ISBN 978-981-14-1684-2
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
(WCSE 2019 SUMMER)
Hong Kong, 15-17 June 2019, pp. 753-759
doi: 10.18178/wcse.2019.06.112

Experimental Design Based Method for Influence Maximization

Yuliang Zhang[1,a] , Ling Chen[1,2,b+]

[1]Department of Computer Science, Yangzhou University,Yangzhou,225009,China
[2]State Key Lab of Novel Software Tech, Nanjing University, Nanjing, 210093, China

Abstract. In this paper, we investigate the influence maximization problem on the independent cascade(IC) model. An experimental design based algorithm is presented to find the seed set to maximize the influence in social network. In the method, we consider each node in the social network as an experiment, and the problem that choosing *k* seed nodes from the social network becomes that choosing the most representative *k* trials from all trials. We also take the situation about the similar nodes into account and try to make the similar nodes not as the seed at the same time. At last, we build the model and use the approach called cross-iteration to solve the problem. Our experimental results show that this method can effectively select the appropriate seed set to maximize the influence in the social network.

Keywords: Social network, influence maximization, experimental design, cross-iteration

1. Introduction

Today, social networks have become an indispensable part of life. The issues related to social networks have been continuously concerned by people, and the influence maximization on social networks is one of the key issues. The problem of maximizing the influence on social networks is that how to find a set of seed nodes in social networks to maximize the spread of information by these diffusion source. We regard the propagation of information as the influence between the nodes. In recent years, social networks have become popular, and the research on the influence maximization has more practical applications, such as marketing, network competition, and information diffusion, etc.

A social network can be described as a directed graph $G=(V,E)$, where V is a set of all nodes, E is the collection of directed edges between nodes, and each $v \in V$ represents a person in the social network. Each a directed edge $(v,u) \in E$ indicates that the node v has an influence on the node u. For each edge, there is not only a direction, but also an associated weight, which is used to indicate the strength of the influence between two nodes. The problem of maximizing the influence on the graph G is to select k seed nodes for information propagation or diffusion. First, the k seed nodes are activated. Then the nodes that have been activated will attempt to activate its inactivated neighbor nodes. This process continues until no new node can be activated.

Our goal is to choose a set S of k activated nodes so that the number of finally activated nodes by influence diffusion can be maximized.

The influence is propagated in the network under a certain propagation model. In this work, we investigate the influence maximization under the independent cascade (IC) model, which is based on the interactive particle system in probability theory. IC model is one of the classic information propagation models. Given the seed set S and the activation probability on each edge $p_{u,v}$, the process of information diffusion under the IC model is as follows:

+ Corresponding author. Tel.: +86-183-6282-2062; fax: +86-514-87887937.
E-mail address: [a] 18362822062@163.com[b]yzulchen@163.com

• At time *t*, if the node *v* is activated, it will try to activate its inactivated neighbor *u* with the probability $p_{u,v}$;

•If *v* cannot successfully activate *u* at time *t*, it no longer has a chance to activate *u*;

•The success of the activation of *u* by *v* is independent of the other neighbors of *u* to activate it;

•If *u* is successfully activated at time *t*, it will activate its neighbor at time *t+1*.

Kempe[1]et al. first proposed a greedy algorithm to obtain an approximate solution, but one of the biggest defects of the greedy algorithm is that it needs to use Monte Carlo method to simulate the influence propagation process, which leads to high time complexity. Leskove.et al. proposed the CELF[2] algorithm and its improved version of CELF++[3]. CELF++ delays the calculation of the marginal gain of the influence diffusion, thus greatly reduces the computational time required. Chen[4] et al. proposed the Degree Discount algorithm on the basis of node degree, which is better than the existing Degree algorithm. IMA and PMIA [5] are also well-known algorithms for IC models. By constructing a tree structure for each node, the influence is propagated and updated only in the tree structure. In recent years, researches not only study efficient algorithms for the IM problem, but also investigate some new influence maximization problems considering a variety of factors, such as considering the direct relationship between two nodes [6], or considering each point has its own geographical location information [7], or considering the influence of the propagation of time has the nature of changes [8] and so on.

The problem of finding the most influential *k* vertices to measure the entire network is very similar to the problem of selecting the most representative *k* experiments in the problem of experiment design [9]. In this work，we first creates a matrix to represent the social network. Each column of the matrix represents a test plan. The problem of maximizing the influence on social networks becomes the optimization problem of extracting a sub-matrix from the original matrix to represent it. This optimization problem can be solved by the cross iteration method. Our experimental results show that this method can effectively select the appropriate seed set to maximize the influence spreading.

2. Concept and Definitions

2.1.The problem of experiment design

Suppose there are *n* test schemes called $Y_1, Y_2, \cdots, Y_n$, and each of which considers some test factors. Now we need to select *k* test schemes from these *n* test schemes to cover the most of the test factors. We suppose the selected scheme is $X_1, X_2, \cdots, X_k$,which can be denoted by a matrix *X* of $m \times k$, where *m* is the number of factors. Then the matrix *Y* of $m \times n$ represents the scheme containing all the experiments. Let Y_i be the *i*-th column of the matrix *Y* representing the *i*-th experiment. $Y_{j,i} = 1$ means that test Y_i considers the *j*-th factor. $Y_{j,i} = 0$ means that the *i*-th test does not consider the *j*-th factor. Matrix *X* is a sub-matrix of *Y* and its each column represents a selected experiment.

We design a selection function $f_i(X, a), (i = 1, 2, \cdots, n)$ to approximate each Y_i. In other words, we try to minimize the formula: $\| Y_i - f_i(X, a_i) \|$.

We define $f_i(X, a_i)$ as the linear function $f_i(X, a_i) = \sum_{j=1}^{m} X_i \cdot a_{j,i} = X \cdot a_i$ .Our goal is to select the appropriate matrix *X* which is a sub-matrix of the original matrix *Y*, and the appropriate vector $a_i (i = 1, 2, \cdots, n)$ to minimize the formula: $\sum_{i=1}^{n} \| Y_i - X \cdot a_i \|^2$.

Therefore, we need to solve the following optimization problems:

$$\min_{X,A} \sum_{i=1}^{n} \| Y_i - X \cdot a_i \|^2 + \lambda \| a_i \|^2 \tag{1}$$

where $a_{j,i} \geq 0$,*A* is an $m \times n$ matrix with $a_1, a_2, \cdots, a_n$ as a column $m \times k$.

The above test design problem is similar to the problem of maximizing influence. We consider each node in the network as a test. The nodes that can be influenced are regarded as factors considered by the test. And the problem of maximizing influence becomes searching out the *k* nodes which covering the most factors. However, the problem of maximizing influence is not exactly the same problem as experimental design. For the network $G = (V, E, P)$,where *P* is the probability matrix of influence propagation, and the element $p_{i,j}$ of

P is the influence of v_i on v_j. To know the propagation range for each vertex, we must first construct an influence matrix Y, where $y_{i,j}$ is the probability for node v_j to influence node v_i.

2.2. Calculation of influence matrix under IC model

We consider the problem of maximizing influence under the independent cascade model. It can be seen from the above that the influence of node v on node u can be calculated by the independent path probability.

Definition 1 Probability of the path: Let the path L from v to u be linked by the edges $e_1, e_2, \cdots, e_l$ in sequence, and the probability of the path L is: $P_l = \prod_{i=1}^{l} p_i$.

Definition 2 Independent path: If two paths L_1 and L_2 does not overlap each other, they are called the mutually independent paths.

According to the above definition, the influence of v on u can be calculated by formula (2):

$$y_{u,v} = 1 - \prod_{\omega \in \Gamma_{in(u)}} (1 - y_{\omega,v} \cdot p_r(\omega, u)) \tag{2}$$

where: $\Gamma_{in(u)} = \{\omega \mid (\omega, u) \in E\}$ is the set of neighbors on the input side of u, and $p_r(\omega, u)$ is the propagation probability on the edge (ω, u).

Here, we propose Algorithm 1 to get the influence matrix:

Algorithm **1** : Calculation of the influence matrix

Input: $P = [p_r(v,u)]$: a matrix of propagation probabilities on each side; L : Length of path considered; $|L| \leqslant |V|$;

Output: $Y = [y_{u,v}]$: Influence matrix;

Begin

1: Begin: Initialization matrix Y is the transpose of matrix P

2: **for** $i \leftarrow 1$ **to** L **do:**

3: Add matrix R as an intermediate variable

4: **for** $u \in V$ **do:**

5: **for** $v \in V$ **and** $u \neq v$ **do:**

6: **for** $\omega \in \Gamma_{in(u)}$ **do:** $R_{u,v} = 1 - \prod_{\omega \in \Gamma_{in(u)}} (1 - y_{\omega,v} \cdot p_r(\omega, u))$ **end for** ω;

7: **end for** v;

8: **end for** u;

9: $Y = R$

10: **end for** i;

End

---

3. Maximizing Influence Based on Experimental Design

We regard Y as the matrix of the experimental scheme in the experimental design problem. We can use formula (1) to find X, so that the column of X is selected from Y and the matrix X can cover the most information of Y. However, the influence matrix Y is different from the test matrix in the experimental design. Its element value is a continuous value between [0, 1] instead of the 0-1 binary value, so we change the optimization problem of (1) into:

$$\min_{A,B} J(A,B) = \min_{A,B} \sum_{i=1}^{n} \left[\left\| y_i - Y \cdot a_i \right\|^2 + \sum_{j=1, b_j \neq 0}^{n} \frac{a_{ij}^2}{b_j} \right] + \lambda \|B\|_1 \tag{3}$$

where: $b_j \geq 0, a_{i,j} \geq 0$, the parameter $\lambda (>0)$ is to control the degree of fit.

In equation (3), we replace the unknown X by the known matrix Y, and introduce a selection vector $B = [b_1, b_2, \cdots, b_n]^T$ to control the selection of the Y column. A larger b_j value indicates that v_j could be selected

as a seed with larger probability. If $b_j = 0$, then $a_{1,j}, a_{2,j}, \cdots, a_{n,j}$ will be 0, and node v_j will not be selected as a seed.

In addition, in the problem of influence maximization, when we select seeds, it is necessary to avoid selecting nodes with similar influence ranges as seeds at the same time. That is to say, we want to select columns from *Y* as different as possible so that the selected columns can be diversified. To this end, we calculate the similarity matrix $S = [S_{i,j}]$ between the column vectors of *Y*, where the element $S_{i,j}$ is the similarity between the *i*-th and the *j*-th columns of *Y*. In this work, we use cosine similarity as a measurement. Then the optimization in formula (3) can be transformed into formula (4):

$$\min_{A,B} J(A,B) = \min_{A,B} \sum_{i=1}^{n} \left[\left\| y_i - Y \cdot a_i \right\|^2 + \sum_{j=1, b_j \neq 0}^{n} \frac{a_{ij}^2}{b_j} \right] + \lambda \left\| B \right\|_1 + \sum_{\substack{i=1}}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} b_i S_{ij} b_j \tag{4}$$

From the last item of (4), we can see that if $S_{i,j}$ has a larger value, which means that the *i*-th column of *Y* is very similar to the *j*-th column of *Y*, then larger values cannot assign to both b_i and b_j. This can ensure that nodes with very similar coverage cannot be selected as seeds at the same time.

4. Optimization Method

To solve the optimization in formula (4), we propose a cross-iterative method. In the method, each iteration consists of two steps: Fixing *B*, update the elements of matrix *A*, and then fixing *A*, update the elements of vector *B*. In this way, the elements of A and B are updated alternately until convergence.

For the objective function *J (A, B)*, we first obtain the partial derivative of *J (A, B)*, by $a_i, (i = 1, 2, \cdots, n)$ which gives:

$$\frac{\partial J(A,B)}{\partial a_i} = -2Y^T \cdot (y_i - Y \cdot a_i) + 2diag(B)^{-1} \cdot a_i$$

Let, μ be the step size. We use the following formula to update the value of a_i:

$$a_i^{'} = \max(0, a_i - \mu \cdot \frac{\partial J}{\partial a_i}) \tag{5}$$

Similarly, we derive the partial derivative of $J(A,B)$ by $b_k, (k = 1, \cdots, n)$, which is:

$$\frac{\partial J}{\partial b_k} = -\frac{\sum_{i=1}^{n} a_{i,k}^2}{b_k^2} + \lambda + 2\sum_{i=1, i \neq k}^{n} S_{i,k} b_k$$

Let the above formula be 0, we can get the update formula of b_k:

$$b_k = \sqrt[2]{\frac{\sum_{i=1}^{n} a_{i,k}^2}{\lambda + 2\sum_{i=1, i \neq k}^{n} S_{i,k} b_k}} \tag{6}$$

Based on the analysis above, we present an experiment design based algorithm for influence maximization. The algorithm first calculates the influence matrix *Y* from probability matrix *P* by algorithm 1. After initializing the values of *A* and *B*, the algorithm performs the cross iteration to update their values. Finally, we select the largest *k* elements in the resulting vector *B* to be a seed set.

Algorithm 2 Experiment Design based IM

Input: *k*: The size of the seed collection;

P: Propagation probability matrix between network nodes;

Output: *Seed*: Seed set consisting of *k* nodes;

Begin:

1: Calculate the influence matrix *Y* by algorithm 1 based on the probability matrix *P*

2: Initialize the values of t matrix A and vector B;

3: **while** not converge **do**:

4: **for** $k \leftarrow 1$ **to** n **do**: Update B according to formula (6); **end for** k;

5: **while** not converge **do**:

6: **for** $i \leftarrow 1$ **to** n **do** Update A according to formula (5); **end for** i ;

7: **end while;**

8: **end while;**

9: Take the node set corresponding to the largest k elements of the B vector as the *seed* set;

10: **return** *seed*

end

5. Experimental Results and Analysis

To evaluate the quality of the predicting results by our algorithm, we tested it on real-world dataset Epinion, (with $n = 5k$, $m = 14k$) and compare the quality of the predicting results with the other similar methods. The algorithms are coded in Python and run on a Windows 10 machine with 2.8GHz, Intel I7 7700HQ and 16GB of RAM.

The experiments in this paper are tested on an independent cascade model. The probability on each side is set to: $p_{v,u} = 1/(\Gamma_u + 1)$, where Γ_u is the degree of ingress of node u.

5.1. Test on the influence spreading by the algorithm

In our experiments, we first test the influence spreading of our influence-maximization algorithm based on the experimental design. We set the parameters $L = 2$,$\lambda = 10$, $\mu = 0.09$, and set the maximum number of iterations as 16. The influence spreading by the algorithm in different sizes of seed sets are shown in Figure 1. From Figure 1， we can see that as the iteration progresses, the range of influence propagation increases, which indicates that the algorithm is effective.

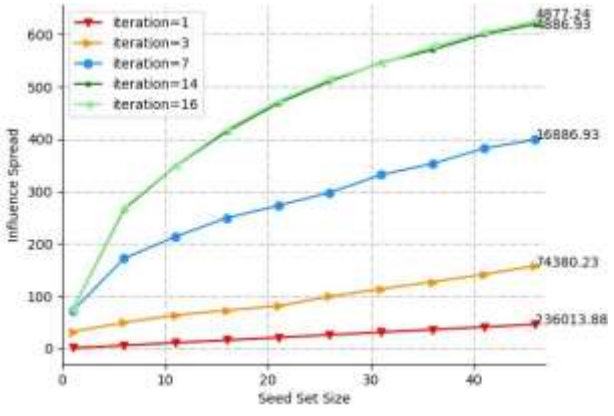

Fig. 1: The influence spreading of the algorithm on seed size

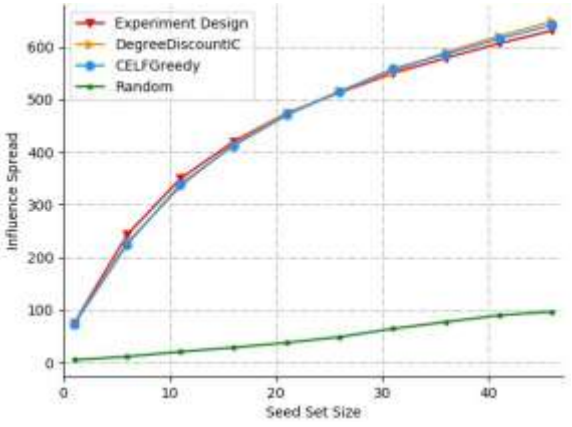

Fig. 2: Influence propagation of different algorithms

5.2. Comparison with other algorithms

In our experiments, we also test the quality of the results by the proposed algorithm and compare them with that of 2 other methods for influence maximization in complex networks: DegreeDiscountIC， Random and CELFGreedy. DegreeDiscountIC considers a simple adaptive strategy. In each round, a node u with the maximum degree is selected as a seed. Then, for each of u's neighbor v, we do not count the edge (u,v) when calculating its degree. In other words, the degree of v will be discounted by 1. This type of degree is named

as discount degree, and used in seed selection. Random is one of the most commonly used methods for influence maximization. In the method, k nodes are randomly selected to construct a seed set. In our experiments, such random seed selection process is repeated several times, and the average of spreading by the different seed sets are output as the result of the algorithm. The CELFGreedy algorithm is an improvement of the greedy algorithm that delays the calculation of the influence diffusion marginal gain.

In our experiments, on Epinion data set, we set the values of the parameters as L= 2, λ= 10, μ= 0.09 for our influence maximization algorithm based on experiment design. The comparison of the influence spreading obtained by different algorithms is shown in Figure 2. From the figure we can see that when the number of selected seeds is relatively small, the influence spreading of our algorithm is larger than other algorithms. When the number of selected seeds is larger, the influence spreading by our algorithm is almost the same as the other algorithms, and is slightly smaller than that of CELFGreedy or DegreeDiscountIC in a few cases. Algorithm Random shows the worst performance in all the four algorithms.

5.3. The influence of different parameters on the algorithm

We also test our algorithm our influence maximization algorithm based on experiment design under different values of parameters such as λ and L.

The parameter λ in the objective function is to control the degree of fitting. Figure 3 shows the effect of different values of λ on the influence spreading algorithm when L = 2 μ =0.09. From the figure, we can see that when the parameter λ is small, the algorithm cannot get large influence spreading, especially when the selected seed number k is small. But if the value of λ is too large, the effect will become worse when the seed set size k increases. Therefore, selection of the value of parameter λ should be moderate.

We also test the performance of our algorithm under different values of L, and Figure 4 shoes the experimental results. In the experiment, we set λ = 10, μ = 0.09. From the figure, we can see that each value of L has its corresponding k value which can obtain the largest influence spreading. For example, when L = 2, the largest influence spreading can be obtained if we set the seed size k as 6. The value of parameter L directly effects the influence matrix Y and is also directly related to the value of k.

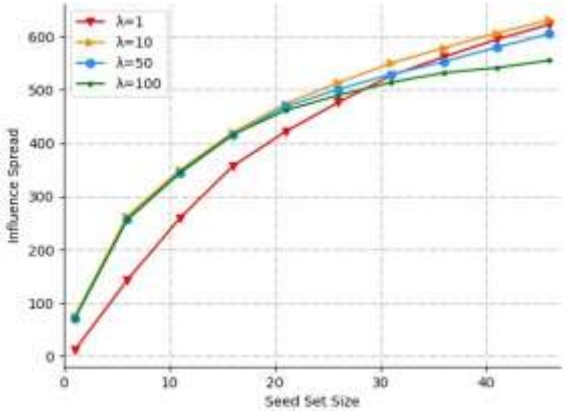

Fig. 3: Effect of the parameter λ on the objective function

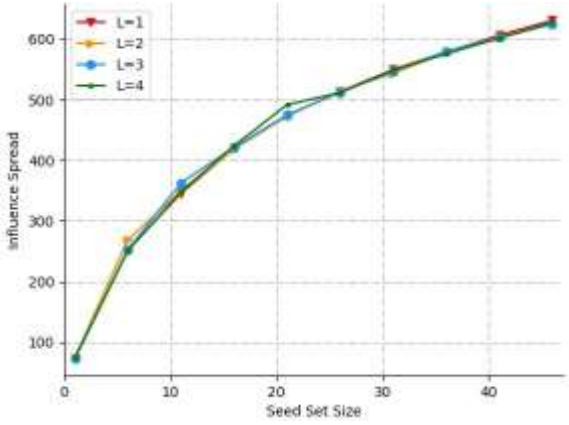

Fig. 4: Effect of the parameter L on the objective function

6. Conclusion

With the development of social networks, research on the problem of influence maximization on social networks is developing rapidly. This paper attempts to present the problem of maximizing influence in a linear model, and transforms the problem of maximizing influence into a linearity with multiple parameters. The regression model solves the influence maximization problem by solving the optimization problem. In this paper, we design the influence maximization matrix, and consider the repeatability of similar nodes. A cross-iteration algorithm is presented to solve the problem. Our experimental results show that the proposed algorithm can spread the influence more effectively under certain condition.

The method above can improve the accuracy on the influence maximization problem, but the time complexity is high because of the matrix calculation. Therefore, one of the possible future research is to optimize the matrix. For example, we can reduce the matrix dimension by just using a little the most important attributes to build the influence matrix. Another future direction is to optimize the model. We can not only use the linear model, but also other similar models to make the problem to be the optimization problem. And then use the relevant methods to solve the problem.

7. Acknowledgements

This research was supported in part by the Chinese National Natural Science Foundation under grant Nos. 61379066, 61379064, 61472344, 61402395, Natural Science Foundation of Jiangsu Province under contracts BK20130452, BK20140492.

8. Reference

[1] David Kempe, Jon Kleinberg, and ÉvaTardos. Maximizing the spread of influence through a social network. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data min- ing, pages 137–146. ACM, 2003.

[2] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In Proceedings of the 13th ACM SIGKDD international con- ference on Knowledge discovery and data mining, pages 420–429. ACM, 2007.

[3] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In Proceedings of the 20th international conference companion on World wide web, pages 47–48. ACM, 2011

[4] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximiza- tion in social networks. In Proceedings of the 15th ACM SIGKDD in- ternational conference on Knowledge discovery and data mining, pages 199–208. ACM, 2009.

[5] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceed- ings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1029–1038. ACM, 2010.

[6] Yuchen Li, Ju Fan, Dongxiang Zhang, and Kian-Lee Tan. Discovering your selling points: Personalized social influential tags exploration. In Proceedings of the 2017 ACM International Conference on Management of Data, pages 619–634. ACM, 2017.

[7] Guoliang Li, Shuo Chen, Jianhua Feng, Kian-lee Tan, and Wen-syan Li. Efficient location-aware influence maximization. In Proceedings of the2014 ACM SIGMOD international conference on Management of data, pages 87–98. ACM, 2014.

[8] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, and Xuemin Lin. Distance- aware influence maximization in geo-social network. In ICDE, pages 1–12, 2016.

[9] Tresp V Yu K, Bi J. Active learning via transductive experimental design. In Proceedings of the 23rd international conference on Machine learning., pages 1081–1088, 2006.