Information Sciences 609 (2022) 578–594

Contents lists available at ScienceDirect

Information Sciences

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

CBIM: Community-based influence maximization in multilayer networks

Venkatakrishna Rao K [a,*], C. Ravindranath Chowdary [a]

[a] *Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi 221005, India*

ARTICLE INFO

Article history:
Received 2 August 2021
Received in revised form 27 April 2022
Accepted 17 July 2022
Available online 20 July 2022

Keywords:
Social networks
Influence maximization
Community detection
Seed nodes
Multilayer networks

ABSTRACT

Selecting seed nodes (most influential nodes) in networks has attracted attention due to seed nodes' ability to influence and spread information. Seed nodes are essential to understanding the spreading and controlling of the information dynamics of the networks. Influence maximization (IM) is predominant in monolayer networks. After the advancements and widespread usage of social networks, applying influence maximization to multilayer networks is gaining popularity. Identifying influential nodes precisely in multilayer networks is a challenging and yet unexplored task. Based on studies, individuals in a community interact frequently and are more likely to influence each other. Motivated by this observation, this paper proposes community-based influence maximization (CBIM) model to find k seed nodes in multilayer networks. CBIM has two phases: In the first phase, CBIM uses the function $FIC(M)$ to find the small communities from a multilayer network based on dice neighborhood similarity. It uses the function $CSC(CS_{init}, \theta)$ to merge smaller communities and generate larger communities to improve communities' quality. In the second phase, CBIM computes Edge Weight Sum (EWS) for each node in a community and ranks the nodes based on EWS. CBIM uses the quota-based approach to select the seed node set from the communities based on the ranks. A comparative study of various influence maximization (IM) algorithms shows that the CBIM algorithm performs better than the state-of-the-art. The simulation studies have shown that CBIM can detect a set of most influential nodes on real-time datasets under various settings and environments.

1. Introduction

With the advent of the Internet, various Web technologies and online social networking services have emerged. Social networks allow users to have connections (friends), and they facilitate sharing thoughts among friends through comments, pictures, etc. Around 68% of online users have a social profile to get news, connect with friends, family, or other interesting users. Many of these users form or join online communities, and they are more powerful marketing tools than regular traditional advertisements in press and media.

Influence maximization (IM) is studied in different domains like rumor control, viral marketing, information diffusion, etc. Kempe et al. [1] formulated the IM problem by selecting the subset of seed nodes with size constraints to influence the largest number of nodes. After selecting seed nodes, they will be passed to a diffusion model. Authors propose the two well-known diffusion models [1], the Independent Cascade (IC) model and the Linear Threshold (LT) model. According

* Corresponding author.
E-mail addresses: vrkatakamsetty.rs.cse18@itbhu.ac.in (K Venkatakrishna Rao), rchowdary.cse@iitbhu.ac.in (C.R. Chowdary).

https://doi.org/10.1016/j.ins.2022.07.103
0020-0255/

to any diffusion model, the expected influence spread is maximized in multilayer networks by activating them. In both these models, a node is assumed to be either in an active state or an inactive state. The diffusion model plays a crucial role in the analysis of influence maximization.

Generally, there are four types of standard IM algorithms for data diffusion; the first is the Heuristic approach, the second is the Approximation approach, the third is the sampling-based approach, and the fourth is the community-based approach. Most of the previous influence maximization studies focused on heuristic approaches, approximation algorithms, and sampling-based algorithms. These types of models ignore the effect of community structure in influence distribution. In traditional methods, data distribution is over the network's partial area; those in another area of the network never get an opportunity to receive the information when the seed set k is limited. In reality, we need to avoid the imbalance distribution, i.e., make the influence distribution evenly over the network. Let us discuss the possible problems as an example. There are general elections every five years to elect the union government in India. In order to win with a high majority, each camp will promote its prime ministerial candidate. However, it is not enough to pursue the maximum total influence because each state has voting rights. Thus, we need to spread the advantage to support the prime ministerial candidate in every state. However, some less populated or less connected states may be ignored in the traditional IM models, which is not advisable.

In addition to this, social media users do not confine to a single social network; they participate in different social networks. In such a case, a user can propagate the information across different social networks, i.e., multilayer networks [2]. However, the majority of the existing studies focus on influence maximization in singlelayer networks and ignore some critical factors such as user engagement across the networks, the network of networks, etc. Therefore, we need to consider multilayer networks, as many users are actively engaged in propagating information, ideas, and innovations across the networks simultaneously. Despite this, relatively little research is dedicated to IM in multilayer networks. Qipeng et al. [3] formulated IM in multilayer networks as a multiobjective optimization problem and employed the classic non dominated Sorting Genetic Algorithm II (NSGA-II) to find a set of Pareto-optimal solutions that provide a wide range of options for decision-makers.

In this paper, we propose a community-based influence maximization (CBIM) in a multilayer network. CBIM's seeding strategy depends on community structure. This paper addresses the influence maximization problem in two phases. In the first phase, CBIM finds the communities based on neighborhood similarity from the multilayer network. In the second phase, CBIM finds the seed nodes from communities by employing a quota-based approach. However, due to the nonavailability of datasets, we compute the neighborhood similarity and the seed nodes independently for each layer in a multilayer network. This is the first attempt to identify seed nodes based on community structure in multilayer networks. The contribution of the proposed work is given as follows.

- We adapted a community detection algorithm [4] to a multilayer network.
- We propose EWS to assign edge weights based on degree and distance concerning that node in multilayer network.
- We propose a seed selection process based on a quota-based approach by considering community structure.
- Through experiments, we compare the performance of the proposed algorithm against the state-of-the-art algorithms under the IC and LT diffusion models on real-world social networks.

The rest of the paper is organized as follows. In Section 2, related works are reviewed. Section 3 discusses the multilayer network model with community-based influence maximization. In Section 4 we present the proposed CBIM model. In Section 5, we evaluate extensive experiments to analyze the performance of the proposed algorithm. In Section 6, we compare the results of our proposed algorithm with various IM algorithms. Finally, Section 7 is devoted to the conclusion and future directions.

2. Related work

Influence maximization is actively studied in the literature [1,5,6]. The influence maximization problem of identifying seed nodes was first proposed by Domingos et al. [7,8] and formulated as optimization problem by Kempe et al. [1]. Since influence maximization problem is NP-hard, several heuristic based [9,10,40], meta-heuristics based [11], sampling based [12,13], location based [44], and several approximation algorithms [14,15] have been proposed.

Because of the low efficiency of the monte-Carlo simulation, Leskovec et al. [16] tried to accelerate the lazy greedy algorithm by ignoring the unnecessary computations and referred to it as a cost-effective lazy forward (CELF), which is nearly 700 times faster than the general greedy algorithm. Goel et al. [17] proposed CELF++, which improves the performance 35 % to 55% than CELF. Although these improved algorithms increased runtime performance, they have poor scalability for more extensive networks. To address the scalability issues, researchers proposed various heuristic methods, including PMIA [18] and degree discount [19] to approximate the influence propagation using node's local structures. Finding seed nodes is a difficult task due to the non-availability of data in text, and due to this, the results will not be impressive. To overcome this, Kim et al. [45] proposed a multimodel deep learning model, influencer profiler (Infprofiler), which uses text and image information for finding seed nodes. Apart from that, Cai et al. [46] proposed Holistic Influence Maximization (HIM) query problem to find the minimum set of seed users who can cover all the targeted users in the network. Most IM problems focus only on cyber interactions and ignore physical interactions, but HIM focuses on both. However, these traditional algorithms ignore the community diversity of activated nodes in social networks and the time cost of the IM problem.

Wang et al. [43] proposed community-based influence maximization (CIM) to find the seed nodes from the communities and address the lack of community diversity and time cost in IM problems. Yi-Chen et al. [20] proposed a new framework called community-based influence maximization. It comprises three phases; the first is community detection using hierarchical clustering; the second is candidate generation, which aims to determine a set of candidate seeds based on community size and connectivity; the third is final seed selection from the communities. Kai Sheng et al. [21] have proposed a similar algorithm with label propagation for community detection called LPIMA. The LeaderRank algorithm was used to quantify community nodes and then assign candidate seed nodes based on quantified values; finally, the submodel characteristic has been used to improve the greedy algorithm and select seed nodes. Jianxin et al. [22] proposed Community-diversified Influence Maximization (CDIM) to find k seed nodes using CPSP-Tree index measure, and it measures the community-diversified influence and addresses a series of computational challenges. However, this method is based on some theoretical models and is not practical for real-world scenarios.

However, the accuracy of selecting seed nodes is still concern, Xiao Li et al. [23] have proposed a community-based seed selection algorithm (CSS) framework for location-aware influence maximization, which depicts finding the seeds by constructing PR-tree-based indexes. But this framework is taking more network processing time after community partition. Wang Y et al. [43] proposed the CGA algorithm (Community-based greedy algorithm), which reduces the processing time of the network. Noha et al. [25] proposed the problem of temporal interaction-based community detection using clique structure. Xiaofei et al. [42] introduced the Multi-Community Influence Maximization (MCIM) problem to maximize influence by identifying seed users in multiple social communities of different properties and characteristics based on a total budget of seed users. Besides this, J. Guo et al. [26] proposed influence maximization with community budget (IMCB). IMCB uses Newman Moore greedy modularity maximization to detect communities, and then a continuous greedy process and pipage rounding are used to find seed nodes from communities.

Most of the methods mentioned above are community-based influence maximization in a single layer. Single layer networks ignore the most critical factors such as user engagements across the networks, the network of networks, etc. Nowadays, users usually have multiple social accounts, such as Facebook, Twitter, etc. Moreover, users have different influences on different platforms. For example, a user may have more followers on Twitter than on Facebook, and vice versa. Therefore we need to consider multilayer networks as many users are actively engaged and propagate ideas, information, and innovation across the networks simultaneously. Inspired by this, we propose the Community-based influence maximization (CBIM for short) model; it comprises two phases. The first phase aims to find communities in the network, and it consists of two parts. The first part of phase one aims to find initial communities in multilayer networks, and the outcome is a large number of small communities. It does not fulfill the fundamental characteristic of the community detection problem. The second part of phase one tries to improve the final quality of communities in multilayer networks through the *community consolidation* ($CSC()$) function using *community conductance*, *communityscale and mergingindex*. The second phase aims to find the seed nodes from various communities generated in phase one using EWS, a quota-based approach. The details of the proposed method are elaborated in Section 4.

3. Preliminaries

This section consists of two parts. The first part is the model of multilayer networks and the second part is for influence maximization. In first part, we present the model of multilayer networks, each layer has the same number of nodes and intra-edges. In second part, we present influence maximization (IM). For IM, we use two well-known diffusion models to disseminate information from the seed nodes.

3.1. Basic model of multilayer networks

Let us consider a multilayer network $M = \{G^1, G^2, \ldots, G^\alpha, .., G^M\}$ [43], where each graph $G^\alpha = (V, E^\alpha)$ is formed by the set of N nodes in each layer. $V = \{i; i = 1, 2, 3..N\}$ and E^α is set of edges in layer α. Each layer consists of same set of nodes N.

The information propagation dynamics of a multilayer network are different from a single layer network. To demonstrate the information spreading in a multilayer network, let us consider a simple multilayer network in Fig. 1. Fig. 1(A) shows two different graphs in a multilayer network, where one graph in layer one and another graph in layer 2; 13 nodes exist in each layer. Fig. 1(B) shows the two graphs after identifying communities in a multilayer network, three communities are identified in layer 1 and two communities are identified in layer 2; layer 1 has $C1 = \{1, 2, 3, 4, 5, 6\}, C2 = \{7, 8, 9\}$ and $C3 = \{10, 11, 12, 13\}$; layer 2 has $C1 = \{1, 2, 3, 5, 6\}, C2 = \{4, 7, 8, 9, 10, 11, 12, 13\}$. Fig. 1(C) shows seed nodes in different communities, i.e., nodes 2, 7, and 12 are selected as seed nodes in layer one, and node 6, 8 are selected as seed nodes in layer 2. These seed nodes will propagate the information to other nodes after running a diffusion model.

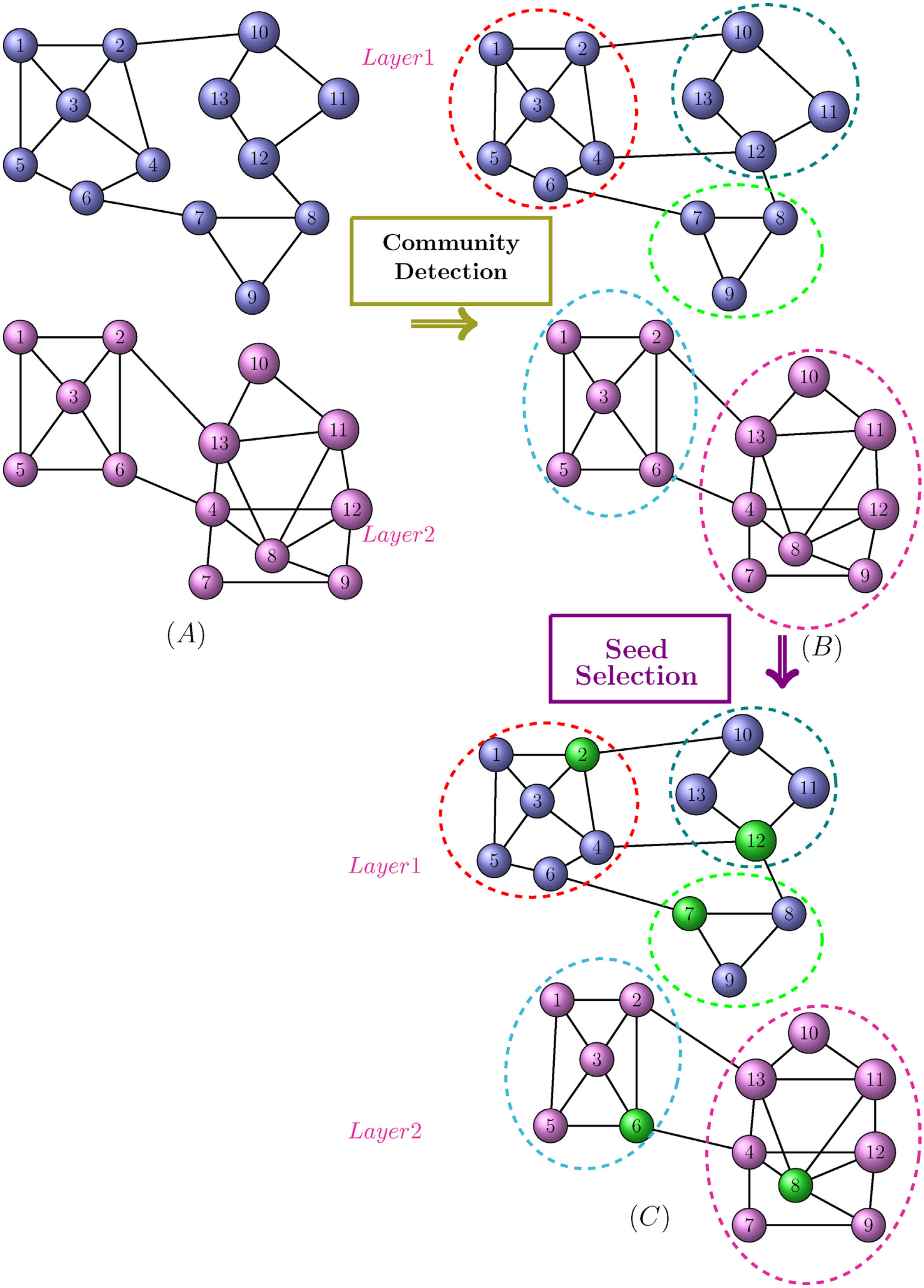


Fig. 1. Overview of CBIM model.

3.2. Influence maximization

Multilayer influence maximization refers to the process of finding out the set of most influential nodes from all the layers, called seed nodes. Diffusion models will maximize the influence across the networks by taking seed nodes as input. Two well-known diffusion models are generally used for influence maximization: the independent cascade (IC) model and the linear threshold (LT) model. In both these models, every node is assumed to be either in active or in inactive state [1].

The information spread in the network can be analyzed in an iterative process. In the first iteration, the seed nodes activate their neighbors using one of the two diffusion models. In the subsequent iteration, the activated nodes will activate their neighbors, and the process continues until no node is left non-participated in action. Heuristics to find the seed nodes in influence maximization of multilayer networks will be discussed in Section 4. In the independent cascade [or IC] model, when a node becomes active at time step t, it will get a single chance to activate its inactive neighbors at time step $t+1$ with propagation probability p. In Linear threshold [or LT] model, every node is associated with an activation threshold between 0 and 1. At any time step t, if the sum of the weight of the incoming edges is greater than the activation threshold of the node, then the node enters into an activated state. The sum of all the incoming edge weights of any node is assumed to be at most 1.

A node v^l influenced by each neighbour w^l according to b_{w^l,v^l} such that

$$\sum_{w^l \in nbrs(v^l)} b_{w^l,v^l} \leqslant 1$$

A node v^l becomes active when at least (weighted) θ_{v^l} fraction of its neighbors are active, i.e.,

$$\sum_{w^l \in actnbrs(v^l)} b_{w^l,v^l} \geqslant \theta_{v^l}$$

In both equations, the model's influence propagation happens until no new node becomes active.

4. Community Based Influence Maximization (CBIM)

In this section, we discuss the proposed model, which consists of two phases. In the first phase, the function $FIC(M)$ finds the small communities from a multilayer network based on dice neighborhood similarity, and then the function $CSC(CS_{init}, \theta)$ merges some small communities of $FIC(M)$ and forms final communities to improve the quality of communities. In the second phase, we find the EWS for each node in a community and assign a ranking to the nodes based on the EWS, then we select the seed node set from the communities based on assigned ranking using the quota-based approach. The proposed method is outlined in Algorithm 1.

Algorithm 1 CBIM: Community-based Influence Maximization

1: ▹ $FIC(M)$ is Function for finding initial communities, CS_{init} is set of initial communities, FC is final community set, θ is community scale, $CSC()$ is community consolidation function

2: Multilayer network $M = \{G^1, G^2, \dots, G^\alpha, .., G^M\}$

3: Form the initial community structure

$CS_{init} \leftarrow FIC(M)$ ▹FIC() is discussed in Section 4.1.2v 4: Combine small communities in CS_init

$FC \leftarrow CSC(CS\_init, \theta)$ ▹ $CSC()$ is discussed in Section 4.1.3

5: Find the EWS of all the nodes.

6: Identify seed nodes from communities based on quota-based approach using EWS.

4.1. Identification of communities

Here we discuss the first phase of CBIM, i.e., finding the communities in multilayer networks. Identification of communities is similar to a standard community detection algorithm. Finding the communities consists of two parts. In the first part, we identify initial communities based on neighborhood similarity. In the second part, we merge small communities based on the merging index factor to form final communities.

4.1.1. Neighborhood similarity selection

Several methods have been proposed in the literature for community detection, i.e., based on the distance between two nodes, paths between nodes, and the number of familiar neighbors, etc. However, those methods depend on global factors such as eigenvalues, euclidean distance, and prior knowledge; getting these factors is hard to acquire due to the large and large networks involved, and most of them are demanding computationally high time complexity. To overcome it, some neighborhood similarity measures have been tested to select the best communities for our influence maximization algorithm, such as the Jaccard similarity coefficient [27], Otsuka similarity coefficient [28], Overlap similarity coefficient [29],

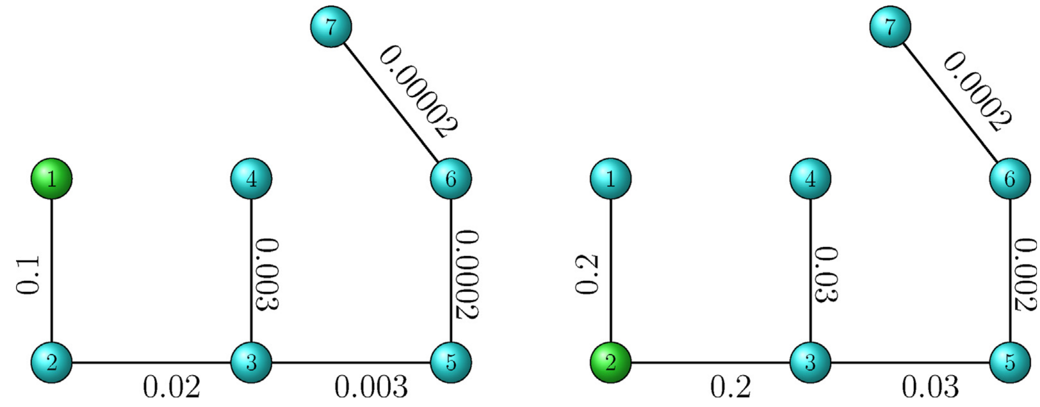


Fig. 2. (A) Weight distribution to the edges for node 1 and EWS for node 1 is 0.12622 (B) Weight distribution to the edges for node 2 and EWS for node 2 is 0.4622.

and Dice similarity coefficient [30]. Among these, dice similarity is giving some impressive results as it considers twice the number of common neighbors of nodes u^l and v^l.

Dice similarity coefficient (DSC)[30]: It is defined as the two times the number of common nodes in both neighbors sets divided by the total number of nodes in two neighbor sets. Dice similarity coefficient is $DSC(u^l, v^l)$,

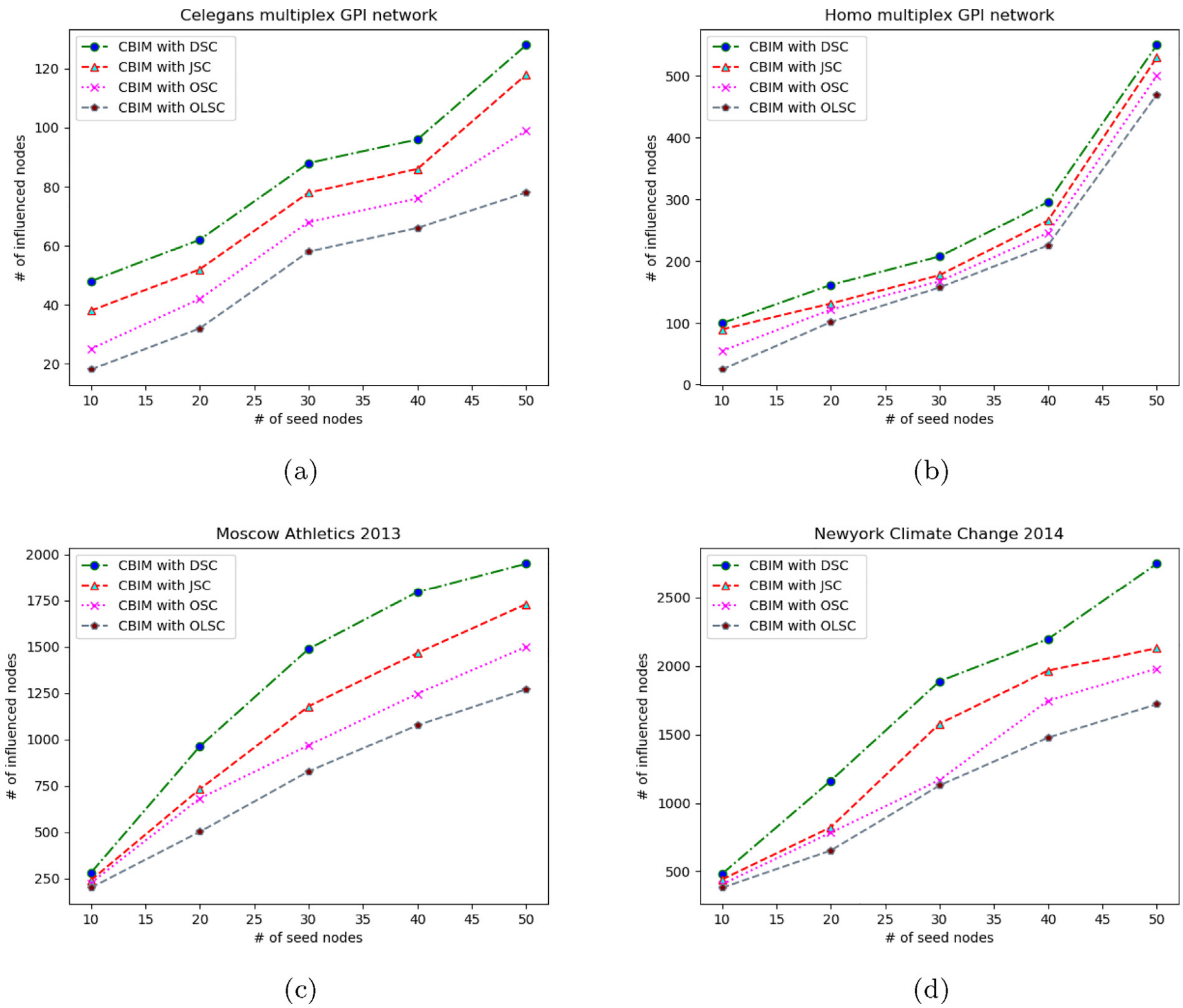


Fig. 3. Influence spread of 50 seed nodes for different similarities with our proposed CBIM algorithm under various datasets.

$$DSC(u^l, v^l) = \frac{2 * |nbrs(u^l) \cap nbrs(v^l)|}{|nbrs(u^l)| + |nbrs(v^l)|} \tag{1}$$

$nbrs(u^l)$ is neighbors of node u in layer l and $nbrs(v^l)$ is neighbors of node v in layer l. After comparing various similarity indexes with our proposed algorithm, we have selected the dice similarity index for finding communities in the network since its performance is better than the remaining similarity measures. Fig. 3 shows the IM performance of different similarity measures with our CBIM model.

4.1.2. Finding initial communities (FIC)

Here, we discuss part one of identifying communities, i.e., the initial community structure from a multilayer network. Cheng et al.[4] has proposed community detection in a single layer network using neighborhood similarity. We adapted [4] and applied it to multilayer networks for community detection. First, select the highest degree node from the network and make it a new community. Next, add the node with the maximum dice similarity coefficient into the new community. After this, select the next highest degree node from the remaining network and find the node with the maximum dice similarity coefficient. If the most similar node is already in a community, add the next highest degree node to that community. Otherwise, we create a new community for the next highest degree node and its most similar node. This process is repeated until every node is in some community. In Algorithm 2, part one explains the procedure of finding the initial community structure.

The initial community structure works based on node selection and node similarity from the above discussion. Generated communities in part one are too small, and a large number of small communities are generated. The idea of community detection will not be fulfilled with small communities, and it fails the fundamental characteristic of the community.

Algorithm 2 Community detection on neighborhood similarity measure

---

/* **Part 1 - Initial communities detection- FIC(M)** */
1: Initialize the variables NS and CS_{init} which are used to record node set and initial community structure.
$NS \leftarrow V, CS_{init} \leftarrow \Phi$
2: Select the highest degree node v^l from the node set (NS)
3: Get the most similar neighbour (sn^l) of v^l using Dice's neighbour similarity.
4: **If** sn^l is not in any community **then**
5: Create a new community and assign v^l and sn^l to it.
$K \leftarrow |CS_{init}|; C^l_{k+1} \leftarrow \{v^l, sn^l\}$
6: Insert the new community into commmunity structure.
7: Remove nodes v^l and sn^l from NS as they are classified
$NS \leftarrow NS - \{v^l, sn^l\}$
8: **else**
9: Find the community to which sn^l belongs to and denote it as C^l_k
$k \leftarrow locate(CS_{init}, sn^l)$
10: Insert node v^l into C^l_k and Remove the node v^l from NS
$C^l_k \leftarrow C^l_k \cup \{v^l\} and NS \leftarrow NS - \{v^l\}$
11: Repeat the steps from 2 to 10, until $NS = \Phi$
/* **Part 2 - Community consolidation- CSC(CS_{init}, θ)** */
12: Intialize Final Communities (FC)
$FC \leftarrow CS_{init}$
13: **Step 1:** Calculate community conductence (γ_i) and community scale (θ_i) for each community C^l_i in CS_{init}.
14: Calculate the merging index (ψ_i) for each community in CS_{init}.
$C^l_i(\psi) = C^l_i(\theta) * C^l_i(\gamma)$
15: Select the community with lowest merging index (C^l_x)
16: **Step 2:** Find the most similar community (C^l_y) to (C^l_x) and merge the two communities to form a new community (C^l_n).
$s \leftarrow argmax_i\{Sim(C^l_x, C^l_y)|i = 1, 2, \ldots CS, x \neq y\} and C^l_n = C^l_x \cup C^l_y$
17: Calculate the merging index (ψ_n) for new community (C^l_n)
18: Replace two communities C^l_x and C^l_y with new community C^l_n in final community set (FC)
19: Repeat the process from 15 to 18 until $\psi^l_i > \delta$
20: return FC

4.1.3. Community consolidation (CSC)

In the first part of community detection, there are many small communities. In these small communities, edges connecting inside the community are less than edges connecting outside. This violates the fundamental characteristic of the community; therefore, it is essential to merge small communities to make a final community structure. In the second part, we have proposed community consolidation, i.e., merging small communities to form sizable communities. In merging small communities, the first step is to find the communities to be merged, and the second step is to select the communities into which each small community is to be merged.

For the first step, we propose a merging index to find the communities that need to be merged. The merging index considers two factors: community conductance and community scale.

Definition 1 (*Community conductance [31]*). Conductance is the measure to use the quality of the community; with this measure, the best communities are a densely connected set of nodes attached to the rest of the network via a few edges.

$$C_i^l(\gamma) = \frac{|E_i^{out}|}{2|E_i^{in}| + |E_i^{out}|} \tag{2}$$

where E_i^{out} is set of edges connecting community C_i^l with other communities and E_i^{in} is set of edges in community C_i^l.

Definition 2 (*Community scale [4]*). Scale of community $C_i^l(\theta)$ is defined as

$$C_i^l(\theta) = \frac{|V_i^l|}{|V|} \tag{3}$$

where V_i is a set of nodes in community C_i^l and V is a set of nodes in a multilayer network based on the above definitions.

Definition 3 (*Merging index*). Merging index of community $C_i^l(\psi)$ is the combination of community conductance and community scale, which is defined for community C_i^l as follows:

$$C_i^l(\psi) = C_i^l(\theta) * C_i^l(\gamma) \tag{4}$$

the first step can be solved by setting the merging index threshold, δ. If the $\psi_i < \delta$ then community C_i^l need to be merged with another community.

The second step is to select the communities into which each small community is merged based on the similarity between communities, i.e., small communities are merged into their adjacent similar communities. The similarity between the two communities C_i^l and C_j^l is given as follows:.

$$Sim(C_i^l, C_j^l) = \frac{\Sigma_{u^l \in C_i^l, v^l \in C_j^l} DSC(u^l, v^l)}{|C_j^l|} \tag{5}$$

where $DSC(u^l, v^l)$ is dice similarity between nodes $u^l \in C_i^l$, $v^l \in C_j^l$. C_i^l needed to be merged in C_j^l. $\Sigma_{u^l \in c_i^l, v^l \in c_j^l} sim(u^l, v^l)$ is sum of dice similarities between node u^l in community C_i and v^l in community C_j in layer l. In, Algorithm 2, Part 2 explains the community consolidation procedure. Part 2 is to improve the quality of community structure by merging small communities generated in Part 1. Most of the time is taken for identifying communities needed to merge with adjacent communities and similarity value.

4.2. Selection of seed nodes

In this Section we discuss the second phase of the proposed method and present how to select seed nodes after finding all the existed communities in a multilayer network. After finding the communities, we calculate the EWS for each node in communities. Based on the EWS, we select the seed nodes from each community using a quota-based approach. Finally, we form a seed node-set after merging all the selected seed nodes from each community.

4.2.1. Finding Edge Weight Sum (EWS)

The seed nodes are determined to maximize the influence over a network. The EWS is similar to Katz centrality [32]. Unlike Katz centrality, EWS considers degree and distance to compute the weight of the edge. If a node j is far away from node i, EWS penalizes for distance, but it rewards based on the degree of j. The main idea of computing EWS is to weigh each node in the community based on the performance, before selecting seed nodes from the communities. The purpose of calculating the EWS is to find the node's connectivity over the network. Node's connectivity depends not only on its degree but also on its neighbor's connections and neighbors of neighbors connections etc. EWS of a node i in layer l is calculated as

$$EWS_i^l = \Sigma_{hl=1}^{\infty}\Sigma_{j=1}^{n}[\alpha^{hl} * d_j^l] * (A^{hl})_{ij} \tag{6}$$

EWS_i^l is the edge weight sum of node i in layer l, n is the number of nodes in community (C_i^l), the adjacency matrix A is a n $\times$ n matrix. $(A^{hl})_{ji}$ is the number of walks of length hl starting from node i to node j. d_j^l is degree of node j in layer l. α is the attenuation factor, and the value is fixed as 0.1 because it gave the best results for all kinds of graphs and experiments. Fig. 2 shows an example of how to distribute the edge weight of a simple graph concerning nodes 1 and 2; it also explains the calculation of the EWS of a node. EWS for node 1 in Fig. 2 is 0.12622, node 2 is 0.4622, node 3 is 0.942, node 4 is 0.1642, node 5 is 0.482, node 6 is 0.4262 and node 7 is 0.12262.

Algorithm 3 Finding EWS and seed node selection

Input: Final Communities (FC)
Ouput: Seed Node set (SN)
1: Intialize the EWS, Seed node set (SN) and Community set (CS).
$SN \leftarrow \Phi, CS \leftarrow FC, EWS \leftarrow \Phi$
2: **for** C_i^l to CS **do**
3: **for** j to C_i^l **do**
4: Calculate the EWS for node j in community (C_i^l).
$EWS_i^l = \Sigma_{hl=1}^{\infty}\Sigma_{j=1}^{n}[\alpha^{hl} * d_j^l] * (A^{hl})_{ij}$
5: Insert the EWS of node j in community C_i^l ($EWS_j(C_i^l)$) factor into EWS of community C_i^l ($EWS(C_i^l)$).
$EWS(C_i^l) \leftarrow EWS(C_i^l) \cup EWS_j(C_i^l)$
6: **end for**
7: **end for**
8: Sort all the nodes in each community based on EWS in descending order.
9: Calculate required seed nodes from each community in quota based approach.
$Quota(C_i^l) = k * \frac{n_i}{|V|}$
10: Select the quota number of highest EWS nodes as seed nodes from each community (C_i^l).
$SN = SN \cup Quota(C_i^l)$
11: return SN

4.2.2. Quota based selection

Here we discuss the selection of seed nodes from the generated communities; after finding the EWS of all the nodes in each community, we sort all the nodes in descending order based on the EWS. Then select the $Quota(C_i^l)$ number of highest EWS nodes from the community C_i^l. Calculation of $Quota(C_i^l)$ for community C_i in layer l is

$$Quota(C_i^l) = k * \frac{n_i}{|V|} \tag{7}$$

where k is the number of seed nodes, n_i is the number of nodes in community C_i^l in a multilayer network. $|V|$ is the number of nodes in multilayer networks. The Sum of all selected seed nodes from communities will form the seed node-set (SN).

4.2.3. Time complexity

Here we discuss the time complexity of the CBIM model. We calculated time complexity for the community detection phase, i.e., for Algorithm 2, and seed node selection phase, i.e., Algorithm 3. Let the time complexity for finding the degree of each node in a multilayer network be $O(V^2)$. The time complexity for finding the highest degree node is $O(VlogV)$, and the time complexity for finding community conductance is $O(2VM)$, where M is the size of the community. The time complexity for the community scale is $O(1)$, and the total time complexity for community detection of CBIM is $O(V^2 + VlogV + 2VM + 1)$. In the seed node selection phase, the time complexity for finding the EWS is $O(CS^3)$, where CS is the size of community. The time complexity to sort the nodes in descending order is $O(MlogM)$. The time complexity to select the seed nodes is $O(K)$. Time complexity for seed node selection phase is $O(CS^3V + MlogM + V)$.

5. Experimental setup

In this section, we first describe the datasets used to evaluate and compare the heuristics. Then, we present the methods used to compare the heuristics. The python programming language is used to evaluate the performance of the proposed algorithm. The platform used for setting up the test-set is an Intel Xeon(R) workstation with 2.20Ghz CPU, 32 GB RAM, and Ubuntu 16.04.

5.1. Datasets

We have used several real-time datasets to evaluate algorithms: (i) Celegans Multiplex GPI Network [33] consists of layers corresponding to different synaptic junctions. (ii) Homo multiplex GPI network [34] consists of genetic and protein interaction data from humans and model organisms. (iii) Moscow Athletics 2013 [35] consists of data collected from Twitter during the 2013 World Championship in Athletics. The multiplex network in the paper uses three layers corresponding to retweets, mentions, and replies between 2013–08-05 and 2013–08-19. (iv) Newyork climate change 2014 [35] consists of data generated from Twitter during the 2014 people's climate March in New York. Data is based on retweets, mentions, and replies from 2014–09-19 to 2014–09-22. (v) ARXIV Net Science Multiplex [36] dataset consists of different Arxiv categories as layers in multiplex generated from the keyword "network" till 2014 journals. (vi) London Multiplex Transport [37] consists of data from the London rail network, nodes are train stations, and routes between train stations are edges. Train stations are connected overground, docklands light railway (DLR), and underground. The LT diffusion model is applied to the first four datasets mentioned above as they are directed graphs. IC diffusion model is applied to the remaining two datasets as they are undirected graphs. We also applied the IC model to the first four datasets by considering them as undirected graphs.

5.2. Methods to compare

In this section, we compare the proposed algorithm with below mentioned algorithms. These algorithms are similar to our proposed algorithm.

- **KSN [38]**: Knapsack seeding of networks (KSN) is another approximation algorithm, which parallelizes the problem in terms of the components of the layer- the difficulty lies in combining the solutions to the influence maximization problem on the separate layers to obtain a solution multiplex influence maximization (MIM). KSN achieves this by approximating the solution to the multiple-choice knapsack problem.
- **IMCS [24]**: Influence Maximization on Community Structure (IMCS) is an IM problem; in this, firstly, they divide the network into communities based on central nodes, the quality of community assessed based on community sparsity and node attribution. Based on community division analysis, the largest degree node selects as seed node from the communities.
- **IM-ELPR [39]**: Influence maximization using Extended H-index, label propagation with relationship matrix (IM-ELPR) has three steps. In the first step, use the h-index centrality to use the seed nodes and the use label propagation technique to detect communities. In the second step, with the help of the relation matrix, merge small communities to form large communities. In the third step, k-influential nodes will be selected from communities.
- **CIM [41]**: Clique-based influence maximization (CIM) is a heuristics-based algorithm; it finds all the available maximal cliques in a multilayer network. After finding the maximal cliques, seed nodes will be selected from the cliques based on their size.

6. Results

In this section, we compare the proposed algorithm, CBIM, with the algorithms described in Section 5.2, using the datasets listed in Section 5.1. We analyze the influence maximization of the heuristics and the execution time for finding the seed nodes. We compare CBIM with two community-based algorithms and two heuristic-based algorithms. We have reported the results of various algorithms on different datasets by fixing the seed set size k as 50. We discuss two scenarios. In the first scenario, we present the performance of the proposed heuristics for influence maximization using IC and LT models. In the second scenario, we report the execution time for finding seed nodes and the execution time for influence spread using IC and LT models.

6.1. Comparison of neighborhood similarities for CBIM algorithm

This section compares the influence maximization of different similarity indexes with the CBIM algorithm on different datasets. Fig. 3(a) presents the IM of various similarities with CBIM algorithm on Celegans multiplex GPI network dataset and dice similarity coefficient has performed better than other neighbor similarities. In addition to this, we also carried out four similarity experiments with some real-time datasets. Table 1.

6.2. Performance of CBIM on Real networks using IC model

Here, we present influence maximization on six real-time datasets for influence maximization. Under each dataset in Table 2, graphs have been used to apply on the independent cascade model. We also consider directed graphs as undirected graphs by ignoring the edge directions and applying them to the IC model for influence maximization. We find the influenced nodes for discussed algorithms under the IC model. Fig. 4(a) shows the influence maximization on London multiplex Transport network, CBIM has performed better than all other algorithms. CIM has performed better than IM-ELPR, KIN, and IMCS.

Table 1
Notations used in this paper.

Notation	Description
[0.5ex] N	Number of nodes per layer in multilayer network
G^{α}	α layer graph in Multilayer network M
E^{α}	edges in α layer in Multilayer network M
K	Number of seed nodes
p	Propagation probability in IC model
$\sigma(S)$	Influence Spread achieved by seed set S
d_v^l	Degree of v in l^{th} layer
$nbrs(v^l)$	Neighbours of node v in l^{th} layer
$actnbrs(v^l)$	Active neighbours of node v in l^{th} layer
t_v^l	Number of neighbours of node v in layer l that are already selected as seed node
b_{w^l,v^l}	Propagation probability of a node v^l influence by w^l
u^l	Node u in layer l

Table 2
Datasets used.

Dataset	# of Nodes	# of Edges	# of Layers
	directed graph		
Celegans multiplex GPI network	3879	8181	6
Homo multiplex GPI network	18222	170899	7
Moscow Athletics 2013	88804	210250	3
Newyork climate change 2014	102439	353495	3
	undirected graph		
London Multiplex Transport Network	369	441	3
EU-Air Transportation multiplex	450	3588	37
Arxiv netscience multiplex	14489	59026	13

IMCS has performed less than other algorithms. Fig. 4(b) shows the influence maximization on Arxiv net Science multiplex, IM-ELPR, and CIM have influenced almost the same number of nodes, KSN and IMCS influenced the same number of nodes. CIM has performed better than other algorithms. Due to the graph connectivity, seed nodes and some nodes may influence the exponential number of nodes, leading to a sudden curve rise. Fig. 5(a) shows the influence maximization on Celegans multiplex GPI network dataset; CBIM performed better than all other algorithms. IMCS has influenced less number of nodes. IM-ELPR has performed better than heuristic algorithms. KSN and CIM performed better than the IMCS. Fig. 5(b) shows the influence maximization on Homo multiplex GPI network dataset, CBIM influenced marginally more number of nodes than CIM and IM-ELPR. KSN and IMCS influenced the same number of nodes.

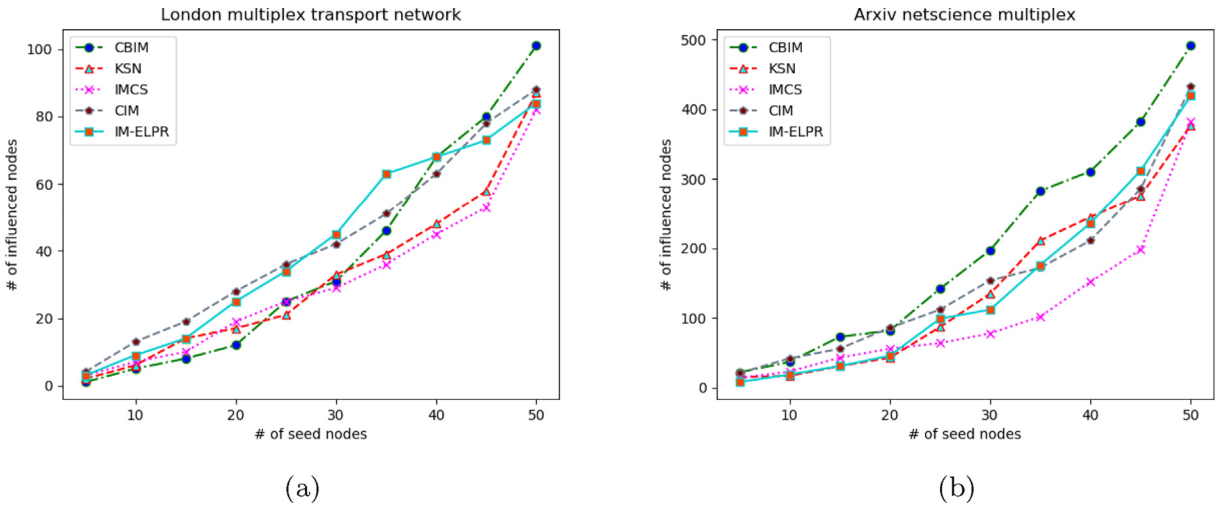
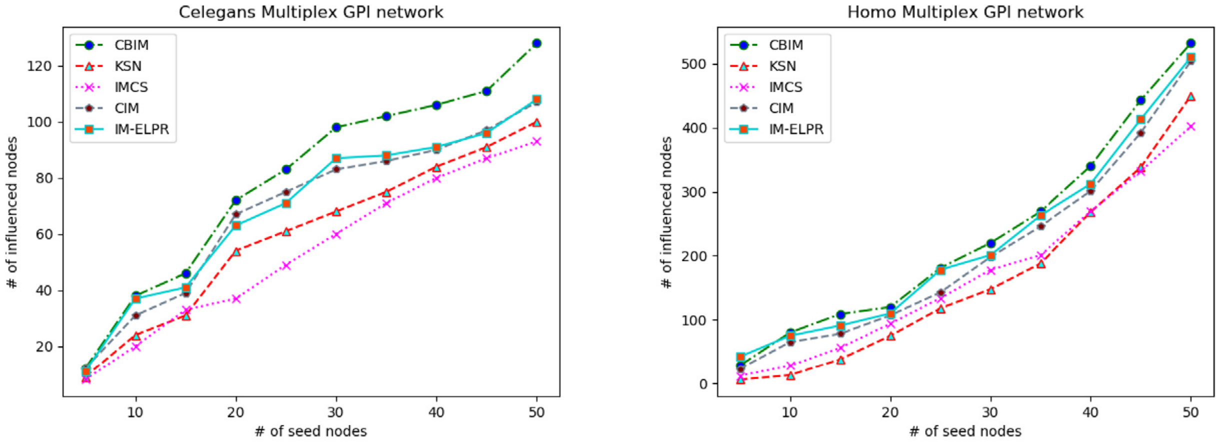


Fig. 4. (a) Influence spread of 50 seed nodes for different heuristics using IC model on *London Multiplex Transport Network*. (b) Influence spread of 50 seed nodes for different heuristics using IC model on *Arxiv netscience multiplex* for influence maximization.



(a) (b)

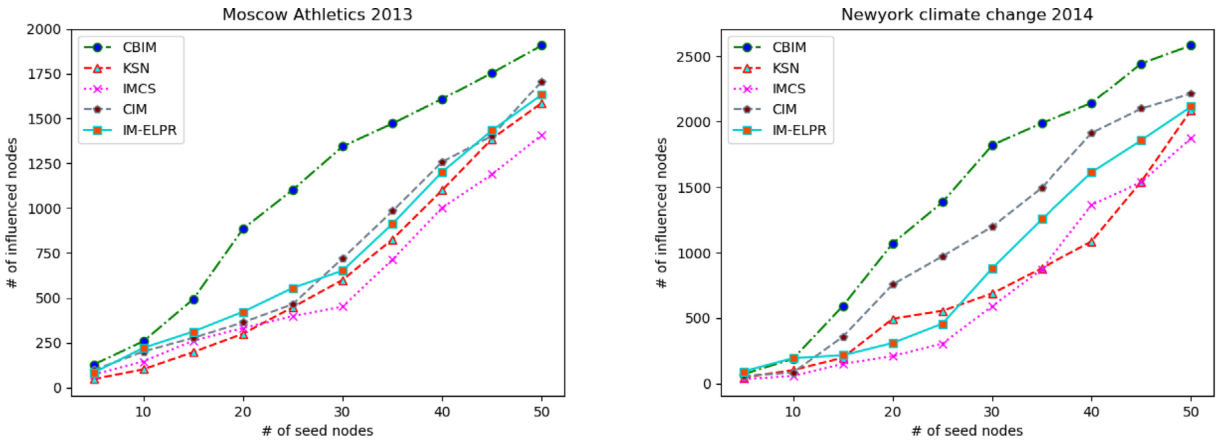
Fig. 5. (a) Influence spread of 50 seed nodes for different heuristics using IC model on *Celegans multiplex GPI network*. (b) Influence spread of 50 seed nodes for different heuristics using IC model on *Homo multiplex GPI network* for influence maximization.

Fig. 6(a) shows the influence maximization on Moscow Athletics 2013 datasets. CBIM has influenced more nodes, IMCS has influenced more nodes than KSN. CIM is performed better than KSN and IMCS. Fig. 6(b) shows the influence maximization on Newyork climate change 2014. CIM has performed better than IM-ELPR, KSN, and IMCS. CBIM performed better than all other algorithms.

6.3. Performance of CBIM on Real networks using LT model

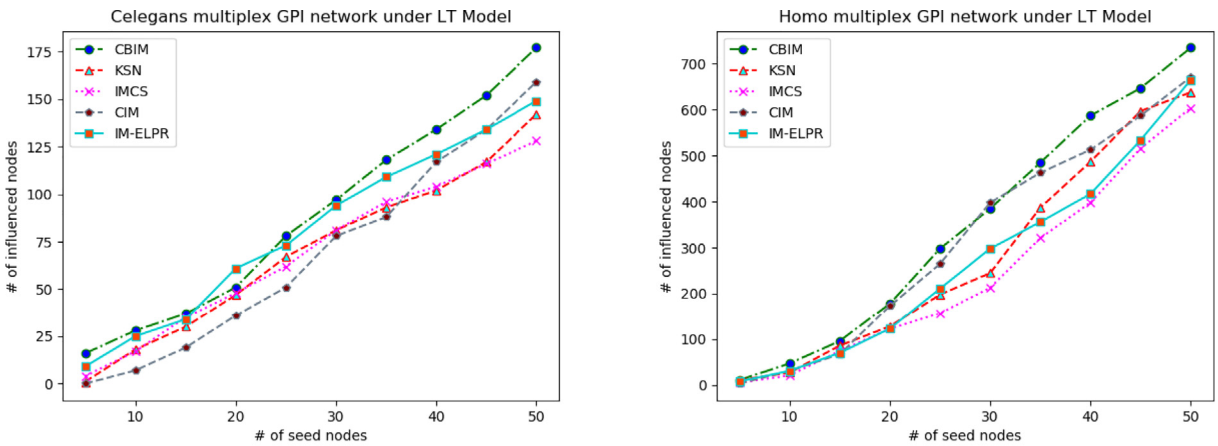
This section discusses the influence maximization of different algorithms under the LT model. In Table 2, directed graph datasets have been used to apply on linear threshold model. Fig. 7(a) shows the influence maximization of discussed algorithms on Celegans multiplex GPI network, for the same seed nodes and same datasets, LT model has influenced marginally more number of nodes than IC model. CBIM, IM-ELPR, and CIM have influenced almost the same number of nodes. IMCS and KSN have influenced the same number of nodes. Fig. 7(b) shows the influence maximization of discussed algorithms on the Homo multiplex GPI network. CBIM has influenced more nodes than other algorithms; CIM and IM-ELPR have performed marginally more than KSN.

Fig. 8(a) shows the influence maximization of discussed algorithms on Moscow Athletics 2013; CIM has performed better than all other algorithms. IM-ELPR has performed better than KSN and IMCS. Fig. 8(b) shows the influence maximization of



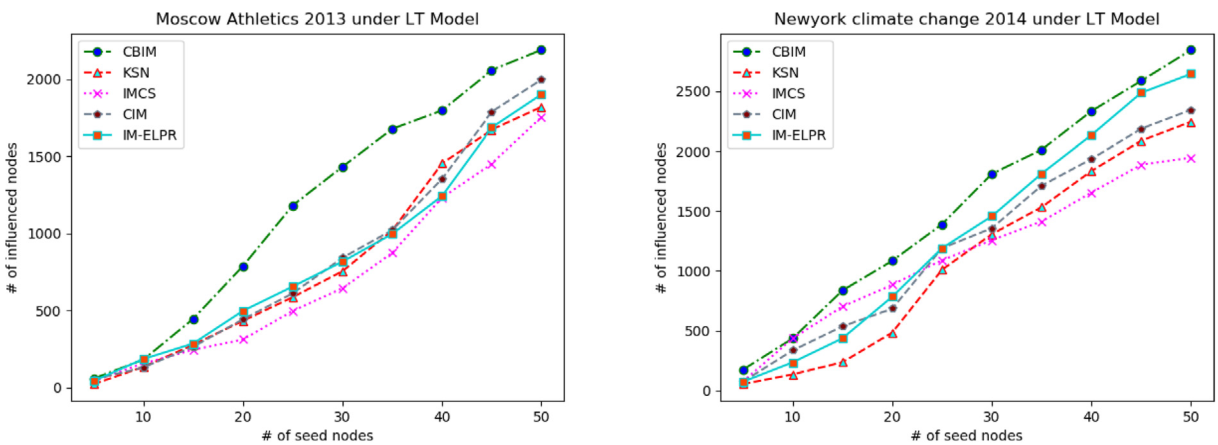
(a) (b)

Fig. 6. (a) Influence spread of 50 seed nodes for different heuristics using IC model on *Moscow Athletics* 2013. (b) Influence spread of 50 seed nodes for different heuristics using IC model on *Newyork climate change* 2014 for influence maximization.



(a) (b)

Fig. 7. (a) Influence spread of 50 seed nodes for different heuristics using LT model on *Celegans multiplex GPI network*. (b) Influence spread of 50 seed nodes for different heuristics using LT model on *Homo multiplex GPI network* for influence maximization.



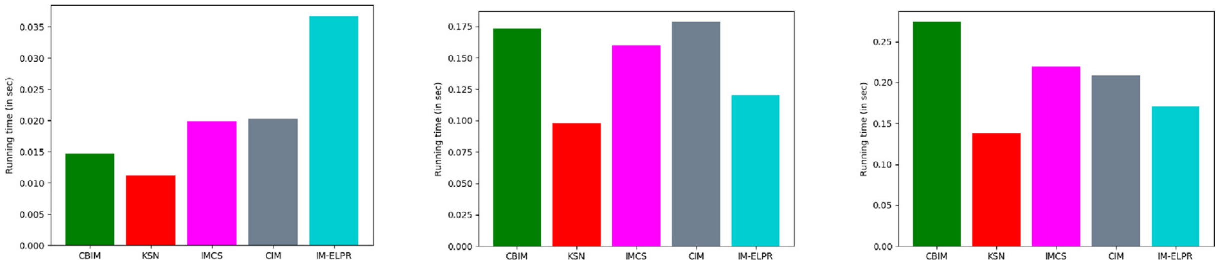
(a) (b)

Fig. 8. (a) Influence spread of 50 seed nodes for different heuristics using LT model on *Moscow Athletics* 2013. (b) Influence spread of 50 seed nodes for different heuristics using LT model on *Newyork climate change* 2014 for influence maximization.

discussed algorithms on the Newyork climate change 2014 dataset; However, IM-EPLR has performed better than CIM at the beginning, IM-EPLR and CIM have influenced the same number of nodes at the end.

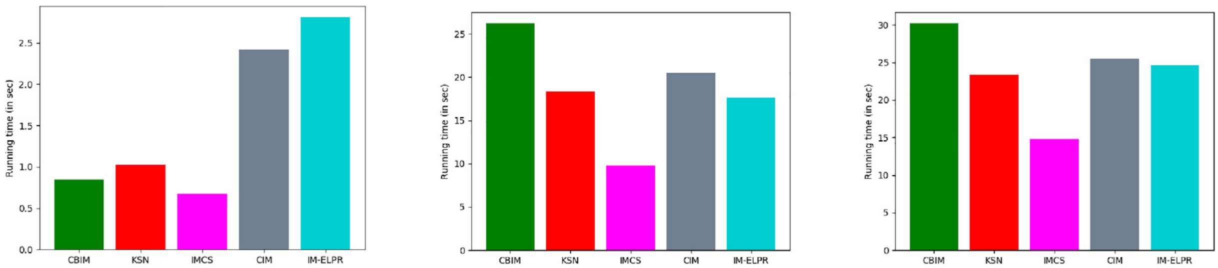
6.4. Execution time on real networks

This section discusses two scenarios; the first is to find seed nodes using different algorithms. The second scenario is the performance of influence maximization using the IC model. In most cases, CBIM is more efficient than other algorithms for finding seed nodes. But, some times CBIM takes more time than other algorithms to spread information as it influences more nodes relatively. Fig. 9(a) shows the execution time of identifying seed nodes using the Celegans multiplex GPI network. IM-EPLR takes more time than other algorithms, and KSN is efficient than other algorithms. IMCS and CIM take almost the same time. CBIM is efficient than IMCS, CIM, and IM-EPLR. Fig. 9(b) shows the execution time of Influence maximization using the IC model on the Celegans multiplex GPI network. CBIM and CIM have taken almost the same time, but they take more time than KSN, IM-EPLR, and IMCS. Fig. 9(c) shows the execution time of influence maximization using the LT model on the Celegans multiplex GPI network. CBIM takes more time, and KSN is efficient than other algorithms. CIM takes more time than IM-EPLR and KSN.



(a) (b) (c)

Fig. 9. (a) Execution time of different heuristics for finding seed nodes under *Celegans multiplex GPI network*. (b) Execution time of different heuristics for spreading of 50 seed nodes using IC model in *Celegans multiplex GPI network*. (c) Execution time of different heuristics for spreading of 50 seed nodes using LT model in *Celegans multiplex GPI network*.

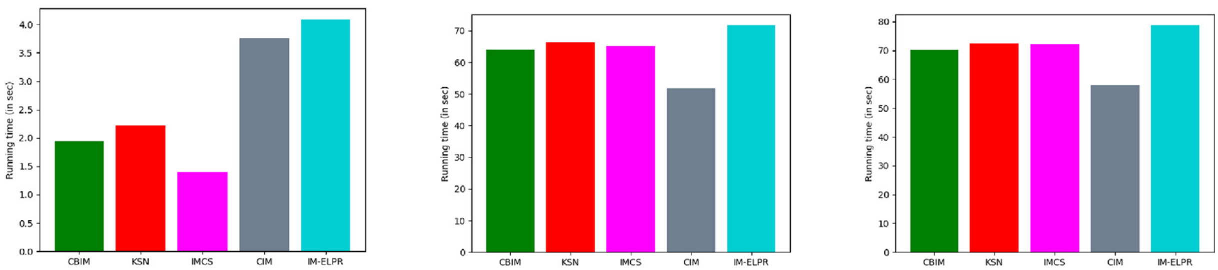


(a) (b) (c)

Fig. 10. (a) Execution time of different heuristics for finding seed nodes under *Homo multiplex GPI network*. (b) Execution time of different heuristics for spreading of 50 seed nodes using IC model in *Homo multiplex GPI network*. (c) Execution time of different heuristics for spreading of 50 seed nodes using LT model in *Homo multiplex GPI network*.

Fig. 10(a) shows the execution time of finding seed nodes by algorithms on the Homo multiplex GPI network. IMCS is efficient than other algorithms. CBIM takes more time than IMCS, but it is efficient than IM-EPLR, CIM, and KSN. Fig. 10 (b) presents the execution time of Influence maximization using the IC model on the Homo multiplex GPI network. IMCS is efficient than other algorithms and CBIM takes more time than all other algorithms. KSN is efficient than CIM, CBIM, and more than IMCS, IM-EPLR. Fig. 10(c) shows the execution time of Influence maximization using the LT model on the Homo multiplex GPI network. IMCS is efficient than other algorithms, and CBIM takes more time than all other algorithms. CIM and IM-EPLR take almost the same time.

Fig. 11(a) shows the execution time of finding seed nodes under-discussed algorithms on Moscow Athletics 2013. IM-EPLR takes more time than other algorithms, and IMCS is efficient than any other algorithms. Fig. 11(b) shows the execution



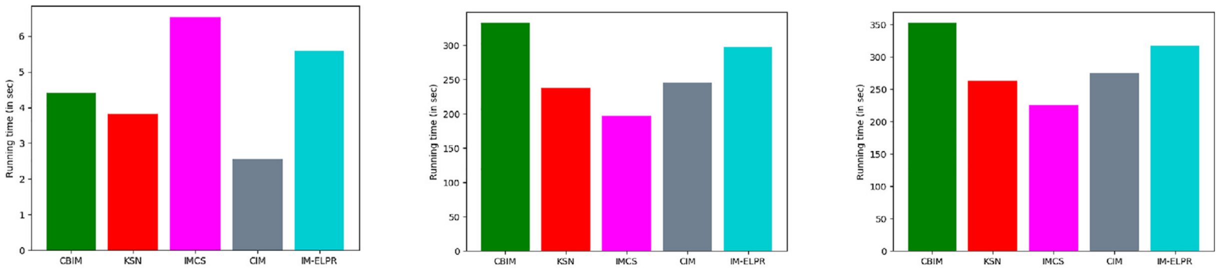
(a) (b) (c)

Fig. 11. (a) Execution time of different heuristics for finding seed nodes under *Moscow Athletics* 2013. (b) Execution time of different heuristics for spreading of 50 seed nodes using IC model in *Moscow Athletics* 2013. (c) Execution time of different heuristics for spreading of 50 seed nodes using LT model in *Moscow Athletics* 2013.

time of Influence maximization using the IC model on Moscow Athletics 2013, CIM takes less time, CBIM and IMCS takes similar time. Fig. 11(c) shows the execution time of Influence maximization using the LT model on the Moscow Athletics 2013 dataset. IM-ELPR takes more time than any other algorithm; KSN, CBIM, and IMCS take almost the same time for influence maximization. CIM is efficient than any other algorithm.

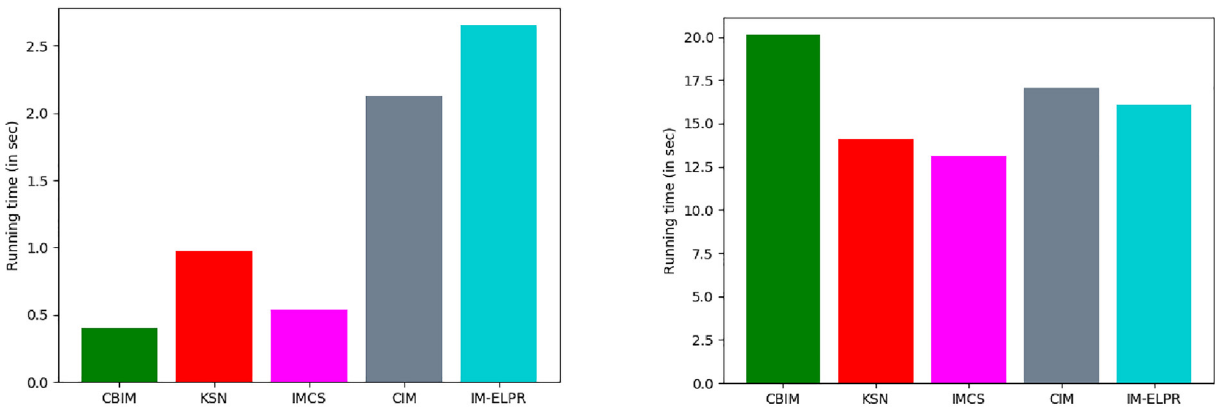
Fig. 12(a) shows the execution time for finding seed nodes of discussed algorithms on the Newyork climate change 2014 dataset. IMCS is efficient than any other algorithm. Fig. 12(b) shows the execution time of influence maximization using the IC model on the Newyork climate change 2014 dataset. KSN is efficient than any other algorithms. CBIM and IMCS take almost a similar time. Fig. 12(c) shows the execution time of influence maximization using the LT model on the Newyork climate change 2014 dataset, IMCS is efficient, and CBIM takes more time, algorithms have taken similar time under the IC model and LT model.

Fig. 13(a) shows the execution time for finding seed nodes of discussed algorithms on Arxiv net science multiplex, CBIM is efficient, and IM-EPLR takes more time than all other algorithms. KSN takes more time than CBIM and IMCS, is efficient than IM-EPLR and CIM. Fig. 13(b) shows the execution time of influence maximization under the IC model on Arxiv net science multiplex, CBIM takes more time, and IMCS takes less time. Fig. 14(a) shows the execution time for finding seed nodes of discussed algorithms on the London Multiplex Transport Network, IM-EPLR takes more time than all other algorithms, CBIM and KSN take almost the same time. Fig. 14(b) shows the execution time of influence maximization using the IC model on London Multiplex Transport Network, CBIM takes more time, KSN and CIM take almost the same time and efficient than all other algorithms.



(a) (b) (c)

Fig. 12. (a) Execution time of different heuristics for finding seed nodes under *Newyork climate change* 2014. (b) Execution time of different heuristics for spreading of 50 seed nodes using IC model in *Newyork climate change* 2014. (c) Execution time of different heuristics for spreading of 50 seed nodes using LT model in *Newyork climate change* 2014.



(a) (b)

Fig. 13. (a) Execution time of different heuristics for finding seed nodes under *Arxiv netscience multiplex*. (b) Execution time of different heuristics for spreading of 50 seed nodes using IC model in *Arxiv netscience multiplex*.

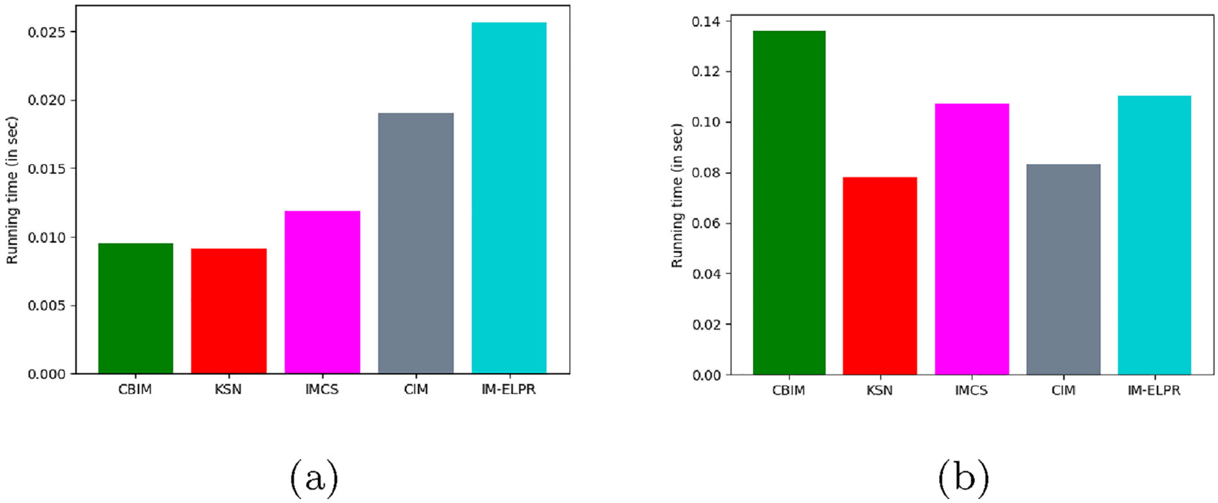


Fig. 14. (a) Execution time of different heuristics for finding seed nodes under *London Multiplex Transport Network*. (b) Execution time of different heuristics for spreading of 50 seed nodes using IC model in *London Multiplex Transport Network*.

7. Conclusion

This paper proposes CBIM, an efficient community-based influence maximization algorithm using IC and LT models in multilayer networks. CBIM selects seed nodes that influence more nodes than the competing algorithms for the same number of seed nodes. The edge density is high in communities generated by CBIM. Due to this, individuals are likely to have frequent interactions and influence each other. Therefore, CBIM is more effective than traditional algorithms. To the best of our knowledge, this is the first attempt to consider both degree and distance for selecting seed nodes from the communities in multilayer networks. The influence spread of CBIM is higher than traditional algorithms. In future, we would like to explore using the concepts of recommender systems to recommend potential seed nodes to users.

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.

References

[1] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, Association for Computing Machinery, New York, NY, USA, 2003, p. 137–146. doi:10.1145/956750.956769. URL:https://doi.org/10.1145/956750.956769.

[2] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter, Multilayer networks, J. Complex Networks 2 (3) (2014) 203–271, https://doi.org/10.1093/comnet/cnu016.

[3] Q. Lu, Z. Bu, Y. Wang, A multiobjective evolutionary approach for influence maximization in multilayer networks, in: Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence, ICCAI '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 431–438, https://doi.org/10.1145/3404555.3404568, URL:https://doi.org/10.1145/3404555.3404568.

[4] Jianjun Cheng, Xing Su, Haijuan Yang, Longjie Li, Jingming Zhang, Shiyan Zhao, Xiaoyun Chen, Neighbor Similarity Based Agglomerative Method for Community Detection in Networks, Complexity, vol. 2019, Article ID 8292485, 16 pages, 2019. doi: 10.1155/2019/8292485.

[5] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions-I, Math. Program. 14 (1978) 265–294, https://doi.org/10.1007/BF01588971.

[6] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 199–208. doi:10.1145/1557019.1557047. URL:https://doi.org/10.1145/1557019.1557047.

[7] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, Association for Computing Machinery, New York, NY, USA, 2001, p. 57–66. doi:10.1145/502512.502525. URL:https://doi.org/10.1145/502512.502525.

[8] M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02, Association for Computing Machinery, New York, NY, USA, 2002, p. 61–70. doi:10.1145/775047.775057. URL:https://doi.org/10.1145/775047.775057.

[9] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 1029–1038, https://doi.org/10.1145/1835804.1835934, URL:https://doi.org/10.1145/1835804.1835934.

[10] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in, IEEE International Conference on Data Mining 2010 (2010) 88–97, https://doi.org/10.1109/ICDM.2010.118.

[11] S.S. Singh, A. Kumar, K. Singh, B. Biswas, Lapso-im: A learning-based influence maximization approach for social networks, Appl. Soft Comput. 82 (2019), https://doi.org/10.1016/j.asoc.2019.105554, URL: https://www.sciencedirect.com/science/article/pii/S1568494619303345 105554.

[12] S. Cheng, H. Shen, J. Huang, G. Zhang, X. Cheng, Staticgreedy: Solving the scalability-accuracy dilemma in influence maximization, in: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 509–518, https://doi.org/10.1145/2505515.2505541.
[13] Ohsaka, N., Akiba, T., Yoshida, Y. and Kawarabayashi, K.- ichi 2014. Fast and Accurate Influence Maximization on Large Networks with Pruned Monte-Carlo Simulations. Proceedings of the AAAI Conference on Artificial Intelligence. 28, 1 (Jun. 2014).
[14] A. Goyal, W. Lu, L.V. Lakshmanan, 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, 2011, p. 47–48. doi:10.1145/1963192.1963217. URL:https://doi.org/10.1145/1963192.1963217.
[15] E. Cohen, D. Delling, T. Pajor, R.F. Werneck, Sketch-based influence maximization and computation: Scaling up with guarantees, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 629–638, https://doi.org/10.1145/2661829.2662077.
[16] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 420–429, https://doi.org/10.1145/1281192.1281239.
[17] A. Goyal, W. Lu, L.V. Lakshmanan, 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, 2011, p. 47–48. doi:10.1145/1963192.1963217. URL:https://doi.org/10.1145/1963192.1963217.
[18] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 1029–1038, https://doi.org/10.1145/1835804.1835934.
[19] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 199–208. doi:10.1145/1557019.1557047. URL:https://doi.org/10.1145/1557019.1557047.
[20] Y.-C. Chen, W.-Y. Zhu, W.-C. Peng, W.-C. Lee, S.-Y. Lee, Cim: Community-based influence maximization in social networks, ACM Trans. Intell. Syst. Technol. 5 (2) (2014), https://doi.org/10.1145/2532549.
[21] K. Sheng, Z. Zhang, Research on the influence maximization based on community detection, in: IEEE Conference on Industrial Electronics and Applications (ICIEA), in: 2018 13th, 2018, pp. 2797–2801, https://doi.org/10.1109/ICIEA.2018.8398185.
[22] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, F. Xia, Community-diversified influence maximization in social networks, Inform. Syst. 92 (2020), https://doi.org/10.1016/j.is.2020.101522, URL: https://www.sciencedirect.com/science/article/pii/S0306437920300326 101522.
[23] X. Li, X. Cheng, S. Su, C. Sun, Community-based seeds selection algorithm for location aware influence maximization, Neurocomput. 275 (C) (2018) 1601–1613, https://doi.org/10.1016/j.neucom.2017.10.007.
[24] J. Chen, J. Liu, Research on a novel influence maximization algorithm based on community structure, J. Phys: Conf. Ser. 1631 (2020), https://doi.org/10.1088/1742-6596/1631/1/012064 012064.
[25] N. Alduaiji, A. Datta, J. Li, Influence propagation model for clique-based community detection in social networks, IEEE Trans. Computat. Soc. Syst. 5 (2) (2018) 563–575, https://doi.org/10.1109/TCSS.2018.2831694.
[26] J. Guo, W. Wu, Influence maximization: Seeding based on community structure, ACM Trans. Knowl. Discov. Data 14 (6) (sep 2020). doi:10.1145/3399661. URL:https://doi.org/10.1145/3399661.
[27] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et du Jura. DOI: 10.5169/seals-266450.
[28] C. Romesburg, Cluster analysis for researchers, Lulu. com (2004).
[29] Kavita Vijayameena, A survey on similarity measures in text mining, Mach. Learn. Appl.: Int. J. 3 (2016) 19–28, https://doi.org/10.5121/mlaij.2016.3103.
[30] L.R. Dice, Measures of the amount of ecologic association between species, Ecology 26 (3) (1945) 297–302, URL: http://www.jstor.org/stable/1932409.
[31] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowl. Inf. Syst. 42 (2015) 181–213, https://doi.org/10.1007/s10115-013-0693-z.
[32] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1953) 39–43, https://doi.org/10.1007/BF02289026.
[33] Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M. BioGRID: a general repository for interaction datasets. Nucl. Acids Res. 2006 Jan 1;34(Database issue):D535-9. doi: 10.1093/nar/gkj109. PMID: 16381927; PMCID: PMC1347471.
[34] Manlio De Domenico, Mason A. Porter, Alex Arenas, MuxViz: a tool for multilayer analysis and visualization of networks, J. Complex Networks 3 (2) (2015) 159–176, https://doi.org/10.1093/comnet/cnu038.
[35] E. Omodei, M. De Domenico, A. Arenas, Characterizing interactions in online social networks during exceptional events, Front. Phys. 3 (2015) 59, https://doi.org/10.3389/fphy.2015.00059, URL:https://www.frontiersin.org/article/10.3389/fphy.2015.00059.
[36] M. De Domenico, A. Lancichinetti, A. Arenas, M. Rosvall, Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems, Phys. Rev. X 5 (2015), https://doi.org/10.1103/PhysRevX.5.011027, URL: https://link.aps.org/doi/10.1103/PhysRevX.5.011027 011027.
[37] M.D. Domenico, A. Solé-Ribalta, S. Gómez, A. Arenas, Navigability of interconnected networks under random failures, Proc. Nat. Acad. Sci. 111 (2014) 8351–8356.
[38] A. Kuhnle, M.A. Alim, X. Li, H. Zhang, M.T. Thai, Multiplex influence maximization in online social networks with heterogeneous diffusion models, IEEE Trans. Comput. Soc. Syst. 5 (2) (2018) 418–429, https://doi.org/10.1109/TCSS.2018.2813262.
[39] S. Kumar, L. Singhla, K. Jindal, K. Grover, B. Panda, Im-elpr: Influence maximization in social networks using label propagation based community structure, Appl. Intell. 51 (2021). doi:10.1007/s10489-021-02266-w.
[40] S. Kianian, M. Rostamnia, An efficient path-based approach for influence maximization in social networks, Expert Syst. Appl. 167 (2021), https://doi.org/10.1016/j.eswa.2020.114168, URL: https://www.sciencedirect.com/science/article/pii/S0957417420309064 114168.
[41] K. Venkatakrishna, M. Katukuri, M. Jagarapu, CIM: clique-based heuristic for finding influential nodes in multilayer networks, Appl. Intell. (2021), https://doi.org/10.1007/s10489-021-02656-0.
[42] X. Wang, X. Tong, H. Fan, C. Wang, J. Li, X. Wang, Multi-community influence maximization in device-to-device social networks, Knowl.-Based Syst. 221 (2021), https://doi.org/10.1016/j.knosys.2021.106944, URL: https://www.sciencedirect.com/science/article/pii/S0950705121002070 106944.
[43] Y. Wang, G. Cong, G. Song, K. Xie, Community-based greedy al22 gorithm for mining top-k influential nodes in mobile social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 1039–1048, https://doi.org/10.1145/1835804.1835935.
[44] Zeng, Qian, Zhon, Ming, Zhu, Yuanyuan, Qian, Tieyun Li, Jianxin. (2021). Business Location Planning based on a Novel Geo-Social Influence Diffusion Model. Inform. Sci. 559. 10.1016/j.ins.2021.01.047.
[45] Seungbae Kim, Jyun-Yu Jiang, Masaki Nakada, Jinyoung Han, Wei Wang, Multimodal Post Attentive Profiling for Influencer Marketing, in: Proceedings of The Web Conference 2020, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2878–2884, https://doi.org/10.1145/3366423.3380052.
[46] T. Cai, J. Li, A. Mian, R.-H. Li, T. Sellis, J.X. Yu, Target-Aware Holistic Influence Maximization in Spatial Social Networks, IEEE Trans. Knowl. Data Eng., 34 (4), pp. 1993–2007, 1 April 2022, doi: 10.1109/TKDE.2020.3003047.