering works include two classical models: Independent Cascade (IC) and Linear Threshold (LT) models [1]. Generally a social network, presented by a directed graph $G(V, E)$ and a seed S set of size K are given. An activated node influences neighboring nodes to adopt any product or service. The influence weight w_{uv} indicates the influence probability of user u upon v and is calculated from the given input graph G. The influence spread $\sigma(S)$ is estimated by the number of activated nodes when all the nodes in S are activated. In the LT model, a node v is activated if the aggregated influence of its activated in-neighbors is no less than a given threshold Θ_v of the node v, that is, if

$$\sum_{u \in n^{-1}(v)} w_{uv} * x_u \geq \theta_v, \qquad (1)$$

where x_u refers to whether an in-neighbor u is active ($x_u = 1$) or not ($x_u = 0$). Then the IM problem under LT model is to find a seed set S of given size K for which the influence spread $\sigma(S)$ is maximized and is defined by [1]:

$$S = \underset{|S|=K}{arg\,max}\ \sigma(S) \qquad (2)$$

It is evident from equation (1) and (2) that node activation depends on the threshold value Θ_v and influence weight w_{uv}. Thus threshold analysis has research significance. We categorize threshold in fixed and variable categories and propose linear threshold estimation models in both categories. Finally we evaluate the proposed algorithms with a real dataset.

2. RELATED WORKS

It has been seen in the most of the influence maximization researches that the authors have used a range of threshold (*e.g.* $\Theta_v = 0.1$ to 0.5) and sometimes some fixed values (e.g. $\Theta = 0.5$) for the simulation of their algorithms without proper explanation of choosing the value(s). In this research we first state an extensive survey on choosing thresholds and then propose multiple threshold estimation models.

A. *Fixed Threshold*

In fixed threshold model, all the nodes v have the same fixed value as threshold Θ. Many authors have set different fixed values. For instance:

- $\Theta = 0.5$ in [2] for contagion. It is well explained in [3].
- $\Theta = 1/320$ in [4] for the algorithm LDAG
- Θ = 1/1000 by Goyal et al. for their SIMPATH algorithm in [5].
- Θ = 0.35 in [6].
- $\Theta = 1/\lambda$ for epidemic threshold in [7] where λ is the largest eigenvalue of the adjacency matrix of the network graph.

B. *Variable Thresholds*

On the other hand, there are different thresholds Θ for each node v of the network. For example:

- $\Theta_v \sim U(0, 1)$ in [1], [2], [8], [9].
- $\Theta_v \sim U(0.03,0.05)$ for IMLT-IOA algorithm in [10].
- $\Theta_v \sim U(0.1,0.5)$ for the algorithm CDH_SHRINK and CDH_Kcut in [6]
- $\Theta_v \sim (1/180, 1/320)$ for LDAG algorithms in[4]

3. PROPOSED THRESHOLD MODELS

A. *Fixed Threshold Model*

This models are quite straightforward and simple yet useful and has been applied by many authors such that [2], [3], [4], [5], [6], [7] etc. as discussed in the section two. Here we propose some fixed threshold models.

1) *Heuristic Average (HA) Model*

In this model threshold value Θ is determined by taking the average of influence values of randomly selected in-neighbors of all the nodes v in the graph G. Firstly, the marginal aggregated influence is derived for all nodes v by:

$$\theta_v^m = \sum_{u \in n^{-1}(v)} w_{uv} * x_u \qquad (3)$$

where m nodes are randomly selected from $n^{-1}(v)$ and for a chosen node u, $x_u = 1$ and for the rest nodes, $x_u = 0$. Then the fixed threshold is finally estimated by taking the expected value as follows:

$$\theta = \mathbb{E}\left[\theta_v^m\right] = \frac{\sum_{v \in \mathcal{V}} \sum_{u \in n^{-1}(v)} w_{uv} * x_u}{|n^{-1}(v)|} \qquad (4)$$

The associated HA model is stated in the *Algorithm 1*.

Algorithm 1: Heuristic Average (HA) Model

Input: $G(V, E), w_{uv}$
Result: θ
for *each* $v \in V$ **do**
  Calculate the set $n^{-1}(v)$;
  V' = Randomly select m nodes from $n^{-1}(v)$;
  $sum = 0.0$;
  for *each* $u \in V'$ **do**
    $sign$ = Randomly select one binary sign $(1, -1)$;
    $sum = sum + w_{uv} + sign \times w_{uv} \times 0.05$;
  end
end
$\theta = \frac{sum}{N}$;
return θ;

2) Sample Average (SA) Model

The HA Model takes the whole population into account and hence the model is a slow process. In order to derive threshold faster, a sampling can be helpful. We employ *Systematic sampling* technique [11]. First we determine the sample size n using Slovin's formula [12]. The formula in the equation (5) with $N = 1000$ and error tolerance $e = 5\% = 0.05$ gives,

$$n = \frac{N}{1 + Ne^2} = \lceil 285.714 \rceil \approx 286 \qquad (5)$$

Algorithm 2: Sample Average (SA) Model

Input: $G(V, E), w_{uv}$
Result: θ
$n = \frac{N}{1+Ne^2}$; /* Sample size */
$incr = \lceil \frac{N}{n} \rceil$; /* Step size of the loop */
$inf = 0.0$;
for $v = 1$ *to* N *step by* $incr$ **do**
  Calculate the set $n^{-1}(v)$;
  V' = Randomly select $m \leq |n^{-1}(v)|$ nodes from $n^{-1}(v)$;
  $inf\_sum = 0.0$;
  for *each* $u \in V'$ **do**
    $sign$ = Randomly select one binary sign $(1, -1)$;
    $inf\_sum = inf\_sum + w_{uv} + sign \times w_{uv} \times 0.05$;
  end
  $\theta_v = inf\_sum$;
  $inf = inf + inf\_sum$;
end
$\theta = inf/n$;
return θ;

Now the *step* = $N/n = 1000/284 \approx 4$. It means that the nodes *1, 5, 9....* will be selected for sampling. The Slovin's formula has great importance when there is no prior idea about the population except the size. Then the m number of in-neighbors are selected from each set $n^{-1}(v)$ where m is also chosen randomly in the range $(1, |n^{-1}(v)|)$. The influence weights w_{uv} of m nodes are then summed up along with *5%* adjustment for each. The SA model is given in the *Algorithm 2*.

B. Variable Thresholds Model

In this model, the threshold is different for each nodes. Actually this model mirrors the real world scenario mode vividly. Since it is expected that different individual has different influence or motivation level.

1) Heuristic Individual (HI) Model

This HI model is nearly same as HA Fixed model expect in the HA model, an average of all N nodes is taken. The model is presented in the *Algorithm 3*.

Algorithm 3: Heuristic Individual (HI) Model

Input: $G(V, E), w_{uv}$
Result: θ
for *each* $v \in V$ **do**
  Calculate the set $n^{-1}(v)$;
  V' = Randomly select m nodes from $n^{-1}(v)$;
  $inf\_sum = 0.0$;
  for *each* $u \in V'$ **do**
    $sign$ = Randomly select one binary sign $(1, -1)$;
    $inf\_sum = inf\_sum + w_{uv} + sign \times w_{uv} \times 0.05$;
  end
  $\theta_v = inf\_sum$;
end

4. EVALUATION

We have simulated the algorithm by a series of Python programs on a machine: Intel(R) Core(TM) i3-4150 CPU @ 3.50 GHz 3.50 GHz machine with 8GB RAM. We have used real dataset named Epinions [13], scaled (first 5,000 nodes and their associated edges) for Monte Carlo (MC) simulation.

Networks	Epinions
Nodes	5,000
Edges	180,493

Table I: *Epinions Dataset*

The *Figure 1* illustrates the comparison of the average threshold calculated by the HA and SA models for 50 to 500 times MC simulations. Here the models show generalized results of the models in [4], [6], and [10].

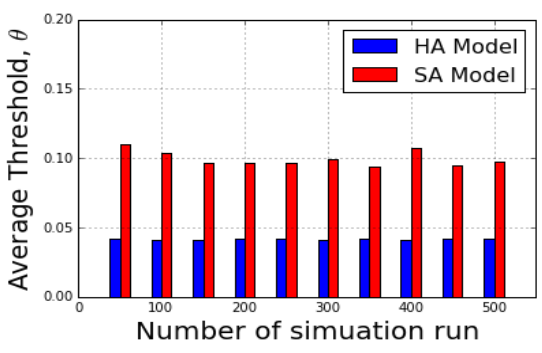

Figure 1: *Calculated thresholds by HA and SA models.*

The threshold value calculation is a sub-problem in the IM problem. Thus it should not take much time. This requirement is achieved and illustrated in the *Figure 2* which shows faster running time of our models. The HA and HI model takes nearly same time as both consider *N* nodes, whereas SA model takes very lesser time than both the rest models due to sampling.

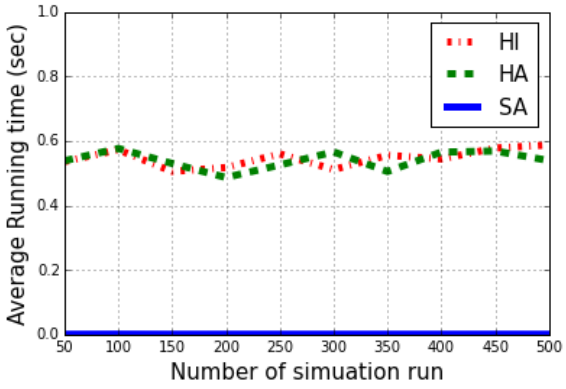

Figure 2: *Running time of HA, SA, and HI models*

5. CONCLUSION

In this research we first have an extensive survey on selecting threshold values for the LT model-based IM problem and we have found that node activation depends on the influence weight. Depending upon application of IM, the threshold may be different as well. We have also categorized the thresholds in two classes: Fixed and variable. We have proposed threshold estimation models covering both the classes. Finally we have evaluated our models with a real dataset. The simulation result suffices the faster running time.

Acknowledgement: *This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0190-16-2013, Development of Access Technology Agnostic Next-Generation Networking Technology for Wired-Wireless Converged Networks) *Dr. CS Hong is the corresponding author.*

6. REFERENCES

[1]. D. Kempe, J. Kleinberg, and E. Tardos., *"Maximizing the spread of influence through a social network,"* In Proc. of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003.

[2]. Wortman, J., 2008. *"Viral marketing and the diffusion of trends on social networks."*

[3]. Morris, S., 2000. *"Contagion," The Review of Economic Studies*, *67*(1), pp.57-78.

[4]. Chen, W., Wang, C. and Wang, Y., 2010, July. *"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* (pp. 1029-1038). ACM.

[5]. Goyal, Amit, Wei Lu, and Laks VS Lakshmanan. *"SIMPATH: An efficient algorithm for influence maximization under the linear threshold model,"* In Data Mining (ICDM), 2011 IEEE 11th International Conference on, pp. 211-220. IEEE, 2011.

[6]. Chen, Y., Chang, S., Chou, C., Peng, W. and Lee, S., 2012, August. "E*xploring community structures for influence maximization in social networks,"* In *Proceedings of the 6th SNA-KDD Workshop on Social Network Mining and Analysis held in conjunction with KDD'12 (SNA-KDD'12)* (pp. 1-6).

[7]. Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J. and Faloutsos, C., 2008. *"Epidemic thresholds in real networks," ACM Transactions on Information and System Security (TISSEC)*, *10*(4), p.1.

[8]. Barbieri, N. and Bonchi, F., 2014. *"Influence Maximization with Viral Product Design,"* In *SDM* (pp. 55-63).

[9]. Mossel, E. and Roch, S., 2007, June. *"On the submodularity of influence in social networks,"* In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing* (pp. 128-134). ACM.

[10]. Li, S., Zhu, Y., Li, D., Kim, D., Ma, H. and Huang, H., 2014, June. *"Influence maximization in social networks with user attitude modification,"* In *2014 IEEE International Conference on Communications (ICC)* (pp. 3913-3918). IEEE.

[11]. Malhotra, N.K., 2008. "*Marketing research: An applied orientation," 5/e*. Pearson Education India.

[12]. Slovin, E., 1960. *"Slovin's formula for sampling technique,"* Retrieved on February, 13, p.2016.

[13]. Richardson, M., Agrawal, R. and Domingos, P., 2003, October. *"Trust management for the semantic web,"* In *International semantic Web conference* (pp. 351-368). Springer Berlin Heidelberg.