World Engineering & Applied Sciences Journal 8 (1): 18-24, 2017
ISSN 2079-2204

DOI: 10.5829/idosi.weasj.2017.18.24

Community Aware Influence Maximization on Large Scale Networks Using Mapreduce

M. Nivethitha and Dojohn Loyd B

SRM University, Ramapuram, Chennai, India

Abstract: Influence maximization problem is a well known problem to find the top-k seed users who can maximize the spread of information in a social network. The primary concern is monte carlo simulations method is suffering with scalability issues while the selection of seed users .It takes days to find potential seed users in large datasets. In this paper, we propose a highly scalable algorithm for identifying Influential nodes on large-scale graph using the Map Reduce framework. We perform a combined community detection algorithm and consider the most influential users in the community as the candidate for the top-k seeds. This approximation allows us to divide the whole graph into multiple sub graphs that can be processed independently. Then, for each sub graph, a Map Reduce based algorithm is designed to identify the minimum-sized influential vertices for the whole graph. This original approach contrasts with previous influence propagation models, which did not use similarity opportunities among members of communities to maximize influence propagation. The performance results show that the model activates a higher number of overall nodes in contemporary social networks, as compared to existing landmark approaches.

Key words: Mapreduce · Community detection · Influence maximization · Influence probagation

INTRODUCTION

Recently the amount of spread of information is steadily increased in online social networks such as Facebook and Twitter. To use online social networks as a marketing platform, there are lots of researches on how to use the spread of influence for viral marketing. Viral marketing is a marketing methodology that uses the word-of mouth effect among the SNS (Social Network Services) users to perform advertisement about a specific product. For example, Hotmail included an advertisement phrase saying "Get your private, free email at http://www.hotmail.com" in the e-mail of Hotmail users. With such advertisement, Hotmail could naturally spread the advertisement among the pre constructed e-mail network. As the result of the viral marketing, Hotmail has gathered 1,200 users in 2 years [1]. Nowadays, social commerce companies such as groupon4 use viral marketing to attract customers. Viral marketing aims to achieve maximum advertisement effect within a given budget. For efficient marketing outcome, it is needed to carefully select the seed users who will initiate the marketing process.

To resolve the problem, Influence Maximization Problem aims to find the k people, who will maximize the marketing outcome when selected as seed users. Many researches until recently have proposed numerous algorithms for solving the Influence Maximization Problem, either by improving the greedy algorithm or proposing new heuristics. However, there are short comings in both approaches that the greedy algorithms' runtime are too large and the heuristics do not take the prominent community structures of the social network. Our contributions in this paper are in threefold.

- Detecting communities in online social networking.
- Finding key nodes within each community (resulting from Step 1)
- Constructing the seed set (made up of top users generated from Step 2) which is used to spread influence over an online social network.

Problem Definition

Social Network Graph: The social network is represented as a weighted directed graph where nodes represent members of the social network and the edges represent relationships or interactions among them. A weighted

Corresponding Author: M. Nivethitha, (CSE), SRM University, Ramapuram, Chennai, India.

directed graph $G = (V, E)$ is comprised of tuples between the set of nodes, V , and the set of edges, E. An edge $e \in E$ can be represented as a pair of two nodes $u, v \in V$ $e = (u, v)$ and the direction from u to v. $e = (u, v)$ has cu, v, the number of interaction between two nodes as weight. The set of neighbors of $u \in V$, $NG(u)$, is defined as follows.

$$N_G(u) = \{v \text{ in } V \mid £(u,v) \in E\}$$

The weight of each edge is normalized by dividing each edge weights by the sum of weights.

Information Spread Model: There are numerous information diffusion models to simulate the spread of information among a network, discuss the "word-of-mouth effect" in the real world. Two of the most basic and widely-studied models will be considered in this paper. Firstly, [2-15] proposes the Independent Cascade(IC) Model. Each edge in the social network has same probability to influence the target node. The information diffusion under the IC model is simulated as follows.

Definition 1: Every node can be either active or inactive. An active node represents an influenced user in the social network. The seed set of active nodes is defined as A0. Newly activated nodes in the ith iteration are defined as Ai. In the i + 1th iteration, a node u in Ai tries to activate its inactive neighbour v with probability of $P_{u,v}$. When u successfully influences v, v is added to A_{i+1} and becomes active. Such iteration is repeated until $A_{i+1} = \epsilon$.

The probability of u influencing v, $P_{u,v}$. is defined as follows.

$$P_{u,v} = 1 - (1 - P)^{c}_{u,v}$$

p in the above formula represents the propagation probability, which is the probability of u influencing v with one interaction.In the IC Model, nodes with high degree have high probability both to influence its neighbor and to be influenced by them. But in some application, nodes with high degree can be less influenced by its neighbor. For example, a person with 100 friends is not easily influenced by one of his friends. However, a person with only one friend can be easily influenced by his only friend. With such intuition, [11] proposed the Weighted Cascade(WC) Model.

Definition 2: In the WC model, the probability of u influencing v, pu, v, is defined as follows.

$$P_{u,v} = \frac{C_{u,v}}{\sum_i \in N_G(v) C_{i,v}}$$

Influence Maximization Problem: Domingos and Richardson [7] were the first to define the Influence Maximization Problem as a probablistic algorithm problem. Kempe [11] defined the Influence Maximization Problem as an optimization problem, and proved thatsuch problem is NP-Hard under the IC Model and the WC Model. The Influence Maximization Problem defined by Kempe [11] is as follows.

Definition 3: *Given a graph $G = (V,E)$ and the weights for each $(u, v) \in E$ representing the probability of u influencing v, the Influence Maximization Problem finds a set of nodes $S \in V$ that maximizes the influence function $f(S)$ and $|S| = k$.*

Algorithms

Algorithm 1: (Common Actions Algorithm)
we proposed this method where we calculate the similarity between two nodes based on the common actions they have. The Jaccard Coefficient measures the commonly active properties of nodes u and v to the number of active properties in u or v. The formula used is

$$JC_{u,v} = \frac{A_{u,v}}{A_u + A_v - A_{u,v}}$$

where A_u is the number of actions performed by node u, A_v is the number of actions performed by node v and $A_{u,v}$ is the number of common actions performed by nodes u and v. Algorithm shows the steps of calculating the common actions between two nodes u and v.

1.Find all $actions_u$;
2.Find all $actions_v$;
3.For $\epsilon a \in actions_u$ do
(a) if a is in $actions_v$ AND $time_a < time_v$do
i. $common_{u,v} = common_{u,v} + 1$;

$$JC_{u,v} = \frac{common_{u,v}}{actions_u + actions_v - commonu,_v}$$

return $JC_{u,v}$;

Degree Centrality:

- CentralityWeight = nodeDegree/totalEdges

where *nodeDegree* is a variable that stores each node's in and out degree, *totalEdges* is the total number of edges in the network. This formula calculates the centrality

weight of each node's in and out degree based on the total number of edges in the network. The resulting values vary between 0 to 1, which reflect the centrality level of each node in the network. Whenever the result is approaching 1, this means the node is probably in a very central location. We will use this value later to make a decision about whether the node could be a key user.

In fluence Weights Avg = sum Influence Weights (node)/ total Nodes

Consider a user A who started a behavior at time T1, and after a while another user B adopts the same behavior at time T2. This means B is influenced by A after a certain time. For the Jaccard Coefficient Based on Common Actions technique, we find out the number of similar actions a user has embraced after his friend had adopted the same behavior. We assume there is only one source for each action.

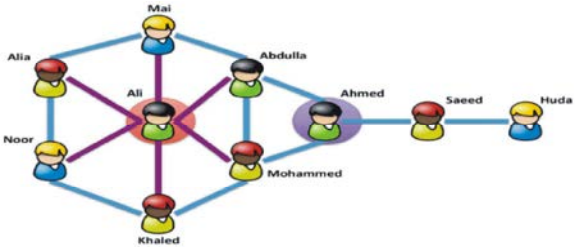
Where *sumInfluenceWeights(node)* is the sum of all influences weights that a specific node has on every node in the network; *totalNodes* is the total number of nodes in the social network.

$Intersection_n$ = min (Centrality $Weight_n$, Influence $Weights\ Avg_n$)

Intersection determines the lowest of the two values between the centrality weight of a node (*Centrality Weight*) and the average weight of influence for that node (*Influence Weight Avg*). Then, we select the maximum membership grade generated from the above intersection process to decide on key users.

$MembershipGrademax = \max (Intersection_n)\ \forall n \in CN$

CN Is the Set of Central Nodes in the Social Network: In Figure shown below node Ali has the highest degree centrality, because it is the node with the highest number of ties or edges. This means he is quite active in the network. However, he is not necessarily the most influential person because he is only directly connected within one degree to people in his group. He has to go through Ahmed to get to other connections.



Algorithm 2 (Similarity Algorithm):

1. Start from single nodes and the original social network.
2. Start generating the similarity social network from the original social network.

(a) For each node a and b do:

i. Calculate similarity based on Equation.

$$Similarity\ (a,b) = \frac{adj_{ab} + cn_{ab}}{n_a + n_b}$$

where adj_{ab} which represents the intersection of row a and column b in the adjacency matrix, is equal to 1 if there is an edge between nodes a and b and 0 otherwise, cn_{ab} are the number of common neighbors of nodes a and b, n_a and n_b are the total neighbors of nodes a and b respectively.

ii. For each node a find the highest similar node b.

- Establish a link between nodes a and b.

iii. Calculate the modularity (Q).

$$Q = \sum_i e_{ii} - a_i^2$$

where e_{ij} is the fraction of edges that connect vertices in group i to vertices in group j and $a_i = \Sigma\ e_{ij}$. Modularity is based on finding the difference between the number of edges within the communities and the expected number of edges (edges are generated randomly).

In this algorithm, nodes are grouped together to create a virtual social network based on high similarity. If two nodes a and b are highly similar, then a link is virtually established between them. The synthetic link is virtually established in the synthetic network, but no actual link is created in the original social network [10].

Algorithm 3: (Community Based Influence Propagation Algorithm):

1. Detect Communities C of (G) (Similarity Algorithm).
2. Threshold-IU ← Number of Important Users.
3. Threshold-S ← Number of Users in the Seed Set (S).
4. For each community C do :

(a) Find Central Users Fuzzy Set

i. CentralUsers ← Find Central Users (Alg: Degree Centrality)
ii. For each CentralUsers do
A. CentralityWeight = nodeDegree/totalEdges//Membership Function
(b) Find Influence Weight Fuzzy Set
i. For each CentralUsers do
A. InfluenceWeight ←
Calculate Influence Weight (Alg:Jaccard Common ActionsAlgorithm)
ii. For each CentralUsers do
A. Inf luenceWeightsAvg=
sumInf luenceWeights(node)/totalNodes
//Membership Function To Calculate Average InfluenceWeight
(c) Find The Important Users (Fuzzy Decision)
i. Intersection =
min(CentralityWeight, Inf luenceWeightsAvg)
ii. ImportantUsers ← Select The Maximum Intersection Grade To Decide The Important Users
5. For each ImportantUsers do
(a) Reachability ← Apply Influence Propagation Method (Alg: Influence Probagation Alg)
6. S←Select The Top Influential Users Based on Threshold-S
7. Return S

This Algorithm Consists of Three Main Steps

Detecting Communities (Step 1): This step determines the success or failure of the adoption of behaviors in the social network, because of the similarity factor within communities that is used for influence propagation. Well structured communities will facilitate a better dissemination of influence. Therefore, the role of this initial community detection step in the influence propagation algorithm is important. Similar users tend to adopt similar behaviors, and detecting these communities in the network will make it easier to find the key nodes in each community to form the seed set for the whole network.

Identifying Key Users in Each Community (Step 4): After dividing the network virtually into groups of similar communities, we then identify the initial set of key users in each community. These users are potential candidates of the final seed set in the network. There are two main characteristics of a key user: 1) location in the network (centrality), and 2) historical influence activity. Indeed, multi-criteria decision making problems consist of (1) a finite set of criteria (or properties) that evaluate the quality of a key user to join the seed set of users used in our influence propagation model, and (2) weights (or importances) of the criteria [12]. These decision-making components map to our problem to identify key users where the criteria are represented by centrality and influence power, and weights are represented by the level associated with these criteria. Based on these criteria and weights, a fuzzy logic process is employed to find key users, and hence the seed set of users who will drive influence propagation.

Central nodes have high connections with other users. Also nodes that reside between two groups play the role of a bridge to convey behavior from one group to another, and so they would be good key user candidates. We used a degree centrality measure to determine users with favorable locations. The process starts by calculating the in-degree and out-degree of each node then summing up these values. Nodes that have higher in and out degree than the threshold are considered central in the social network.

After calculating the influence weights, we propose to calculate the average influence weight to discriminate nodes with the highest historical influence activity in the network.

3) Finding the seed set to propagate influence across the entire social network. (Steps 5 and 6).

E. Algorithm 4: (Influence Propagation Algorithm)

1. For $\forall u \in$ KeyUsers do
(a) At step t = 0,
activate $u \in$ KeyUsers and added it to $Coverage_0$
(b) At each step t > 0, For $\forall u \in Coverage_{t-1}$ do
i. For $\forall v_{inactive}$ if $InfluenceWeight_{u,v} \geq$ InfluenceThreshold
A. Activate v
B. ActiveList = ActiveList $\cup \{v\}$

C. TotalCoverage = TotalCoverage+1

ii. All the nodes activated at this step are added to $Coverage_t$

iii. This process ends at a step t if $Coverage_t = 0$ /*no more nodes to activate*/

2. Add nodes u with highest TotalCoverage to S

3. Return S

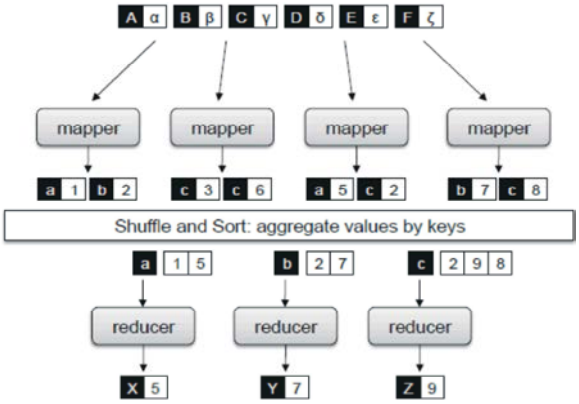
After finding the key users in each community, we select the top ones which have highest influence propagation in the social network to form the seed set of nodes that are used to diffuse influence across the network. This algorithm shows our influence propagation method which is based on an independent cascade propagation model.

Our algorithm takes into consideration the influence power that a node might have on other nodes indirectly. This means a node might activate nodes other than its direct neighbors. This is inferred through historical common actions as a method to estimate weights probabilities. The complexity of this approach is similar to the IC model since the propagation process uses the same steps but the nodes selected for propagating the influence are different in our approach from the IC one.

Algorithm 5: (MapReduce Algorithm): The mappers emit distances to reachable nodes, while the reducers select the minimum of those distances for each destination node. Each iteration (one MapReduce job) of the algorithm expands the \search frontier" by one hop.

```
1: class Mapper
2: method Map(nid n; node N)
3: d ¬N:Distance
4: Emit(nid n;N) .              // Pass along graph structure
5: for all nodeid m ∈ N.AdjacencyList do
6: Emit(nid m; d + 1) .  //Emit distances to reachable nodes

1: class Reducer
2: method Reduce(nid m; [d1; d2; : : :])
3: dmin ¬ 8
4: M ¬ ∈ ;
5: for all d  ∈ counts [d1; d2; : : :] do
6:   if IsNode(d) then
7:      M ¬d .                   //Recover graph structure
8:   else if d < dmin then .   //Look for shorter distance
9:     dmin ¬d
10: M:Distance ¬dmin . //Update Number Of Important
                                Users
11: Emit(nid m; node M)
```



RESULTS AND DISCUSSION

As an experimental platform, we used Flickr real-world social network dataset. Flickr is a photo sharing social network. On Flickr users can share and embed photographs in their own blogs. We used the dataset provided in [15], which consists of 2,570,535 nodes and 33,140,018 links between the nodes. Due to computational constraints and as part of our preliminary experiments, we randomly selected 500 nodes, to run our experiments. We are planning to increase the size of the sample data set in the future to run further experiments. Community Aware Influence Probagation Algorithm was implemented using Matlab and C++. The experiments were performed on an Apple iMac with Mac OS X version 10.6.8, processor 2.66GHz intel Core i5 and 4GB memory.

Figure 1 shows the results generated by both candidate algorithms when the seed set maximally contains 5 nodes. Based on the original propagation of the IC model, the 5 seed set nodes will activate 33 nodes in the social network. While in our approach, the same number of seed set nodes activates 134 nodes. This shows that by using the social relations that are available in the network, more additional nodes are reached by the influence propagation process. In doing so, the initial nodes are more successful in persuading neighbors or neighbors-of-neighbors to adopt the propagated behavior.

In another experiment, we increased the size of the seed set to 30 nodes. Using the IC model, 30 nodes in the seed set activate 98 nodes in the social network as shown in Figure 2. On the other hand, in this experiment our approach first discovers the top 6 nodes out of the 30 nodes and then selects them as potential candidates for the seed set. The 6 nodes activate 135 nodes in the social network.

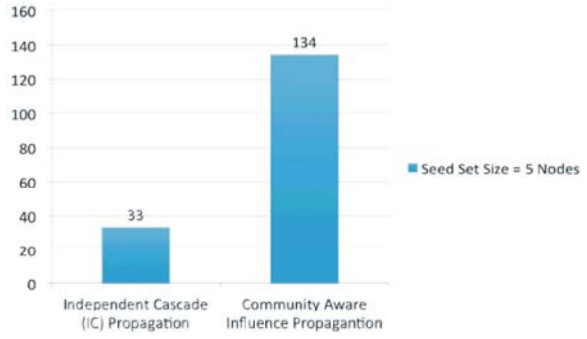


Fig. 1: Activated Nodes When Seed Set = 5

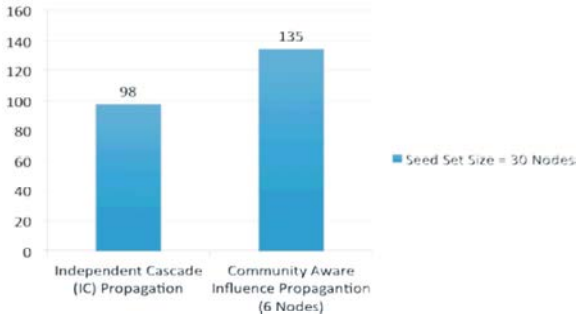


Fig. 2: Activated Nodes When Seed Set = 30)

We noticed more interesting results by changing the threshold Influence Threshold of similarity probability used as weight of influence propagation. When the threshold decreases, our approach reveals that 1 node can activate 144 other nodes in the network. By decreasing this threshold, we increase the number of nodes that are similar to the initial nodes. This means instead of finding 30 initial nodes we can find only 1 node which has connections that can activate about 29% of the total nodes in our sample social network as seen in Figure 2.

CONCLUSION

We designed and implemented an efficient community-detection solution for large social networks using MapReduce. A set of algorithms proposed to significantly save running time and storage space while improve the detection accuracy. We use a similarity metric to achieve low convergence time and increase robustness in practice. Experimental results demonstrate that our solution achieves one order of magnitude improvement in both processing time and storage space. Furthermore, our experiments indicate a great scaling characteristic with more machines in a Map Reduce cluster. Our evaluation shows that the input parameters of our detection solution are robust for real-world social network data.

REFERENCES

1. Pasupathy R., S.H. Kim, A. Tolk, R. Hill and M. E. Kuhl, 1696. A Simulation-based Approach to Analyze the Information Diffusion in Microblogging Online Social Network, In Proceedings of the 2013 Winter Simulation Conference, edited by 1685-Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
2. Cosley, D., D.P. Huttenlocher, J.M. Kleinberg, X. LAN and S. Suri, 2010. Sequential influence models in social networks, In Proc. 4th International Conference on Weblogs and Social Media.
3. Kempe, D., J. Kleinberg and É. Tardos, 2003. Maximizing the spread of influence through a social network, in Proceedings of 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '03. New York, NY, USA: ACM, pp: 137-146.
4. Dean, J. and S. Ghemawat, 2008. MapReduce: Simplified Data Processing on Large Clusters, Communications of the ACM, 51(1): 107-113.
5. Sevilla, M., I. Nassi, K. Ioannidou, S. Brandt and C. Maltzahn, 2013. A Framework for an In-depth Comparison of Scale-up and Scale-out, In Proceedings of the 2013 International Workshop on Data- Intensive Scalable Computing Systems, pp: 13-18.
6. Nguyen, N.P., T.N. Dinh, Y. Xuan and M.T. Thai, 2011. Adaptive algorithms for detecting community structure in dynamic social networks, in Proc. IEEE INFOCOM, pp: 2282-2290.
7. Neto, S.M.B., M.A.C. Gatti, P.R. Cavalin, C.S. Pinhanez, C.N. Santos and A.P. Appel, 2013. Reaction Times for User Behavior Models in Microblogging Online Social Networks, In Proceedings of the 2013 Workshop on Data-Driven User Behavioral Modelling and Mining From Social Media, pp: 17-20.
8. Pastor-Satorras, R. and A. Vespignan, 2001. Epidemic spreading in scale-free networks, Physical Review Letters, 86(14): 3200-3203.
9. Boyd, D. and N. Ellison, 2007. Social network sites: Definition, history, and scholarship, Journal of Computer-Mediated Communication, 13(1): 210-230.
10. Half of the world's online population uses facebook. [Online]. Available: http:// www.statista.com/ topics/ 1164/ social-networks/ chart/1103/top-10-social-networks-in-q1-2013/

11. Fortunate, S., 2010. B Community detection in graphs, [Phys. Rep., 486(3-5): 75-174.
12. Pons, A.P. and M. Latapy, 2005. Computing communities in large networks using random walks, in Proc. ISCIS, pp: 284-293.
13. Zhen, L., H.T. Song and J.T. He, 2012. Recommender systems for personal knowledge management in collaborative environments, Expert Systems with Applications: An International Journal, 39: 16.
14. Xue, W., J. Shi and B. Yang, 2010. BX-RIME: Cloud-based large scale social network analysis, in Proc. IEEE Int. Conf. Services Comput., pp: 506-513.
15. Lin, J. and M. Schatz, 2010. Design patterns for efficient graph algorithms in Map Reduce, in Proc. ACM 8th Workshop Mining Learn. Graphs, pp: 78-85.