Revenue Maximization of Influence Diffusion in Social Networks

Jiayi Zhang
Shanghai, China
zjyhjz@sjtu.edu.cn

***Abstract*—The rapid development of social network has drawn a lot of attention to the research field. How to achieve influence maximization with some initial nodes to start diffusion has been studied a lot. However, no one has concern about the cost and payoff of the nodes which may have great impact on the revenue. We may choose less nodes or nodes with high quality to realize revenue maximization which is the paramount aim of companies. In this paper, I proposed some algorithms based on the existing works using Independent Cascade diffusion model. Some algorithms are of great performance on revenue and some on time complexity. Furthermore, I designed experiments to verify the proposed algorithms' performance and compared to some baseline algorithms.**

***Index Terms*—social network, influence maximization, revenue maximization, diffusion model**

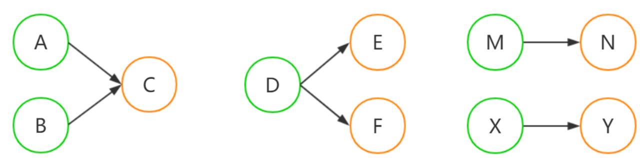
I. INTRODUCTION

With the development of online social network with a surprisingly high speed, it plays a significant role in everyone's daily life. Traditionally, information spread over social network by social interaction. Nowadays, online social network of a unprecedented scale, such as Twitter and Facebook, are effective tools in connecting people and bringing small and disconnected offline social network together. Since people use them to exchange their opinions about new products, companies can easily access corresponding data on people's reactions to the new products, which provides useful insights and opportunities on their marketing strategies.However, to fully utilize these social networks as marketing and information dissemination platforms, many challenges have to be met. In this paper, I focus my work mainly from the perspective of the advertise companies, trying to find influential individuals efficiently in a large-scale social network with the maximization of revenue.

To address this problem, we can use a toy example about it.A company develops a new product and wants to market it through social network. It has a limited budget such that it can only select a small number of initial users in the network to use it with incentves such as payments or presents. The company wishes that these initial users would love the product and start influencing their friends on the social network to use it, and their friends would influence their friends friends and so on, and thus through the word-of-mouth effect [4] a large population in the social network would buy the product. The key problem is how to select initial users so that after balancing the cost of them and the influence they brought, the company makes the best revenue.

The problem of selecting a subset of influential individuals, called seeding problem, has been extensively studied in the last decade, where the objective is to trigger the largest adoption of new products over social networks by seeding the influential subset, called seed set, i.e., providing some additional incentive for them to pre-adopt the new product. Various diffusion models have been proposed to address the problem.In all the models, the cost of the nodes are same and the payoff of each node has no difference.

However, in pratice, the maximization of the total active nodes in the event lifespan or in a limited time period is not equal to the best revenue.But as a company, their paramount goal is to maximize revenue. The following simple example demonstrates how the object revenue maximization influnce our choice of the seed set.



(a) Network 1

(b) Network 2

Fig. 1. Simple example of a social network.

In Fig. 1(a) and Fig. 1(b), we use green circles to represent seed nodes, orange circles as nodes being influenced. Obviously, if we want to maximize total active nodes with least nodes, we will choose node A or B with no difference in Network 1 and choose node D in Network 2. However, by adding the different cost of the node (the payment company need to pay to motivate the diffuse), we will choose the node with least cost between node A and B in Network 1. As for Network 2, consider the situation that cost of D is greater than the sum of node costs of M and X. We will choose node M and X instead of choosing node D. To get a better revenue from the adverstising, we should take both node cost and node payoff into consideration.

To this end, we formulate a new problem named *Revenue Maximization* to address this issue.

II. RELATED WORK

The diffusion models in literature can be broadly classified into: epidemic-based ones, e.g., [1], [3], [5], [7] and gamebased ones, e.g., [2], [6] depending on how diffusion dynamically occurs. In our work, we mainly discuss the epidemic-based diffusion. A lot of studies and several models have been proposed to describe the problem, such as Independent Cascade (IC) model [1],[3], [7], [9] and Linear Threshold (LT) model [3], [7], a data-based credit distribution model [10] and linear social influence model [11]. Among these models, IC and LT models are stochastic diffusion models which specify the randomized process of information propagation. These models offered algorithm to select a k-subset of the vertices. Their purpose was to show an algorithmically-effient (i.e., polynomial) mechanism with a finite approximation ratio, but never consider about the problem of the difference of node cost and node payoff.

Influence maximization, which aims to maximize the expected number of active nodes in a given diffusion model, is another main research direction of the analysis of information propagation in social networks. At first, people proved the problem is NP-hard in both IC and LT models and proposed a greedy framework to solve it[1]. The following researchers focused on developing both efficient and effective algorithms[1][8], such as Cost-Effective Lazy Forward (CELF) scheme which use submodularity property of the influence maximization object to greatly reduce the number of evaluations on the influence spread of vertices, PMIA, StaticGreedy, Linear and Bound and IMRank. In [1], the anthors take a new kind of node into consideration. The nodes were informed by the active nodes, but don not continue diffusing the information which is more realistic in practice.

However, all the existing work never consider the revenue maximization problem in the model. In [2], the authors talks about the balance between seeding quality (the willingness to be a seed) and the quantity (the range of the seeding) with a limit budget. In [6], the authors design mechanism to ensure the incentive compatible which means no node can get more utility by cheating about his/her social network. These are works that take practical scenarios and people's strategies into consideration. In our model, we introduce the node cost and payoff into the existing models, and find out a solution to solve it with efficiency and effectiveness.

III. Model

A. Inflence Diffusion Models

Social network can be imagined as people are nodes and their connection is edges in network. Whole social network can be divided into mainly three parts: (1) Social network as Graph G= (V, E) where V is nodes and E connection between nodes. (2) Diffusion model which decide in which pattern information will diffuse in network; (3) activation probability or weight on edges for independent cascade and linear threshold model respectively. There are mainly two diffusion models: Linear threshold model and Independent cascade model. In network there are mainly three types of nodes: Active, Informed and Inactive. Some of the node receives information from their neighbor and change state as Informed. If node adopts the new information and spread it, then state will be Informed to Active. These models spread influence in network and inactive nodes become active. Every node gets single chance to activate their neighbor nodes. An example of such information propagation in social network is shown in Fig. 2 [1].

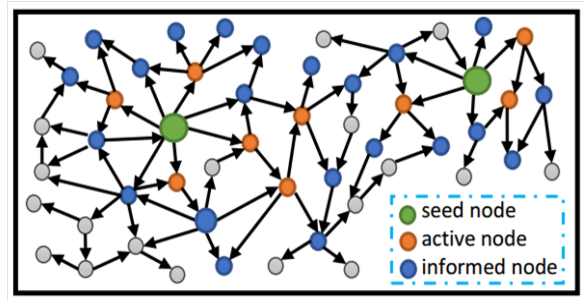


Fig. 2. Information propagation in social network.

- Linear Threshold Model (LT)
  Linear threshold model works based on threshold value of nodes. If node receives information from threshold number of nodes then it will change state and start to diffuse influence. i.e. If node v has threshold value 100 then it gets influenced byits neighboring 100 or more nodes, hence v will become active and diffuse influence in network.
- Independent Cascade Model (IC)
  Independent cascade model works based on activation probability of other neighbor nodes and node have single chance to get activated. Each edge has probability of p so node u has single chance to activate node v with probability value p. If edges have some weight assigned then weighted cascade model is taken in to account.

B. Assumptions

Based on the existing works, concepts and basic models, we construct my own model. To make my model more reasonable, it should satisfy the following assumptions.

- **Closed World assumption.**
  The major observation about modeling information diffusion is certainly that all the models work under a closed world assumption that information can only propagate from node to node via the network edges and that nodes cannot be influenced by external sources. In other words, we assume that people can only be influenced by other members of the network and that information spreads because of informational cascades. [7]
- **Independent diffusion process.**
  This assumption is that diffusion processes are independent, i.e. each information spreads in isolation. In contrast, the situations will be the spreading processes have cooperation and competition. Competing contagions decrease each others probability of diffusion, while cooperating ones help each other in being adopted. We ignored this factor to simplify our problem.
- **Independent Cascade Model based.** Although many different stochastic diffusion models can be used to

describe the information propagation process, we adopt IC model in this paper as it has been shown as one of the most suitable models for the diffusion of information.

- **No time delay diffusion**
  The users who are actived are assumed to immediately become seeds to continue the diffusion process. Meanwhile, they have the same willingness of diffusion.

C. Problem Definition

Consider the undirected graph $G = (V, E, P)$, where $V = \{1, 2, ..., n\}$ is the set of nodes in the graph and E denotes all edges between two nodes. We use n to denote the size of nodes $|V|$ and m to denote the size of nodes $|E|$ respectively. And for each edge, we have a diffusion probability $p_{i,j}$ to represent the probability of information transfer from i to j. All the probability information is stored in matrix P.

We use S to denote the set of seed nodes, A to denote the set of active nodes and I of informed nodes. After seed nodes are select, they spread the information to their neighbours and try to activate them. If the neighbour node is actived, it becomes part of node set A and start to act like the seed node. Otherwise, it will become an informed node and stop the diffusion process.

First we suppose that the cost of each node is identical, denoted as c, means that every seed node needs same incentive to start the propagation. However, as an imformed node, it may just adopt the information or product once but have no willing to adopt it again not to mention to spread it to others. So it not only stop the propagation process, but also has a lower payoff than the active node. So I define the payoff different between the two type of nodes, p_a as the payoff of active node and p_i as the payoff of informed node.

If the company has a budget limit, we can at most select k nodes as seed nodes. In the existing work, all the algorithms choose to select exactly k nodes so that they can achieve more nodes which are not inactive. However, our target is to maximize the revenue, if the increase payoff of nodes cannot cover the cost of the added node, the company has no motivation to pay for that seed. To this end, we can formulate our *Revenue Maximization Problem* as follows:

$$\begin{aligned} &\arg\max_{S} r(S) = p_a \cdot |A| + p_i \cdot |I| - c \cdot |S| \\ &s.t. \quad |S| \le k \end{aligned} \tag{1}$$

Since the target function is very different from before, we need to design new algorithms to target of revenue maximization.

IV. Revenue Maximization Algorithm

In this section, we make impoves on existing algorithms and propose two new algorithms.

Algorithm 1 describes the general greedy algorithm given a random process Random(). In each round i, the algorithm adds one vertex into the selected set S such that this vertex together with current set S maximizes the influence spread.To do so, for each vertex not in the seed set, the influence spread of adding it is estimated with R repeated simulations of $Random(S \cup \{v\})$

Algorithm 1: GeneralGreedy Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev
initialize $S = \emptyset$ and R;
for $i = 1$ *to* k **do**
    for *each node* $n \in V \backslash S$ **do**
        $s_n = 0$;
        for $i = 1$ *to* R **do**
            $s_n+ = |Random(S \cup \{n\})|$;
        end
        $s_n = s_n / R$;
    end
    $S = S \cup \{\arg\max_{n \in V \backslash S} \{s_n\}$;
end
$rev = r(S)$;
return S,rev;

. Each calculation of $Random(S)$ takes $O(m)$ time, and thus Algorithm 1 takes $O(knRm)$ time to complete.

Algorithm 2: ImproveICGreedy Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev
initialize $S = \emptyset$ and R;
for $i = 1$ *to* k **do**
    for *each node* $n \in V \backslash S$ **do**
        $s_n = 0$;
    end
    for $i = 1$ *to* R **do**
        compute G' by removing each edge from G with probability $1 - p$;
        compute $N_{G'}(S)$;
        compute $|N_{G'}(n)|$ for all $n \in V$;
        for *each node* $n \in V \backslash S$ **do**
            if $n \notin N_{G'}(S)$ **then**
                $s_n+ = |N_{G'}(n)|$;
            else
            end
        end
    end
    $s_n = s_n / R$ for all $n \in V \backslash S$;
    $S = S \cup \{\arg\max_{n \in V \backslash S} \{s_n\}$;
end
$rev = r(S)$;
return S,rev;

The time complexity of Algorithm 1 is too high and unrealistic. Consider that in the independent cascade (IC) model, RanCas(S) works as follows. Let L_i be the set of vertices that are activated in the ith round, and $L_0 = S$. For any edge in E, such that $u \in L_i$ and v is an inactive node. If v is not a informed node, v will be activated by u in the A_{i+1} round with an independent probability p, which we call the propagation probability. In other words, if there are n neighbors of v that are in L_i, v will be informed or actived with probability $1(1p)^n$. This process is repeated until L_{i+1} is empty. Notice that in the random process $Random(S)$, each edge is determined once, either from u to v or from v to u, on whether the influence is propagated through this edge. Moreover, now the probability on either direction is the same propagation probability p. Therefore, we may determine first whether the edge is selected for propagation or not, and remove all edges not for propagation from G to obtain a new graph G'. With this treatment, the random set is simply the set of vertices which are union of neighbours in set S in G'. Let $N_{G'}(S)$ denote the set of vertices neighbous' union from S in graph G'.

Thus, by randomly generating G' for R times, and each time computing s_v as stated above for all $v \in V \backslash S$ by a linear scan of graph G', we can select the next best candidate vertex v with the best average s_v. Algorithm 2 gives the details of the above improved algorithm. Since computing $N_{G'}(S)$ and $N_{G'}(\{v\})$ for all vertices$v \in V$ takes $O(m)$ time, the running time of the algorithm is $O(kRm)$ where R is the number of simulations. Therefore, our improvement in Algorithm 2 provides $O(n)$ speedup to the original greedy Algorithm 1.

Moreover, to make the algorithm more suitable to our problem, we can define $|N_G(S)|$ denote the set of vertices reachable from S in graph G that are consisted of informed nodes and active nodes. Different nodes have different weights when computing the best average s_v. We assert the weight being proportions to the payoff according to the node type. So $s_v = |A| + \frac{pi}{pa}|I|$ as our new algorithm which is part of the experiment.

Although the algorithm above has a speedup when comparing to Algorithm, it is still of low efficiency. What's more, we try to design a algrithm which is more efficient and more related to our problem scenario. We try to take the target function as our heuristic function.Due to the submodularity of the problem, this update scheme can reduce the times of estimating $r(S)$. We find out the node that by adding it in the seed set, we can get the most revenue gain. Then we compare the current revenue with the existing maximum revenue. If the current revenue is bigger, we will update the seed set and the maximum revenue. We only focus on the revenue maximization within the budget, but we do not guarantee the seed set size is the biggest. More details about the update scheme are shown in Algorithm 3. From the algorithm, we can see that it needs $(n+k)$ times of information coverage estimations, where $\ll n$ is the expected number of information coverage estimations in each iteration. Thus the total time cost is $O(nRm + kRm)$, where R is the number of rounds of simulations in each estimation.

Although Algorithm 3 update scheme reduces the time cost dramatically, it is still intractable for large scale networks. In the real world, there are often thousands of nodes and millions of edges in a social network. To address the scalability issue, we develop an efficient heuristic algorithm. When we revisit the objective function, we can find that a nodes contribution to

Algorithm 3: Revenue Difference Greedy Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev
initialize $S = \emptyset$;
initialize $max\_r = 0$;
initialize $S' = \emptyset$;
for *each node n in V* **do**
  compute $\Delta(n) = r(n)$;
  $flag_n = 0$;
end
while $|S'| < k$ **do**
  $n = \arg\max_{n \in V \backslash S'} \Delta(n)$;
  if $flag_n == |S'|$ **then**
    $S' = S' \cup n$;
    if $r(S') > max\_r$ **then**
      $S = S'$;
      $max\_r = r(S')$;
    else
      continue;
    end
  else
    compute $\Delta(n) = r(S' \cup n) - r(S')$;
    $flag_n = |S'|$;
  end
end
$rev = max\_r$;
return S,rev;

Algorithm 4: Degree Rank Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev
initialize $S = \emptyset$;
initialize $C = \emptyset$;
for *each node n in V* **do**
  $Degree(n) = OutDegree(n)$;
end
while $k > 0$ **do**
  while $|S| < k$ **do**
    $n = \arg\max_{n \in V \backslash S} Degree(n)$;
    $S = S \cup n$;
    $C = C \cup OutNeighbour(n)$;
    for *each node n in* $V \backslash S$ **do**
      $Degree(n) = OutDegree(n) - |C \cap OutNeighbour(n)|$;
    end
  end
  k = k-1;
end
$rev = r(S)$;
return S,rev;

the information coverage is highly dependent on its out degree. Thus if we rank the nodes according to their out degrees and take top-k nodes as the seed nodes, we can probably get a good result. Furthermore, when a node is selected, its out neighbours will be informed. This will result in a decrease of other nodes effective out degrees, as their out neighbours may have been informed. This observation means that we can benefit from adjusting each nodes effective out degree dynamically. This heuristic is summarized in Algorithm 4. From the algorithm, we can see that it takes only $O(k(n+m))$ time to complete if we store the graph G and the covered nodes set C with appropriate data structures.

Algorithm 5: Degree Discount Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev
initialize $S = \emptyset$;
for *each node n in V* **do**
    compute $Degree(n)$;
    $dDgree(n) = Degree(n)$;
    $flag_n = 0$;
end
for $i = 1$ *to* k **do**
    $u = \arg\max_n \{dDegree(n) | n \in V \backslash S\}$;
    $S = S \cup u$;
    for *each neighbour* n *of* u *and* $n \in V \backslash S$ **do**
        $flag_n = flag_n + 1$;
        $dDegree(n) = Degree(n) - 2flag_n - (Degree(n) - flag_n) flag_n p$;
    end
end
$rev = r(S)$;
return S,rev;

As for the Degree Rank Algorithm, we can make a further improvement by using a more accurate degree discount heuristic according to some property of IC model. Since v is a neighbor of u that has been selected into the seed set, with probability at least p, v will be influenced by u, in which case we do not need toselect v into the seed set. This is the reason why further discount is more accurate. When p is small, we may ignore indirect influence of v to multi-hop neighbors and focus on the direct influence of v to its immediate neighbors, which makes degree discount calculation manageable. This forms the guideline for us to compute the degree discount amount. By the Theorem 2 in [8],

In the IC model with propagation probability p, suppose that $d_v = O(1/p)$ and $t_v = o(1/p)$ for a vertex v. The expected number of additional vertices in $Star(v)$ influenced by selecting v into the seed set is:

$$1 + (d_v - 2t_v - (d_v - t_v)t_v p + o(t_v) \qquad (2)$$

By all the knowledge above, we can design Algorithm 5.Using Fibonacci heap, the running time of Algorithm 5 is $O(klogn + m)$.

V. Experiments

We conduct experiments for various algorithms on a real-life network. We try to find out the effectiveness and efficiency of them by observing the performance (revenue) changing by payoff, cost and budget.

A. Experiment Setup

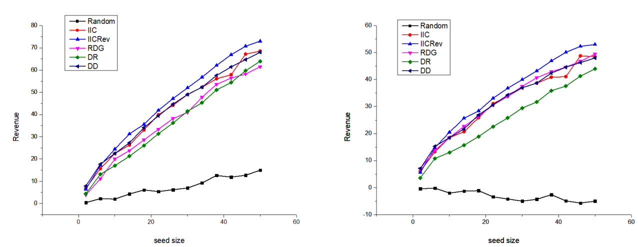
The data we use in this experiment is a collaboration graph crawled from arXiv.org, High Energy Physics Theory section, from year 1991 to year 2003. It has $n = 15233$ nodes and $m = 58891$ edges. The graph is available for download on the web at http://research.microsoft.com/enus/people/weic/graphdata.zip.

We run the following set of algorithms under the IC models on the network.

- **Random** As a baseline comparison, simply select k random nodes in the graph.
- **IIC** The ImprovedICGreedy algorithm in Algorithm 2 with $R = 20$
- **IICRev** Change the heuristic function in Algorithm 2 and make it related to the revenue with $R = 20$
- **RDG** Revenue Difference Greedy heuristic algorithm in Algorithm 3.
- **DR** Degree Rank heuristic algorithm in Algorithm 4.
- **DD** Degree Discount heuristic algorithm in Algorithm 5.

To obtain the influence spread of the heuristic algorithms, for each seed set, we run the simulation of the IC model on the networks 2000 times and take the average of the influence spread.

B. Revenue and Seed Set Size



(a) seed node cost = 0.8 (b) seed node cost = 1.2

Fig. 3. Realtionship between revenue and seed size.

Fig. 3(a) and Fig. 3(b) repersent the corresponding changes of revenue when seed size increase at seed cost 0.8 and 1.2 respectively. We fix the cost of seed node is 0.5, the payoff of active node is 1. IICRev Outperforms other algorithms greatly. IIC, DD and RGD have similar performance and DD is the most stable one among them. DR has a poor revenue gain compared to others.

C. Time Complexity

Fig 4 reports the running times of different algorithms for selecting k = 50 seeds in the two graphs. All results are measured on reasonably efficient implementation of the

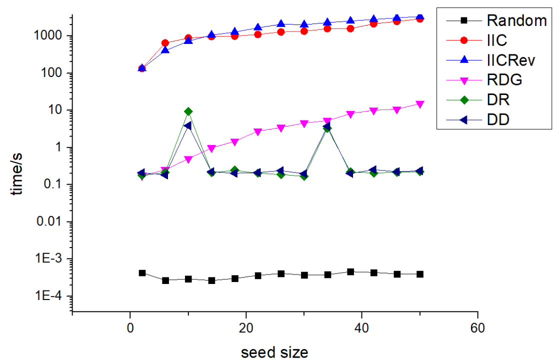


Fig. 4. Running time of different algorithms in log scale y-axis.

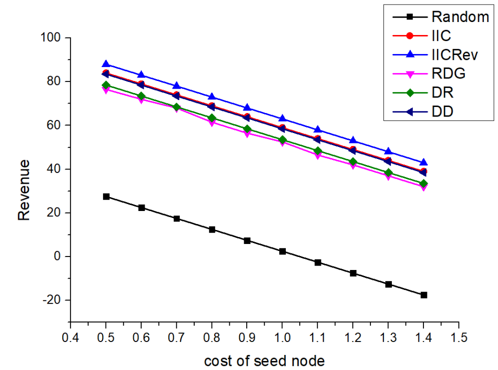


Fig. 6. Realtionship between revenue and cost of seed node.

various algorithms. Notice that the y-axis is in log scale. So the running time of IIC and IICRev is thousand time higer than other algorithms. Even in this experiment with small network and very small iteration time, it took me several days to finish the experiment. So it is not tractable in practice. RDG has a not bad time complexity, but it increases much as seed size increases. DR and DD with no doubt have great performance on time complexity.

D. Revenue and Payoff/Cost Value

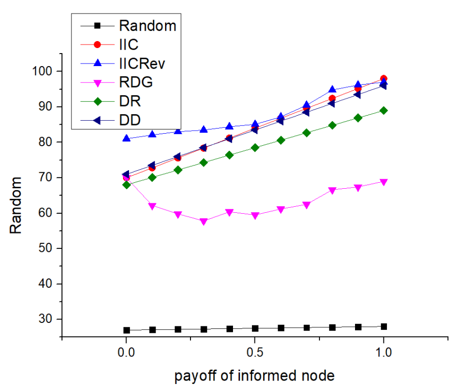


Fig. 5. Realtionship between revenue and payoff of informed node.

Fig 5 reports the relationship between the revenue and the change of payoff of the informed node. In this experiment, we fix the cost of seed node is 0.5, the payoff of active node is 1 and the size of seed set is 50. The payoff of informed node range from 0.0 to 1.0. The proportion of informed node and active node changes. IICRev outperforms other algorithms. IIC and DD has similar performance better than DR and RDG.

Fig 6 reports the relationship between the revenue and the change of cost of the seed node. In this experiment, we fix the payoff of informed node is 0.5, the payoff of active node is 1 and the size of seed set is 50. The cost range from 0.5 to 1.5. The Random algorithm performs bad even with negative revenue. IICRev outperforms other algorithms. IIC and DD has similar performance better than DR and RDG.

E. Conclusion

If we want to adopt the algorithm in a small network or have strong computation ability or do not consider the time problem, we should choose IICRev. However, it is usually intractable. In pratice, we can use RDG or DD to maximize our revenue. If there is strict limit on time complexity, DD is a best choice with great performance.

ACKNOWLEDGMENT

We would like to thank our teachers and teacher assistants for giving me excellent lectures so that giving me an inspiration of possible ways to carry out my research and offering consistent help to the project. We also thank the teachers and teacher assistants for taking time to read our article.

REFERENCES

[1] Z. Wang, E. Chen, Q. Liu, Y. Yang, Y. Ge, B. Chang, “Maximizing the coverage of information propagation in social networks,” Proc. 29th AAAI Conf. Artif. Intell. pp. 2104-2110 2015.

[2] J. Ok, J Shin, Y Yi, “Incentivizing strategic users for social diffusion: Quantity or quality?,” IEEE Conference on Computer Communications, INFOCOM 2017, Atlanta, GA, USA, 1–9.

[3] K Rabadiya, A Makwana, S Jardosh, IC Changa “Performance Analysis and a Survey on Influence Maximization,” INTERNATIONAL CONFERENCE ON TELECOMMUNICATION, POWER ANALYSIS AND COMPUTING TECHNIQUES-2017, At Bharath University, Chennai.

[4] D. J. Eaton, ”e-Word-of-Mouth Marketing,” [Online]. Available: $http://college.cengage.com/business/modules/eWOM_secure.pdf$. [Accessed 17 August 2016].

[5] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, Scalable influence estimation in continuous-time diffusion networks, in Proc. of NIPS,2013.

[6] Y. Babichenko, O. Dean, and Moshe Tennenholtz, ”Incentive – Compatible Diffusion”, in The World Wide Web Conference 2018, April 2327, 2018, Lyon, France.

[7] A. Guille, H. Hacid, C. Favre, D. A. Zighed, ”Information Diffusion in Online Social Networks: A Survey,” SIGMODRecord, June 2013 (Vol. 42, No. 2)

[8] Wei Chen , Yajun Wang , Siyu Yang, Efficient influence maximization in social networks, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, June 28-July 01, 2009, Paris, France

[9] D. Kempe, J. M. Kleinberg, and . Tardos. Maximizing the spread of influence through a social network. In Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 137146, 2003.

[10] A. Goyal, F. Bonchi, and L. VS Lakshmanan, ”A data-based approach to social influence maximization,” Proceedings of the VLDB Endowment, 5(1):7384, 2011.
[11] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, ”Pagerank with priors: An influence propagation perspective,” In IJCAI, 2013.