RLIM: representation learning method for influence maximization in social networks

Chengai Sun · Xiuliang Duan · Liqing Qiu · Qiang Shi · Tengteng Li

Received: 29 March 2021 / Accepted: 22 June 2022 / Published online: 29 July 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
A core issue in influence propagation is influence maximization, which aims to find a set of nodes that maximize influence spread by adopting a specific information diffusion model. The limitation of the existing algorithms is they excessively depend on the information diffusion model and randomly set the propagation ability. Therefore, most algorithms are difficult to apply in large-scale social networks. A method to solve the problem is neural network architecture. Based on the architecture, the paper proposes Representation Learning for Influence Maximization (RLIM) algorithm. The algorithm consists of three main parts: the influence cascade of each source node is the premise; the multi-task deep learning neural network to classify influenced nodes and predict propagation ability is the fundamental bridge; the prediction model applying to the influence maximization problem by the greedy strategy is the purpose. Furthermore, the experimental results show that the RLIM algorithm has greater influence spread than the state-of-the-art algorithms in different online social network datasets, and the information diffusion 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 multiple iterations of this information transmission are information diffusion. The Independent Cascade (IC) model and Linear Threshold (LT) model are often used to simulate information diffusion. The core issue of information diffusion research is to predict the propagation probabilities. One of the purposes of studying information diffusion is to solve the Influence Maximization (IM) [4, 5] problem. The IM problem is to find a fixed number of active nodes and uses a specific diffusion model to maximize the number of active nodes.

Over the past years, plenty of research has been conducted 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 express the IM problem and prove that the optimal solution to this problem is NP-hard. Moreover, they present a general greedy hill-climbing algorithm with the error bounded by \(1 - 1/e - \epsilon\), where \(\epsilon\) denotes the error generated by using the Monte Carlo (MC) simulation to evaluate the influence spread. Similarly, to solve the efficiency problem in the IM, many heuristic algorithms have been proposed. The most famous method of selecting seeds is based on the degree of nodes [11]. However, in terms of the number of final activated nodes, the heuristic algorithms usually perform worse than the greedy algorithm. Furthermore, the propagation probability in the process of information diffusion is usually set a random or even a fixed value, which is not accurate. RLIM algorithm predicts the propagation probability by integrating the properties and structural characteristics of nodes, which can ensure that the information propagation is closer to the actual situation.

In information diffusion, the user will have a certain tendency of forwarding behavior when receiving information.
This propagation probability to the other user is not set by a fixed or random, but by modeling and predicting this trend. Saito et al. [14] focus on the IC model and propose a method to predict the propagation probability based on the logarithm of the past propagation. Cao et al. [15] study the IM problem under the LT model with unknown diffusion model parameters. Goyal et al. [16] estimate propagation probability by utilizing historical data, thereby avoiding the need for learning influence probabilities through expensive MC simulations. These approaches adopt a pair-wise manner to simulate the influence probability without considering other factors, such as the user interest. RLIM algorithm considers the user’s interest in events.

In the previous works about the IM algorithm, the running time of heuristic algorithms is the fundamental research factor. 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 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 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 real-world information diffusion may occur between unknown connections. Second, the activation probability between nodes cannot be set according to the influence ability of the nodes. These problems have greatly interfered with the estimated influence spread. RLIM algorithm uses \( P \) and \( Q \) parameters to extract node influence cascades from space network, instead of traditional diffusion model to simulate the process of information propagation.

To break through the limitations of existing algorithms, this paper proposes the RLIM algorithm. This algorithm requires three aspects of research 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 ability, and it is the key to narrowing the gap between estimated and actual influence spread. Influence diffusion is a way to maximize influence spread based on the vectorized influence ability.

This paper proposed a novel method for constructing influence cascades. The node’s vector representation obtained by random walk often ignores the local property. In other words, only one node among multiple direct neighbors of the initial node may participate in the context cascade to construct the influence cascades. Therefore, it is necessary to control the cascade direction to ensure that the source node can consider both global property and local property. Inspired by node2vec, the influence cascades of the initial node are constructed by the combination of BFS and DFS. The RLIM algorithm controls the cascade direction through the \( P \) and \( Q \) parameters. In addition, considering the actual information propagation, the process of information transforms not only occurs in the existing connections. If the users’ influence is relatively large, even if the connection between the user does not exist, a new connection may be created to spread the information more widely.

RLIM algorithm absorbs the core technology of current social network research and becomes a powerful tool for solving IM problems in large-scale social networks. Currently, the application of representation learning technology [17, 18] is quite mature. In OSNs, node representation learning generally adopts a vector to represent a node. The key of this technique is to generate the nodes’ context appropriately and use it to realize the vectorization of nodes. Based on the technology, the influence cascades can be constructed by analogy to the nodes’ context.

Although the traditional representation learning [19, 20] method can achieve vector representation of nodes, it still has application limitations for the IM problem such as the millions of influence cascades and optimal solution. Therefore, another vectorized representation method (neural network architecture) is considered. The Neural Network Architecture (NNA) [21–24] has three advantages to realizing the vectorized representation, including the adaptation to massive data, the super computing power, and the advanced algorithm support. With these advantages, the RLIM algorithm can easily classify influenced nodes and predict propagation ability.

In summary, the NNA is used to solve the problem of vectorized representation of nodes in a large-scale social network, which is the most suitable method for the development of this field. 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 construction of influence cascade, the prediction of the propagation probability and the simulation of the information diffusion. Among them, the key factor of the framework is how to construct 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 indicate the propagation behavior that has not occurred. By this way, the actual propagation situation is reflected by the proposed algorithm.
- The representation learning method is used to maximize influence spread and solve the problem that traditional
IM algorithms cannot be applied to large-scale OSNs. Moreover, the paper designs classification visualization experiment and influence spread experiment, which respectively prove that the RLIM algorithm is close to actual 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, Sect. 6 concludes the paper and gives some directions of the future works.

2 Motivations and related works

2.1 Motivation

With the vigorous development of online social networks, a large amount of real-world data is produced, which is a huge challenge for traditional research algorithms. The traditional IM algorithms include 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 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 [25] sets the propagation probability to a fixed value of 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.

It is excellent that representation learning methods solve the influence maximization problem, to obtain the maximum influence spread and stable application in large social networks. First, the information propagation process of nodes no longer depends on a fixed diffusion model, and can be realized by the vectorized representation of the node’s cascade. Second, the controllability of the cascade direction ensures that the information propagation is closer to the actual process. Finally, the propagation ability between nodes can be realized through model prediction, which gets rid of the manual set.

2.2 Related work

Influence maximization problem. 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 the existing connections and the 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).

Figure 1(a) and (c) are two classic IC models Figure 1(a) presents the synchronous IC model [26], 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 [27], 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 the basic models that many algorithms rely on. However, in the process of the traditional information diffusion, how the propagation probability reflects the reality remains a difficult problem. Therefore, the construction of models that reflect actual information diffusion is a key research direction.

Figure 1(b) and (d) as Space Network are subsequently proposed

Figure 1(b) presents the synchronous space network [28], 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 an 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.

In addition, the original selection method of the seed set is the 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
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. In addition, the application of estimation techniques based on classic statistical tools (martingales) to solve IM problems is a brand-new framework. Such as the IMM algorithm [29], which can not only provide accurate results with low computational complexity, 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 [30–32] still cannot play an important role in large social networks. Therefore, to solve the IM problem, a method to meet the development of OSNs is urgently proposed.

**Representation learning method.** Currently, representation learning is widely applied in the analysis of social networks [33–35]. The biggest feature of presentation learning is that the network structure and node properties can be captured by the node vector. This is why presentation learning has become a desirable 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. [36] proposed the DeepWalk algorithm, which is an algorithm that generates context with random walks and then updates the representations with skip-gram [37]. 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 [38], 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.

Simultaneously, a large number of research methods have emerged on information diffusion. To simulate the information diffusion in OSNs, the most important problem is to deduce the propagation probability between nodes, which is fundamental to the IM problem. Goyal et al. proposed a method using co-occurrence counting to estimate the probability of propagation. Another method adopts the word2vec
technique, which improves word representation learning. It is called word embedding in the Natural Language Processing (NLP) [39]. Tang et al. designed the LINE algorithm [40], which preserved both the local and global network structure by using first-order and second-order proximity. To consider influence propagation and similarity of user interest, Feng et al. proposed the Inf2vec algorithm [41], 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 proposed by Panagopoulos et al. [42]. This algorithm utilizes multi-task neural network architecture to calculate, and can vectorize the node sequence and sequence length. Inspired by this, the proposed RLIM algorithm combined neural network architecture and similarity of user interest. Moreover, for the information propagation to be more suitable for the actual situation, the RLIM algorithm used the parameters of $P$ and $Q$ to extract the influence cascade of nodes in the asynchronous space network.

**Greedy method.** 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 sub-modularity 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 a process of forming a node sequence using $P$ and $Q$ control direction. Moreover, the propagation probability is calculated utilizing a combination of the NNA and node interest similarity. The NNA can 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 candidate node takes 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 with the same interest. Similarly, the construction of node influence cascade is also a fundamental part of the proposed method. Furthermore, the second part calculates the vectorized representation of influence cascade and propagation probability. In this section, the term for propagation possibility is called cascade length. Finally, the ultimate purpose of the proposed method is to maximize the influence spread. Especially, the combination of the asynchronous space network and the neural network is the focus of the paper. To better illustrate the proposed method framework, Fig. 2 shows a toy example. For better illustration, the key notations are summarized in Table 1.

#### 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 an 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 a multifarious
influence cascade is a key issue. Suppose there is an initial cascade node $u$ and an unfixed cascade length, where $u \in V$. The direction of the cascade is determined by $P$ and $Q$. Moreover, the depth or width of the node walk conforms to the following binomial distribution.

**Theorem 1** The paper assumes that the cascade is $n$ times, and the probability of the deep walk ($P$) occurring once is $p$.

$$P(p | n) = \frac{p^n}{\int_0^1 p^d d_p} = (n + 1)p^n. \quad (1)$$

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

$$P(k | n, p) = C_n^k p^k (1 - p)^{n-k}. \quad (2)$$

Therefore,

$$P(p | n, k) \propto P(p)P(k | n, p), \quad (3)$$

taking conjugate prior,

$$p \sim Be(\alpha, \beta). \quad (4)$$

Finally,

$$P(p | n, k) \propto p^{(\alpha + n - 1)}(1 - p)^{\beta - 1}. \quad (5)$$

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

$$\int_0^1 p \cdot (n + 1)p^n = \frac{n + 1}{n + 2}. \quad (6)$$

The wide walks ($Q$) appears one time is $q$, where $q = 1 - p$.

When the selection of the walking direction encounters special cases, the algorithm is determined according to the following definition. Moreover, the detailed description will be shown in Sect. 4.1.

$$W_{nex} = \begin{cases} Q, P \ast q \geq Q \ast p \\ P, P \ast q < Q \ast p \end{cases} \quad (7)$$

where $W_{nex}$ represents the direction of the next cascade. The purpose of this is to maximize the influence of initial cascade nodes.

### 3.2 Vectorized representation

The subsection implements the vectorization of cascade nodes and cascade capabilities. In particular, the cascade capacity refers to the cascade length. Given a message $m$, we construct $G^m = (u, v, t)$, where each tuple $(u, v, t)$ denotes node $v$ receives the information from node $u$ at time $t$. Under the same interesting information $m$ 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 $C$ represents the cascade length of the user. When applying the NNA, we define the hidden layer function, output function, and loss function respectively.

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

**Theorem 2** The hidden layer function of the cascade node is defined as follows.

$$g_{v,w} = I_u T + b_v, \quad (8)$$

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

$$f_{vc}(t_v) = \frac{\exp(g_{v,c})}{\sum_{w \in G^m} \exp(g_{v,w})}, \quad (9)$$

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

$$\mathcal{L}(y, p(y|x)) = -\log p(y|x). \quad (10)$$

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 $y$, the probability of $y$ is $p(y|x)$.
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:

\[ \mathcal{L}_c = -\log (f_c(I_u)) \]  
(11)

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

**Theorem 3** The hidden layer function of the cascade node is defined as follows.

\[ g_{c,u} = I_u C + b_c, \]  
(12)

where \( b_c \) denotes bias. Moreover, the output layer function \( f_c \) utilizes Sigmoid function.

\[ f_c(I_u) = \frac{1}{1 + \exp(g_{c,u})}, \]  
(13)

Furthermore, the paper uses the Quadratic Loss Function in the definition of loss function.

\[ \mathcal{L}_c = (y_c - f_c(I_u))^2, \]  
(14)

where \( y_c \) denotes the cascade ability of the initial node \( u \).

### 3.3 Influence spread

After the NNA is trained on the datasets, the paper gets the function \( f_{ic} \) that reflects the influence of the cascade and the function \( f_{cl} \) that reflects the cascade length.

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

\[ Pr_u = \begin{bmatrix} f_{ic}(I_u T)_{1,1:N} \\ \vdots \\ f_{ic}(I_u T)_{1,1:N} \end{bmatrix}, \]  
(15)

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 \( \alpha \).

\[ \alpha_n = \frac{N|I_u|^2}{\sum_{\nu \in \mathcal{M}}|I_u|^2}, \]  
(16)

where \( \mathcal{M} \) 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 \( \sigma(S) \).

\[ \sigma(S) = \sum_j Pr_{ij}, \]  
(17)

where \( Z = \alpha_n \). The RLIM algorithm maximizes \( \sigma(S) \) in a greedy manner.

To prove the convergence of the RLIM algorithm, we give a proof that the algorithm is monotonic increasing.

**Proof** The influence propagation is monotonic increasing.

\[ \sigma(S^k \cup \{v\}) = \sum_{s \in S^k \cup \{v\}} \sum_j Pr_{sj} = \sum_{s \in S^k} \sum_j Pr_{sj} + \sum_j Pr_{vj} \]

\[ = \sigma(S^k) + \sum_j Pr_{vj} \geq \sigma(S^k). \]

(18)

If and only if the addition of node \( v \) as a seed to the seed set does not produce new influence propagation, \( \sum_j Pr_{vj} = 0 \), Eqs. (18) takes the equal.

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

### 4 Proposed RLIM algorithm

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. First, the influence cascade including cascade capability is produced by setting \( P \) and \( Q \), which is fundamental in the RLIM algorithm. Second, the main component of RLIM algorithm is the bridge that connects vectorized representation and influence spread.

#### 4.1 Constructing influence cascade

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 \( Q \) can control node width cascade. In the cascading process, considering the later IM problems, the high-degree 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. Simultaneously, the parameters \( P \) and \( Q \) are set, where \( P = 1, Q = 0 \), which means that the 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 $P = 1, Q = 0$. Figure 3 shows this case, where the green line indicates the constructed influence cascade line, and the red line indicates the connection that will be added to the influence cascade.

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 $P = 0, Q = 1$. Figure 4 shows this case, where the dashed line represents an unknown connection in the data set.

Let us now consider the special case (initial node $s$) where the degree of the node $w$ is the same as that of the 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. Figure 5 shows this case. 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. Figure 6 shows this case. 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 $P = 1, Q = 0$.

The influence cascaded is $s \rightarrow u \rightarrow v \rightarrow w$.

For the specific algorithm, please refer to Algorithm 1.
The 2th to 9th 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) by Eqs. (7). 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 RLIM algorithm

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 application of the former is to obtain the objective function that reflects the influence cascade and the cascade capability, 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 implements 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:

\[
J(\theta) = \frac{\sum_{i=1}^{T} J_i(\theta)}{T},
\]

\[
J_i(\theta) = \log \sigma(\theta_{0i}^T v_c) + \sum_{i=1}^{k} E_{jP(ni)} \left[ \log \sigma\left(-u_j^T v_c\right) \right],
\]

where \( k \) denotes 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 \( T \) was trained by Eqs. (8)–(12), but to optimize only the node vectors involved in the cost calculation process. Therefore, we also need to follow the new gradient.

\[
\frac{\partial J}{\partial v_c} = (\sigma(\theta_{0i}^T v_c) - 1) u_0 - \sum_{k=1}^{K} (\sigma(-u_k^T v_c) - 1) u_k,
\]

\[
\frac{\partial J}{\partial v_0} = (\sigma(\theta_{0i}^T v_c) - 1) v_c,
\]

\[
\frac{\partial J}{\partial u_k} = -(\sigma(-u_k^T v_c) - 1) v_c, \text{ for all } k = 1, 2, \cdots, K,
\]

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 \( f_c \) and \( f_{sl} \) on the train set (line 2–line 4), and the other is to calculate the influence spread on the test set (line 6–line 9).
13:3425–3440

Table 2

|          | 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 |

**Time complexity.** Before the iteration starts, because the propagation ability of each candidate seed needs to be sorted, the time complexity required by the algorithm to sort the nodes is $O(M \cdot \log M)$, which is the constant time. The first step of the algorithm is to compute the information diffusion in $Pr$ for $M$ candidate seeds, which requires $M \cdot N \log N$. In addition, considering the size $K$ of the seed set, the time complexity of the whole algorithm is $O(K \cdot M \cdot \log M) \cdot (N \log N))$.

### 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 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.

### 6 Datasets

Flickr [43] is a social network where users share pictures and videos. In this dataset, each node is a user, and each edge represents the friendship between users. Moreover, each node has a label to identify the interest group.

Digg [44] 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 [45] dataset was crawled in the following ways. To begin with, 100 random users were selected as seed nodes, 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 recent 1,000 messages (including tweets and retweets) by Weibo.

The MAG [46] 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 2, especially the cascade of each dataset.

|          | 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 |
6.1 Parameter setting

To achieve the best experimental results, the paper set 80% real-world dataset as the training dataset and the remaining 20% as the testing dataset. Moreover, in the NNA, the learning rate \( lr \) defaulted to 0.1, the train epochs \( te \) defaulted to 100, the embedding size \( es \) defaulted to 50, and the negative sampling rate \( ns \) defaulted to 10. Finally, the candidate rate \( p \) that indicates the rate of nodes involved in the influence spread was set to 40.

To visualize the classification effect of different algorithms, the paper uses the t-SNE algorithm. This algorithm involves three parameters tuning: \( \text{Perp}, \eta \), and \( \text{Mom} \). \text{Perp} represents the perplexity of the conditional probability distribution induced by a Gaussian kernel and \( \text{Perp} \in [5, 50] \). \( \eta \) represents the learning rate and \( \eta \in [100, 1000] \). \text{Mom} represents the momentum at each iteration and \( \text{Mom} \in [0.1, 0.9] \).

The algorithms in the paper are based on the following assumptions: (a) the influence between nodes is independent and decreases as the distance increases; (b) each node has only one category attribute, which does not change over time; (c) the nodes with the same type have possible to construct new connections.

All algorithms were implemented in Python 3.6. All experiments were conducted on a windows server with a 2.90 GHz quad core Intel i5-10,400 CPU machine with 8.00 Gb memory.

6.2 Evaluated algorithms

DeepWalk [36]: 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. Moreover, the skip-grammar model is used to realize the node vectorized representation.

LINE (2nd) [40]: LINE is a node representation method based on neighborhood similarity. The algorithm adopts Breadth-First-Search to construct node’s network structure attributes. LINE algorithm considers the first-order and second-order neighborhood similarities. However, it insufficiently utilizes high-order information.

Node2vec [38]: Node2vec is a further step based on DeepWalk. By adjusting the search method of the graph, the embedding results have balanced between homophily and structural equivalence. Furthermore, in the task of node classification, Node2vec’s effect is better than the previous algorithm.

NPIR [43]: NPIR is a clustering algorithm based on the nearest distance between the assigned point and a point using an election operation. This algorithm mainly solves the clustering problem on the 2-dimensional data set. To compare with the evaluation algorithms appearing in the paper, we first adopt the t-SNE algorithm to reduce the dimensionality of the large-scale OSNs dataset, and then applies the dataset to the NPIR algorithm.

KCC [44]: The KCC algorithm is an improvement of the K-means algorithm that integrates high-dimensional data statistical methods and the duality in the data. This algorithm adopts a higher-order walks framework to solve the problem of large-scale data sets. In the clustering process, the weighted random walks are considered as a measure of the similarity between nodes.

CELF [7]: The CELF algorithm takes advantage of the submodular 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 [29]: 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 [41]: 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 [42]: 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.

6.3 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 using t-SNE algorithm made a visual classification map. The experiment mainly discussed the \( \text{Perp} \) (perplexity of the conditional probability distribution) and \( \text{Mom} \) (momentum at each iteration) involved in the t-SNE algorithm. 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 Fig. 7.

Figure 7 presents the optimal classification visualization of Deepwalk, KCC, LINE, Node2vec, NPIR and RLIM algorithms.

In Fig. 7(a), the distribution of users with the same type is discrete. Those users with different types are mixed distribution, and there is almost no boundary.

In Fig. 7(b), users with the same type have clearly gathered. However, users of different types appear a large number of stacking phenomena.

In Fig. 7(c) and (d), users with the same type tend to concentrate. Although users of different types are still partially mixed, they have blurred boundaries.

In Fig. 7(e), the distribution of users with the same type is basically clustered, but the boundaries between users with different types are not obvious.

In Fig. 7(f), 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 are mixed.

It is our RLIM algorithm that is the best in the result of classification visualization. The NPIR and node2vec algorithms followed it. 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 standards. Table 3 shows the results of classification on the part Digg network.

Figure 8 shows the changes of Macro-F1 and Micro-F1 of the RLIM algorithm with Perp and Mom. Although the trend of change is ups and downs, in terms of statistical values, it seems that the change is not too great. The range of change is [0.07,0.08]. This also reflects that the RLIM algorithm is robust.

From the above analysis, it can be concluded that to achieve a better classification effect, in addition to the basic network structure, the properties 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 properties, so satisfactory results have not been achieved.

In terms of clustering, KCC and NPIR algorithms have an exceptional performance. However, they still have certain shortcomings in solving the classification problem of the complex OSNs.

On the whole, the RLIM algorithm performs well in the classification of large-scale OSNs datasets because it estimates user interests. 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 seed sets to spread the influence on different datasets. Figure 9 shows the different experimental results that indicate the influence spread of different algorithms. The lines with different colors in the figure represent different algorithms. Moreover, the higher the line, the wider the influence spread of the algorithm in each dataset. The detailed influence spread is shown in Fig. 9.

Figure 9 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.

As shown in Fig. 9, there are many obvious inflection points. However, with the increase of the seed set size, the influence spread generated by different algorithms increases. The RLIM algorithm represented by the red line has a relatively satisfying influence spread in each dataset.

Figure 9(a) shows the influence spread 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 Fig. 9(b), the influence spread of the RLIM algorithm has an absolute advantage, regardless of the seed set size 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 Fig. 9(c) and (d), when the seed set size 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 represented in a relatively small dataset, 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 Fig. 9(c) and about 16% on average in Fig. 9(d).

In general, the performance of the RLIM algorithm in the four datasets is quite satisfactory compared to the cutting-edge algorithms. From Fig. 9, 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.

7 Conclusion

The proposed new algorithm RLIM, which includes the construction of influence cascade, predicting 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 length. 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 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.

Acknowledgements The whole author first thanks every reviewer who provided valuable comments and feedback. Second, we thank the researchers Xiangbo Tian, Jianyi Zhang, Yuying Liu who provided guidance for the writing of the paper. Finally, the author would like to thank Qiang Shi, the researcher who collected the data for the experiment.
References

1. Vega L, Mendez-Vazquez A, Lopez-Cuevas A (2021) Probabilistic reasoning system for social influence analysis in online social networks. Soc Netw Anal Min 11(1):1–20

2. Coscia Michele (2021) Noise corrected sampling of online social networks. ACM Trans Knowl Discov Data 15(2):1–21. https://doi.org/10.1145/3434749

3. Chen T, Liu B, Liu W, Fang Q, Yuan J, Wu W (2020) A random algorithm for profit maximization in online social networks. Theor Comput Sci 803:36–47

4. Güney E, Leitner M, Ruthmair M, Sinnl M (2021) Large-scale influence maximization via maximal covering location. Eur J Operat Res 289:144

5. Chen SJ, Chen WK, Dai YH, Yuan JH, Zhang HS (2021) Efficient presolving methods for influence maximization problem in social networks. arXiv:2101.06399

6. Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 137–146

7. Leskovec J, Krause A, Guestrin, C, Faloutsos, C, Glance NS (2007) Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose, California, USA, August 12–15, 2007. ACM

8. Goyal A, Lu W, Lakshmanan LVS (2011) CELF++: optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th international conference companion on World wide web (WWW ‘11). Association for Computing Machinery, New York, NY, USA, pp 47–48. https://doi.org/10.1145/1963192.1963217

9. Galhotra S, Arora A, Roy S (2016) Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In: Proceedings of the 2016 international conference on management of data, pp 743–758

10. Peng S, Wang G, Xie D (2017) Social influence analysis in social networking big data: opportunities and challenges. IEEE Network 31(1):11–17

11. Pal SK, Kundu S, Murthy CA (2014) Centrality measures, upper bound, and influence maximization in large scale directed social networks. Fundamenta Informaticae 130(3):317–342

12. Wang F, Li J, Jiang W, Wang G (2017) Temporal topic-based multi-dimensional social influence evaluation in online social networks. Wirel Pers Commun 95(3):2143–2171

13. Deng X, Dou Y, Lv T, Nguyen QVH (2017) A novel centrality cascading based edge parameter evaluation method for robust influence maximization. IEEE Access 5:22119–22131

14. Saito K, Nakano R, Kimura M (2008) Prediction of information diffusion probabilities for independent cascade model. Springer, Berlin, Heidelberg

15. Cao T, Wu X, Hu TX, Wang S (2011) Active learning of model parameters for influence maximization. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, pp 280–295

16. Goyal A, Bonchi F, Lakshmanan L (2011) A data-based approach to social influence maximization. Proc VLDB Endow 5(1):73–84

17. Özdelen Z, Eldeeb S, Demir A, Erdğmuş D, Akçayaka M (2021) EGG-based texture roughness classification in active tactile exploration with invariant representation learning networks. Biomed Signal Process Control 67:102507

18. Sriman A, Mackley M, Sinha K, Shamout F, Pineau J, Geras KJ, Azour L, Aphinyanaphongs Y, Yakubova N, Moore W (2021) Covid-19 prognosis via self-supervised representation learning and multi-image prediction. arXiv:2101.04909

19. Mandal RK, Anand N, Sahu N, Kale P (2020) PV system fault classification using SVM accelerated by dimension reduction using PCA. In: 2020 IEEE 9th power india international conference (PIICON). IEEE, pp 1–6

20. Oral C, Sezgin H (2013) Effects of dimension reduction in mamograms classification. In: International conference on electrical & electronics engineering. IEEE, pp 630–633

21. Wang J, Cherian A (2019) Discriminative video representation learning using support vector classifiers. IEEE Trans Pattern Anal Mach Intell 43(2):420–433

22. Goyal P, Benner P (2021) LQResNet: a deep neural network architecture for learning dynamic processes. arXiv:2103.02249

23. Urda D, Veredas FJ, González-Enrique J, Ruiz-Aguilar JJ, Jerez JM, Turias JJ (2021) Deep neural networks architecture driven by problem-specific information. Neural Comput Appl 33(15):9403–9423

24. Picone RA, Webb D, Obierefu F, Lentz J (2021). New methods for metastimuli: architecture, embeddings, and neural network optimization. In: International conference on human-computer interaction. Springer, Cham, pp 288–304

25. Adineh M, Nouri M (2019) High quality degree based heuristics for the influence maximization problem. arxiv:1904.12164

26. Guille A, Hacid H, Favre C, Zighed DA (2013) Information diffusion in online social networks. Indiana University, Bloomington

27. Saito K, Kimura M, Ohara K, Motoda H (2010) Selecting information diffusion models over social networks for behavioral analysis. In: Joint european conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, pp 180–195

28. Bourigault S, Lamacz D, Gallinari P (2016) Representation learning for information diffusion through social networks: an embedded cascade model. In: Proceedings of the ninth ACM international conference on web search and data mining. Association for Computing Machinery, New York, NY, USA

29. Tang, Y., Shi, Y., & Xiao, X. (2015, May). Influence maximization in near-linear time: A martingale approach. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data, pp 1539–1554

30. Sivaganesan D (2021) Novel influence maximization algorithm for social network behavior management. J ISMAC 3(01):60–68

31. Gong Y, Liu S, Bai Y (2021) Efficient parallel computing on the game theory-aware robust influence maximization problem. Knowl-Based Syst 220:106942

32. Duan X, Qiu L, Sun C, Shi Q (2022) Multi-hop analysis method for rich-club phenomenon of influence maximization in social networks. Appl Intell 52(8):8721–8734

33. Wang W, Yin H, Du X, Hua W, Li Y, Nguyen QVH (2019) Online user representation learning across heterogeneous social networks. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 545–554

34. Yin H, Wang Q, Zheng K, Li Z, Yang J, Zhou X (2019) Social influence-based group representation learning for group recommendation. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE). IEEE, pp 566–577

35. Chen J, Zhong M, Li J, Wang D, Tu H (2021) Effective deep attributed network representation learning with topology adapted smoothing. IEEE Trans Cybern 52(7):5935–5946

36. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 701–710

37. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv:1301.3781

38. Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. The 22nd ACM SIGKDD International Conference
39. Kumhar SH, Kirmani MM, Sheetlani J, Hassan M (2021) Word embedding generation for urdu language using word2vec model. Mater Today Proc. https://doi.org/10.1016/j.matpr.2020.11.766
40. Jian T, Meng Q, Wang M, Ming Z, Mei Q (2015) LINE: large-scale information network embedding. International World Wide Web Conferences Steering Committee
41. Feng S, Cong G, Khan A, Li X, Chee YM (2018) Inf2vec: latent representation model for social influence embedding. 2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE
42. Panagopoulos G, Vazirgiannis M, Fragkiskos DM (2019) Influence maximization via representation learning. arXiv:1904.08804
43. http://www.datatang.com/data/13785.html. Accessed 20 Feb 2021
44. https://www.isi.edu/~lerman/downloads/digg2009.html. Accessed 15 Dec 2020
45. https://www.aminer.cn/influencelocality. Accessed 5 Jan 2021
46. https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/. Accessed 12 Mar 2021

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.