Information Sciences 602 (2022) 86–105

Contents lists available at ScienceDirect

Information Sciences

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

Information diffusion-aware likelihood maximization optimization for community detection

Zheng Zhang [a,b], Jun Wan [a], Mingyang Zhou [a], Kezhong Lu [a,*], Guoliang Chen [a], Hao Liao [a,*]

[a] National Engineering Laboratory for Big data System Computing Technology, Guangdong Province Key Laboratory of Popular High Performance Computers, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[b] School of Computer and Software, Nanyang Institute of Technology, Nanyang 473004, China

ARTICLE INFO

Article history:
Received 10 August 2021
Received in revised form 6 March 2022
Accepted 3 April 2022
Available online 7 April 2022

Keywords:
Community detection
Information diffusion
Likelihood maximization
Network inference

ABSTRACT

As a hot research topic in network science, community detection has attracted much attention of scholars. In recent years, many methods have emerged to discover the underlying community structure in the network. However, most of these methods need to take the network topology information as prior knowledge that is not feasible in practical cases. When information diffusion occurs in the network, one can observe the cascade data in which nodes participate in the propagation process, which reflects the network's community structure to some extent. In this paper, we build a likelihood maximization model by utilizing the diffusion information and propose two different optimization algorithms to obtain community division of the network. Extensive experiments on various datasets show that our proposed methods achieve significant improvements in terms of accuracy, scalability, and efficiency of community detection compared with the existing state-of-the-art methods.

1. Introduction

The graph structure is usually used to represent the social network, rating network, and citation network in the real world. Take Twitter as an example, the nodes in the graph represent the users, and the edges between the nodes represent the social relationships of the users [57]. Community structure is regarded as one of the most valuable characteristics in the real-world network [27,29]. Community is characterized that highly interconnected sets of vertices being well separated from other sets of the vertex in the network [15,23]. For instance, in World Wide Web, a community is made up of a collection of web pages that handle the related tasks. In social networks, users with common interests form the community. In citation networks, articles in similar research fields are divided into the same community [18,2]. The existence of community structure dramatically influences the propagation process of information arising on the network, e.g. [21], the spreading of infectious disease, the retweet of a blog, and the recommendation of products [47]. Therefore, by identifying the communities of the network, community detection methods can offer insight into how the network is organized and reveal the dynamics of information spreading processes across the network [59,22,61].

Currently, the common assumption of community detection algorithms is that the network topology is known. Most community detection methods are based on the assumption that the network topology can be obtained. These methods

* Corresponding authors.
E-mail addresses: kzlu@szu.edu.cn (K. Lu), haoliao@szu.edu.cn (H. Liao).

https://doi.org/10.1016/j.ins.2022.04.009
0020-0255/

[41,55,7,10,30] get the final community division by analyzing and mining the structural features of the network. However, due to the strict requirements for data privacy and security in practical applications, the data about the network topology can not be completely obtained, which limits the performance of this kind of community detection algorithm, and restricts its scalability in the real-world network [46]. Although the underlying network is hidden, we can observe the phenomenon of information diffusion in the network, including whether each node participates in the diffusion process, as well as the time of participation [28]. For example, during the spread of the epidemic, the infection time of a person is known, but we do not know who infected him [32]. For each retweet in Twitter, the information about the source node of the tweet is available, but the social graph is unavailable [58]. These propagation cascades consisting of time series of nodes' activation reflect the structure of the underlying network.

Most traditional community detection algorithms often need to take the network topology as the prior knowledge or obtain the network structure indirectly through the inference graph methods [17,35], and there exist some problems with these methods. First, it is not feasible to obtain the network topology in the real-world network. Second, the inference process of network topology suffers from high computational complexity that seriously affects the scalability in large-scale networks. Moreover, methods based on network structure can not exploit the behavioral features between nodes in the same or different communities. The traceable information diffusion processes are considered to be the presentation of the relationship and behavioral feature between nodes and are influenced by the community structure of the network [14,53,51]. Recently, some community detection methods based on deep learning have emerged. Line Graph Neural Network (LGNN) [6] is a supervised community detection method, which integrates the non-backtracking operator with belief propagation rules. Deep Attentional Embedded Graph Clustering (DAEGC) [56] exploits high-order neighbors to obtain the community division under self-training. Community-Aware Network Embedding (CANE) [52] integrates a community detection model into the adversarial node representation process to capture the global structural features. Although the community detection methods based on deep learning have progressed, the application of deep learning in community detection still faces several limitations and challenges. First, for community detection in the real-world scenario, most data are unlabeled due to the high-cost acquisition. The supervised and semi-supervised deep learning models require labeled data for training, which reduces the applicability of deep learning based community detection methods. Second, in addition to the network topology, the methods based on the deep learning model need to embed non-structural features as the prior knowledge, such as the attributes of nodes and edges, to increase the knowledge of community memberships and achieve effective community detection results. However, due to the privacy protection and security consideration in practical applications, the metadata about the attributes of nodes and edges in the real-world network is usually unavailable. For example, it is difficult to obtain the personal data of user nodes and the actual relationship between users in Twitter network. As a result, the performance of the deep learning methods for community detection are largely reduced. Third, current community detection methods based on the deep learning model are not doing well in terms of the interpretability of results.

Different from existing community detection methods, our research work in this paper focuses on the observed propagation cascades of information over the network, which reflects the characteristics of community structure to some extent. We model the propagation cascades by utilizing likelihood maximization without knowledge about the network topology. Moreover, to verify the effectiveness of this likelihood maximization model in the presentation of community structure, we proposed two different optimization methods for community detection based on the likelihood maximization model, one is implemented based on Expectation–Maximization by using an iterative optimization strategy, the other adopts a greedy optimization strategy based on the improvement of the Louvain algorithm. Through the comparative analysis of two different optimization methods, we can verify the superiority of the proposed model and find the better optimization method to solve the optimal parameters of the likelihood maximization model of propagation cascade. By optimizing the model's parameters to match the propagation cascades, the network's community structure can be detected. Compared with the existing research work, the optimization algorithms proposed from information infusion have the following three advantages. First, the proposed optimization algorithms do not need network topology and metadata of nodes and edges as the prior knowledge. Second, the unsupervised optimization algorithm does not depend on the labeled dataset. Third, the optimization method based on the likelihood model has better interpretability for community detection results. To the best of our knowledge, there is little research on community detection only by information diffusion. The contributions of our work are summarized as follows:

- We build a probability inferring model by utilizing the likelihood maximization method based on the information propagation with the integration of community structure, which can effectively capture the behavioral features of nodes' participation in information propagation at the community level of the network.
- We propose two different optimization algorithms based on the likelihood maximization model: the EM-CD algorithm adopts an iterative likelihood optimization strategy based on Expectation–Maximization, and the L-Louvain algorithm modifies the Louvain algorithm by replacing modularity optimization with likelihood maximization as a criterion and performs community repartition greedily to maximize the likelihood probability of propagation cascades.
- To verify the superiority of the proposed methods, experiments have been done on synthetic and real datasets, and we compare our proposed algorithms with state-of-the-art methods. The experimental results show that our proposed two optimization algorithms have better performance on different evaluation metrics. In addition, no prior knowledge is required for our proposed algorithms which guarantees better scalability in practical applications.

The rest of this paper is organized as follows. In Section 2, we give an overview of the related work. In Section 3, we define the problem. In Section 4, we build a likelihood maximization model of propagation cascade and propose two different optimization algorithms for community detection. In Section 5, we aim to demonstrate the efficiency and robustness of our proposed algorithms through extensive experiments. Finally, we summarize our work and discuss future study in Section 6.

2. Related works

2.1. Information diffusion models

Independent cascade model (IC) [26] is a commonly used probability model, in which an activated node v will attempt to activate its inactive neighbor node w with the probability p_{vw} and the activation behaviors between nodes are independent of each other. In the process of information diffusion, regardless of whether a node v can successfully activate its neighbor nodes at t time, it does not even have a chance to activate other nodes. When no new node is activated, the propagation process ends. In the Susceptible-Infected-Recovered model (SIR) [25], all nodes are in one of these states at a certain time. A node in the infected state can infect its neighbor nodes in the susceptible state with a probability α, and the infected nodes recover with a probability β. The infection process is simultaneous and independent of each other. Linear threshold model (LT) [26] is a value accumulation model, unlike the IC model, LT has an activation threshold $\theta_v \in [0, 1]$. When an activated node attempts to activate a neighbor node without success, its influence on the neighbor node is accumulated rather than abandoned until the node's activation or the propagation process's end.

2.2. Community detection

There have been many methods for community detection, the most commonly used of which is the optimization approach, and modularity is the most dominant metric used in community detection to assess the quality of community division. Many researchers have used different optimization methods to optimize the modularity to obtain the community division. Louvain [40] is a greedy optimization algorithm that iteratively assigns a new community to each node to maximize modularity value. Louvain has been proved to be fast and provides high-quality community partitions. Guimera et al. [16] firstly applied the simulated annealing method to modularity optimization. The method includes local moves where nodes are shifted from one community to another and global moves that merge and split communities, which has low efficiency and is not suitable for large-scale datasets. Spectral optimization [37,50] is another well-known method based on modularity optimization, which proposes the concept of modularity matrix.

Inference graph algorithm is another effective method for community detection. This kind of algorithm infers the topology of the underlying network by observing the activation time of nodes in the propagation cascade and then detects the communities by using inferred edges between nodes. NETINF [20] analyzes the time series of information diffusion and uses the likelihood maximization method for cascading data to infer the network topology. MultiTree [49] optimizes NETINF, and the directed tree corresponding to each cascade is retained. When the cascade data of information diffusion is relatively small, MultiTree shows great performance advantages compared with NETINF. CONNIE algorithm [36] calculates the probability of each edge in the process of information diffusion by using the convex optimization method. NETRATE algorithm [44] builds on an epidemic model and infers edges with weights. The implementation of the algorithm is to turn the likelihood optimization into a convex problem. The DANI algorithm [48] infers the underlying network structure while preserving the properties of the network, and the experiments show that DANI has higher accuracy and lower run time. Du et al. [12] proposed a kernel method called KernelCascade that models the latent diffusion processes and infers the hidden network with no restrictive assumption on the transmission mode of edge.

Traditional community detection methods, such as statistical inference, spectral clustering, graph partition, etc., mainly explore communities from network structures. However, capturing topology information alone may lead to sub-optimal community detection results. Recently, deep learning has been introduced to community detection for uncovering deep network information and modeling complex relationships, thus improving the performance of community detection results. Park et al. [42] proposed an unsupervised method for embedding attributed multiplex networks, which can jointly integrate the embeddings from multiple types of relations between nodes through the consensus regularization framework and the universal discriminator. Jing et al. [24] designed a novel High-order Deep Multiplex Infomax (HDMI) to learn network embedding for multiplex networks via self-supervised learning, which simultaneously captures the extrinsic signal (i.e., the mutual dependence between node embedding and the global summary), the intrinsic signal (i.e., the mutual dependence between node embedding and attributes), and the interaction between these two signals. ProGAN [19] is a novel proximity generative adversarial network for network embedding, and the ProGAN can generate proximities, which can help to discover the complicated underlying relationship between different nodes. LPA_NI [62] is a novel label propagation algorithm for community detection based on node importance and label influence in networks. LPA_NI measures node importance using the semantic information of networks, and the prior importance of nodes are learned by expert knowledge from the Bayesian network.

Recently, researchers have focused on community inference directly from cascades that require less prior information and lower running time. Barbieri et al. [3,4] models the influence of individual nodes in cascading and infers the community

assignment of nodes using the maximum likelihood approach. In [3,4], two different algorithms are proposed, of which C-IC only consider whether nodes participate in the propagation process and C-Rate also consider the sequence of the time participated in cascade. [46] believes that the propagation cascade in the network has the characteristics of Markov and proposes two different community detection algorithms based on likelihood optimization, of which the R-CoDi algorithm is initialized by random community division and obtain the community structure from the cascade. In contrast, the D-CoDi algorithm uses DANI [48] to realize the initial community partition.

3. Problem statement

The interactions between online web services constitute a network represented by a Graph $G = (V, E)$ with $| V | = n$ vertices and $| E | = m$ edges. The network could be a social network like Twitter and Weibo, or a blog network made up of reprints of blogs. The goal of community detection is to assign the vertices to different communities. However, the network topology is unknown. The only observable data is the cascade data in the process of information diffusion that is a set of cascades $\mathscr{C} = \{C_1, \ldots, C_t\}$ that propagated independently on the underlying network G. Each cascade $C_i \in \mathscr{C}$ is an observed time series in which nodes are activated in the ith propagation process, i.e., $C_i = \left\{\left(v_1, t_{v_1}^{C_i}\right), \left(v_2, t_{v_2}^{C_i}\right), \ldots, \left(v_j, t_{v_j}^{C_i}\right)\right\}$, where v_j is a node, $t_{v_j}^{C_i}$ is the node v_j's activation time in propagation process of C_i. Note that when a node v_i is not involved in the jth cascade, we can denote the activation time of a node as: $t_{v_j}^{C_i} = \infty$. Here we assume that G has a community structure with k different communities which can be denoted as: $\mathscr{A} = \{A_1, A_2, \ldots, A_{k-1}, A_k\}$, $U_{n=1}^{k} A_n = V$, $A_n \cap A_m = \varnothing$ *for* $n \neq m$. In our research, we expect to find a community division similar to the ground truth partitions $\mathscr{A}$ that are densely connected internally and loosely connected. Here, we adopt the method of a maximum posterior with the hypothesis that activation behaviors of nodes follow a mathematical model which contains a parameter set denoted by Φ. Hence, our problem becomes equal to find $\hat{\Phi}$ that maximizes the likelihood of the set of cascades $\mathscr{C}$ as shown in Eq. (1). As a result, the optimal value of community division $\hat{\mathscr{A}}$ can be obtained.

$$\hat{\Phi} = argmax_{\Phi} \mathscr{L}(\Phi, \mathscr{A}, \mathscr{C}) \tag{1}$$

4. Methods

4.1. Likelihood maximization model of propagation cascade

Given that only propagation cascades of the underlying network with no information about the network topology are known, we propose our community detection algorithms based on the probability inferring model utilizing the likelihood maximization method where there is a latent variable $\mathscr{A}$ representing the relationship between nodes and communities. Here each propagation trace is independent of each other, then the likelihood function in Eq. (2) can be expanded as:

$$\mathscr{L}(\Phi, \mathscr{A}, \mathscr{C}) = \prod_{C_i \in \mathscr{C}} P(C_i | \mathscr{A}, \Phi) \tag{2}$$

where $P(C_i|\mathscr{A}, \Phi)$ represents the probability of ith propagation occurring in $\mathscr{C}$. The specific propagation cascade C_i can be deemed relative to the contribution of every node that participates in it. In practice, the participation behaviors of nodes in the propagation process are consistent with the properties of the Markov probability chain, so $P(C_i|\mathscr{A}, \Phi)$ is denoted as:

$$P(C_i | \mathscr{A}, \Phi) = \prod_{v \in V} P(a_{v,i}, C_{v,i} | \mathscr{A}, \Phi) \tag{3}$$

where $a_{v,i}$ is the activation behavior $\left(v, t_v^{C_i}\right)$ of node v in cascade C_i at time $t_v^{C_i}$ and $C_{v,i}$ is the sequence of activation actions that occur before the node v participates in C_i. $\mathscr{A}$ represents the underlying community division of graph that divides nodes into K subsets $\{A_1, \ldots, A_k\}$. This community division affects the activation behavior of nodes in the propagation process to some extent. We define a binary variable $Y_{v,k}$ that describes the membership of node v to community A_k:

$$Y_{v,k} = \begin{cases} 1, & (v \in A_k) \\ 0, & (v \notin A_k) \end{cases} \tag{4}$$

the probability that the node v is located in community A_k is defined as $\rho_k = P(Y_{v,k} = 1)$ and the likelihood function is finally written as Eq. (5) which could be optimized by resorting to the optimization algorithm proposed below.

$$\mathscr{L}(\Phi, \mathscr{A}, \mathscr{C}) = \prod_{C_i \in \mathscr{C}} \prod_{v \in V} \sum_{k=1}^{K} P\left(a_{v,i}, C_{v,i} | Y_{v,k}, \Phi\right) \rho_k \tag{5}$$

4.2. Likelihood optimization methods for community detection

Our proposed likelihood optimization methods are based on the probabilistic graphical model, which is a probability model that uses graphs to represent conditional dependencies between random variables. Here, the undirected graph model constructed by cascade data generated by the propagation model can be represented by Markov Random Field (MRF). Therefore, we can decompose the joint probability distribution by factorization with the conditional independence hypothesis. The complete likelihood can be rewritten as:

$$\mathscr{L}(\Phi,\mathscr{A},\mathscr{C}) = P(\mathscr{C}|\mathscr{A},\Phi)P(\mathscr{A}|\Phi)P(\Phi) \tag{6}$$

$$P(\mathscr{C}|\mathscr{A},\Phi) = \prod_{v\in V}\prod_{k=1}^{K} P(v|\Phi)^{Y_{v,k}} \tag{7}$$

$$P(\mathscr{A}|\Phi) = \prod_{v\in V}\prod_{k=1}^{K} \rho_k^{Y_{v,k}} \tag{8}$$

$P(\Phi)$ is the prior probability of the parameter set Φ, we use Dirichlet prior distribution to model the prior probability:

$$P(\Phi) \propto exp\sum_{k=1}^{K}\omega_k \log\rho_k = \prod_{k=1}^{K}\rho_k^{\omega_k} \tag{9}$$

where $\omega_k = -\sqrt{|\Phi_k|}/2$ is a weight factor and $|\Phi_k|$ represents the size of parameter set relative to the kth community.

Iterative likelihood optimization method based on Expectation–Maximization (EM-CD). In this section, we adopt an iterative method to solve the optimal parameter set $\hat{\Phi}$, in each iteration of which the iterative estimation $\Phi^{(k)}$ is constantly approaching the parameter estimation $\hat{\Phi}$. Specifically, we modify the general stochastic framework in [3,4] and propose a likelihood optimization method for community detection (EM-CD) based on Expectation–Maximization (EM) algorithm. The optimizations of the EM-CD method are as follows: First, since the convergence result of the EM algorithm is closely related to the initial community partition, we use the inference graph method to replace the original random generation method to achieve the division of the initial community. Second, in the original framework, the optimal parameters are learned based on two specific cascade models, and the result of community division is obtained. The limitations of the specific cascade model result in poor performance. We model the propagation process in the real network, taking into account independent activation behaviors and temporal dynamic activation behaviors to improve the performance of community detection. Third, we rewrite the execution logic of the E step and M step to speed up the convergence of the algorithm and greatly reduce the running time of the algorithm. The specific implementation steps of Algorithm 1 (EM-CD optimization algorithm) are described as follows:

Step (1): We start from the initial parameter set $\Phi^{(0)}$ and the large number of communities K.

Step (2): In the initialization phase of community division, the EM-CD optimization algorithm generates a directed graph $G^0 = \left(V^0, E^0\right)$ with edge weights using the DANI inference graph method. The k-1 nodes with maximum node degree are selected as the central nodes of the corresponding initial community and are assigned the initial labels from 1 to k-1. Then we start with the k-1 central nodes, traverse the rest of the nodes in the graph G^0, and assign the initial community labels to the nodes. The specific assignment method is as follows:

$$Y^{(0)}_{i\in V^0} = \begin{cases} Y^{(0)}_j & if \left(\underset{m\in N(i)}{\forall}\left((w_{ij}+w_{ji}) > (w_{im}+w_{mi})\right)\right) \\ Y^{(0)}_i & else \end{cases} \tag{10}$$

where $Y^{(0)}_{i\in V^0}$ is the initial community assignment for node i and w_{ij} is the weight of the edge of node i and j. After the above steps, there are still some nodes without community labels, and we uniformly divide these nodes into the kth community.

Step (3): The posterior probability of the latent variable $\mathscr{A}$ is calculated according to the parameters' initial value or the parameters' value of the previous iteration, that is the expectation of the latent variable as current estimates in step E. First, construct the expectation likelihood function $\mathscr{Q}(\Phi,\Phi_k)$ as shown in Eq. (11).

$$\mathscr{Q}\left(\Phi,\Phi^{(t)}\right) = E_A\left(\log P(\Phi,\mathscr{A},\mathscr{C})|C,\Phi^{(t)}\right) \propto \sum_{v\in V}\sum_{k=1}^{K}\mu_{v,k}\left\{\log P\left(v|\Phi_k^{(t)}\right) + \log\rho_k\right\} + \log P(\Phi) \tag{11}$$

where $\mu_{v,k} = P\left(Y_{v,k}=1|\Phi_k^{(t)}\right)$ is the specific implementation of the latent variable $\mathscr{A}$, which represents the probability that node v is located in community k given the parameter $\Phi_k^{(t)}$ and $\log P(\Phi) = \sum_{k=1}^{K}\omega_k\log\rho_k$ is the prior probability of the parameter set Φ. Then we optimize the likelihood function $\mathscr{Q}\left(\Phi,\Phi^{(t)}\right)$ with respect to ρ_k considering the restriction conditions $\sum_{k=1}^{K}\rho_k = 1, 0 \leqslant \rho_k \leq 1$ to estimate $\mu_{v,k}$ as:

$$\mu_{v,k} = \frac{P\left(v|\Phi_k^{(t)}\right)\rho_k}{\sum_{k=1}^{K} P\left(P\left(v|\Phi_k^{(t)}\right)\rho_k\right)} \tag{12}$$

Step (4): In step M, we compute the maximum likelihood estimation for function $\mathcal{Q}\left(\Phi, \Phi^{(t)}\right)$ given the value of $\mu_{v,k}$ obtained in the previous step and yields ρ_k.

$$\rho_k = \frac{max\left\{0, \sum_{v\in V}\mu_{v,k} + \omega_k\right\}}{\sum_{k=1}^{K} max\left\{0, \sum_{v\in V}\mu_{v,k} + \omega_k\right\}} \tag{13}$$

$$\Phi^{(t+1)} = argmax_{\Phi}\left(\mathcal{Q}\left(\Phi, \Phi^{(t)}\right)\right) \tag{14}$$

The general framework of EM-CD algorithm above is parametric to $P\left(v|\Phi_k^{(t)}\right)$ which is related to the probabilistic model of node participation in the propagation cascade. In [3,4], the C-Rate algorithm is built on NETRATE epidemic model [44] with a assumption that the influence of a node only works on nodes within the same community, ignoring the effect of node interaction in different communities. However, in the real cascade observation, a node can be activated by nodes from different communities, and the C-Rate obviously does not follow the actual law of cascade propagation. Therefore, we define the information diffusion from node v to u as $f(t_u|t_v, \bar{\alpha}_{v,k})$ when v is in the k-th community and u is in a different community, and $f(t_u|t_v, \alpha_{v,k})$ represents the information diffusion within the k-th community, where $\bar{\alpha}_{v,k}$ represents the time delay of activation that node v triggers across the k-th community and $\alpha_{v,k}$ within the k-th community. To maintain generality, we set $\alpha_{v,k} = (\lambda + 1)\bar{\alpha}_{v,k}$, and the higher the value of the parameter $\alpha_{v,k}$, the shorter the delay of activation.

Based on the above cascade propagation, we adapt the modified Community-Rate propagation model to fit the framework of the EM-CD algorithm, and $P(v|\Phi_k)$ is rewritten as:

$$\begin{aligned} P(v|\Phi_k) = & \prod_{C_i\in\mathscr{C}, v\notin C_i} \prod_{u\in C_i} S(T|t_u(i), \alpha_{u,k}) \prod_{C_i\in\mathscr{C}, v\in C_i} \prod_{u\in C_i, t_u(i)<t_v(i)} S\left(t_v(i)|t_u(i), \alpha_{u,k}\right) \\ & \left(\sum_{u\in C_i, t_u(i)<t_v(i), v\in A_k} H\left(t_v(i)|t_u(i), \alpha_{u,k}\right) + \sum_{u\in C_i, t_u(i)<t_v(i), v\notin A_k} H\left(t_v(i)|t_u(i), \bar{\alpha}_{u,k}\right)\right) \end{aligned} \tag{15}$$

Algorithm 1: EM-CD optimization algorithm

Input : Propagation cascade $\mathcal{C} = \{C_1, ..., C_t\}$,
Initial number of communities K.
Output: Optimization parameter set $\hat{\Phi}$,
Community division $\mathcal{A} = \{A_1, ..., A_k\}$.

$\Phi^{(0)} = \text{init}(\hat{\Phi})$; // Initialization of parameters;
repeat //Iterative optimization
  for ***all*** *the* $k \in [1, K]$ **do**
    for ***all*** *the* $v, u \in \mathcal{V}$ **do**
      //compute $\sigma_{i,v,u,k}$ according to eq. (19)
      $\sigma_{i,v,u,k} = P(\omega_{i,v,u} = 1|Y_{v,k} = 1, i, v, \Phi^{(t)}) = \frac{\lambda\cdot H(t_v(i)|t_u(i),\alpha_{u,k}) + H(t_v(i)|t_u(i),\bar{\alpha}_{u,k})}{(\lambda+1)\cdot\sum_{u'\in C_i, t_{u'}(i)<t_v(i)} H(t_v(i)|t_{u'}(i),\alpha_{u',k})}$
  for ***all*** *the* $k \in [1, K]$ **do**
    compute ρ_k according to eq. (13)
    if $\rho_k > 0$ **then**
      //compute $\alpha_{u,k}$ according to eq. (20)
      $\alpha_{u,k} = \frac{\sum_{u,v\in C_i, t_u(i)<t_v(i)}(\sigma_{i,v,u,k}\cdot\mu_{v,k} + \lambda\cdot\mu_{v,k})}{(1+\lambda)\cdot(\sum_{v\notin C_i, u\in C_i}\mu_{v,k}\Delta_u + \sum_{u,v\in C_i}\mu_{v,k}\Delta_{u,v})}$
      //compute maximum likelihood estimation
      $\Phi^{(t+1)} = \text{argmax}_{\Phi}(\mathcal{Q}\left(\Phi, \Phi^{(t)}\right))$
    else
      K=K-1;
until *Convergence*;
compute $\mathcal{A} = \{A_1, ..., A_k\}$ according to optimization parameter set $\hat{\Phi}$

where [0,T] represents the time window of propagation, $P(v|\Phi_k)$ considers the likelihood of the time at which node v is activated or the likelihood that activation does not happen within T. $S(T|t_u(i),\alpha_{u,k})$ models the probability that a node is still not activated until time T, $S(t_v(i)|t_u(i),\alpha_{u,k})$ models the probability that a node is not activated by node u until time $t_v(i), H(t_v(i)|t_u(i),\alpha_{u,k})$ and $H(t_v(i)|t_u(i),\bar{\alpha}_{u,k})$ model instantaneous infections from node u within or across the kth community.

In order to simplify the optimization procedure of $\mathscr{Q}\left(\Phi,\Phi^{(t)}\right)$, we introduce the latent binary variable $\omega_{i,v,u}$ denoting the fact that v is activated by u in the propagation cascade C_i, and let W denote the set of all possible $\omega_{i,v,u}$. Then we can rewrite the complete likelihood relative to W as

$$\mathscr{L}(W,\Phi,\mathscr{A},\mathscr{C}) = P(\mathscr{C},W|\mathscr{A},\Phi)P(\mathscr{A}|\Phi)P(\Phi) \tag{16}$$

where

$$\begin{aligned} P(\mathscr{C},W|\mathscr{A},\Phi) = & \prod_{C_i\in\mathscr{C},v\notin C_i}\prod_{k}\prod_{u\in C_i} S(T|t_u(i),\alpha_{u,k})^{Y_{v,k}} \prod_{C_i\in\mathscr{C},v\in C_i}\prod_{k}\prod_{u\in C_i,t_u(i)<t_v(i)} \\ & H\left(t_v(i)|t_u(i),\alpha_{u,k}\right)^{\omega_{i,v,u}Y_{v,k}}\cdot H\left(t_v(i)|t_u(i),\bar{\alpha}_{u,k}\right)^{\omega_{i,v,u}\left(1-Y_{v,k}\right)}\cdot S\left(t_v(i)|t_u(i),\alpha_{u,k}\right)^{Y_{v,k}} \end{aligned} \tag{17}$$

Then, we adopt the exponential distribution to model the conditional likelihood of transmission between a node u and node v, and $f\left(t_v|t_u,\alpha_{u,k}\right)=\alpha_{u,k}\cdot e^{-\alpha_{u,k}\Delta_{u,v}}$ which enables $S\left(t_v|t_u,\alpha_{u,k}\right)=e^{-\alpha_{u,k}\Delta_{u,v}}, H\left(t_v|t_u,\alpha_{u,k}\right)=\alpha_{u,k}$ and $H\left(t_v|t_u,\bar{\alpha}_{u,k}\right)=\bar{\alpha}_{u,k}=\frac{1}{\lambda+1}\cdot\alpha_{u,k},\Delta_{u,v}$ is the activation time delay between nodes u and v. As a consequence, $\mathscr{Q}\left(\Phi,\Phi^{(t)}\right)$ can be rewritten as:

$$\begin{aligned} \mathscr{Q}\left(\Phi,\Phi^{(t)}\right) \propto & \sum_{v\in V}\sum_{k}\mu_{v,k}\log\rho_k - \sum_{C_i\in\mathscr{C},v\notin C_i}\sum_{k}\sum_{u\in C_i}\mu_{v,k}\Delta_u\alpha_{u,k} + \sum_{C_i\in\mathscr{C},v\in C_i}\sum_{k}\sum_{u\in C_i,t_u(i)<t_v(i)} \\ & \left(\tfrac{\lambda}{\lambda+1}\cdot\sigma_{i,v,u,k}\cdot\mu_{v,k}\cdot\log\alpha_{u,k}+\tfrac{1}{\lambda+1}\cdot\mu_{v,k}\cdot\log\alpha_{u,k}\right) - \sum_{C_i\in\mathscr{C},v\in C_i}\sum_{k}\sum_{u\in C_i,t_u(i)<t_v(i)}\mu_{v,k}\cdot\Delta_{u,v}\alpha_{u,k} \end{aligned} \tag{18}$$

where $\sigma_{i,v,u,k}$ is the contribution of the node u in triggering v's activation in the context of the k-th community:

$$\sigma_{i,v,u,k}=P\left(\omega_{i,v,u}=1|Y_{v,k}=1,i,v,\Phi^{(t)}\right)=\frac{\lambda\cdot H\left(t_v(i)|t_u(i),\alpha_{u,k}\right)+H\left(t_v(i)|t_u(i),\bar{\alpha}_{u,k}\right)}{(\lambda+1)\cdot\sum\limits_{u'\in C_i,t_{u'}(i)<t_v(i)}H\left(t_v(i)|t_{u'}(i),\alpha_{u',k}\right)} \tag{19}$$

$$\alpha_{u,k}=\frac{\sum\limits_{u,v\in C_i,t_u(i)<t_v(i)}\left(\sigma_{i,v,u,k}\cdot\mu_{v,k}+\lambda\cdot\mu_{v,k}\right)}{(1+\lambda)\cdot\left(\sum\limits_{v\notin C_i,u\in C_i}\mu_{v,k}\Delta_u+\sum\limits_{u,v\in C_i}\mu_{v,k}\Delta_{u,v}\right)} \tag{20}$$

Greedy optimization method for maximizing the likelihood based on Louvain (L-Louvain). In community detection, modularity is usually used to measure the quality of the partitions obtained by different community detection algorithms, but also as an objective function to optimize for detecting the optimal partitions [5]. The modularity of a partition is a scalar value between −1 and 1 that measures the fraction of the edges inside communities minus the expected one if edges are distributed at random [40], and the modularity is defined as

$$\mathscr{Q}=\frac{1}{2m}\sum_{i,j}\left[A_{ij}-\frac{k_ik_j}{2m}\right]\cdot\delta(c_i,c_j) \tag{21}$$

$$\delta(c_i,c_j) = \begin{cases} 1, & where\ (c_i==c_j) \\ 0, & else \end{cases} \tag{22}$$

where A_{ij} is the weight of the edge between nodes i and j, when the network is an unweighted graph, the weight of all edges is 1, $m=\frac{1}{2}\sum_{ij}A_{ij}$ denotes the sum of the weights of all edges, i.e., the number of edges, $k_i=\sum_j A_{ij}$ represents the sum of the weights of all the edges connected to node i, which is the degree of node i, c_i and c_j respectively denote the community to which nodes i and j belong. Many community detection algorithms are based on direct modularity optimization. Louvain is the most widely used greedy optimization algorithm based on modularity. It divides the community by merging node pairs, which increases the modularity step by step that is fast and suitable for large-scale data sets. In [39], Newman demonstrates an equivalence between modularity optimization and maximum likelihood methods for community detection. Inspired by Newman, we modify the Louvain algorithm by replacing modularity optimization with a likelihood maximization model of

propagation cascades, and propose a novel greedy optimization algorithm for maximizing the likelihood based on Louvain, namely L-Louvain. Precisely, we rewrite the quality function $\mathcal{Q}$ that the L-Louvain algorithm needs to optimize as shown in Eq. (23).

$$\mathcal{Q} = \mathcal{Q}(\mathcal{A}, \Phi) = log\mathcal{L}(\Phi, \mathcal{A}, \mathcal{C}) = \sum_{v \in V}\sum_{k=1}^{K} Y_{v,k} \log P(v|\Phi) + \log \rho_k + \sum_{k=1}^{K} \omega_k log \rho_k \tag{23}$$

In each iteration updating the community partition, we compute the gain in likelihood function based on the previously obtained communities and pick the new community division with the largest likelihood gain. This process is repeated until the likelihood function is maximized. The modified Louvain-based optimization algorithm has the same time complexity $\mathcal{O}(nlogn)$ as the original Louvain. Let us now describe the specific implementation of Algorithm 2 (L-Louvain optimization algorithm).

Step (1): We start from initial parameter set $\Phi_{(0)}$ and initial community partition $\mathcal{A}_{(0)}$. In the initialization phase, every node forms its own community and the probability of each node participating in cascading $P(a_{v,C_i}|v \in V, C_{i \in} \mathcal{C})$ in parameter set $\Phi_{(0)}$ is randomly initialized.

Step (2): According to the set of cascades $\mathcal{C} = \{C_1, \ldots, C_t\}$, we construct a weighted graph $\widehat{G}$ which represent an underlying network and could capture the community structure. The $\widehat{G}$ servers as the basis for community renewal in the following steps.

Algorithm 2: L-Louvain optimization algorithm

Input : Propagation cascade $C = \{C_1, ..., C_t\}$
Output: Optimization parameter set $\hat{\Phi}$,
Community division $\hat{\mathcal{A}} = \{A_1, ..., A_k\}$.

```
//Initialization of parameters and community partition
Φ^(0) = init(Φ̂)
A_(0) = init(Â)
// construct weighted graph Ĝ
Ĝ = InferGraph(C = {C_1, ..., C_t})
repeat //Iterative optimization
    compute Φ^(t) according to eq. (22)
    for v ∈ V and v ∈ A do
        //Find the best community for vertex v
        A' = argmax ΔQ_{v→A'}
        if ΔQ_{v→A'} > 0 then
            //Update the community assignment for vertex v
            A' ← A' ∪ {v};
            A ← A − {v};
    //Update the community division
    A_(t+1) ← A_(t) ;
until |Φ^(t) − Φ^(t−1)| < τ;
return Â = {A_1, ..., A_k}
```

Step (3): We compute the optimal parameters $\Phi^{(t)}$ based on the community division obtained in the (i-1) th iteration to maximize likelihood quality function $\mathcal{Q}(\mathcal{A}^{(t-1)}, \Phi)$.

$$\Phi^{(t)} = argmax_{\Phi} \mathcal{Q}(\mathcal{A}^{(t-1)}, \Phi) \tag{24}$$

Step (4): Under the fixed optimal parameters $\Phi^{(t)}$, we update the community division until the likelihood function reaches maximum.

Step (5): Repeat Step (3) and Step (4) until the difference between the optimal parameters obtained by two iterations is less than the threshold.

Based on the likelihood maximization model of propagation cascade, we propose two different optimization algorithms (EM-CD and L-Louvain) to further verify the model's effectiveness in the representation of community structure. In subsequent experiments, both EM-CD and L-Louvain algorithms show significant performance advantages compared with other baseline algorithms, proving that our proposed model represents well the features of community structure in the network. Moreover, EM-CD and L-Louvain algorithms adopt completely different optimization strategies to solve the optimal parameter set of the model, and the results of community detection reflect the corresponding differences. To help understand EM-CD and L-Louvain algorithms, we comparatively analyze the advantages and disadvantages of these two algorithms from the following three aspects: (1) Convergence of algorithms. The Expectation–Maximization algorithm has been proven to be

convergent [11]. The proposed EM-CD algorithm rewrites the optimization objective function $\mathscr{L}(\Phi, \mathscr{A}, \mathscr{C})$ for community detection and still maintains the convergence of the algorithm. However, the log-likelihood objective function of propagation cascade is not convex, which results in the optimal solution of the EM-CD algorithm is not globally optimal but locally optimal. Louvain algorithm is a fast convergent community discovery algorithm based on modularity [5]. The proposed L-Louvain algorithm replaces modularity optimization with likelihood maximization as a criterion, and performs community repartition greedily to obtain the global optimal solution of community detection. Therefore, the L-Louvain algorithm performs better than the EM-CD algorithm in terms of the convergence effect. (2) Sensitivity of algorithms to initialization parameters. EM-CD algorithm needs to initialize the parameter set Φ, and the values of the initial parameter set $\Phi^{(0)}$ directly affect the convergence efficiency and optimization effect of the EM-CD algorithm. In the specific implementation, we use Dirichlet distribution to initialize the prior probability of the parameter set Φ to ensure excellent convergence efficiency and optimization effects. The L-Louvain algorithm does not depend on the initialization parameter set, which is initialized randomly. In summary, the L-Louvain algorithm is superior to the EM-CD algorithm in terms of sensitivity of the initialization parameters. (3) Stability of algorithms. Each iteration in the execution of the EM-CD algorithm is divided into two steps, one is the Expectation step, which calculates the posterior probability of the latent variable, and the other is the Maximization step, which maximizes the objective likelihood function to obtain the parameter values for the next iteration. This iterative optimization strategy ensures a stable and reliable approach to the optimal convergence value. However, the number of communities detected by the L-Louvain algorithm is usually unstable, and there is an imbalance in community size. Therefore, the stability of the EM-CD algorithm is better than the L-Louvain algorithm.

5. Experiments

This section conducts extensive experiments to demonstrate that our proposed diffusion-aware algorithms based on likelihood optimization for community detection are effective and robust. First, the datasets and evaluation metrics used in the experiment are introduced. Next, we compare our proposed algorithms with the current state-of-the-art methods on synthetic and real-world networks. Finally, systematic explanations are given about the experimental results.

5.1. Datasets

Synthetic networks. We use the LFR model [33] to generate artificial networks, which is a benchmark generating graph algorithm for testing community detection. The LFR algorithm has the following parameters to control the process of generating synthetic network: the number of nodes n, the average node degree k, the maximum node degree max_k, the exponent of the power-law degree distribution β, the exponent of the power-law community size distribution γ, the minimum of community size min_c, the maximum of community size max_c and the mixing parameter μ. Firstly, we calculate the degree of each node in the network according to the parameters β and k. Then, we generate the edges for each node, where the proportion of edges connected to nodes in the same community is $1-\mu$, and the proportion of edges connected to other nodes of the network is μ. Next, we generate the size of each community according to the parameter γ with the condition that the sum of all community sizes is equal to the parameter n. Finally, each node's community ownership relationship is assigned with the required degree sequences of nodes and the required fraction of inter-community edges.

In our experiments, we set the parameters as follows: n = 1000, k = 15, max_k = 50, β = 2, γ = 1, min_c = 20, max_c = 50. Since the mixing parameter μ directly influences the obvious degree of community structure in the network, we generate ten different synthetic networks (S1-S10), changing the parameter μ from 0.05 to 0.5. When the value of μ is small, different communities are well-separated. When the value of μ is large, the communities of the network overlap with each other.

For the synthetic network, we generate the propagation cascades using three different propagation modes, including independent cascade model (IC), linear threshold model (LT), and susceptible-infected-recovered (SIR) model [15]. In order to generate cascading data that conforms to the rules of information diffusion in the real network, we compare the cascade data based on these three propagation models on the synthetic network of mixing parameter μ = 0.1, and the distribution of

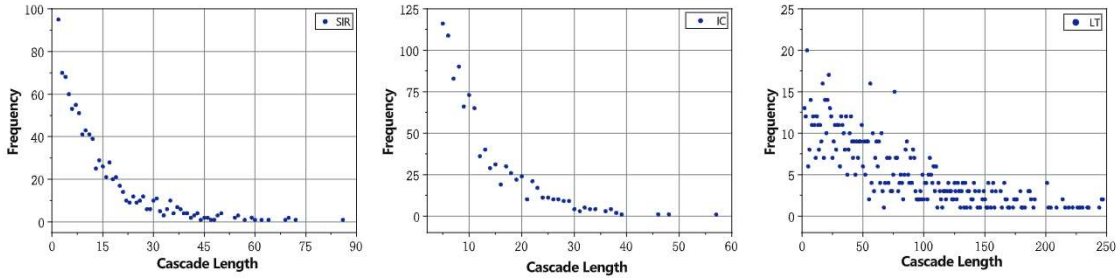

Fig. 1. The distribution of cascades generated by different models (SIR, IC and LT) on the synthetic network.

cascade size of synthetic data is described as shown in Fig. 1. It can be found that the cascades generated by IC and SIR propagation model are similar to the cascade distribution in the real-world networks.

Real networks. To prove the performance advantages of the proposed algorithm on real networks, three different real datasets are applied in our experiments. The Twitter dataset includes Twitter retweet data which was collected from September to November in 2010 [8]. The authors in [8] discover two political communities in the Twitter dataset and manually classify Twitter users based on political alignment into the political left and right communities. In our experiment, these two political communities are used as the ground truth of the Twitter dataset, where the left community contains 9358 users, and the right community contains 9112 users. Therefore, the Twitter data contains the ground truth of community division. The following attribute fields are available in the original data, such as the id of the source user of Twitter, the id of the retweeting user of this Twitter, the timestamp of retweet action, the hashtag list, and the number of hyperlinks. Although there is no cascade data available, we assume that the retweet actions of the same Twitter constitute a cascade. The three attributes of source user id, hashtag list, and the number of hyperlinks together form the primary key and uniquely distinguish each cascade. Then we pre-processed the raw Twitter data based on the above assumptions and obtained a dataset with 18470 nodes and 31143 cascades; the average cascade length is 2.85.

The second dataset comprises six real networks commonly used in the field of community detection and network analysis, which have different network sizes and structural properties with the ground truth of community division. The detailed descriptions of these datasets are shown in Table 1.

Blog dataset contains cascade data for different topics and world news for the 5000 most active sites from four million sites from March 2011 to February 2012 without ground truth community assignments. In the blog datasets, a cascade is created when a site quotes or republishes an article from another site. Each topic contains several memes, and each cascade represents a meme's propagation process. In our experiments, we focus on the cascading data for four different topics, as shown in Table 2.

Amazon dataset is provided by Amazon website to analyze the co-purchased products where nodes represent products, and edges link any two products which are co-purchased frequently. Each product belongs to one or more product categories, and products from the same category form a ground-truth community. In the raw Amazon dataset, the number of nodes is 334863, the number of edges is 925872, and the given number of communities is 75149. We pre-process the Amazon dataset and extract a subset containing only the nodes in communities and their external connected nodes. The statistical information of the pre-processed Amazon dataset is shown in Table 3.

5.2. Evaluation metrics

To evaluate the performance of the community detection algorithms, different evaluation metrics are adopted according to the different characteristics of the datasets. In some datasets, we know the ground truth community assignments and can assess the quality of community detection by comparing the detected community division with the ground truth. In some others, the ground truth of community division is unknown, we adopted the clustering evaluation measures to evaluate the quality of detected communities. Based on the presence or absence of ground truth in the dataset, and we can classify the evaluation metrics used in the experiment as follows:

Metrics for dataset with ground truth. In this case, the following metrics are adopted for evaluating the performance of all the methods.

- **NMI** [15] is a widely used similarity measure that evaluates the difference between the detected and actual community divisions in the network. In our experiments, the set of nodes that participated in cascade is denoted by V_0, and the rest of the nodes in the network are denoted by $V \setminus V_0$, divided into the community labeled "unknown" in the calculation of NMI.
- **F-measure** [23] is the weighted harmonic mean of Precision and Recall that is defined as $F=\frac{(1+\alpha^2)P \cdot R}{\alpha^2(P+R)}$, where α is the harmonic parameter, P is the Precision and R is the Recall.

Table 1
Six different real datasets with ground truth community assignment. n is the number of nodes. m is the number of edges. k is the real number of communities in the datasets.

Dataset	n	m	k
Karate club [60]	34	78	2
Dolphins [34]	62	159	2
Football [45]	115	13	11
Political books [38]	105	441	3
email-Eu-core [31]	986	16064	42
Political blogs [1]	1224	16715	2

Table 2
The description of blog datasets for four topics with cascades.

Dataset	#Sites	#Links	#Cascades	#Average length
Baseball	2187	362519	498448	5.438
Bomb	2364	469243	581158	5.098
Cancer	2068	388926	291686	4.900
NewsOftheWorld	2348	618014	885395	5.332

Table 3
The Statistics of Amazon dataset. #Nodes denotes the number of nodes. #Edges denotes the number of edges. #Communities denotes the number of communities. S_{max} denotes the largest community size. S_{avg} denotes the average community size.

Dataset	#Nodes	#Edges	#Communities	S_{max}	S_{avg}
Amazon	13178	33767	4517	30	9.3

- **ARI** [54] is an adjustment of the Rand Index to solve the problem that the score of Rand index is generally high. ARI is defined as $ARI = \frac{RI - E(RI)}{max(RI) - E(RI)}$. RI is Rand Index that is defined as $RI(X, Y) = \frac{a_{00}+a_{11}}{C_2^n}$, where a_{00} is the number of point pairs that do not belong to the same community in both real and actual community division, and a_{11} is the number of point pairs that belong to the same community in both real and actual community division, and C_2^n is the total number of point pairs.
- **Jaccard** [9] index is a metrics for comparing the similarities and differences between finite sample sets that is defined as $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$. The higher the Jaccard, the higher the sample similarity.

Metrics for datasets without ground truth. Since there are no real communities to compare, we utilize clustering indices to measure the quality of detected communities as follows:

- **Modularity** [49] is a commonly used method to measure the stability of communities in the network that is computed as $\mathcal{Q} = \frac{1}{2m}\sum_{c=1}^{n}\left[2l_c - \frac{(d_c)^2}{2m}\right]$, where n is the number of communities, l_c is the total number of edges of community c, d_c is the sum of the degree of nodes of the community c and m is the total number of edges of the network.
- **Conductance** [44] is used to evaluate the closeness of a single community that measures as Conductance(S) = $\frac{c^s}{2m^s + c^s}$, where c^s is the number of edges connecting the nodes within the community S and the external nodes and m^s is the number of internal edges within community S, the smaller, the better.
- **Internal Density** [43] is the ratio of the actual number of connected edges in a community to the total number of all possible connected edges.
- **Cut Ratio** [13] is another effective measure that comprehensively considers the compactness of nodes in the community and the looseness among the nodes in different communities. Precisely, Cut Ratio is calculated as Cut_Ratio(s) = $\frac{c^s}{n^s(n-n^s)}$, where n is the total number of nodes in the network, and n^s is the number of nodes within community S, the smaller, the better as Conductance.

5.3. Baselines

We compare our proposed algorithms with the following several state-of-the-art community detection methods.

- **MultiTree** [49] is an inference graph algorithm for community detection which first obtains an inferred graph $\hat{\mathcal{G}}$ by using the likelihood maximization method and then uses the Louvain algorithm to cluster the graph $\hat{\mathcal{G}}$.
- **C-IC** [3] is a community detection algorithm based on the likelihood maximization method and utilizes the community-level IC diffusion model, which only considers whether a node participates in cascading or not.
- **C-Rate** [3] is different from C-IC, which takes into account the time when nodes participate in the cascade.
- **R-CoDi and D-CoDi** [46] are advanced community detection methods that utilize the Conditional Random Fields (CRF). The R-CoDi algorithm is initialized with a random community partition, while D-CoDi is initialized with a community partition obtained by an inference graph algorithm DANI [48].

We use the optimal parameters according to their original papers and compare the best results with our proposed algorithms for all mentioned baseline methods.

5.4. Results and discussion

Evaluation on synthetic networks. First, all the algorithms are compared on synthetic LFR networks with the ground truth community assignments. The IC model generates the cascade data. All baseline methods include the inference graph

algorithm (MultiTree and DANI) the related algorithms for community detection by using information diffusion. In the experiment, we vary the mixing parameter μ of the LFR network from 0.05 to 0.5; for each μ, we generated ten random samples of LFR and averaged the results of evaluation metrics for all the comparison algorithms as shown Fig. 2. Based on the experimental results, several important conclusions can be drawn. With the increase of parameter μ, the performance of all algorithms decreases to varying degrees, and the running time has increased. For all values of μ and most evaluation metrics, our proposed EM-CD and L-Louvain optimization algorithms perform better than other baseline algorithms. For the two algorithms proposed, the greedy method based on Louvain is slightly better than the iterative method of EM on the optimization problem of likelihood maximization. Since the inference graph algorithm (MultiTree and DANI) can no longer identify the communities in the network when the parameter μ is greater than 0.3, we only plot the evaluation results when μ is less than or equal to 0.3. The performance of inference graph algorithms (MultiTree and DANI) is much lower than other algorithms using information diffusion. D-CoDi has the largest time complexity of all algorithms except MultiTree and DANI, while MultiTree and DANI are not comparable because they require too long a running time. Therefore, MultiTree and DANI algorithms are excluded in the comparison experiments on the running time metric.

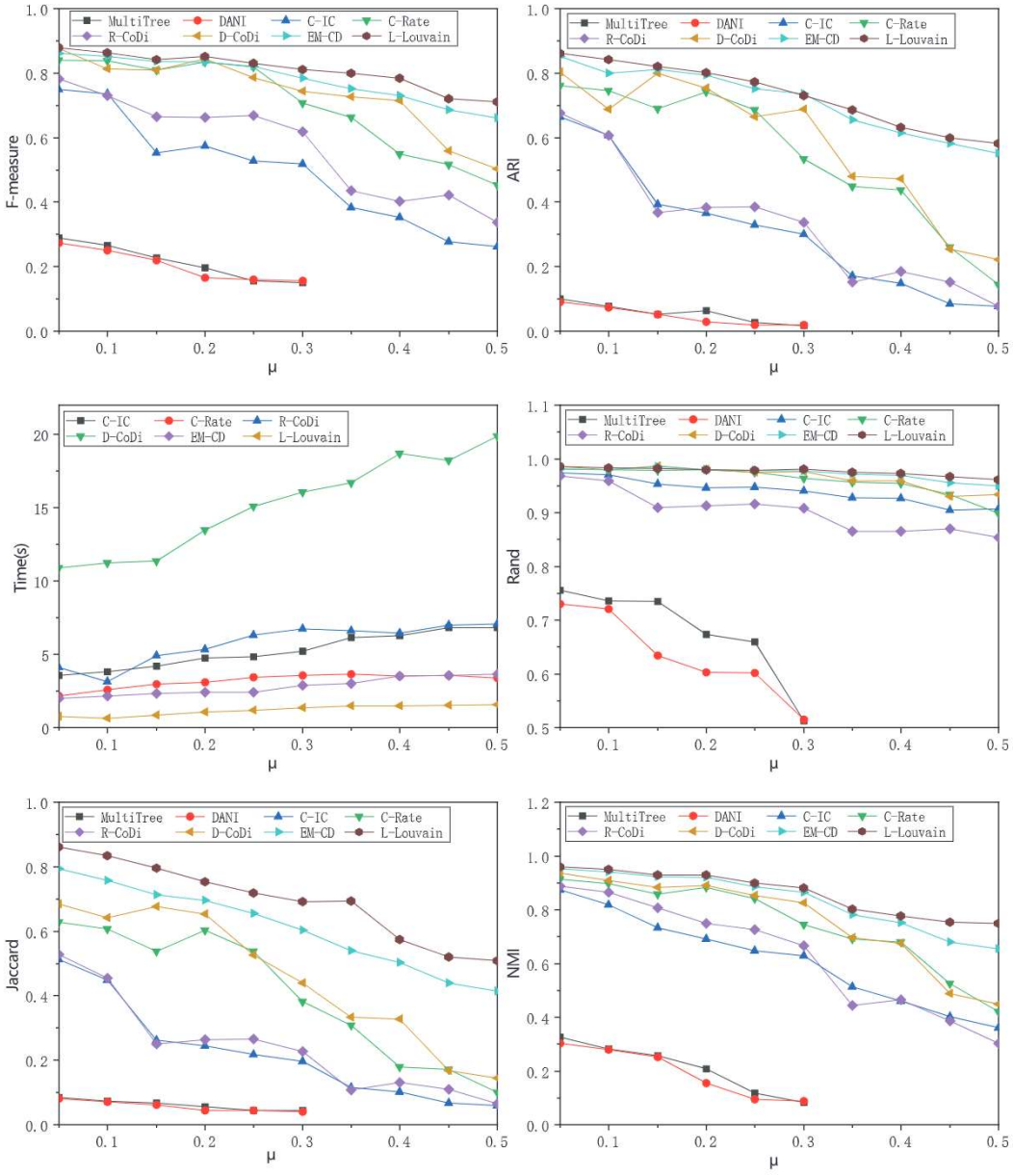

Fig. 2. Comparison of algorithms on synthetic LFR network with IC cascades.

We studied the correlation between the algorithm and the number of cascades as essential performance criteria. Specifically, we fixed the parameter μ of the synthetic network to 0.4 and varied the cascades from 1000 to 10000. Fig. 3 shows the results on different evaluation metrics. All experimental results are averaged over ten cascade samples generated. First, we note that the L-Louvain optimization algorithm performs consistently well for the complete cascade data. Meanwhile, with the increase in the number of cascades, the EM-CD and L-Louvain algorithms significantly improve each evaluation metric and stabilize at a value. Also, the overall performance of the D-CoDi algorithm is quite good, which is very close to that of the EM-CD algorithm. Unexpectedly, the C-IC, C-Rate, and R-CoDi algorithms do not perform very well. A possible reason is that C-IC and C-Rate algorithms are strongly based on the specific models which are not suitable for cascades generated by the IC model in our experiments, and the community initialization assignment of the R-CoDi algorithm seriously affects its performance. In terms of the running time of the algorithms, the two optimization algorithms proposed and C-Rate always maintain a very low running time with the increase of the number of cascades. However, the running time of other algorithms has increased dramatically. Note that the inference graph algorithms can no longer detect the communities in the network when the cascade number is greater than 2000, so the comparison algorithms for this set of experiments do not include MultiTree and DANI.

For the synthetic LFR network, the ground truth community assignments are known. In our experiments, we use the number of detected communities as a criterion for evaluating the algorithms. However, C-IC and C-Rate algorithms require

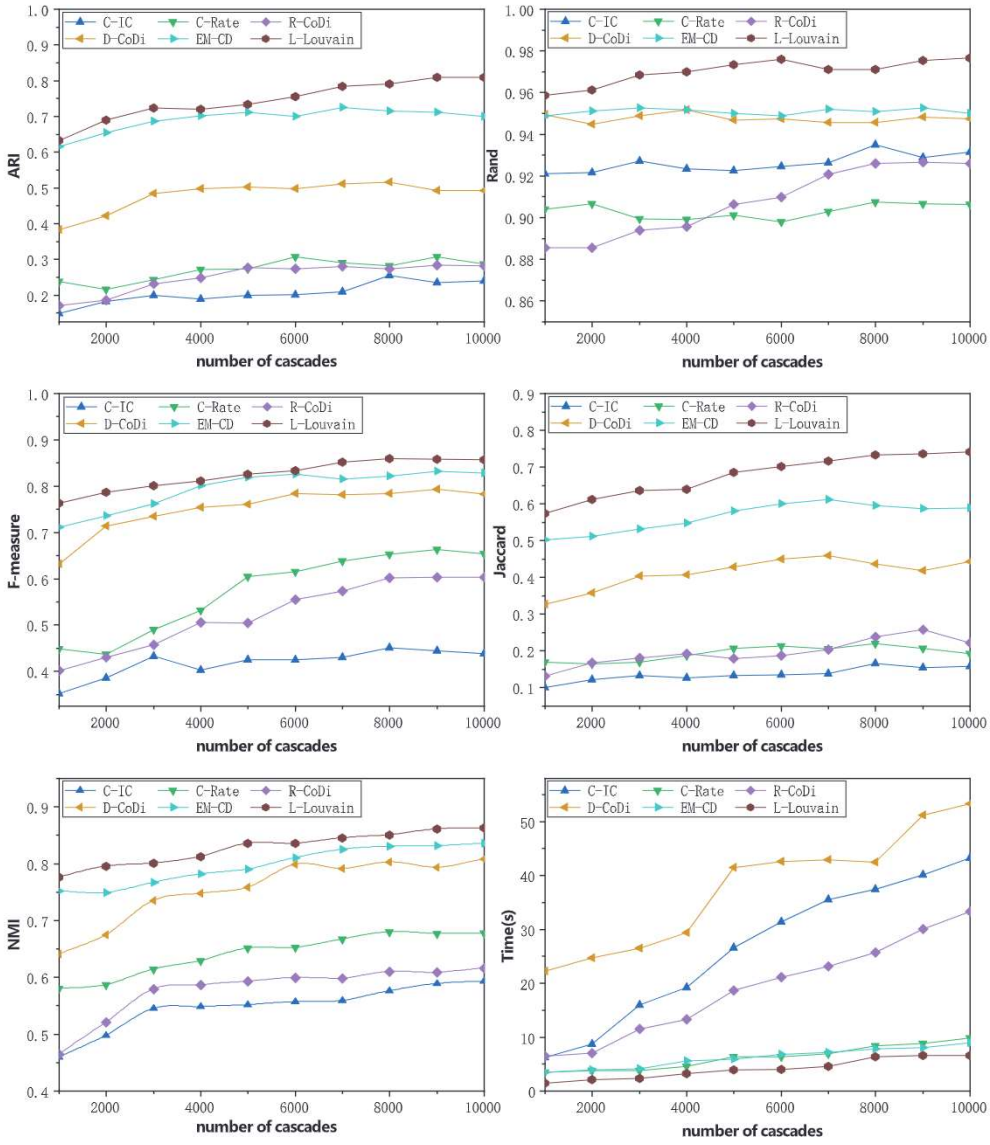

Fig. 3. Comparison of algorithms versus the number of cascades over the synthetic LFR network.

the number of communities as prior knowledge. We use the actual number of communities in the network as our experiments' prior knowledge of these two algorithms. The results obtained by C-IC and C-Rate are the ground truth number of communities which is unfair to other algorithms. So, these two algorithms are excluded from this set of experiments. Fig. 4 demonstrates that D-CoDi, EM-CD, and L-Louvain algorithms can better detect the communities in the network which are similar to the actual number of communities. In contrast, R-CoDi does not detect a sufficient number of communities and the inference graph algorithms (MultiTree and DANI) turn out to be the worst.

Evaluation on real networks with ground truth. All the algorithms are compared in the experiment based on Twitter dataset, including the ground-truth community assignments and real cascades. First, all the metrics of the algorithms are compared on the complete cascade data of Twitter, and the results are presented in Fig. 5. The L-Louvain optimization algorithm is stably the best, and the EM-CD is slightly lower than the L-Louvain, which has been confirmed by previous experiments on the synthetic LFR network. Other algorithms perform unsteadily on different metrics. Specifically, C-IC performs better than DANI and D-CoDi according to F-measure and Jaccard metrics, and DANI performs better than C-IC and D-CoDi according to NMI, Rand and ARI metrics. C-Rate and R-Codi are always the worst, according to all the metrics. Since some evaluation metrics are biased, showing the preference for larger or smaller communities. This causes the volatilities of experimental results. Note that the MultiTree algorithm has poor computational efficiency and does not obtain the results of community division within the acceptable time. So the MultiTree algorithm is excluded from this set of comparison experiments. In addition, we gradually increase the number of cascades and observe the performance trend of all the algorithms. The results are presented in Fig. 6. We notice that the L-Louvain and EM-CD algorithms perform stably well for the complete cascade data, and the performance of these two algorithms are improved with the increase of the number of cascades. Surprisingly, the performance improvement of DANI is the most significant, which verifies the conclusion that the inference graph algorithm needs more cascades to improve its performance. However, for C-Rate and R-CoDi, the cascade number increase almost does not significantly improve F-measure, NMI, and Jaccard metrics. We notice that C-IC increases the most in running time, proving that the C-IC algorithm's time complexity is very sensitive to the number of cascades. There is also a significant increase for DANI. In contrast, the running time of other algorithms is relatively low with the increase in the number of cascades. Nota that the MultiTree is excluded from this set of experiments because of its unacceptable running time.

To explore the impact of the dynamic development of information diffusion on the community detection performance, we divide the retweet cascade process on the Twitter dataset into three stages based on the retweet time of the twitters. Specifically, the first retweet time is denoted as t_s, and the last retweet time is denoted as t_e. Then the duration of the retweet cascade process is denoted as $t_{interval}$. We respectively define the retweet cascade occurring in the time periods of $[t_s,\ t_s + \frac{1}{3}t_{interval}]$, $[t_s,\ t_e - \frac{1}{3}t_{interval}]$, $[t_s,\ t_e]$ as the beginning stage, the growth stage and the mature stage of the complete cascade process. We process the original cascade data according to the above rules, and obtain cascade datasets of three different stages, where the beginning stage contains 19258 cascades with an average cascade length of 2.15. The growth stage contains 27956 cascades with an average cascade length of 2.69, and the maturity stage contains the complete 31143 cascades with an average cascade length of 2.85. Comparison experiments are conducted on the datasets of three stages separately, and the results are presented in Table 4. We notice that the L-Louvain and EM-CD algorithms stably outperform other baseline algorithms, and the performance of these two algorithms is improved with the development of the three different stages. Among all the baseline algorithms, the D-CoDi algorithm performs best on the F-measure metric, the DANI algorithm performs best on the NMI metric, and the C-IC algorithm performs best on the Jaccard metric. In addition, we can find a phenomenon that the performance of not all methods will be improved in the propagation of the retweet cascade. The C-Rate and R-CoDi algorithms show the performance degradation on all three metrics from the growth stage to the mature stage, and the performance of the C-IC algorithm decreases from the growth stage to the mature stage on the NMI metric. The possible reason is that although more retweet data can be used in cascade propagation, more noise data

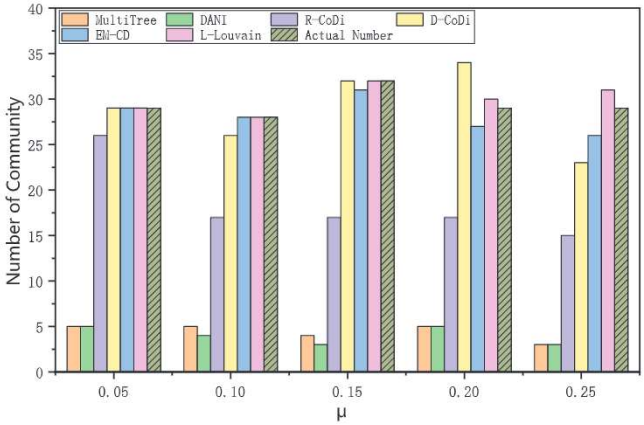

Fig. 4. Comparison of the number of detected communities on different synthetic LFR networks.

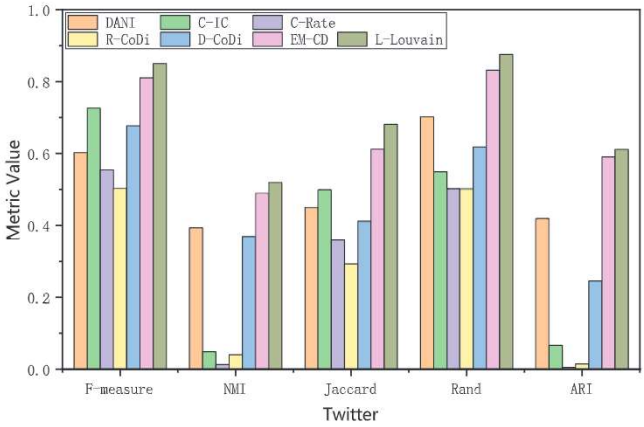

Fig. 5. Comparison of algorithms on Twitter network with complete cascade data.

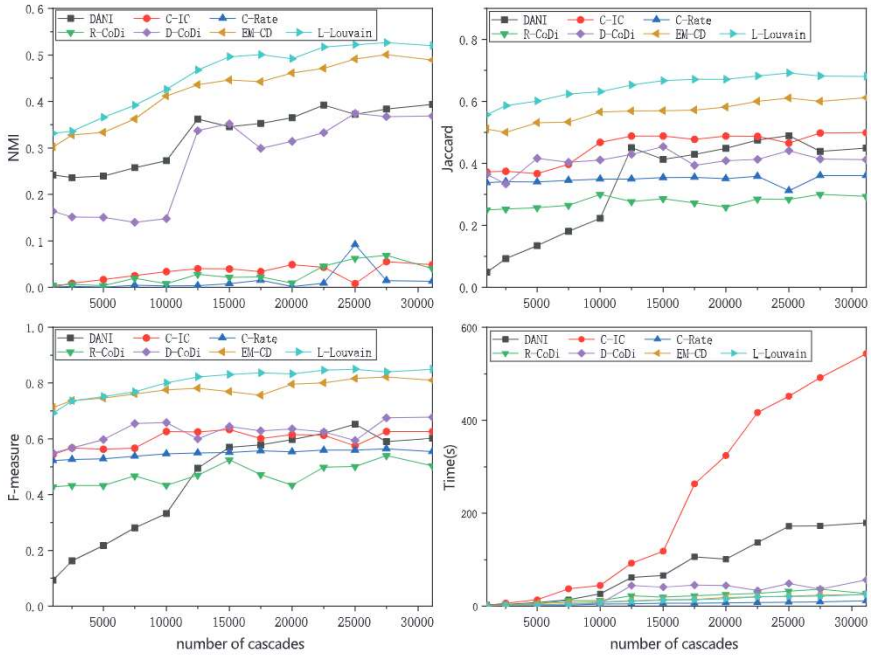

Fig. 6. Comparison of algorithms versus the number of cascades over the Twitter network.

Table 4
Comparison of algorithms in the three different propagation stages of the Twitter dataset. B represents the beginning stage, G represents the growth stage, and M represents the mature stage. '–' indicates that this method does not obtain the results on the corresponding metrics. The optimal value of all the methods is bolded, and the optimal value of all the baseline methods is underlined. '↑' represents an increase in metric value for the current propagation stage compared with the previous stage. '↓' represents a decrease in metric value for the current propagation stage compared with the previous stage.

Method	F-measure			NMI			Jaccard		
	B	G	M	B	G	M	B	G	M
MultiTree	–	–	–	–	–	–	–	–	–
DANI	0.588	0.590↑	0.603↑	0.365	0.384↑	0.393↑	0.431	0.439↑	0.449↑
C-IC	0.605	0.626↑	0.626	0.045	0.055↑	0.049↓	0.488	0.497↑	0.499↑
C-Rate	0.553	0.564↑	0.554↓	0.007	0.015↑	0.013↓	0.351	0.361↑	0.360↓
R-CoDi	0.434	0.539↑	0.503↓	0.010	0.069↑	0.040↓	0.259	0.301↑	0.293↓
D-CoDi	0.635	0.675↑	0.677↑	0.314	0.367↑	0.369↑	0.409	0.412↑	0.414↑
EM-CD	0.796	0.811↑	0.829↑	0.461	0.489↑	0.501↑	0.582	0.600↑	0.612↑
L-Louvain	**0.833**	**0.841**↑	**0.850**↑	**0.491**	**0.520**↑	**0.526**↑	**0.671**	**0.680**↑	**0.680**

is also introduced for these algorithms, which causes performance degradation. Note that the MultiTree algorithm has poor computational efficiency and does not obtain the results of community division within the acceptable time.

We compared our proposed methods with all the baselines on six different real datasets listed in Table 1, the IC model generates the cascades. The results are presented in Table 5 and are mostly consistent with that obtained on the Twitter network. We can observe that L-Louvain is the best in most of the metrics of all six networks, and the performance of EM-CD is very close to that of L-Louvain, and it is better than L-Louvain in some evaluation metrics. The results of inference graph algorithms (MultiTree and DANI) on small networks (Dolphins and Karate club) are the same, and we do not obtain the results of the MultiTree algorithm on large networks (Football, Political blogs, and email-Eu-core) due to long-running time. DANI performs better than some complicated methods (C-IC, C-Rate, R-CoDi, and D-CoDi) on Political books and Football networks. Also, as discussed above, D-CoDi is an excellent algorithm, and its performance is close to that of EM-CD on Dolphins, Football, and Political blogs networks.

To evaluate the applicability and scalability of the proposed optimization algorithms, we conducted experiments on the large-scale Amazon dataset. The results are presented in Table 6. We can observe that the proposed EM-CD and L-Louvain algorithms are significantly better than other baseline algorithms in all metrics. Compared with the optimal values in the baseline algorithms, the optimal values of EM-CD and L-Louvain algorithms improve by 54.73%, 79.22%, 3.31%, 60.24%, 134.31% respectively on F-measure, Jaccard, Rand, NMI, ARI evaluation metrics. In addition, the L-Louvain algorithm is superior to the EM-CD algorithm on most metrics, which is consistent with the previous analysis of the advantages and disadvantages of the two optimization algorithms. D-CoDi is an excellent algorithm, which performs better than other baseline algorithms. We do not obtain the results of the MultiTree algorithm on a large-scale Amazon dataset due to long-running time. In summary, the experimental results on the Amazon dataset demonstrate the applicability of the proposed optimization algorithms in large-scale real-world application scenarios.

We study the correlation between the algorithm's running time and the number of cascades as an essential criterion to evaluate the algorithm's scalability. In specific experiments, the number of cascades generated by the IC model on the Amazon dataset varies from 1000 to 10000, and the running time variation of all the algorithms with the number of cascades is shown in Fig. 7. We notice that the L-Louvain algorithm maintains the lowest running time with increasing the number of

Table 5
Comparison of algorithms on six real networks with ground truth and different structural properties (F-measure/ Jaccard/ Rand/ NMI/ ARI).

Method	Dolphins	Karate club	Political books
MultiTree	0.611/0.392/0.654/0.494/0.346	0.658/0.463/0.731/0.567/0.454	0.872/0.615/0.787/0.519/0.574
DANI	0.611/0.392/0.654/0.494/0.346	0.658/0.463/0.731/0.567/0.454	0.882/0.637/0.802/0.544/0.604
C-IC	0.586/0.373/0.492/0.027/0.028	0.695/0.413/0.549/0.164/0.103	0.790/0.542/0.755/0.490/0.495
C-Rate	0.759/0.527/0.556/0.011/0.030	0.650/0.470/0.485/0.046/0.007	0.715/0.516/0.766/0.444/0.498
R-CoDi	0.694/0.532/0.707/0.477/0.427	0.491/0.273/0.565/0.261/0.122	0.772/0.578/0.803/0.498/0.578
D-CoDi	0.866/0.824/0.894/0.674/0.787	0.729/0.531/0.743/0.465/0.483	0.807/0.620/0.816/0.508/0.614
EM-CD	0.891/0.855/**0.913**/0.740/0.801	0.811/**0.796**/0.861/0.728/0.706	0.866/0.673/**0.851**/0.611/0.720
L-Louvain	**0.915**/**0.881**/0.911/**0.769**/**0.826**	**0.836**/0.790/**0.895**/**0.755**/**0.726**	**0.891**/**0.725**/0.844/**0.669**/**0.771**
Method	**Football**	**Political blogs**	**email-Eu-core**
MultiTree	-/-/-/-/-	-/-/-/-/-	-/-/-/-/-
DANI	0.885/0.701/0.958/0.879/0.775	0.554/0.334/0.508/0.014/0.016	0.222/0.056/0.441/0.117/0.022
C-IC	0.776/0.505/0.943/0.806/0.640	0.739/0.484/0.587/0.258/0.173	0.342/0.120/0.921/0.525/0.173
C-Rate	0.868/0.663/0.967/0.874/0.779	0.600/0.388/0.511/0.018/0.022	0.352/0.094/0.890/0.543/0.118
R-CoDi	0.841/0.570/0.945/0.867/0.697	0.847/0.728/0.855/0.583/0.710	0.347/0.122/0.914/0.474/0.183
D-CoDi	0.886/0.633/0.958/0.873/0.753	0.870/0.747/0.865/0.617/0.729	0.487/0.201/0.905/0.544/0.290
EM-CD	0.911/0.731/0.966/0.905/**0.847**	0.875/**0.791**/0.871/0.713/**0.756**	0.711/**0.621**/**0.924**/0.771/0.639
L-Louvain	**0.929**/**0.758**/**0.976**/**0.924**/0.839	**0.892**/0.780/**0.905**/**0.739**/0.752	**0.736**/0.600/0.921/**0.795**/**0.681**

Table 6
Experimental results on Amazon dataset. The optimal value of all the methods is bolded, and the optimal value of all the baseline methods is underlined. '–' indicates that this method does not obtain the results on the corresponding metrics.

Method	F-measure	Jaccard	Rand	NMI	ARI
MultiTree	–	–	–	–	–
DANI	0.386	0.291	0.561	0.195	0.018
C-IC	0.461	0.335	0.876	0.473	0.265
C-Rate	0.487	0.405	0.853	0.491	0.214
R-CoDi	0.452	0.429	0.866	0.422	0.191
D-CoDi	0.550	0.462	0.872	0.498	0.306
EM-CD	0.836	0.811	**0.905**	0.781	0.691
L-Louvain	**0.851**	**0.828**	0.890	**0.798**	**0.717**

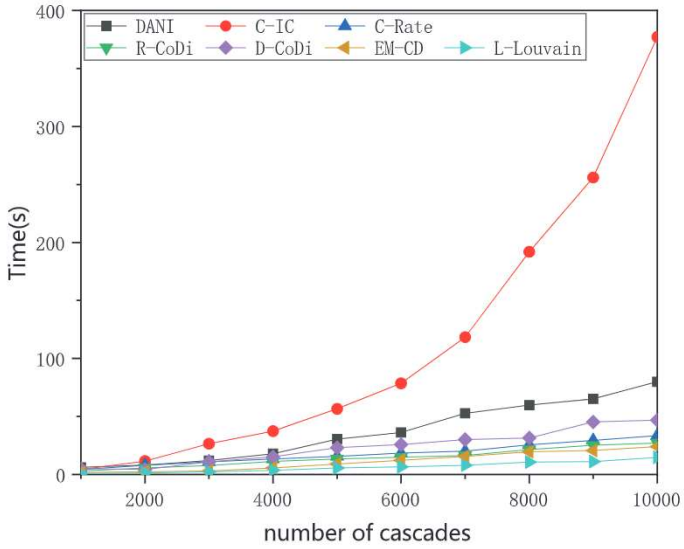

Fig. 7. The running time of algorithms versus the number of cascades over Amazon dataset.

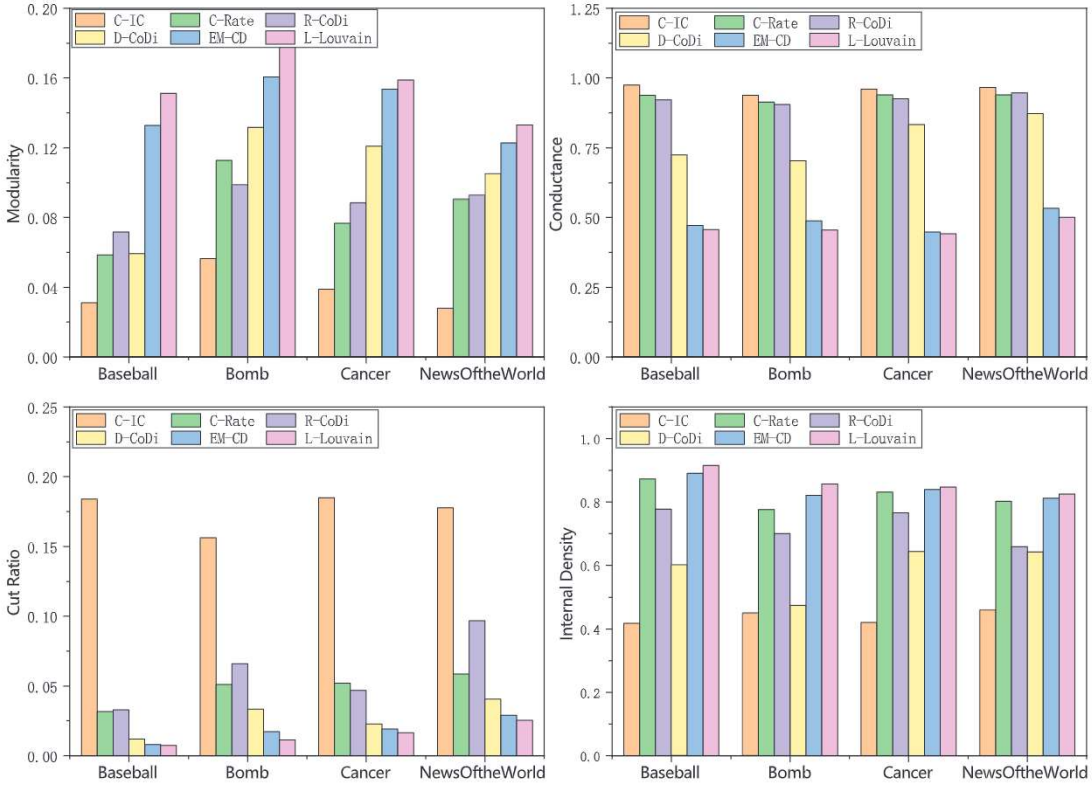

Fig. 8. Comparison of algorithms on real blog networks without ground truth.

cascades. The running time of EM-CD, R-CoDi, and C-Rate algorithms are relatively close, showing good stability. In contrast, C-IC increases the most in running time, proving that the C-IC algorithm's time complexity is very sensitive to the number of cascades. Note that MultiTree is excluded from this experiment because of its unacceptable running time. Based on the above analysis, we can conclude that the high runtime efficiency of the proposed L-Louvain and EM-CD algorithms guarantees their scalability for large-scale practical applications.

Evaluation on real networks without ground truth. We compare all the algorithms on real networks without the ground-truth community assignments described in Table 2. Since the blog networks do not include any information about the ground truth community assignment, we use another kind of metrics to evaluate the performance of the algorithms, including modularity, conductance, internal density, and cut ratio. High modularity and internal density with low conductance and cut ratio show excellent performance in evaluating algorithms in the field of community detection. The experimental results of different algorithms based on the blog datasets are shown in Fig. 8. The EM-CD and L-Louvain algorithms can achieve high modularity and internal density with low conductance and internal density. This proves that the EM-CD and L-Louvain algorithms have the best performance on the blog networks. Also, we notice that D-CoDi performs better than C-IC, C-Rate, and R-CoDi in modularity, conductance, and cut ratio metrics. Meanwhile, the performance of C-Rate in internal density metric is better than C-IC, R-CoDi, and D-CoDi and is close to EM-CD and L-Louvain algorithms. Therefore, the performance of all the algorithms can be sorted from best to worst as L-Louvain, EM-CD, D-CoDi, R-CoDi, C-Rate, and C-IC on real blog networks.

6. Conclusions and future work

This paper first built a likelihood maximization model of propagation cascade without any prior knowledge about the network topology. Then we proposed two optimization algorithms based on the likelihood maximization model for community detection, and finally compared them with the state-of-the-art methods. Extensive experiments on artificial and real networks show that the proposed algorithms (EM-CD and L-Louvain) have better performance for community detection on different evaluation metrics with higher efficiency. In the future, the procedure and principle of information diffusion under the influence of overlapping communities of the network need to be investigated. In addition, the likelihood maximization method to discover the overlapping communities and the model of the internal relationship between information diffusion and the structure of overlapping communities would also be worth studying.

CRediT authorship contribution statement

Zheng Zhang: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Jun Wan:** Visualization, Writing – review & editing. **Mingyang Zhou:** Data curation, Writing – review & editing. **Kezhong Lu:** Formal analysis, Funding acquisition, Writing – review & editing. **Guoliang Chen:** Writing – review & editing. **Hao Liao:** Investigation, Resources, Supervision, Project administration, Funding acquisition, Writing – review & editing.

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

The authors acknowledge financial support from the National Natural Science Foundation of China (Grant Nos. 61803266,61703281,91846301,71790615), Guangdong Province Natural Science Foundation (Grant Nos. 2019A1515011173,2019A1515011064,2017B030314073), Shenzhen Fundamental Research-general project (JCYJ20190808162601658), Natural Science Foundation of Henan (No.2012BAJ05B07).

References

[1] Lada.A. Adamic, Natalie Glance, The political blogosphere and the 2004 us election: divided they blog, in: Proceedings of the 3rd international workshop on Link discovery, 2005, pp. 36–43.
[2] Nicola Barbieri, Francesco Bonchi, Giuseppe Manco, Cascade-based community detection, in: Proceedings of the sixth ACM international conference on Web search and data mining, 2013, pp. 33–42.
[3] Nicola Barbieri, Francesco Bonchi, Giuseppe Manco, Influence-based network-oblivious community detection, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 955–960.
[4] Nicola Barbieri, Francesco Bonchi, Giuseppe Manco, Efficient methods for influence-based network-oblivious community detection, ACM Transactions on Intelligent Systems and Technology (TIST) 8 (2) (2016) 1–31.
[5] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech.: Theory Exp. (2008) (10):P10008, 2008..
[6] Zhengdao Chen, Joan Bruna, Lisha Li, Supervised community detection with line graph neural networks, in: 7th International Conference on Learning Representations, 2019.
[7] Michele Coscia, Fosca Giannotti, Dino Pedreschi, A classification for community discovery methods in complex networks, Stat. Anal. Data Min. 4 (5) (2011) 512–546.
[8] Michael D. Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini, Political polarization on twitter, in: Fifth international AAAI conference on weblogs and social media, 2011..
[9] Sandro Cavallari, Vincent W. Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, Erik Cambria, Learning community embedding with community detection and node embedding on graphs, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 377–386.
[10] Hadi Daneshmand, Manuel Gomez-Rodriguez, Le Song, Bernhard Schoelkopf, Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm, in: International conference on machine learning PMLR, 2014, pp. 793–801.

[11] Arthur P. Dempster, Nan M. Laird, Donald B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. Roy. Stat. Soc.: Ser. B (Methodol.) 39 (1) (1977) 1–22.
[12] Nan Du, Le Song, Ming Yuan, Alex Smola, Learning networks of heterogeneous influence, Adv. Neural Inf. Process. Syst. 25 (2012) 2780–2788.
[13] Santo Fortunato, Darko Hric, Community detection in networks: A user guide, Phys. Rep. 659 (2016) 1–44.
[14] Mario A.T. Figueiredo and Anil K. Jain, Unsupervised learning of finite mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 24(3) (2002) 381–396..
[15] Santo Fortunato, Community detection in graphs, Phys. Rep. 486 (3–5) (2010) 75–174.
[16] Roger Guimera, Luis A Nunes v, Functional cartography of complex metabolic networks, Nature 433 (7028) (2005) 895–900.
[17] Hao Liao, Xiaoming Huang, Ziqiang Wu, et al, Network-splitter: a network feature extraction algorithm based on overlapping community and its application in link prediction, Sci Sin Inform 51 (2021) 1116–1130.
[18] Adrien Guille, Hakim Hacid, Cecile Favre, Djamel A. Zighed, Information diffusion in online social networks: A survey, ACM Sigmod. Record 42 (2) (2013) 17–28.
[19] Hongchang Gao, Jian Pei, Heng Huang, Progan: Network embedding via proximity generative adversarial network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1308–1316.
[20] Manuel Gomez-Rodriguez, Jure Leskovec, Andreas Krause, Inferring networks of diffusion and influence, ACM Transactions on Knowledge Discovery from Data (TKDD) 5 (4) (2012) 1–37.
[21] Manuel Gomez-Rodriguez, Jure Leskovec, Bernhard Schölkopf, Modeling information propagation with survival theory, in: International conference on machine learning, PMLR, 2013, pp. 666–674.
[22] Manuel Gomez Rodriguez, Jure Leskovec, Bernhard Schölkopf, Structure and dynamics of information pathways in online media, in: Proceedings of the sixth ACM international conference on Web search and data mining, 2013, pp. 23–32.
[23] Di Jin, Ziyang Liu, Weihao Li, Dongxiao He, Weixiong Zhang, Graph convolutional networks meet markov random fields: Semi-supervised community detection in attribute networks, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 152–159.
[24] Baoyu Jing, Chanyoung Park, Hanghang Tong, Hdmi: High-order deep multiplex infomax, in: Proceedings of the Web Conference 2021, 2021, pp. 2414–2424.
[25] Matt J. Keeling, Ken T.D. Eames, Networks and epidemic models, J. R. Soc. Interface 2 (4) (2005) 295–307.
[26] David Kempe, Jon Kleinberg, Éva 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, 2003, pp. 137–146.
[27] Masahiro Kimura, Kazumasa Yamakawa, Kazumi Saito, Hiroshi Motoda, Community analysis of influential nodes for information diffusion on a social network, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1358–1363.
[28] Yanhua Li, Wei Chen, Yajun Wang, Zhi-Li Zhang, Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships, in: Proceedings of the sixth ACM international conference on Web search and data mining, 2013, pp. 657–666.
[29] Zhifang Li, Hu. Yanqing, Xu. Beishan, Zengru Di, Ying Fan, Detecting the optimal number of communities in complex networks, Physica A 391 (4) (2012) 1770–1776.
[30] Shudong Li, Laiyuan Jiang, Wu. Xiaobo, Weihong Han, Dawei Zhao, Zhen Wang, A weighted network community detection algorithm based on deep learning, Appl. Math. Comput. 401 (2021) 126012.
[31] Jure Leskovec, Jon Kleinberg, Christos Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (1) (2007), 2-es.
[32] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, Michael W. Mahoney, Statistical properties of community structure in large social and information networks, in: Proceedings of the 17th international conference on World Wide Web, 2008, pp. 695–704.
[33] Jure Leskovec, Kevin J. Lang, Michael Mahoney, Empirical comparison of algorithms for network community detection, in: Proceedings of the 19th international conference on World wide web, 2010, pp. 631–640.
[34] David Lusseau, Karsten Schneider, Oliver J. Boisseau, Patti Haase, Elisabeth Slooten, Steve M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. 54 (4) (2003) 396–405.
[35] Yasir Mehmood, Nicola Barbieri, Francesco Bonchi, Antti Ukkonen, Csi: Community-level social influence analysis, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 48–63.
[36] Seth A. Myers and Jure Leskovec. On the convexity of latent social network inference. arXiv preprint arXiv:1010.5504, 2010..
[37] Mark E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006) 036104.
[38] Mark E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.
[39] Mark E.J. Newman, Equivalence between modularity optimization and maximum likelihood methods for community detection, Phys. Rev. E 94 (5) (2016) 052315.
[40] Mark E.J. Newman, Michelle Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[41] Tiago P. Peixoto, Network reconstruction and community detection from dynamics, Phys. Rev. Lett. 123(12) (2019) 128301..
[42] Chanyoung Park, Donghyun Kim, Jiawei Han, Hwanjo Yu, Unsupervised attributed multiplex network embedding, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 5371–5378.
[43] Liudmila Prokhorenkova and Alexey Tikhonov, Community detection through likelihood optimization: in search of a sound model, in: The World Wide Web Conference, 2019, pp. 1498–1508..
[44] Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf, Uncovering the temporal dynamics of diffusion networks. arXiv preprint arXiv:1105.0697, 2011..
[45] Yiye Ruan, David Fuhry, Srinivasan Parthasarathy, Efficient community detection in large networks using content and links, in: Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 1089–1098.
[46] Maryam Ramezani, Ali Khodadadi, and Hamid R. Rabiee, Community detection using diffusion information, ACM Transactions on Knowledge Discovery from Data (TKDD) 12(2) (2018) 1–22..
[47] Manuel Gomez Rodriguez, Jure Leskovec, David Balduzzi, Bernhard Schölkopf, Uncovering the structure and temporal dynamics of information propagation, Network Sci. 2 (1) (2014) 26–65.
[48] Maryam Ramezani, Hamid R. Rabiee, Maryam Tahani, and Arezoo Rajabi, Dani: A fast diffusion aware network inference algorithm. arXiv preprint arXiv:1706.00941, 2017..
[49] Manuel Gomez Rodriguez and Bernhard Schölkopf. Submodular inference of diffusion networks from multiple trees. arXiv preprint arXiv:1205.1671, 2012..
[50] Yudong Sun, Bogdan Danila, K. Josić, and Kevin E. Bassler. Improved community structure detection using a modified fine-tuning strategy, Europhys. Lett. 86(2) (2009) 28004..
[51] Yuyao Wang, Bu. Zhan, Huan Yang, Hui-Jia Li, Jie Cao, An effective and scalable overlapping community detection approach: Integrating social identity model and game theory, Appl. Math. Comput. 390 (2021) 125601.
[52] Jia Wang, Jiannong Cao, Wei Li, Senzhang Wang, Cane: community-aware network embedding via adversarial training, Knowl. Inf. Syst. 63 (2) (2021) 411–438.
[53] Wang Yu, Gao Cong, Guojie Song, Kunqing Xie, Community-based greedy algorithm 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, 2010, pp. 1039–1048.
[54] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang, Community preserving network embedding, in: Thirty-first AAAI conference on artificial intelligence, 2017..

[55] Liaoruo Wang, Stefano Ermon, John E. Hopcroft, Feature-enhanced probabilistic models for diffusion network inference, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2012, pp. 499–514.
[56] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, C. Zhang, Attributed graph clustering: A deep attentional embedding approach, in: International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence, 2019.
[57] Lilian Weng, Jacob Ratkiewicz, Nicola Perra, Bruno Gonçalves, Carlos Castillo, Francesco Bonchi, Rossano Schifanella, Filippo Menczer, and Alessandro Flammini, The role of information diffusion in the evolution of social networks, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013, pp. 356–364..
[58] Yongqing Wang, Huawei Shen, Shenghua Liu, and Xueqi Cheng, Learning user-specific latent influence and susceptibility from information cascades, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015..
[59] Meng Wang, Chaokun Wang, Jeffrey Xu Yu, Jun Zhang, Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework, Proce. VLDB Endowment 8 (10) (2015) 998–1009.
[60] Wayne W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.
[61] Ju-ping Zhang, Zhen Jin, Epidemic spreading on complex networks with community structure, Appl. Math. Comput. 219 (6) (2012) 2829–2838.
[62] Xian-Kun Zhang, Jing Ren, Chen Song, Jia Jia, Qian Zhang, Label propagation algorithm for community detection based on node importance and label influence, Phys. Lett. A 381 (33) (2017) 2691–2698.