RLIM: Representation Learning Method for Influence Maximization in social networks

Sun Chengai
Shandong University of Science and Technology
Duan Xiuliang
Shandong University of Science and Technology
Qiu Liqing (qiuliqing2020@163.com)
Shandong University of Science and Technology
Shi Qiang
Shandong University of Science and Technology
Li Tengteng
Shandong University of Science and Technology

Research Article

Keywords: Influence maximization, information diffusion model, propagation probability, neural network architecture, representation learning

Posted Date: April 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-381918/v1

RLIM: Representation Learning Method for Influence Maximization in social networks

Sun Chengai [a] , Duan Xiuliang [a] , Qiu Liqing [a,1] , Shi Qiang [b] and Li Tengteng [a]
[a] *College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[b] *College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, Qingdao 266590, China*

Abstract. A core issue in influence propagation is influence maximization, which aims to find a group of nodes under a specific information diffusion model and maximize the final influence of this group of nodes. The limitation of the existing researches is that they excessively depend on the information diffusion model and randomly set the propagation ability (probability). Therefore, most of the algorithms for solving the influence maximization problem are basically difficult to expand in large social networks. Another challenge is that fewer researchers have paid attention to the problem of the large difference between the estimated influence spread and the actual influence spread. A measure to solve the influence maximization problem is applying advanced neural network architecture also represents learning method. Based on this idea, the paper proposes Representation Learning for Influence Maximization (RLIM) algorithm. The premise of this algorithm is to construct the influence cascade of each source node. The key is to adopt neural network architecture to realize the prediction of propagation ability. The purpose is to apply the propagation ability to the influence maximization problem by representation learning. Furthermore, the results of the experiments show that RLIM algorithm has greater diffusion ability than the state-of-the-art algorithms on different online social network data sets, and the diffusion of information is more accurate.

Keywords. Influence maximization, information diffusion model, propagation probability, neural network architecture, representation learning

1. Introduction

In Online Social Networks (OSNs) [1-3], a variety of information is transmitted between individuals and groups. The continuous iteration of this information transmission is the information diffusion in social networks. One of the purposes of studying information diffusion is to solve the Influence Maximization (IM) [4,5] problem. The core issue of information diffusion research is to predict the possibility of information diffusion. The IM problem is to find a fixed number of active nodes and uses a specific diffusion model

[1] Corresponding Author, Corresponding author, Book Department, IOS Press, Nieuwe Hemweg 6B, 1013 BG Amsterdam, The Netherlands; E-mail: bookproduction@iospress.nl.

to maximize the final number of active nodes. The Independent Cascade (IC) model and Linear Threshold (LT) model are often used to simulate the information diffusion..

Over the past years, A number of researches have been done to solve the IM problem. The typical solutions can be divided into two categories: the greedy algorithms [6-10] and the heuristic algorithms [11-13]. Kempe et al. [6] formally expressed the IM problem and proved that the optimal solution of this problem is NP-hard. Moreover, they presented a general greedy hill climbing algorithm with the error bounded by $(1-1/e-\epsilon)$ where ϵ denotes the error generated by using the Monte Carlo (MC) simulations to evaluate the influence spread. Similarly, to solve the efficiency problem in influence maximization, a number of heuristic algorithms have been proposed. The most famous approach is to select seeds based on their degrees [11]. However, the heuristic algorithms usually perform much worse than the greedy algorithm. Furthermore, the propagation probability in the process of information transmission is usually set a random or even a fixed value, which is not accurate.

In the information diffusion, the user will have a certain tendency of forwarding behavior when receiving information. This tendency modeling of forwarding is to predict the propagation probability. A number of researches have been done in this area. Kempe et al. [9] supposed a uniform propagation probability in the seed selection process. Saito et al. [14] focused on the IC model and proposed a method to predict the propagation probability based on the logarithm of the past propagation. Cao et al. [15] studied the IM problem under the LT model with unknown diffusion model parameters. Goyal et al. [16] estimated propagation probability by utilizing historical data, thereby avoiding the need for learning influence probabilities through expensive MC simulations. These approaches all adopt pair-wise manner to simulate the influence probability without considering other factors, such as the content of information transmission.

In the previous works about the IM algorithm, the running time of heuristic algorithms is the most important factor for researches. Efficiency is the core factor of the research task, especially in fast-tracking OSNs. However, these types of algorithms have some limitations. For example, the degree-based heuristic algorithm [11] is generally simulated on the IC model, and the propagation probability is set to a fixed value or a random value. The basic assumption of the IC model is that whether node u tries to activate its neighboring node v is an event with probability p, where p is set to a fixed value. After analyzing the basic principles of the IC model, two core problems can be found. First, the information diffusion process of the IC model occurs on the nodes that have connections (neighbor nodes), but the real-world information diffusion may occur between unknown connections. Secondly, the activation probability between nodes cannot be set according to the influence ability of the nodes. These problems have greatly interfered with the influence spread estimated.

To break through the limitations of existing research methods, this paper proposes the RLIM algorithm. The proposed algorithm requires three aspects of researches on nodes, including the influence cascade, the vectorized representation, and the information diffusion. The influence cascade is a sequence of nodes affected by the source node, and is the raw material for vectorized representation. Vectorized representation is a tool for predicting the influence capacity, which is the key to narrowing the gap between estimated influence spread and actual influence spread. Influence diffusion is a way to maximize influence spread based on the vectorized influence ability.

RLIM algorithm absorbs the core technology of current social network research and becomes a powerful tool that can be applied to large-scale social networks to solve the

IM problems. Currently, the application of representation learning technology [17-19] is quite mature. In OSNs, representation learning generally means that adopting a vector to represent a node. The key to this technique is to generate the context of nodes appropriately and uses the Deep Learning [20] to realize the vector representation of nodes. Inspired by this idea, to facilitate the subsequent information diffusion, the influence cascade can be realized by analogy to the context of nodes. Although representation learning method can achieve vector representation of nodes, it still has limitations for the IM problem. Therefore, another vectorized representation method (neural network architecture) is considered to be added to the RLIM algorithm. The Neural Network Architecture (NNA) [21-23] has three advantages to realize the vectorized representation, including the adaptation to massive data, the super computing power and the advanced algorithm support. In summary, NNA is used to solve the problem of vectorized representation of nodes in a huge social network and representation learning is used to solve the problem of information diffusion. Furthermore, the experimental results show that the proposed algorithm outperforms the state-of-the-art methods. The paper makes the following contributions:

- The paper proposes a new framework to solve the IM problem, which includes the establishment of influence cascade, the predict of the propagation probability and the simulation of the information diffusion. Among them, the key factor of the framework is how to produce influence cascade.
- NNA is used to compute the vectorized representation of nodes. Specifically, each node vector represents that the propagation ability is not limited to the existing communication links, and can also indicates the propagation behavior that has not occurred. By this way, the actual propagation situation is reflected by the proposed algorithm.
- Representation learning method is used to maximize information diffusion and solve the problem that traditional IM algorithms cannot be applied to large social networks. Moreover, the paper designs a classification visualization experiment and an influence communication experiment, which respectively prove that the RLIM algorithm is close to real communication and has advantages in large-scale social networks.

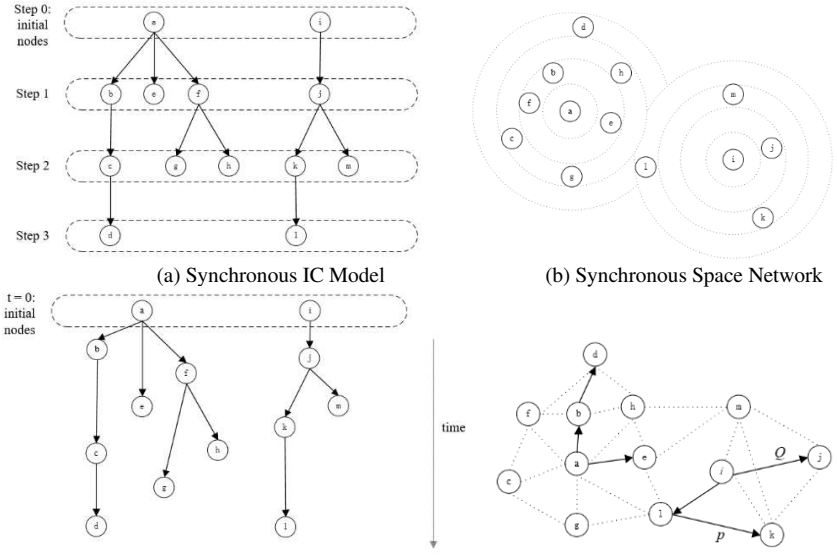
The rest of the paper is organized as follows. Section 2 presents motivations and related works for the proposed algorithm. Section 3 introduces the three main parts of the proposed method framework. Section 4 presents the details of the proposed RLIM algorithm. Section 5 conducts related experiments and case studies on the real-world data set. Finally, Section 6 concludes the paper and gives some directions of the future works..

2. Motivations and related works

With the vigorous development of online social networks, a large amount of researchable real-world data is produced, which is a huge challenge for traditional research algorithms. The traditional IM algorithms basically includes two parts: the information diffusion model and the selection method of the seed set. In the study field of the IM algorithm, it has emerged an awkward situation that the traditional diffusion model is not suitable to study the large datasets. Moreover, the limitation of the existing algorithms is that they excessively depend on the information diffusion model and randomly set the propagation probability. For example, the degree-based heuristic algorithm [24] sets the propagation

probability to a fixed value 0.01 or 0.1, which is very inconsistent with the actual diffusion situation. As a result, the influence spread is very different between the estimated value and the actual value. Therefore, the traditional IM algorithm requires optimization.

The traditional diffusion models mostly simulate the information propagation process based on the existing connections between nodes, such as the IC model. However, to diffuse information, real-world social networks may establish a new connection between nodes, which requires prediction. To make the information diffusion close to the real situation, researchers put forward the concept of space network. Immediately, the space network constructed by existing connections and predicted connections becomes a hot spot in current social network research. Figure 1 compares the difference between the traditional IC model (left) and the new space network (right).



(a) Synchronous IC Model (b) Synchronous Space Network

(c) Asynchronous IC Model (d) Asynchronous Space Network

Figure 1. (a) The forwarding of information is carried out within a time step represented by a natural number. (b) The positions of nodes in the network are relative, and nodes in the same circle have the same level of influence probability. (c) The time interval for the information to be forwarded to neighboring nodes is a continuous random variable. (d) The information flow between nodes is not fixed, and the propagation direction is determined by P and Q.

Figure 1 (a) and (c) are two classic IC models. Figure 1 (a) presents the synchronous IC model [25], which assumes that the forwarding of information is carried out in a time step represented by a natural number, and the forwarding of information in each time step is carried out synchronously. Another IC model is called the asynchronous IC model [26], which assumes that the time interval for forwarding information to adjacent nodes is a continuous random variable. Figure 1 (c) presents this model, which usually supposes that this random variable is exponentially distributed or approximately normal distribution. These two propagation models are widely used to solve the IM problem and have become models that many algorithms rely on. However, in the process of the traditional information diffusion model, how the propagation probability reflects the reality is a difficult problem. Therefore, the construction of models that reflect actual

information diffusion is a key research direction. Figure 1 (b) and (d) are proposed Space Network. Figure 1 (b) presents the synchronous space network [27], which can be regarded as the space representation of the synchronous IC model that removed the known connected edges. In this model, the positions of nodes are relative, and nodes in the same circle have the same level of propagation probability. Moreover, these propagation probabilities are calculated by a function of the relative positions of the two nodes. Figure 1 (d) presents asynchronous space network, where the direction of information flow between nodes is not fixed, which is determined by P and Q. Moreover, these propagation probabilities are obtained by the function including the influence cascade vector and the cascade length vector. The combination of the asynchronous space network and the neural network is the focus of the paper.

The original selection method of the seed set is greedy method. Because this method is time-consuming and inconvenient to be applied to real data sets, it has not made impressive progress. However, it is worth mentioning that CELE algorithm [7], the representative algorithm of the greedy method, reduces time consumption by reducing MC simulation. In this way, the efficiency problem of the greedy method is slightly alleviated. Therefore, the proposed algorithm that adopts the greedy method can shorten the time consumption by reducing the number of simulations or selecting valuable candidate nodes. Moreover, the estimation technology based on the classic statistical tool (martingales) is applied to solve the IM problem, which is a brand-new framework. Such as the IMM algorithm [28], which can not only provide accurate results with small calculations, but also can be applied to various types of information diffusion models. However, based on the analysis of the experimental results, this paper found that these algorithms still cannot play an important role in large social networks. Therefore, to solve the IM problem, the method that meets the development of OSNs need to be proposed urgently.

Currently, representation learning is widely applied in the analysis of social networks. The biggest feature of presentation learning is that the network structure and node properties can be captured by the node vector. It is the reason why the presentation learning has become a sought-after object for many researchers at the moment. In the many research methods, the biggest change is the way of obtaining node context. Perozzi et al. proposed DeepWalk algorithm [29], which is an algorithm that generated context with random walks and then updated the representations with skip-gram [30]. Although this algorithm creates a precedent for learning node representation using short-term random walks, there is still a problem that high-order nodes cannot be learned by low-dimensional representation. To solve this problem, Aditya et al. proposed the node2vec algorithm [31], which is an improved algorithm that uses the second-order random walks method to generate the influence cascade. The proposed method of generating influence cascade is an improvement on it, which introduces two parameters P and Q to construct a high-order influence cascade.

At the same time, a large number of research methods have emerged on information diffusion. To model the information diffusion in OSNs, the most important problem is to infer the propagation probability between nodes, which is fundamental to the IM problem. Goyal et al. proposed a method that estimates the propagation probabilities by utilizing the co-occurrence counting. Another method adopts the word2vec technique, which is improved for the word representation learning. It is called word embedding in the Natural Language Processing (NLP) [32]. Tang et al. designed the LINE algorithm [33], which preserved both the local and global network structure by using the first-order and second-order proximity. To consider influence propagation and similarity of user

interest, Feng et al. proposed the Inf2vec algorithm [34], which combined node2vec model and global user similarity to learning the representations. Moreover, the most valuable method for solving the IM problem is the IMINFECTOR algorithm [35] proposed by Panagopoulos et al. This algorithm utilizes multi-task neural network architecture to calculate, and can vectorize the node sequence and the sequence length at the same time. Inspired by this, the proposed RLIM algorithm combines neural network architecture and similarity of user interest.

For the IM problem, the ultimate purpose is to maximize the influence spread. When the optimal solution of a problem contains the optimal solution of its sub-problems, it can be solved with the key features of dynamic programming algorithm or greedy algorithm. According to the submodular of the IM problem, the RLIM algorithm adopts the greedy method to maximize the influence propagation. The greedy method executes relatively time-consuming, however, it can be improved by reducing the number of candidate seed nodes according to the influence ability. The RLIM algorithm keeps $\beta\%$ test nodes as participating in information diffusion.

To sum up, different from the existing research methods, the proposed algorithm includes three innovations. First, the influence cascade adopts the high-order random walks, which is the process of using P and Q to control the direction to form nodes' sequence. Moreover, the propagation probability utilizes the combination between combines neural network architecture and similarity of user interest. The NNA can simultaneously realize node vectorized representation of the influence cascade and propagation ability. Besides, the consideration of user interest is to improve the authenticity of influence propagation. Finally, the termination of RLIM continuously optimizes the marginal gain through the greedy method. The vectorized nodes use the greedy method to calculate the marginal gain, which is radically improving in time consumption.

3. Influence maximization representation learning

The section introduces the three main parts of the proposed method framework. The basis of all work is to extract the asynchronous space network, which includes all the possible connections of users that have the same interest. Similarly, the construction of the node influence cascade is also the fundamental part of the proposed method. Furthermore, the second part calculating the vectorized representation of the influence cascade and propagation probability. In this section, the term of propagation possibility is called cascade length. Finally, the ultimate purpose of the proposed method is to maximize the influence spread. To better illustrate the proposed method framework, Figure 2 shows a simple example.

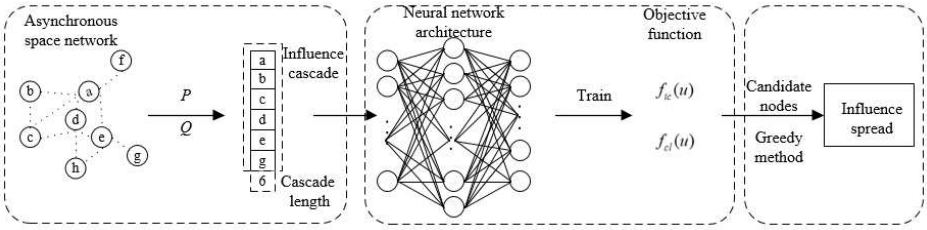


Figure 2. Representation learning method framework.

3.1. influence cascade

According to these data sets obtained from real OSNs, we construct an asynchronous space network $AS = (V, E)$, where V denotes all the nodes with the same interest in the transferring information process and E denotes the all-possible connections of nodes. Although asynchronous space network has many common links, it is different from the traditional network. The space network depicts that the flow of information diffusion is not only between the nodes that are already known but also between the nodes that are predicted to connect. Therefore, how to establish an influence cascade in this network is a key issue. Suppose there are an initial node u and an unfixed cascade length. The direction of the cascade is determined by P and Q. Whether the depth or width of the node walk conforms to the following binomial distribution.

Definition 1: The paper assume that the cascade is n times, and the probability of the depth walk (P) appears one times is p.

$$P(p|n) = \frac{p^n}{\int_0^1 p^n dp} = (n+1)p^n$$

Proof: The conjugate prior of the binomial distribution is the beta distribution. The paper assume that the cascade is n times, and the probability of the depth walk appears k times is p:

$$P(k|n,p) = C_n^k p^k (1-p)^{n-k}$$

Therefore,

$$P(p|n,k) \propto P(p)P(k|n,p).$$

Taking conjugate prior,

$$p \sim Be(\alpha, \beta)$$

Finally,

$$P(p|n,k) \propto p^{n+\alpha-1}(1-p)^{\beta-1}$$

The direction of the next time cascade is the depth walk, the probability is as follows:

$$\int_0^1 p \cdot (n+1)p^n = \frac{n+1}{n+2}$$

3.2. vectorized representation

The subsection implements the vectorization of cascade nodes and cascade capabilities. Especially, the cascade capacity refers to the length of the cascade. Given a message m, we construct $G^m = (u, v, t_v)$, where each tuple (u, v, t_v) denotes node v receives the information from node u at time t_v. Under the same interesting information transfer, we define two vectors: initial vector I_u and target vector T. Initial vector I_u denotes the user u who first receives the information at time t_0, and target vector T indicates all users who receive the information after time t_0. Similarly, the vector C is defined to represent the propagation capability of the user. When applying the neural network architecture, we define the hidden layer function, output function, and loss function respectively.

First, the vectorization of the cascade node is given below.

Definition 2: The hidden layer function of the cascade node is defined as follows.

$$g_{v,u} = I_u T + b_v$$

where b_v denotes bias. Moreover, the output layer function f_{ic} utilizes SoftMax function.

$$f_v(I_u) = \frac{\exp\,(g_{v,u})}{\sum_{w \in G^m} \exp\,(g_{v,w})}$$

where $w \in G^m$ denotes nodes in the influence cascade. Furthermore, the paper uses the Logarithmic Loss Function

$$\mathcal{L}\big(y, p(y|x)\big) = -\log\, p(y|x)$$

In fact, the loss function uses the idea of maximum likelihood estimation. $p(y|x)$common explanation is: based on the current model, for the sample x with the predicted value is y, which is the probability that the prediction is correct. Finally, because it is a loss function, the higher the probability of correct prediction, the smaller the loss value should be, so add a negative sign to get the opposite result. We define the Loss Function by vectorization:

$$\mathcal{L}_v = -\log\,(f_v(I_u))$$

Second, the vectorization of the cascade length is given below.

Definition 3: The hidden layer function of the cascade node is defined as follows.

$$g_{c,u} = I_u C + b_c$$

where b_c denotes bias. Moreover, the output layer function f_{cl} utilizes Sigmoid function.

$$f_c(I_u) = \frac{1}{1 + \exp\,(g_{c,u})}$$

where $w \in G^m$ denotes nodes in the influence cascade. Furthermore, The paper use Quadratic Loss Function in the definition of loss function.

$$\mathcal{L}_c = (y_c - f_c(I_u))^2$$

where y_c the cascade ability of the initial node u.

3.3. influence spread

After the neural network architecture is trained on the datasets, the paper get the function f_{ic} that reflects the influence of the cascade and the function f_{cl} that reflects the cascade ability.

Definition 4: Through these two basic objective functions, the influence cascade and cascade capabilities of the nodes in the test datasets can be predicted:

$$Pr_u = \begin{bmatrix} f_{ic}(I_u T)_{1,1:N} \\ \vdots \\ f_{cl}(I_u T)_{l,1:N} \end{bmatrix}$$

where N denotes the embedding size. To reduce time consumption, the nodes need to be sorted. The ordering of nodes is based on the value of α.

$$\alpha_u = \frac{N|I_u|^2}{\sum_{w \in \mathrm{M}} |I_w|^2}$$

where M denotes the node-set used for testing. Select $\beta\%$ nodes in the test set to participate in the influence propagation process, and calculate the marginal gain $\sigma(S)$.

$$\sigma(S) = \sum_{\gamma}^{\mathrm{Z}} Pr_{s,r}$$

where Z= M$\beta\%$. The RLIM algorithm maximizes $\sigma(S)$ in greedy manner.

Finally, the maximization influence spread of the initial node can be calculated in the test data. We adopt the representation learning method as a bridge to solve the IM problem, which can greatly expand the spread of influence and make the influence closer to the real spread.

4. Influence maximization representation learning

The section provides the Representation Learning Method for IM problem. This part provides pseudocodes of related algorithms to explain the RLIM algorithm in detail. The algorithm has two steps. Firstly, the influence cascade including cascade ability is produced by setting P and Q, which is fundamental in the RLIM algorithm. Moreover, the main components of the RLIM algorithm are also realized by the bridge that connects vectorized representation and influence spread.

4.1. influence spread

In an OSN, to achieve the cascade of node depth and width, the proposed RLIM algorithm regulates by setting two parameters P and Q, where P can control node depth cascade and Q control node width cascade. In the cascading process, taking into account the later IM problems, high nodes are given priority. Starting from the initial node u, the neighbor node v with a high degree is given priority and added to the influence cascade. At this time, the parameters P and Q are set, where $P = 1, Q = 0$, which means that node has performed a depth cascade. Then the maximum degree neighbor node w of node v is considered. If the degree of node w is greater than the degree of node u, then node w is selected as the next node to be added to the influence cascade, and the two parameters are set, where $P = 1, Q = 0$. If the degree of node w is smaller than the degree of the node u, then another neighbor n of the node u is selected as the next node to add the influence cascade, and the two parameters are set, where $P = 0, Q = 1$.

Let us now consider the case where the degree of node w is the same as that of node u, the values of P and Q are considered. If P=1, Q=0, it means that the last time the node was cascaded in depth, then this secondary cascade should be cascaded in width. The neighbor node n of node u should be considered to be added to the influence cascade, and the values of P and Q should be set at the same time, where $P = 0, Q = 1$. Moreover, if P=0, Q=1, it means that the last time the node was cascaded in width, then this secondary cascade should be cascaded in depth. The neighbor node w of node v should be considered to be added to the influence cascade, and the values of P and Q should be set at the same time, where $P = 1, Q = 0$. For the specific algorithm, please refer to Algorithm 1.

Algorithm 1: Influence cascade

Input: graph G= *(*V, E, t*), initial node* u
Output: influence cascade ic
1 Initialize ic *to [*u*],* P=Q=*0*
2 While true: # The termination condition is that the end node of the deep cascade is node u.
3 curr=ic *[-1]*
4 V_{curr}, D_{curr}=*Neighbor_maxdeg(*curr, G*) #Get the node with the largest degree among neighbor nodes.*
5 *Compare (*$D_{V_{curr}}$, D_{curr}*)*
6 *Determine (*P, Q*)*
7 *Append* V_{curr} *to* ic
8 *Set* P *and* Q
9 Return ic, len(ic)

The 4th to 7th lines of the algorithm pseudocode are the key parts. First, the algorithm gets the node with the largest degree among neighbor nodes (line 4). Second, the degree of the node V_{curr} just obtained is compared with the degree of the node *curr* to determine the kind of situation mentioned above (line 5), and the values of P and Q are determined (line 6). Finally, If the degree of node V_{curr} is greater than that of *curr*, and the values of P and Q meet the conditions, then V_{curr} is added to the influence cascade *ic* (line 7). The termination condition is that the end node of the deep cascade is node *u* (line 2).

4.2. influence spread

The network is preprocessed on the RLIM algorithm. First, the real network is extracted into two parts including the initial node sets with different interests: the train space network and the test space network. The former is to obtain the objective function that reflects the influence of the cascade and the cascade ability, and the latter is to maximize the influence spread. In the train space network, to construct the influence cascade, the proposed algorithm categorizes nodes and finds the initial node set. The influence cascade inputs into the neural network architecture. Pass pieces of training, the two objective functions that can express the node's influence cascade and cascade capability are generated. Finally, these objective functions are applied to the test space network to maximize the influence of the initial nodes in this network.

To improve the performance of the algorithm, the paper implemented a negative sampling method. Because RLIM uses the SoftMax function, the denominator needs to calculate the "scores" of all nodes in the window and then sum them. However, the core idea of negative sampling method is to calculate the real node pair "score" of the target node and the node in the window, and plus some "noise", which is the random data in the vocabulary and the "score" of the target node. The real node pairs "score" and "noise" as a cost function. Each time the parameters are optimized, only the node vectors involved in the cost function are concerned. The formula is given below:

$$J(\theta) = \frac{\sum_{t=1}^{T} J_t(\theta)}{T}$$

$$J_t(\theta) = log\sigma(u_o^T v_c) + \sum_{i=1}^{k} \mathrm{E}_{jP(w)}[log\sigma(-u_j^T v_c)]$$

Where k denotes that the number of samples to be sampled, u_0 denotes the vector of the initial node, and v_c denotes the vector of the target node.

The purpose of negative sampling is not to optimize the entire vector matrix I or T, but to optimize only the node vectors involved in the cost calculation process. Therefore, we also need to follow the new gradient.

$$\frac{\partial J}{\partial v_c} = (\sigma(u_o^T v_c) - 1)u_0 - \sum_{k=1}^{K} (\sigma(-u_k^T v_c) - 1)u_k$$

$$\frac{\partial J}{\partial v_0} = (\sigma(u_o^T v_c) - 1)v_c$$

$$\frac{\partial J}{\partial u_k} = -(\sigma(-u_k^T v_c) - 1)v_c, for\ all\ k = 1,2, \dots, K$$

Where u_k denotes the node vector randomly selected during negative sampling. The specific algorithm pseudocode is shown in Algorithm 2. The pseudocode consists of two parts, one is to generate the objective function f_{ic} and f_{cl} on the train set (line 2 - line 4), and the other is to calculate the influence spread on the test set (line 5 – line 8).

Algorithm 2: RLIM

Input: graph G= *(*V, E, t*), learning rate* lr*, train epochs* te*, embedding size* es*, num-neg-samples* ns*, candidate rate* β
Output: influence spread Ins
1 G_{tr}, G_{te}=*Extract_net(G)*
2 For i *in* G_{tr}*.initial_node:*
3 ic, cl=*influence cascade (*i, G_{tr}*)*
4 f_{ic}, f_{cl}=*NNG (*ic, cl, G_{tr}, lr, te, es, ns*)*
5 Candidate node_set= $\beta(G_{te}$*.initial_node)/100*
6 For j *in Candidate node_set:*
7 $\sigma(f_{ic}[j], f_{cl}[j])$
8 Ins=*Greedy (*σ*)*
9 Return Ins

5. Experiments

This section introduces the advantages of the proposed algorithm in classification visualization and influence propagation. A total of four parts are described. Firstly, four OSNs datasets are introduced, including Digg, Flickr, Sina Weibo and Microsoft Academic Graph (MAG). The number of nodes in the data set ranges from 170 thousand to 1.4 million, and the number of connected edges is between 10 million and 20 million. Secondly, the setting parameters involved in the proposed algorithm and the operating environment of the experiment are introduced. Moreover, we introduced the seven algorithms involved in the comparison. The Deepwalk, LINE and node2vec algorithms are used for classification visualization comparison experiments, and the IMINFECTOR, Inf2vec, CELF and IMM algorithms are used for influence spread comparison experiments. Finally, the results of the classification visualization and influence spread are displayed in the form of graphs and tables. We describe the results of the experiment in detail, analyze the reasons for the results, and make a summary based on these experimental results.

5.1. Datasets

Flickr [36] is a social network where users share pictures and videos. In this data set, each node is a user in Flickr, and each edge represents the friendship between users. Moreover, each node has a label to identify the user's interest group.

Digg [37] dataset contains data about stories promoted to Digg's front page over a month. For each story, the dataset collected the list of all Digg users who have voted for the story up to the time of data collection, and the timestamp of each vote. Moreover, the voters' friendship links were also retrieved.

Sina Weibo [38] dataset was crawled in the following ways. To begin with, 100 random users were selected as seed users, and then their followers. The crawling process produced a total of 1.1 million users and 0.2 billion following relationships among them, with an average of 200 followers per user. For each user, the crawler collected her 1,000 most recent microblogs (including tweets and retweets).

The MAG [39] is a heterogeneous graph containing scientific publication records, citation relationships between those publications, as well as authors, institutions, journals, conferences, and fields of study.

These detailed dataset contents are shown in Table 1.

Table 1. The content about real datasets

	Flickr	**Digg**	**Sina Weibo**	**MAG**
nodes	162,663	279,631	1,170,689	1,436,158
edges	10,226,532	2,251,166	225,877,808	20,456,480
cascades	2,173	3,553	115,686	181,020
avg cascade size	63	148	847	29

5.2. Parameter setting

To achieve the best experimental results, the paper sets 80% real-world dataset as the training dataset, and the remaining 20% as the test dataset. Moreover, In the NNA, the learning rate *lr* defaults to 0.1, the train epochs *te* defaults to 100, the embedding size *es* defaults to 50, and the negative sampling rate *ns* defaults to 10. Finally, the candidate rate β that can indicate the rate of nodes involved in the influence spread is set to 40.

All algorithms are implemented in Python and experiments are conducted on a windows server with a 2.90 GHz quad core Intel i5-10400 CPU machine with 8.00 Gb memory.

5.3. Evaluated algorithms

Deepwalk [29]: DeepWalk is a proposed method for node representation learning in social networks. This method is only applicable to pure social networks, not to the OSNs that include node properties. For each node, a random walk is used to generate the context, and a skip-grammar model that can realize the learning of word vector representation is used to realize the vectorized representation of the node.

LINE (2nd) [33]: LINE is a method based on the assumption of neighborhood similarity, which can be seen as an algorithm that uses BFS to construct neighborhoods. Moreover, LINE can also be used in weighted graphs. However, the LINE method only considers the first-order and second-order similarities, and insufficiently utilizes high-order information.

Node2vec [31]: Node2vec is a further step based on DeepWalk. By adjusting the weight of random walk, the results of graph embedding are weighed in the homophily of the network and structural equivalence. Furthermore, in the task of node classification, Node2vec's effect is better than the previous algorithm.

CELF [7]: The CELF algorithm takes advantage of the submodular of the function. When the seed node is selected in the first round, the marginal revenue of all nodes in the network is calculated, but in the subsequent process, the marginal revenue of the network node will not be double-calculated. Compared with the traditional greedy algorithm, it will get a very obvious improvement in time.

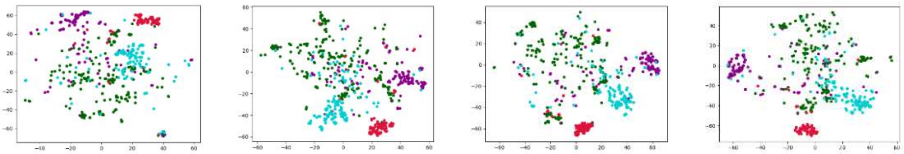
IMM [28]: IMM has higher empirical efficiency compared with the many algorithms, but achieves the same approximation guarantee through the algorithm based on martingales. In large social networks, both the scope and efficiency of influence spread must be taken into account. Only a unilateral improvement cannot get a good response in practical applications.

Inf2vec [34]: Inf2vec algorithm is a method to learn node representation. The novelty of the algorithm is that the generated context combines local influence and global user similarity. Previous work did not consider user interest in learning influence parameters. However, the application of this algorithm to IM problems is not particularly ideal.

IMINFECTOR [35]: IMINFECTOR realizes the ability to connect influence expression and influence maximization by the representation learning method. In the paper where the algorithm is located, it is proposed that there is a gap between the estimated influence propagation and the actual influence propagation. The flaw of this algorithm is that it uses a normal random walk method to obtain the cascade of nodes.

5.4. Experiment results

Classification visualization. The results of classification visualization can reflect the distribution of different types of users, and the aggregation of users of the same type is a standard to measure the learning algorithm. To compare with different representation learning, the paper made a visual classification map. The part Digg dataset was selected as the network graph for classification visualization. Especially, we categorize people who voted on the same story and have the same label. On this basis, the voting users of the four stories are shown in the experiment as the target of classification. Moreover, the colors of the nodes refer to different types of users in figure 3.



(a)Deepwalk (b)LINE (c)Node2vec (d)RLIM

Figure 3. Visualization results of the part dataset of Digg network. Each participating user is mapped to the 2-D space using t-SNE package. The colors of the nodes refer to different types of users.

Figure 3 presents the classification visualization of Deepwalk, LINE, node2vec and RLIM algorithms. In figure 3 (a), the distribution of users with the same type is discrete, and users with different types are mixed distribution, and there is almost no boundary. In figure 3 (b) and (c), users of with same the type tend to concentrate. Although users of different types are still partially mixed, they have blurred boundaries. In Figure 3 (d), the distribution of users with the same type is relatively concentrated, and different types of users have relatively clear boundaries, and only a small number of users be mixed. It can be seen that the result of our RLIM algorithm is the best. Followed by the LINE and node2vec algorithms. Therefore, the Deepwalk algorithm performs poorly in classification tasks. To analyze the classification of different algorithms more accurately, we introduce the Micro-F1 and Macro-F1 standard. Table 2 shows the results of classification on the part Digg network.

Table 2. Results of classification on the part Digg network

	Deepwalk	**LINE**	**Node2vec**	**RLIM**
Micro-F1	0.165	0.181	0.193	0.201
Macro-F1	0.179	0.187	0.190	0.194

From the above analysis, it can be concluded that to achieve a better classification effect, in addition to the basic network structure, the proprieties of each user, such as user interests, must also be considered. In terms of node representation learning, although the

Deepwalk, LINE and node2vec algorithms have been continuously strengthened, these algorithms have not taken into account the user's proprieties, so satisfactory results have not been achieved. On the other hand, the RLIM algorithm performs well in the classification effect because of the user's interest. From the side, the vector representation obtained by the RLIM algorithm is closer to the real situation, so that a more real influence spread can be obtained.

Influence spread. The experiment selected 10 to 100 size seed sets to spread the influence on different data sets. The difference in the results of the experiment indicates the difference in the effect of the algorithm. Moreover, In the experiment performed, the wider the influence spread, the more advantageous the algorithm is in the case of the same seed size. Figure 4 shows the experimental results of influence propagation. The lines with different colors in the figure represent different algorithms. Furthermore, the higher the line in each picture, the wider the spread of the algorithm represented by this line. The detailed influence spread is shown in figure 4.

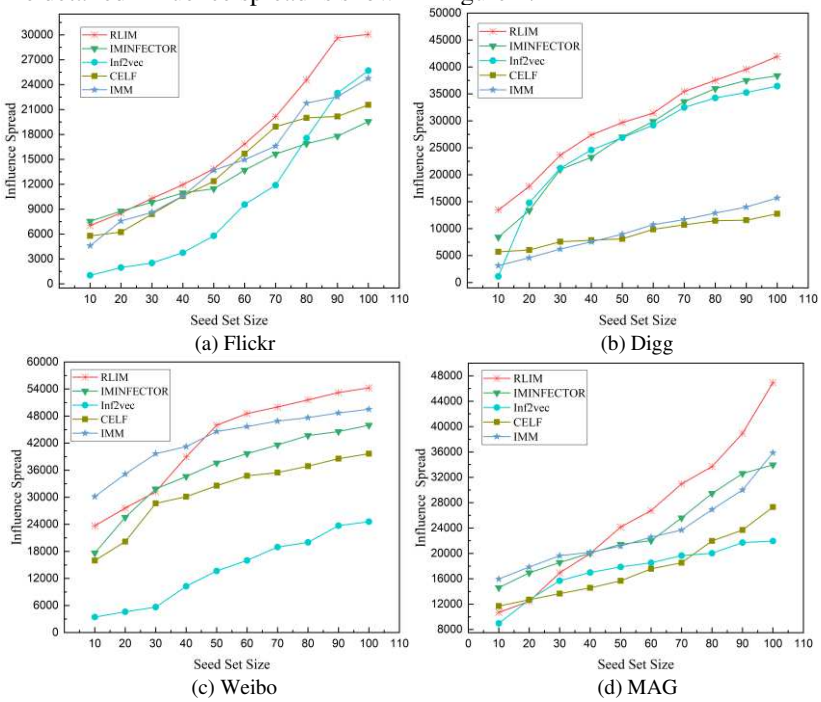


Figure 4. The influence spread of different algorithms on the four OSNs.

Figure 4 shows the influence spread of the RLIM, IMINFECTOR, Inf2vec, CELF and IMM algorithms on the Flickr, Digg, Weibo and MAG datasets. The red line represents the influence spread performance of the RLIM algorithm, the green line represents the influence spread performance of the IMINFECTOR algorithm, the blue line represents the influence spread performance of the Inf2vec algorithm, the brown line represents the influence spread performance of the CELF algorithm, and the dark blue line represents the influence spread of the IMM algorithm. Influence spread performance. Moreover, from the perspective of changing trends, the influence spread generated by different algorithms increases as the size of the seed set increases. On the whole, the RLIM algorithm represented by the red line has a relatively good influence on spreading performance in each data set.

Figure 4 (a) shows the influence spread performance of different algorithms on the Flickr dataset. The influence spreading ability of the RLIM algorithm is always at a high level except when the seed set size is 10 and 20. Moreover, compared with the Inf2vec algorithm, the influence spreading ability of the RLIM algorithm is increased by about 5 times at the highest and 16% at the lowest.

In figure 4 (b), the influence spread of the RLIM algorithm has an absolute advantage, regardless of the size of the seed set on the Digg dataset. Moreover, compared with the IMM algorithm, the influence spreading ability of the RLIM algorithm is increased by about 3.2 times at the highest and 1.6 times at the lowest.

In figure 4 (c) and (d), when the size of the seed set is relatively small, the RLIM algorithm shows a small flaw, which is the influence spread ability that is not particularly strong. The reason for this phenomenon is that the seed set of the same size can be representative in a relatively small data set, but this representativeness will be weakened as the data set increases. However, compared with the CELF algorithm, the influence spreading ability of the RLIM algorithm is improved by about 41% on average in figure 4 (c) and about 16% on average in figure 4 (d).

In general, the performance of the RLIM algorithm in the four datasets is quite satisfactory compared to the cutting-edge algorithms. Through the proposed algorithm, influence spread has been greatly improved. From figure 4, we can find that the influence spread obtained by the RLIM algorithm not only spreads well in the existing seed set size, but also from the change trend analysis, the influence spread effect will be better for the larger seed set.

6. Conclusion

The proposed new algorithm RLIM, which includes the establishment of influence cascade, predicts propagation ability and maximizing influence spread. The key is to adopt NNA to realize the prediction of propagation ability, including the vectorized representation of cascade nodes and cascade capabilities. Furthermore, we conduct experiments on four OSNs data sets. The experimental results show that the RLIM algorithm can not only be closer to the actual situation in the vectorized representation of nodes, but also can achieve the optimal influence spread in large social networks.

Several interesting directions for future work are shown below. First, In the process of influencing the cascade, multiple node attributes can be considered, such as the input degree and output degree of the node. Second, the direct combination of representation learning technology and influence maximization is the focus of future research work.

References

[1] Vega L , Mendez-Vazquez A , A López-Cuevas. Probabilistic reasoning system for social influence analysis in online social networks[J]. Social Network Analysis and Mining, 2021, 11(1):1-20.

[2] CosciaMichele. Noise Corrected Sampling of Online Social Networks[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2021.

[3] Chen T , Liu B , Liu W , et al. A Random Algorithm for Profit Maximization with Multiple Adoptions in Online Social Networks[J]. 2021.

[4] E Güney, Leitner M , Ruthmair M , et al. Large-scale influence maximization via maximal covering location[J]. European Journal of Operational Research, 2021, 289.

[5] Chen S J , Chen W K , Dai Y H , et al. Efficient presolving methods for influence maximization problem in social networks[J]. 2021.

[6] Kempe D , Kleinberg J , Tardos E . Maximizing the Spread of Influence through a Social Network[J]. Theory of Computing, 2003, 137-146(4).
[7] Leskovec J , Krause A , Guestrin C E , et al. Cost-effective outbreak detection in networks. 2007.
[8] Goyal A , Wei L , Lakshmanan L . CELF++: optimizing the greedy algorithm for influence maximization in social networks. ACM, 2011.
[9] Galhotra S , Arora A , Roy S . Holistic Influence Maximization: Combining Scalability and Efficiency with Opinion-Aware Models. ACM, 2016.
[10] Peng S , Wang G , Xie D . Social Influence Analysis in Social Networking Big Data: Opportunities and Challenges[J]. IEEE Network, 2017, PP(1):12-18.
[11] Pal S K , Kundu S , Murthy C A . Centrality Measures, Upper Bound, and Influence Maximization in Large Scale Directed Social Networks[J]. Fundamenta Informaticae, 2014, 130(3):317-342.
[12] Wang F , Li J , Jiang W , et al. Temporal Topic-Based Multi-Dimensional Social Influence Evaluation in Online Social Networks[J]. Wireless Personal Communications, 2017.
[13] Deng X , Dou Y , Lv T , et al. A Novel Centrality Cascading Based Edge Parameter Evaluation Method for Robust Influence Maximization[J]. IEEE Access, 2017:1-1.
[14] Saito K , Nakano R , Kimura M . Prediction of Information Diffusion Probabilities for Independent Cascade Model[J]. Springer-Verlag, 2008.
[15] Cao T , Wu X , Hu T X , et al. Active Learning of Model Parameters for Influence Maximization[J]. Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2011.
[16] Goyal A , F Bonchi, Lakshmanan L . A Data-Based Approach to Social Influence Maximization[J]. Proc. VLDB Endow. 2011, 5(1):73-84.
[17] Ozdenizci O , Eldeeb S , Demir A , et al. EEG-based Texture Roughness Classification in Active Tactile Exploration with Invariant Representation Learning Networks[J]. 2021.
[18] Sriram A , Muckley M , Sinha K , et al. COVID-19 Deterioration Prediction via Self-Supervised Representation Learning and Multi-Image Prediction. 2021.
[19] Li Z , X Wang, Li J , et al. Deep attributed network representation learning of complex coupling and interaction[J]. Knowledge-Based Systems, 2021, 212(1):106618.
[20] Wang J , Cherian A . Discriminative Video Representation Learning Using Support Vector Classifiers. IEEE, 2021.
[21] Goyal P , Benner P . LQResNet: A Deep Neural Network Architecture for Learning Dynamic Processes. 2021.
[22] Urda D , Veredas F J , J González-Enrique, et al. Deep neural networks architecture driven by problem-specific information[J]. Neural Computing and Applications, 2021(7553).
[23] Picone R , Webb D , F Obierefu, et al. New methods for metastimuli: architecture, embeddings, and neural network optimization[J]. 2021.
[24] Adineh M , Nouri-Baygi M . High Quality Degree Based Heuristics for the Influence Maximization Problem. 2019.
[25] Guille A , Hacid H , Favre C , et al. Information Diffusion in Online Social Networks: A Survey[C]// ACM. ACM, 2013:17-28.
[26] Saito K , Kimura M , Ohara K , et al. Selecting Information Diffusion Models over Social Networks for Behavioral Analysis. Machine Learning and Knowledge Discovery in Databases, 2010.
[27] Bourigault S , Lamprier S , Gallinari P . Representation Learning for Information Diffusion through Social Networks: an Embedded Cascade Model[C]// the Ninth ACM International Conference. ACM, 2016.
[28] Tang Y , Shi Y , Xiao X . Influence Maximization in Near-Linear Time: A Martingale Approach. ACM, 2015.
[29] Perozzi B , Al-Rfou R , Skiena S . DeepWalk: Online Learning of Social Representations. ACM, 2014.
[30] Mikolov T , Chen K , Corrado G , et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.
[31] Grover A , Leskovec J . node2vec: Scalable Feature Learning for Networks[J]. ACM, 2016.
[32] Kumhar S H , Kirmani M M , Sheetlani J , et al. Word Embedding Generation for Urdu Language using Word2vec model[J]. Materials Today: Proceedings, 2021(8).
[33] Tang J , Qu M , Wang M , et al. LINE: Large-scale information network embedding[J]. International Conference on World Wide Web Www, 2015.
[34] Feng S , Cong G , Khan A , et al. Inf2vec: Latent Representation Model for Social Influence Embedding[C]// 2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE, 2018.
[35] Panagopoulos G , Malliaros F D , Vazirgiannis M . Influence Maximization via Representation Learning[J]. 2019. [30] Mikolov, T. , Chen, K. , Corrado, G. , & Dean, J. . (2013). Efficient estimation of word representations in vector space. Computer Science.
[36] http://www.datatang.com/data/13785.html
[37] https://www.isi.edu/~lerman/downloads/digg2009.html

[38] https://www.aminer.cn/influencelocality
[39] https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

Figures

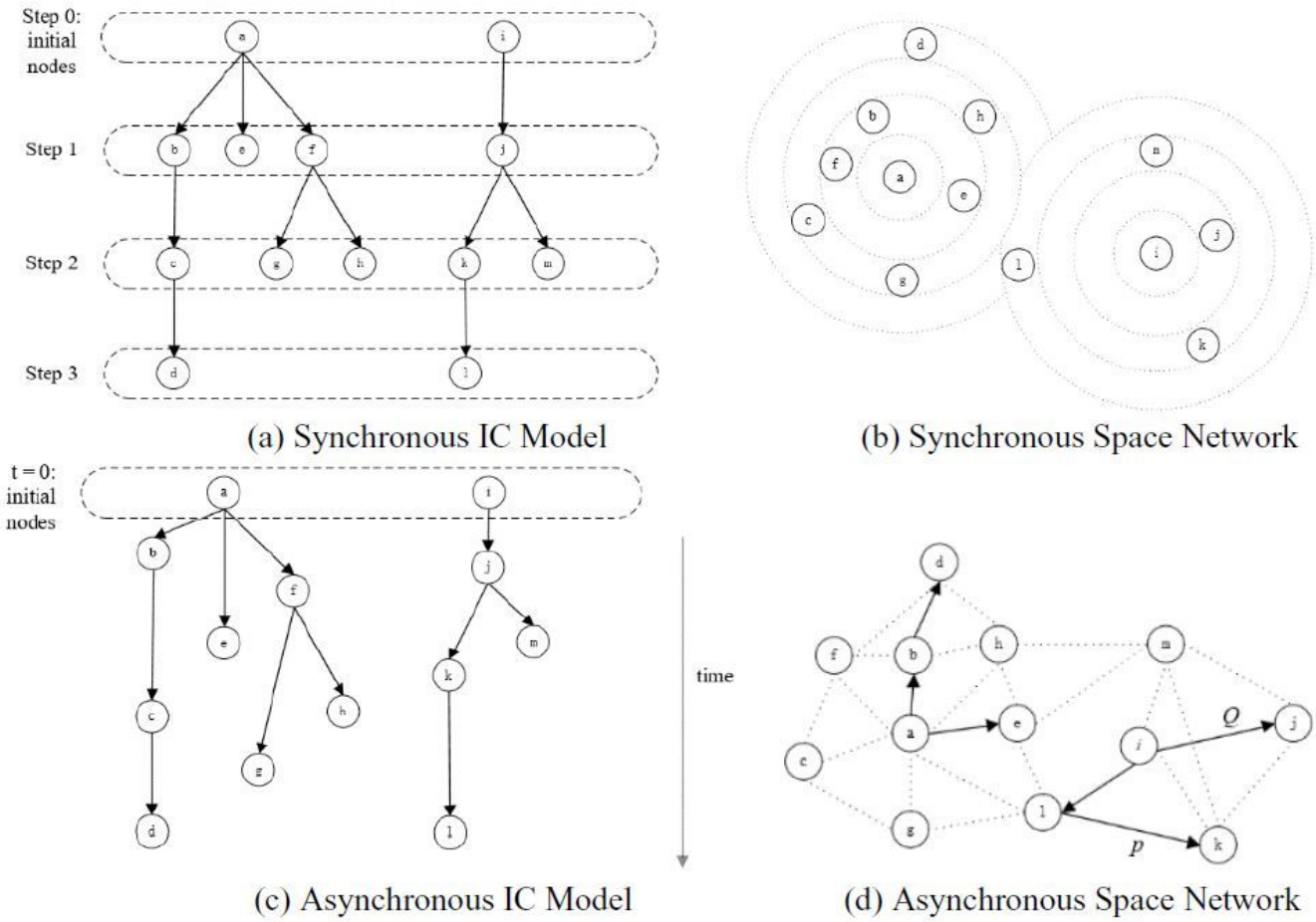


Figure 1

(a) The forwarding of information is carried out within a time step represented by a natural number. (b) The positions of nodes in the network are relative, and nodes in the same circle have the same level of influence probability. (c) The time interval for the information to be forwarded to neighboring nodes is a continuous random variable. (d) The information flow between nodes is not fixed, and the propagation direction is determined by P and Q.

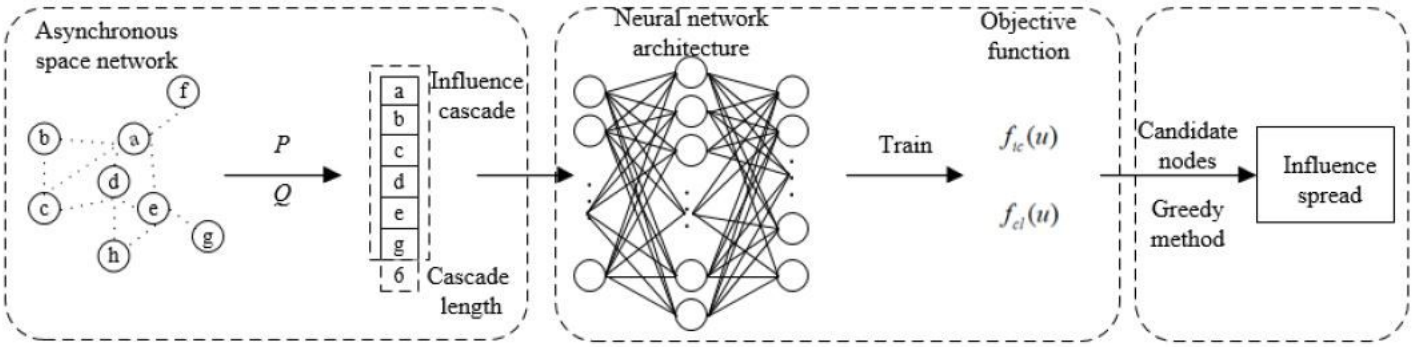


Figure 2

Representation learning method framework.

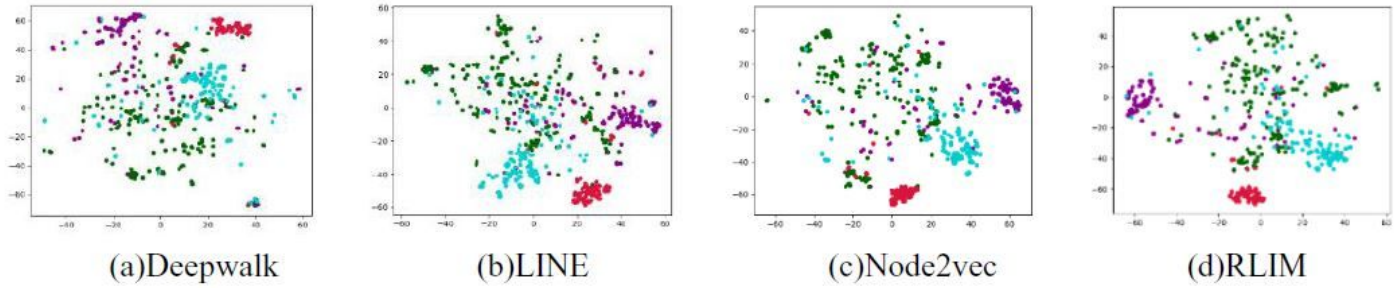


Figure 3

Visualization results of the part dataset of Digg network. Each participating user is mapped to the 2-D space using t-SNE package. The colors of the nodes refer to different types of users.

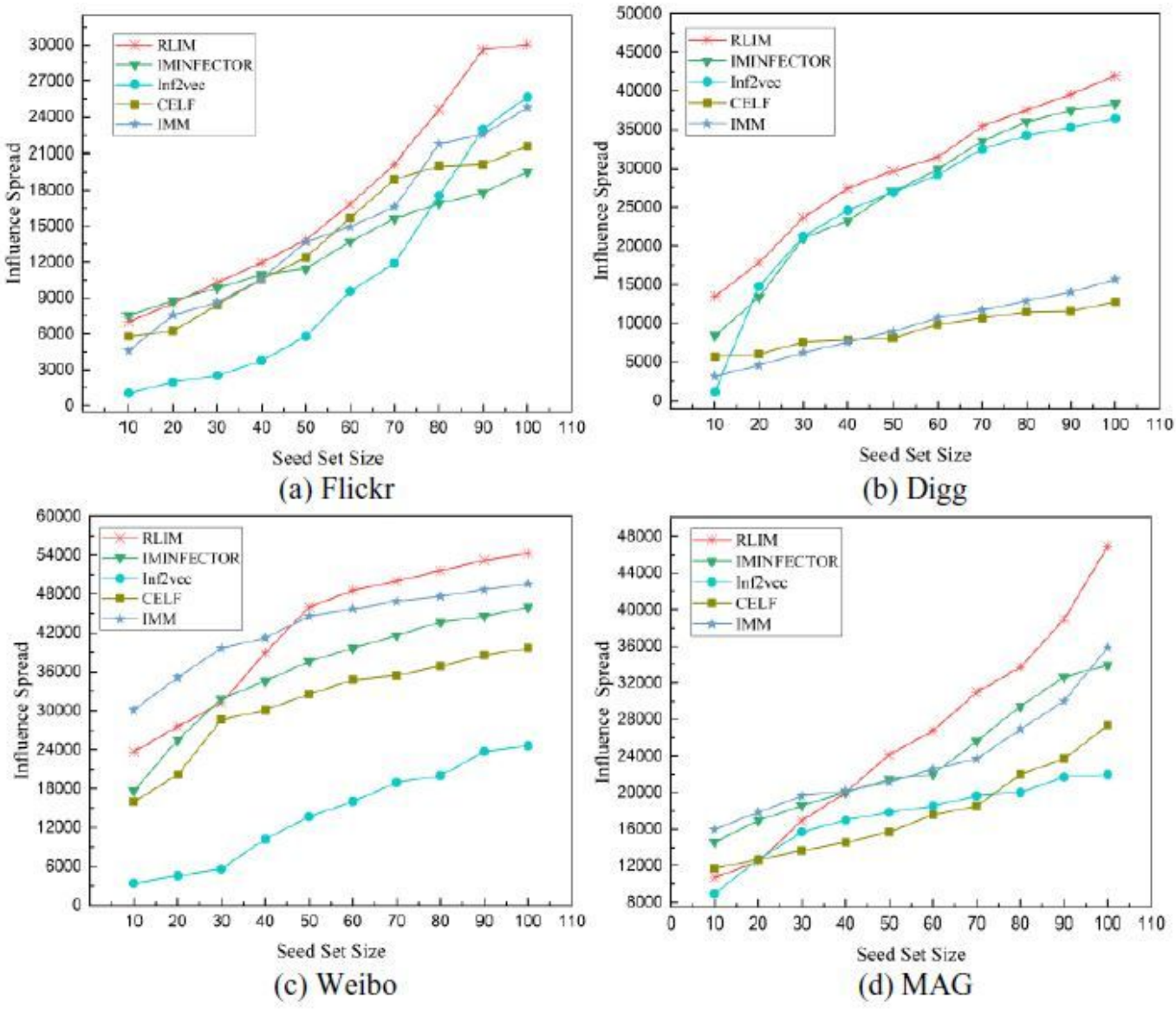


Figure 4

The influence spread of different algorithms on the four OSNs.