Expert Systems With Applications 169 (2021) 114207

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A dynamic algorithm based on cohesive entropy for influence maximization in social networks

Weimin Li [a], Kexin Zhong [a], Jianjia Wang [a], Dehua Chen [b]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai, China
[b] Donghua University, Shanghai, China

ARTICLE INFO

Keywords:
Influence maximization
Dynamic
Cohesive entropy
Overlapping community discovery

ABSTRACT

The problem of influence maximization in social networks has been widely investigated, but most previous studies have usually ignored the dynamic nature of propagation and the effects of local aggregation factors on diffusion. This paper presents a Dynamic algorithm based on cohesive Entropy for Influence Maximization (DEIM), the goal of which is to find the most influential nodes in social networks. Firstly, the Community Overlap Propagation Algorithm based on Cohesive Entropy (CECOPA) is put forward for the discovery of overlapping communities in networks, and potential nodes in the gathering area are selected to construct the candidate seed set. Then, the Optional Dynamic influence Propagation algorithm (ODP) is designed based on narrowing down the selection range of seeds. It utilizes a variety of entropy calculations to obtain the cohesive power between neighboring nodes and then determines whether the node has the ability to become a propagable pioneer of another node; thus, information continues to diffuse effectively. Finally, via many times experiments on several data sets, it is confirmed that the proposed DEIM algorithm in this paper can successfully affect the ideal number of users in different scenarios.

1. Introduction

With the evolution of network technology, various social software, such as Facebook, YouTube, and Twitter, have become the mainstream form of online communication. This has triggered a massive amount of network data, making the research of influence maximization more broad and vital. Influence maximization is the task of finding a set of seeds in a social network that can influence the rest of the nodes in this network to the greatest extent under a specific diffusion model. A widely-used marketing strategy is the creation of a chain reaction via word-of-mouth to make more people buy a company's products. However, determining how to achieve the best publicity effect with minimal overhead, i.e., the choice of the initial user set, is the challenge of maximizing influence.

The goal of Influence maximization is to determine K influential nodes as seeds, but the features of complex networks complicate this task. The personal attributes and structural characteristics of users in a network play vital roles in the seed selection process, and the community structure exists precisely because of these features. It is a reasonable choice to use the community to reflect the topological characteristics of nodes. Moreover, particular users in the community can also provide a good start for the diffusion of information; therefore, the accuracy of community division directly affects the impact of seeds. The COPRA algorithm (Gregory, 2010) is designed for the discovery of overlapping community discovery, and the results are judged by the belonging coefficients of communities. In 2008, the community structure was divided by the attractiveness of self-organized node (Hu, Chen, & Zhang et al., 2008). Banerjee et al. argued that the community was decided via budget allocation, and the seed nodes were determined via budget transfer (Banerjee, Jenamani, & Pratihar, 2019). Nevertheless, none of these algorithms quantifies the social distance between users, which affects the results of community division. Moreover, the community structure (Banerjee, Jenamani, & Pratihar, 2019) is considered to be a non-overlapping, which is not in line with reality. Thus, determining how to obtain an accurate community structure and integrate it into the influence maximization process to make the results more ideal and improve the efficiency of the algorithm is a direction worth exploring.

At present, much related research involves the problem of influence maximization. For example, the two fundamental propagation models in the field of influence maximization are the independent cascade model and linear threshold model (Kempe & Kleinberg et al., 2003), based on which several other models (Yang, Brenner, & Giua, 2019; Shi, Wang, &

E-mail addresses: wmli@shu.edu.cn (W. Li), zhongzhongx@shu.edu.cn (K. Zhong), jianjiawang@shu.edu.cn (J. Wang), chendehua@dhu.edu.cn (D. Chen).

https://doi.org/10.1016/j.eswa.2020.114207
Received 1 April 2020; Received in revised form 9 September 2020; Accepted 31 October 2020
Available online 7 December 2020

Chen, 2019) have been studied. However, most of the existing algorithms have certain limitations; they do not take into account some uncertainties of the diffusion process in real social networks, and ignore the autonomy of users to choose the sharing object. In reality, users can subjectively select who they share information with, e.g., they may freely talk to their best friends, but only have work exchanges with colleagues. The decision of users regarding with whom they share resources is the starting point for information diffusion. Multiple paths are radically formed around each user by observing users from a spatial perspective, and information flows along the path to other people. Because the user autonomously selects the passing point of information flow, the length and direction of the propagation path starting from the user are uncertain. Based on these features, the modeling of the dynamics of the propagation path caused by individual autonomy is a challenge.

This paper presents an influence maximization algorithm that integrates users' dynamic selection of sharing objects. To reduce the scope of the seed search based on the social network community structure, cohesive entropy is formulated to quantify the social distances between users. Combined with the label propagation algorithm, an overlapping community discovery algorithm based on close distances is defined; this algorithms utilizes node position information to narrow down the seed selection range. Moreover, to preserve the autonomy of individuals and improve the realism of the information dissemination process, an optional dynamic influence propagation algorithm is put forward to analyze the close distances between users and distinguish their effects. This not only effectively reduces the time overhead, but also reflects the autonomy and dynamics during the process of user propagation. The main contributions of this work are summarized as follows.

1) The Community Overlap PRopagation Algorithm based on Cohesive Entropy (CeCOPRA) is presented to divide overlapping communities. The close distance between people is defined by using the local topology information of nodes and the concept of cohesive entropy. To a certain extent, the influence of randomness caused by ignoring the relationships between users and improper threshold selection is eliminated. Additionally, the aggregation bridge and aggregation focus are selected as potential seed nodes, which can significantly improve the algorithm efficiency.
2) The Optional Dynamic influence Propagation algorithm (ODP) is designed and propagation control factors α is added, which are used to represent the lower limit of the propagation condition, i.e., to regulate the process. Additionally, a cohesive force that combines both self-information and cohesive entropy is formulated to determine whether a user can become a propagable pioneer and thereby influence others. Only when the cohesive power reaches a threshold does the communicator have opportunities to express their views; otherwise, the influence of diffusion will end. This makes the actual propagation path more realistic. Moreover, the provided condition can improve the algorithm efficiency, and also avoid unnecessary and time-consuming diffusion attempts.
3) Multiple experiments were conducted on disparate datasets, and the results show that conditional propagation using community structures can significantly improve efficiency and guarantee an acceptable loss of accuracy.

The structure of the remainder of this article is as follows. Section 2 summarizes the current work on influence maximization. The specific implementation of the DEIM algorithm is presented in Section 3. The experimental results on different datasets are reported in Section 4. Finally, Section 5 presents the conclusions of this article.

2. Related work

Domingos and Richardson et al. (Domingos & Richardson, 2001; Richardson & Domingos, 2002) argued the influence maximization problem. They think that some data can be used to estimate the degree of influence between users in social networks to promote new products, and ultimately make these products acceptable to most users. Kempe and Kleinberg et al. presented a greedy hill-climbing algorithm (Kempe & Kleinberg et al., 2003) to solve this problem, with $(1-1/e-\varepsilon)$ performance close to the optimal solution. The key to this influence maximization algorithm is to use the Monte Carlo approach to estimate the influence of any node. It allows the greedy algorithm to obtain a highly accurate solution, but it requires at least 10,000 times Monte Carlo simulations to evaluate the influence of each node. Therefore, the computational cost is very expensive for large networks. Also, Kempe and Kleinberg summarized the IC model and the LT model, and proved that the influence function under these models satisfies the sub-model characteristic. J. Leskovec et al. (Leskovec, Krause, & Guestrin et al., 2007) devised an algorithm named Cost_Effective Lazy Forward (CELF) to solve the inefficiency of the greedy algorithm by using submodularity. To avoid the simulate of a large number of Monte Carlo approach, Zhang et al. (Zhang, Gu, & Zheng et al., 2010) investigated Greedy Estimate-Expectation (GEE), which uses the expectation to calculate the influence of the node and greedily selects the node that obtains the maximum marginal benefit. If the distance between the infection point and the receiving point exceeds a certain distance, the small influences could be ignored (Tang, Tang, & Yuan et al., 2018; Li & Fan, 2020). So an influence propagation algorithm based on hops is raised to reduce the propagation range. However, the above mentioned technologies still require a higher order of computing time and still cannot be extended to large graphs.

Heuristic algorithm has better efficiency and scalability compared with greedy framework. Qin, Xu et al. used PageRank algorithm to rank the importance of the nodes in the network for ranking web pages (Qin, Xu, & Hu et al., 2005). Chen et al. discussed Single Discount and Degree Discount algorithms based on degree centrality (Chen, Wang, & Yang, 2009) to reduce the overlap of influence between seed nodes. The innovation of these algorithms is that if a node becomes a seed, the degree centrality of its neighbor nodes will decrease. Zareie A et al. (Zareie, Sheikhahmadi, & Khamforoosh, 2018) utilized the TOPSIS method (Hwang & Yoon, 1981) to comprehensively consider the direct and indirect influences of nodes. The goal is to maximize the global influence while minimizing the overlapping influence between the seeds, but the complexity of the algorithm is still high. For this kind of combinatorial optimization problem, the warning propagation algorithm (Wang, Wang, & Li, 2019) is also a solution with distinct advantages in calculation speed. The similarity of nodes is used frequently in the heuristic algorithm, and its common measurement methods include degree centrality (Freeman, 1978), closeness centrality (Okamoto, Chen, & Li, 2008), eigenvector centrality (Ruhnau, 2000), K-shell decomposition (Kitsak, Gallos, & Havlin et al., 2010), intermediate metric (Barthelemy, 2004) and so on. Most above heuristic algorithms have strong scalability, but the performance close to optimal solution achieved by the method of Kempe et al. cannot be retained because of ignoring various surrounding features.

In addition to the above two types of algorithms, the idea of narrowing the selection range of seeds has gradually become a hot spot in the research of influence. He Q et al. devised a two-stage iterative framework TIFIM (He, Wang, & Lei et al., 2019). In the first stage, the nodes with less influence are excluded to generate a candidate seed set. In the second stage, vertex advantage and influence overlap are considered for final selection. To balance efficiency and accuracy, Singh S.S. et al. discussed that the influence probability between users is not only related to their degree but also affected by the aggregation coefficient of neighbors (Singh, Singh, & Kumar et al., 2018). The community partition method Louvain algorithm (Blondel, Guillaume, & Lambiotte et al., 2008) is also used to narrow the seed selection range. None of these step-by-step algorithms consider a key point. It is important to narrow the scope of seed selection, but the stability of the node relationship is also very important for the dissemination of information.

Because information sharing usually occurs between people who are familiar with or trusted each other. In this paper, we attempt to discuss these factors and use it for the dissemination of information.

3. Dynamic algorithm based on cohesive entropy for influence maximization

The application scenarios for influence maximization are pervasive, and include viral marketing, recommendation systems, information diffusion, expert discovery, and link prediction. Given a social network graph $G = (V, E)$, V denotes the node set and E denotes the edge set.

In the IC model, a seed set S attempts to activate its neighbors, and the activation probabilities are known. When a new node is activated, it will continue to maintain the activated state and try to activate its inactive neighbors. This process iterates until no new nodes in the network are activated. All nodes have two states, namely activate and inactive. Once a node is activated, it will remain active. An activated node has only one chance to try to activate all its inactive neighbors with a given probability. In the LT model, each node in the network has a threshold that represents the difficulty of the node being affected. Rather than disappearing immediately, as in the IC model, the effects of all activated neighbors of a node are cumulative. When the accumulated value reaches the threshold, the node is activated, after which the node can contribute influence and activate its neighbors. This process stops when no more nodes are activated.

Influence maximization is a problem of finding K seed nodes from graph G and obtaining the maximum influence under a given diffusion model.

$$S^{*} = argmax_{s\subseteq V}\{\sigma(S)\||S| = K\} \tag{1}$$

The DEIM algorithm proposed in this paper makes full use of the community structure and considers the individual characteristics of users. In the real-world social network, the community structure is widespread, and resources are allocated to users based on the communities. Information also tends to spread within the unit. The possibility of information transmission among internal community members is higher than that of users in different communities. Due to differences in hobbies and regions, some users could receive resources from multiple fields and become part of overlapping communities. Cross-community nodes can also bring information to members of different communities and become important information transfer stations. Users are connected due to specific connections, and different levels of relationships lead to differences in close distance. The difference in close distance is the reason why the community structure exists and is relatively stable, which is very important for the result of the community discovery. Therefore, in the DEIM algorithm, we first quantifies the social distance between the user and each neighbor, and then designes the process of calculating the cohesive entropy between the nodes based on the relative entropy formula. On this basis, the CECOPA algorithm is devised to obtain a more accurate community division. In order to reduce the range of seeds selection, shorten the time of seeds selection, and improve the efficiency of algorithms based on greedy thinking, candidate seed nodes are selected based on the community structure. Also, the individual has autonomy. The process of choosing the sharing target by the user is dynamic, so resource sharing does not necessarily occur between adjacent users. In order to simulate this dynamic propagation process, an optional dynamic influence propagation algorithm is presented.

The notations in this paper are shown in Table 1.

3.1. Community overlap propagation algorithm based on cohesive entropy

Influence maximization refers to the selection of the most influential users in a social network. However, real social networks are characterized by complex structures, diverse nodes, and various connections, and determining a small-scale seed set from these networks is a complicated task. Moreover, the power-law distribution phenomenon is common; for the entire network, few nodes are closely connected to other nodes and are in critical positions, so it is reasonable to narrow the selectable range of seed nodes. In the present work, the community structure is used to select the nodes that have the potential to become a seed; thus, the quality of community division directly affects the impact of the last seeds. The concept of cohesive entropy is used to distinguish the social distances between users and each neighbor node, and different relationship distances correspond to varying levels of influence. The belonging coefficient of the node to the communities in which the neighbor node belongs is calculated. Based on this, the CECOPA method is proposed.

Table 1
Notations.

Notation	Definition
G(V, E)	Social graph
N	Number of nodes in G
M	Number of edges in G
N_u	Neighbor set of node u
D(i)	Degree of node i
$D_{local}(j)$	Degree of node j in the adjacent area
p	Number of communities
C	Community structure of social networks
C_j	Communities tagged with j
α	Propagation control factor
C(v)	Community set to which node v belongs
$\delta^{v}_{C_j}$	Node v's belonging coefficient toC_j
$G^{i}_{local}(N^{i}_{local}, DI^{i}_{local})$	Adjacent area structure of node i
N^{i}_{local}	Node set within the adjacent area of node i
DI^{i}_{local}	Adjacent area structure information distribution of node i

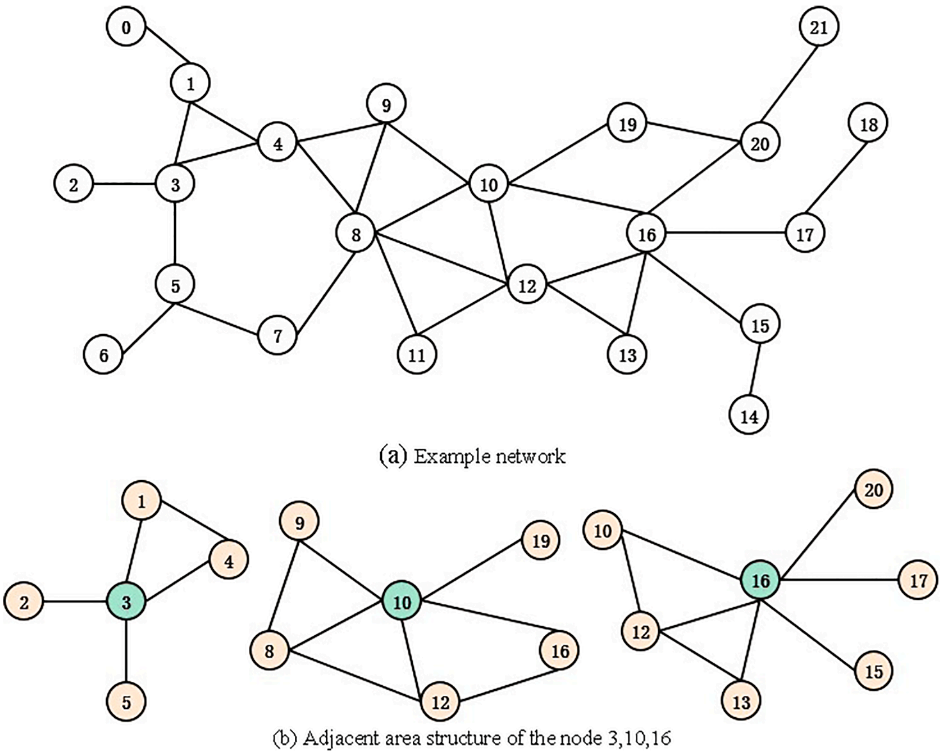
3.1.1. Calculation of cohesive entropy in adjacent areas

In social networks, due to the existence of individual characteristics, the differences and similarities between users are also different. The greater the similarity, the closer the connections between users may be. This varying degree of connection leads to the emergence of communities within the network. Relative entropy is a calculation that measures the difference between probability distributions; it is suitable for measuring the difference between nodes and then converting this difference into similarity, which is of great significance for improving the accuracy of community division. Zhang et al. calculated the similarity of nodes in a network via relative entropy in the network by using the local topology of the nodes (Zhang, Li, & Deng, 2018), i.e., the degree of distribution of each node and all adjacent nodes. The structural similarity between nodes was studied, but this method does not distinguish the importance of the node's own attributes from the attributes of its neighbors. The characterization of a node mainly depends on its attributes, and the attributes of adjacent nodes play an auxiliary role. In addition, the degrees of the neighbors cannot accurately represent the characteristics of the local structure, and the edges of some adjacent nodes are not related to the representative nodes. Therefore, in this work, a new adjacent area structure is defined, and a method by which to find the similarity between nodes is presented.

First, the specific composition of the adjacent area structure of node i is given as $G^{i}_{local}(N^{i}_{local}, DI^{i}_{local})$, N^{i}_{local} consists of node i and its adjacent nodes, and DI^{i}_{local} indicates the distribution of structural information in the adjacent area, which is the proportion of the number of connected edges between each node in N^{i}_{local} and other nodes in N^{i}_{local}. Fig. 1(b) presents an example of the adjacent area structure, and the information distribution of the node's adjacent area structure is given by Eq. (2).

$$DI^{i}_{local} = [p(i, 1), p(i, 2), \cdots, p(i, b), \cdots, p(i, m)] \tag{2}$$

where $\mathrm{m} = D(i) + 1, l \in \{1, \cdots, b, \cdots, m\}$is the representation of each node



(a) Example network

(b) Adjacent area structure of the node 3,10,16

Fig. 1. An example of a network is presented in the subgraph a). The adjacent area structures of the nodes are shown in the subgraph b).

in the adjacent areas, and $p(i,l) = \frac{D_{local}(l)}{\sum_{b=1}^{m} D_{local}(b)}$.

The relative entropy formula calculates the elements at the same position in the two probability sets and accumulates the results. The elements in the adjacent area of the nodes are then sorted in descending order. The distribution of the sorted adjacent area information is given by Eq. (3).

$$DI_{local}^{i\,'} = [p^{'}(i,1), p^{'}(i,2), \cdots, p^{'}(i,b), \cdots, p^{'}(i,m)] \tag{3}$$

where $p'(i,1)$ is same as $p(i,1)$ in formula(2), it is the self-characteristics of the representative node, which is the main attribute in the similarity calculation. And $p^{'}(i,l)$ is the element after descending sort,$l \in \{2,\cdots,b,\cdots,m\}$. As shown in Fig. 1 (b), where$N_{local}^{3} = [1,2,3,4,5]$, the corresponding degree in the local area is [2,1,4,2,1]. $DI_{local}^{3} = [\frac{2}{10},\frac{1}{10}, \frac{4}{10},\frac{2}{10},\frac{1}{10}]$, $DI_{local}^{3\,'} = [\frac{4}{10},\frac{2}{10},\frac{2}{10},\frac{1}{10},\frac{1}{10}]$.

In this work, the topological similarity analysis of the nodes is performed based on the relative entropy, instead of the Euclidean distance. This is because, in social networks, the similarity between users is the similarity of the information carried by the users. The input of the relative entropy formula is precisely the information distribution, which is more suitable for the similarity calculation of social network information. The measure of node similarity can be regarded as the quantification of the difference in the topology, which is conducted to find the difference between the local structures. Therefore, the relative entropy is a calculation for the quantification of the discrepancy between each pair of nodes. If the difference between two nodes is small, then their similarity is substantial, and vice versa. The specific use of the relative entropy to ascertain the similarity between nodes is subsequently presented.

Definition 1. (***Cohesive entropy***):*Cohesive entropy is a measure of the similarity between two nodes regarding the distribution of adjacent area information. The attributes of the nodes are first considered, after which the tightness of the connecting edges between the nodes in the adjacent area is used as an auxiliary attribute, and the adjacent area structure information of the nodes is then determined to calculate the cohesive entropy. The calculation formula for the cohesive entropy CE_{ij} of nodes i and j is given as follows.*

$$CE_{ij} = 1 - \frac{r_{ij}}{max(r_{ij})}$$

where r_{ij} is the sum of the relative entropy of the adjacent area information distribution of node i and node j, called degree of dispersion. Because the relative entropy is a measure of asymmetry, and because the similarity of every two nodes should be equal, the discrepancy between the two nodes is converted into an asymmetric value by the variable r_{ij}; the larger the value, the greater the local structural difference between the two nodes. The calculation formula is as follows.e

$$r_{ij} = D_{kl}\left(DI_{local}^{i\,'} \| DI_{local}^{j\,'}\right) + D_{kl}(DI_{local}^{j\,'} \| DI_{local}^{i\,'}) \tag{5}$$

where D_{kl} is the relative entropy calculation formula, which is expressed as follows:

$$D_{kl}(DI_{local}^{i\,'} \| DI_{local}^{j\,'}) = \sum_{b=1}^{B} P'(i,b) \ln \frac{P'(i,b)}{P'(j,b)} \tag{6}$$

where $B = \min(D(i)+1, D(j)+1)$ to ensure that the two information distributions are the same size.

Cohesive entropy is used to measure the relationships between users. First, the similarity between nodes is converted into a difference calculation. The distribution of the adjacent area information distribution of the nodes is then used as the input of relative entropy to determine the difference between the nodes, which is the degree of dispersion. If the dispersion is small, the cohesive entropy is vast, and vice versa. When the adjacent area structure information of two nodes is the same, their dispersion degree is 0, and the cohesive entropy is 1; when this information of two nodes is very different, their dispersion degree is close to 1, and the cohesive entropy is close to 0.

3.1.2. Community overlap propagation algorithm based on cohesive entropy

The use of the COPRA algorithm (Gregory, 2010) to discover overlapping communities has been discussed. The main concept of this algorithm is that the community to which a node belongs is determined by the neighbors' community distribution, i.e., the distance of a node from all neighboring nodes and its level of influence are the same. However, in reality, the amount of information diffused or received by different users is not the same. The probability of sharing information between close friends is much higher than that of sharing information between ordinary friends, and affected users also trust other users who have similar preferences. Moreover, the numbers of neighboring nodes in different communities are likely to be the same, and the random strategy in the algorithm dramatically reduces the accuracy of the results; it is therefore more reasonable to use cohesive entropy to distinguish the effects from different neighbors. In the process of cohesive entropy calculation, the node's own attributes and the factors of the surrounding environment that are closely related to the node are considered, and information distribution is used in the relative entropy formula. The information differences between the internal elements can be reflected and accumulated successively, and the result is more accurate. Therefore, the CECOPA algorithm is proposed, as described in Algorithm 1.

Algorithm 1 CECOPA Algorithm

```
Input: G(V,E)
Output: Community structure C
1: Set a unique community label for each node
2: While not termination condition:
3: for each $v \in V$:
4:   $t=\frac{1}{D(v)}$
5:     Set J←the communities which the neighbor nodes belong
6      for each j∈ J:
7:         update the belonging coefficient $\delta^{v}_{C_j} = \sum_{u\in N_v, u\in C_j} CE_{vu}$
8:   $\delta^{v}_{C_j\leftarrow}$←normalize the belonging coefficient
9:   for each j∈ J:
10:       if $\delta^{v}_{C_j} < t$:
11:           exclude the community label
12:       else:
13:           update the community labels of node v
14:    if all community label $\delta^{v}_{C_j} < t$:
15:       select the community label with maximal belonging coefficient
16: Return community structure
```

The first line in the algorithm sets a unique community label for each node. In line 4, the threshold *t* is used to limit the minimum value of the belonging coefficient that meets the conditions. For each node, *t* is set as the inverse of the node degree. Lines 5 to 8 update the belonging coefficient to the communities in which the neighbors are located and normalized. Lines 9 to 14 are used to select the label of the community to which the node belongs. When all the belonging coefficients are less than *t*, the community with the highest belonging coefficient is selected as the community to which the node belongs.

The time complexity is related to the number of nodes of the network and the number of neighbors of each node. For a node, the number of neighbors is approximately equal to its degree, so the time complexity of the algorithm is O(ND), where N is the total number of nodes in the network, and D is the maximum degree of the network..

3.2. Selection of candidate nodes

The candidate seed set is constructed to select individual nodes in the network that have the potential to become seed nodes, narrow the search range of seed nodes, and speed up the selection process of seeds. The community structure can help evaluate the importance of the nodes and comprehensively consider the position of the nodes in the community or between communities. It also allows for the consideration of the relationships between the nodes to compare the spreading capabilities of nodes. In this paper, we believe that nodes with higher degrees have more opportunities to influence the rest of the members within a community, while users located in the overlapping area have the ability to spread information to different communities. We mainly evaluate the node's degree and connectivity between communities to eliminate unimportant nodes. In this work, the attributes and locations of nodes in a community-based network are discussed, and potential nodes are selected to form a candidate set.

Definition 2. (***aggregation bridge***)**:***This article considers each community to be an aggregation area, and overlapping nodes are located at the aggregation intersection area, in which the aggregation bridge is generated. The aggregation bridge N_{hinge} is a set of user representatives that spans multiple fields and is defined as follows.*

$$N_{hinge} = \cup_{i=1}^{p} Community^{i}_{bridge} \tag{7}$$

where $Community^{i}_{bridge}$ indicates node sets in community i that are located in six or more communities at the same time. These points are closely connected to multiple clusters. The number of communities can ensure that users in the cluster have sufficient opportunities to influence others, and can guarantee a certain number of influence diffusion paths. To avoid a situation in which the number of communities is too small and the size of a community is too large, according to the concept of the six degrees of separation, each node belongs to a maximum of six communities after the community is divided. Thus, the nodes in the aggregation bridge can simultaneously belong to a maximum of six communities.

Definition 3. (***aggregation focus***)**:***The non-overlapping nodes of each community form the centralized aggregation area of the community. The node with the highest degree of centrality has the closest connection with other nodes in this area. This is called the aggregation focus, and is expressed as follows.*

$$N_{core} = \cup_{i=1}^{p} argmax_{v\in c_i} D(v) \tag{8}$$

Fig. 2 presents an example of candidate seed selection. The social network in the figure is divided into three communities, in which nodes 8 and 16 each have a degree of 6, forming the aggregation bridge {8,16} (because the network in the example is small, the standard of the aggregation bridge is reduced to the most moderate node across multiple community nodes. In a realistic network, the nodes that meet the requirements of the aggregation bridge are generally present). For each community, the aggregation focus is separately selected; $N^{1}_{core} = \{3, 4\}$, $N^{2}_{core} = \{12\}$. For the third community, excluding overlapping nodes, the remaining nodes have a degree of 1. This case is not common for large networks.

Based on the concepts introduced previously, the candidate seed set is generated by the Candidate seed Set based on Two Key Regions (TKRCS) algorithm. The specific process is reflected in Algorithm 2 below.

Algorithm 2 TKRCS Algorithm

```
Input: Community structure$C$
Output: Candidate seed set CS
1:$N_{hinge}$←∅,$N_{core}$←∅
2: for each $c \in C$:
3:    $v' \leftarrow argmax_{v\in c} D(v)$
4:    $N_{core} \cup \{v'\}$
5:    $N_{hinge} \cup Community^{c}_{bridge}$
6: Return$N_{core} \cup N_{hinge}$
```

Line 1 of the algorithm assigns initial values to the aggregation bridge set and the aggregation focus set, and lines 3 and 4 search for the aggregation focus for each community. Line 5 searches for aggregation bridges between communities. The two components together form the candidate seed set CS.

The time complexity is related to the number of communities of the

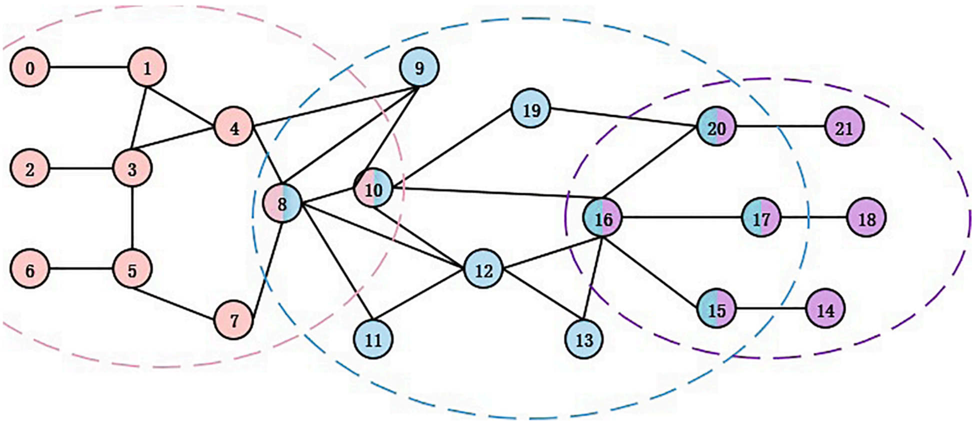


Fig. 2. Example of candidate seeds selecting.

network and the number of nodes of each community. Let N be the number of nodes and C be the number of communities, the time complexity of the TKRCS algorithm is O(NC).

3.3. Optional dynamic influence propagation algorithm

The selection of the candidate seed set has been completed, and nodes with the potential to become seeds have been selected. Based on this, an optional dynamic influence propagation algorithm is designed to determine the most influential nodes based on greedy logic and the IC model. The user's autonomy, i.e., the user's right to choose who to share information with, is considered. Users tend to connect with friends at close distances, and have a lower probability of choosing to connect with users from further social distances. Therefore, based on the IC model, the propagation control factor α, which represents the lower limit of the propagation conditions, is added to the proposed algorithm. The node can be infected only when the conditions are met, and it then attempts to activate its neighbors. In the IC model, active users do not need prerequisites when they try to influence others. If the relationship between users is estranged and they are in a state of distrust, such an attempt is considered to be meaningless.

Definition 4. (***self-informationentropy***)**:***The amount of information obtained by the node is positively related to the amount of diffusion amount of the node, and the formula is as follows.*

$$H_u = -\frac{D_u}{M}log_2\frac{D_u}{M} \tag{9}$$

where M is the total number of edges, and D_u is the degree of node u. Information entropy is a quantification of information. Here, self-information entropy measures the amount of information carried by a node via the ratio of the node degree and the total number of edges.

Definition 5. (***cohesive power***)**:***The cohesive power between the node* u ∈ V *and its neighbor node w is given by Eq.* (10).

$$CP_{uw} = H_u - \frac{1}{CE_{uw}} \tag{10}$$

where H_u is the self-information entropy of the node to be propagated, and CE_{uw} is the cohesive entropy of node u and w. The higher the cohesive power, the closer the nodes are.

Definition 6. (***propagable pioneer***)**:***For edge* $(u,v) \in E$*, when the cohesive power of nodes u and v reaches the value of the propagation control factor* α*, node u has the ability to try to affect node v. In other words, u becomes the propagable pioneer of node v , which it then tries to influence node v. An example is given in* Fig. 3*, in which node* 12 *has been successfully activated, and five nodes may be affected. The cohesive power strength is respectively calculated. Here,* $\alpha = 0.1$ *is set.*$CP_{12,10}$, $CP_{12,11}$, *and* $CP_{12,16}$ *exceed* α*, so node* 12 *continues to try to activate them with a certain probability. However, the cohesive power of node* 12 *and nodes* 8 *and* 13 *has not reached* α*, so the two propagation paths are terminated.*

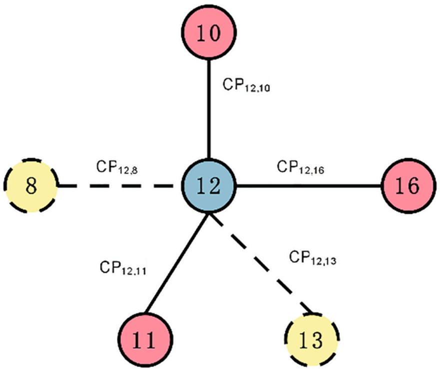


Fig. 3. Example of propagable pioneer.

Unlike previous communication algorithms, the threshold α is used in the proposed algorithm to indicate the lower bound for propagation. When the cohesive power between two users reaches the threshold, one user can become the propagable pioneer of the other user, i.e., the node has the ability to try to influence another node, and the influence continues to spread outward. In this paper, the activation probability between users is the magnitude of the reciprocal of the degree of the node to be activated. The specific steps are shown in Algorithm 3.

Algorithm 3 ODP Algorithm

```
Input: G(V,E),node n
Output: the influence of node n
1: Inf←0
2: for each w∈ $N_n$ :
3:   if w ∈ seedsetS:
4:   continue
5:   if $CE_{nw}>\alpha$:
6:   n try to activate w with$P_{nw}$
7:   if success:
8:      ODP(G, w)
9:      Inf ++
10:    else:
11:       return Inf
12:    else:
13:       continue
14: return Inf
```

Line 1 initializes node influence. Lines 5 to 13 represent that, when the cohesive power reaches the propagation control factor, the node continues to try to activate other nodes; otherwise, the propagation stops. This process is recursive. When the next node is successfully activated, the steps are repeated until the activation fails and the value of influence is returned.

The time used by the algorithm is related to the number of neighbors of the input node, i.e., the degree of this node. The time complexity of the ODP algorithm is $O(D^2)$, where D is the maximal degree in the network.

3.4. Dynamic algorithm based on cohesive entropy for influence maximization

The Dynamic Algorithm based on Cohesive Entropy for Influence Maximization (DEIM) algorithm proposed in this article is an influence maximization algorithm based on community discovery and the process of the dynamic selection and sharing of objects by fusion users. Real social networks have features such as complexity and diversity, which lead to the difficult and time-consuming nature of seed set selection. First, to improve efficiency, cohesive entropy is formulated based on the community structure to quantify the social distances between users. Combined with the label propagation algorithm, an overlapping community discovery algorithm is constructed based on cohesive entropy, in which the node position information is used to narrow the scope of seed selection. Then, to reflect the dynamic process of the self-selection of sharing objects, an optional dynamic influence propagation algorithm is investigated to evaluate the influence of nodes. It analyzes the differences in the influences due to the different intimate distances between users and then determines the seed set. This not only effectively reduces the time overhead, but also reflects the autonomy and dynamics of the user propagation process. The details of the DEIM algorithm are presented in Algorithm 4.

Algorithm 4 DEIM Algorithm

```
Input: G(V, E), t, K
Output: seed set S
1:S←∅
2: Community structure C←CECOPA (G,t)
3: Candidate seed set CS ←TKRCS (C)
4: for i = 1 to K:
5:     for each n ∈ CS:
6:     Inf(n) ←use ODP Algorithm(G,n)
7:     sort the influence of the node in CS
8:     w←argmax_{n∈CS} Inf(n)
9:     S∪{w}
10: Return S
```

In lines 2 and 3 of the algorithm, the candidate seed set CS is obtained by dividing the community. Then, in lines 5 to 9, the optional dynamic influence propagation algorithm is used to calculate the influence of each node in the candidate set. After sorting, the node with the highest return is selected as the seed node.

Combining the various stages, the total time of the DEIM algorithm is: $O(ND + NC + C'D^2)$, where N is the total number of nodes in the network, C is the number of communities after division, and D is the value of the largest degree in the network , C' represents the number of nodes in the candidate seed set and C' is far less than the number N of nodes.

4. Experiment

4.1. Setup

Experiments were performed on five datasets of different sizes, and the experimental results are presented in Section 4.2. The diffusion model adopted for all evaluation experiments was the IC model, in which the influence probability of each edge is set as the reciprocal of the degree of the end node of the edge.

a) The influence spread corresponding to seeds selected by different algorithms
b) The running time of different algorithms for seed selection
c) The setting of propagation control factors

Datasets:1) The DBLP dataset is a comprehensive list of computer science research papers provided by the Computer Science Bibliography. It has established a co-author network with a total of 954 nodes and 3798 edges. If two authors publish at least one paper together, they are connected. 2) The Facebook dataset is derived from the friend list of social software Facebook. It has a total of 4024 users and 87,887 connected edges. The connected edges reflect the friendship between each other. 3) wiki-Vote Wikipedia is a web encyclopedia project. Nodes in the network represent Wikipedia users, while edges represent votes among users. There are 7115 nodes and 103,689 edges. 4) CA-HepPh - collaboration network is from the e-print arXiv and covers scientific collaborations between author's papers submitted to High Energy Physics-Phenomenology category. 5) The Amazon dataset was collected by crawling Amazon website. If a product is frequently purchased with another product, the graph contains an undirected edge between them. The summaries of all data sets are shown in Table 2.

The DEIM algorithm was compared with heuristic algorithms and the greedy algorithm to demonstrate its advantages of the greedy algorithm and the efficiency of the heuristic algorithm.

Algorithm. ***s for comparison***: 1) *Greedy*: *A classic seed selection strategy, whose approximation to the optimal solution is known, can be called one of the criteria for the influence maximization algorithm. The algorithm selects the node with the largest marginal profit at each step to join the seed set. It uses the Monte Carlo simulation to calculate the influence of each node with high accuracy*. 2) *Degree*: *A classic heuristic algorithm that uses the centrality of network nodes. This algorithm selects the node with the largest degree in the network as the seed node, which is a most intuitive and simple indicator to measure the influence of nodes.* 3) *PageRank*: *It is also a classic heuristic algorithm used to rank the importance of each node in the network. We set the damping factor is* 0.85. *Originally used in Google's web page ranking, and it is now used to find influential seed nodes in social networks.* 4) *IMM*: *One of the advanced sampling methods, using reverse reachable set to find the seeds.*

4.2. Experimental results

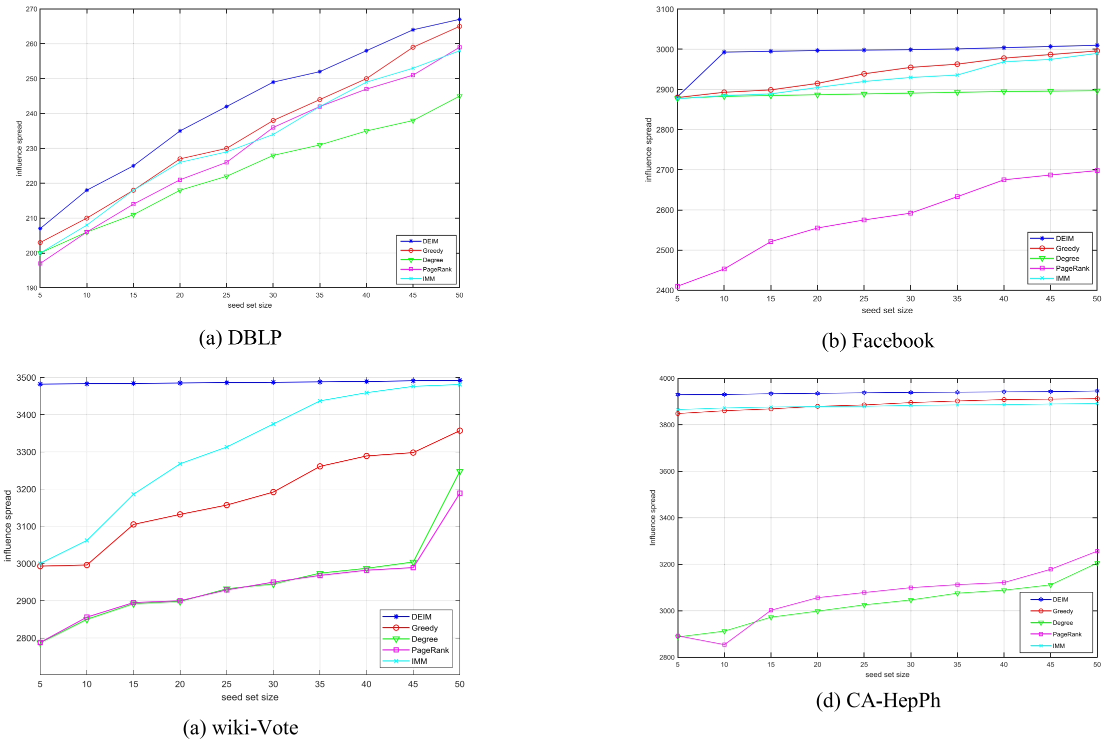
4.2.1. Influence spread

The proposed DEIM algorithm was compared with four other classical algorithms on five datasets with very different characteristics. In the DEIM algorithm, $\alpha = 0.001$. The seed sets obtained by each algorithm were simulated on the IC model for propagation. The effect is shown in Fig. 4, and it is evident that the proposed algorithm performed well overall; its influence propagation range was found to be superior to those of the other algorithms. Meanwhile, the accuracy of the algorithm in different seed sizes is compared.

For the DBLP dataset, as plotted in Fig. 4(a), with the increase of the number of seeds, the influence of each algorithm increased steadily, and the DEIM algorithm performed prominently. For the Facebook, viki-Vote, and CA-HepPh datasets, as shown in Fig. 4(b-d), the proposed DEIM algorithm selected the decisive seed set when the number of seeds was small, and the effect was always better than those of the other algorithms. This is because DEIM eliminates the nodes with little influence

Table 2
Summary of the datasets.

Network	Nodes	Edges
DBLP	954	3798
Facebook	4024	87,887
Wiki-Vote	7115	103,689
CA-HepPh	12,008	237,010
Amazon	334,863	925,872



(a) DBLP

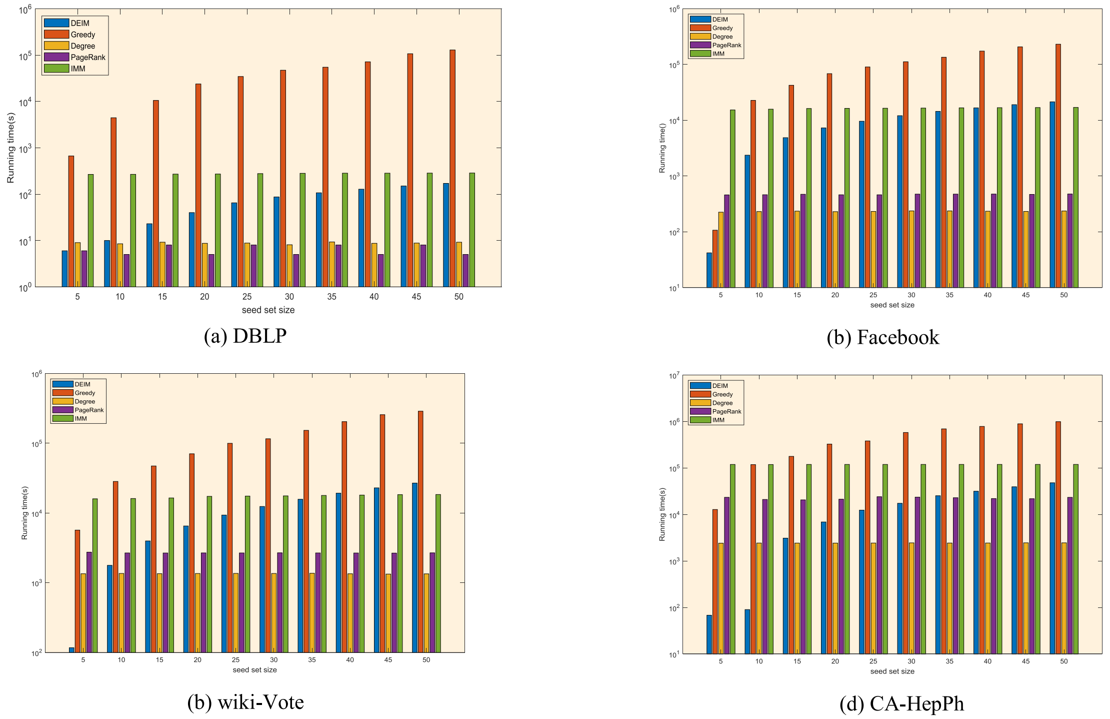
(b) Facebook

(a) wiki-Vote

(d) CA-HepPh

Fig. 4. Impacts of different algorithms.

before determining the seed, and considering that users selectively share information, the result will only generate propagation paths with a high probability; therefore, when seed users disseminated information, these paths performed significantly better than the Contrast algorithm. For the larger-scale dataset Amazon, the DEIM algorithm also has a good performance, as shown in Fig. 4 (e), indicating that the performance of the algorithm in this paper is relatively stable for networks of different sizes. The result of IMM algorithm is unstable, which may be caused by random selection of nodes to generate reverse reachable set. The simple heuristic PageRank algorithm and Degree algorithm performed well for small datasets; however, with the increase of the size of the dataset, the scale-free nature of the network gradually strengthened; the chosen



(a) DBLP

(b) Facebook

(b) wiki-Vote

(d) CA-HepPh

Fig. 5. Running time of different algorithms.

seeds may have appeared to be aggregated, and the effect gradually decreased. The validity of the DEIM algorithm was relatively stable, which demonstrates that it is universal for networks of different types and sizes.

4.2.2. Running time

Fig. 5 illustrates the running times on the five datasets when different algorithms selected different numbers of seeds. The seed set size is set to 5 to 50 with an interval of 5. In the DEIM algorithm, α = 0.001 was set.

It can be seen from Fig. 5 that the DEIM algorithm exhibited obvious advantages in efficiency when the number of selected targets was small, and the result was not inferior to that of the heuristic algorithm. This is because the candidate seed set greatly narrows the selection range of seeds and eliminates the network edge nodes with little influence. As the number of seeds increased, the running time increased, but it was still less than the running times of the greedy algorithm and the IMM algorithm. In the greedy logic-based algorithm, the simulation of influence propagation is performed on all nodes in the candidate seed set to obtain the influence value. As the number of seeds increased, the number of influence simulation calculations increased, and the algorithm became increasingly more time-consuming. As shown in Fig. 5(a-d), the time efficiency of the DEIM algorithm was found to be significantly higher than that of the greedy algorithm, but it was not found to have an advantage as compared with the two heuristic algorithms. This is because the Degree algorithm and the PageRank algorithm only consider a certain characteristic of the network, and do not consider the actual propagation characteristics. However, as the size of the dataset increased, the running times of these two algorithms also increased significantly, as shown in Fig. 5(b) and 5(c). For the large-scale dataset Amazon as shown in Fig. 5(e), the efficiency advantage of the DEIM algorithm is seriously reduced compared with other algorithms. In the overlapping community division stage, multiple iterations are required to ensure that the community division in the network achieves a stable structure, thereby ensuring the accuracy of the results. In large-scale networks, iterative methods are extremely time-consuming. Although the running time has increased significantly, it still has certain efficiency advantages compared with the greedy-thought algorithms.

4.2.3. Setting of propagation control factor α

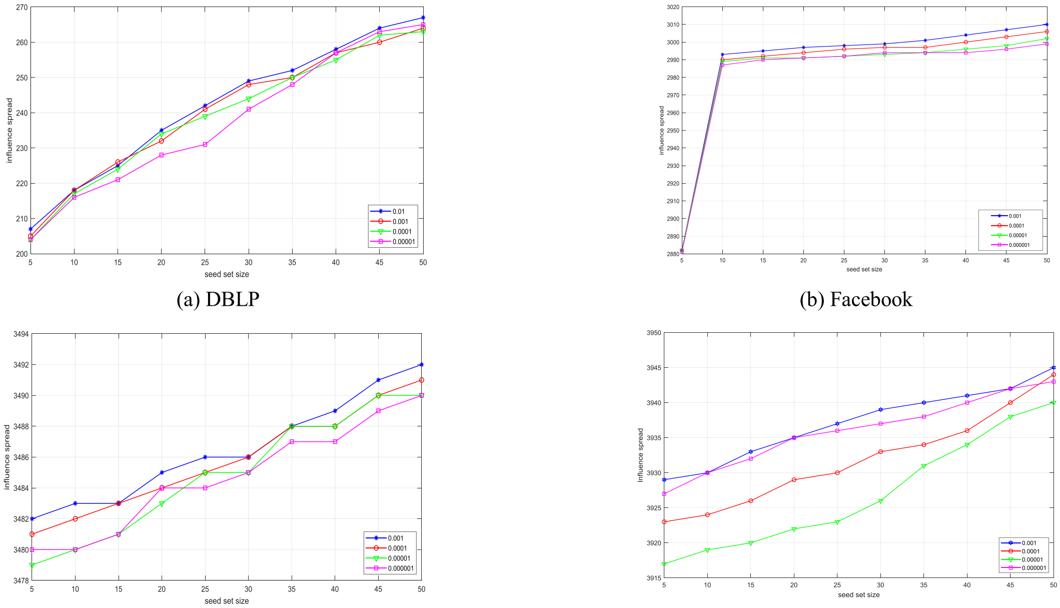
The propagation control factor α is a parameter that determines whether a user shares a message in the influence diffusion stage, as well as the length of the influence diffusion path. α will constrain the scope of the influence propagation process, and will also directly affect the selection of seeds and the running time of the algorithm. According to the distribution of the cohesive power of the nodes in each network, various values of α were respectively set, and the corresponding results are plotted in Figs. 6 and 7. The experiments were compared in terms of effect and efficiency.

The respective values of α in Fig. 6(a) are 0.01, 0.001, 0.0001, and 0.00001. The performance was found to be better when α = 0.01 and α = 0.001; the propagation control requirements between nodes were relatively high, and the cohesive power between users was relatively large. In contrast, when$\alpha = 0.0001$ and $\alpha = 0.00001$, the influence range was relatively low, and redundant activation attempts may have been made. In Fig. 6(b-d), the respective values of α are 0.001, 0.0001, 0.00001, and 0.000001. As can be seen from Fig. 6(b) and 6(c), on each dataset, the performance was still better when α was larger, and the situation was similar to that presented in Fig. 6(a). As shown in Fig. 6(d), when α was set to the maximum and minimum values, the effect was significant, and when α was set to the intermediate values of 0.0001 and 0.00001, the performance was poor. When α = 0.000001, the relative requirements were the lowest, the range of the spreading attempts was larger, and there were more opportunities for activation. However unnecessary attempts were easily produced, and the running time was increased.

The value of the threshold directly affects the length of the diffusion path of the node, and was also found to result in a significant difference in the algorithm running time, as presented in Fig. 7. When the value of α was large, the length of the diffusion path was shortened significantly, thereby avoiding repeated partial path activation attempts and greatly reducing the running time. This phenomenon will become more obvious for denser networks.

5. Conclusion

This paper investigated the issue of influence maximization in social networks. Information tends to spread within communities, and



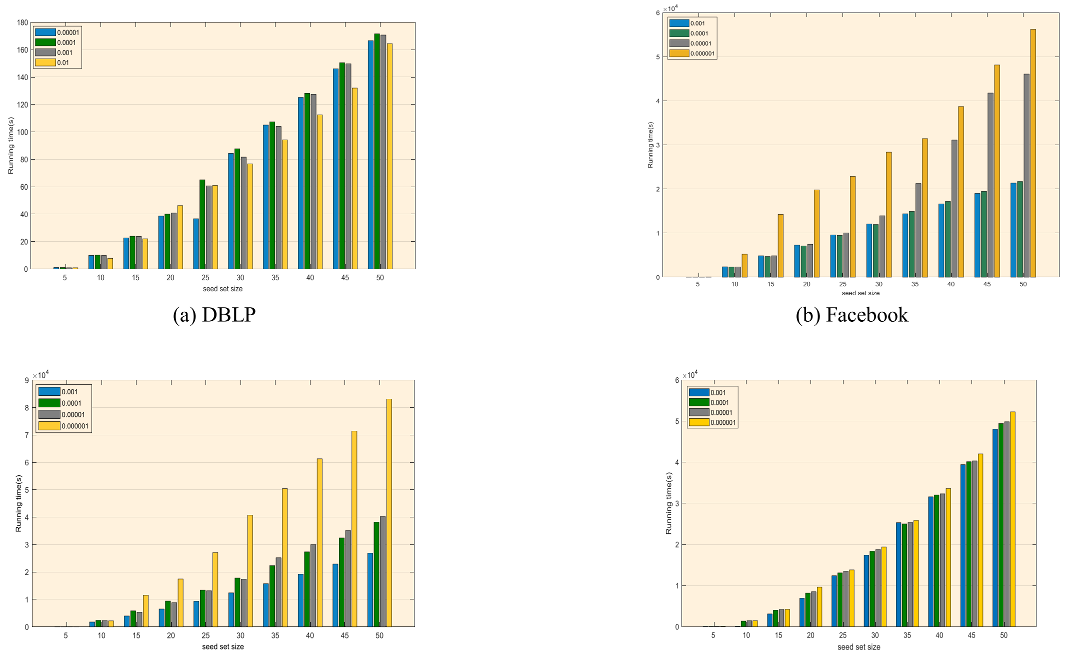
(a) DBLP

(b) Facebook

(c) viki-Vote

(d) CA-HepPh

Fig. 6. Impacts of different propagation control factors.



(a) DBLP

(b) Facebook

(c) viki-Vote

(d) CA-HepPh

Fig. 7. Running time of different propagation control factors.

individuals have autonomy. Considering these factors, as well as by taking into account a balance between accuracy and efficiency, an influence maximization algorithm that combines the propagation dynamics DEIM algorithm was designed in this work. To simplify the process of seed selection, the overlapping community partition results obtained by the CeCOPRA algorithm are used in the DEIM algorithm to remove insignificant nodes, and a candidate set with the potential to become a seed is selected. The CeCOPRA algorithm defines the concept of cohesive entropy to accurately quantify the similarity between users. Unlike previous similarity measurements based on the distance formula, the cohesive entropy is used in the proposed algorithm to calculate the similarity of the information carried between users, thereby improving the accuracy of the community division result. Then, by defining the cohesive power between users and the propagation control factors to imitate the dynamic process of the message dissemination, an ODP algorithm was proposed to determine the final seeds. Experimental results indicate that the proposed DEIM algorithm is more effective while ensuring the influence of the propagation range. But for large-scale data, the advantage of DEIM algorithm is not obvious. In future work, we will improve this point and avoid using the iterative approach to community division. In addition, multiple relationships between users and numerous types of influences will be considered. Additionally, the application characteristics of different scenarios will be analyzed to ensure that the selected seed node has the greatest influence after integrating various situations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by National Key Research and Development Program of China (No. 2017YFE0117500) and the Natural Science Foundation of China (No 61762002).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2020.114207.

References

Banerjee, S., Jenamani, M., & Pratihar, D. K. (2019). ComBIM: A community-based solution approach for the Budgeted Influence Maximization Problem. *Expert Systems with Applications, 125*, 1–13.

Barthelemy, M. (2004). Betweenness centrality in large complex networks. *The European physical journal B, 38*(2), 163–168.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., et al. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment, 2008*(10), P10008.

Chen, W., Wang, Y., & Yang, S. (2009). Efficient influence maximization in social networks[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. *ACM*, 199–208.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers[C]// Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. *ACM*, 57–66.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks, 1*(3), 215–239.

Gregory, S. (2010). Finding overlapping communities in networks by label propagation. *New Journal of Physics, 12*(10).

He, Q., Wang, X., Lei, Z., et al. (2019). TIFIM: A Two-stage iterative framework for influence maximization in social networks. *Applied Mathematics and Computation, 354*, 338–352.

Hu, Y., Chen, H., Zhang, P., et al. (2008). Comparative definition of community and corresponding identifying algorithm. *Physical Review E, 78*(2).

Hwang, C. L., & Yoon, K. (1981). *Methods for multiple attribute decision making[M]// Multiple attribute decision making* (pp. 58–191). Berlin, Heidelberg: Springer.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network[C]//Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. *ACM*, 137–146.

Kitsak, M., Gallos, L. K., Havlin, S., et al. (2010). Identification of influential spreaders in complex networks. *Nature physics, 6*(11), 888.

Leskovec, J., Krause, A., Guestrin, C., et al. (2007). Cost-effective outbreak detection in networks[C]//Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. *ACM*, 420–429.

Li, W., Fan, Y., Mo, J., et al. (2020). Three-hop velocity attenuation propagation model for influence maximization in social networks. *World Wide Web, 23*(2), 1261–1273.

Okamoto, K., Chen, W., & Li, X. Y. (2008). *Ranking of closeness centrality for large-scale social networks[C]//International Workshop on Frontiers in Algorithmics* (pp. 186–195). Berlin, Heidelberg: Springer.

Qin, J., Xu, J. J., Hu, D., et al. (2005). *Analyzing terrorist networks: A case study of the global salafi jihad network[C]//International Conference on Intelligence and Security Informatics* (pp. 287–304). Berlin, Heidelberg: Springer.

Richardson, M., & Domingos, P. (2002). Mining knowledge-sharing sites for viral marketing[C]//Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. *ACM*, 61–70.

Ruhnau, B. (2000). Eigenvector-centrality—a node-centrality? *Social networks, 22*(4), 357–365.

Shi, Q., Wang, C., Chen, J., et al. (2019). Post and repost: A holistic view of budgeted influence maximization. *Neurocomputing, 338*, 92–100.

Singh, S. S., Singh, K., Kumar, A., et al. (2018). *Coim: Community-based influence maximization in social networks[C]//International Conference on Advanced Informatics for Computing Research* (pp. 440–453). Singapore: Springer.

Tang, J., Tang, X., & Yuan, J. (2018). An efficient and effective hop-based approach for influence maximization in social networks. *Social Network Analysis and Mining, 8*(1), 10.

Wang, X., Wang, X., & Li, W. (2019). A Warning Propagation Algorithm for Solving Minimum Cut. *ACTA ELECTRONICA SINICA, 47*(11), 2386–2391.

Yang, W., Brenner, L., & Giua, A. (2019). Influence maximization in independent cascade networks based on activation probability computation. *IEEE Access, 7*, 13745–13757.

Zareie, A., Sheikhahmadi, A., & Khamforoosh, K. (2018). Influence maximization in social networks based on TOPSIS. *Expert Systems with Applications, 108*, 96–107.

Zhang, Q., Li, M., & Deng, Y. (2018). Measure the structure similarity of nodes in complex networks based on relative entropy. *Physica A: Statistical Mechanics and its Applications, 491*, 749–763.

Zhang, Y., Gu, Q., Zheng, J., et al. (2010). *Estimate on expectation for influence maximization in social networks[C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 99–106). Berlin, Heidelberg: Springer.