Knowledge-Based Systems 205 (2020) 106256

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Community detection based on the Matthew effect

Zejun Sun [a,*], Yanan Sun [b], Xinfeng Chang [a], Qiming Wang [a], Xuetong Yan [c], Zhongqiang Pan [b], Zong-ping Li [c]

[a] School of Information Engineering, Pingdingshan University, Pingdingshan, Henan, China
[b] Department of Network Center, Pingdingshan University, Pingdingshan, Henan, China
[c] ZhongYe ChangTian International Engineering Co., Ltd., Changsha, Hunan, China

ARTICLE INFO

Article history:
Received 3 February 2020
Received in revised form 20 June 2020
Accepted 11 July 2020
Available online 16 July 2020

Keywords:
Community detection
Matthew effect
Complex network
Cluster

ABSTRACT

Community structure exists in most real-world networks, such as social networks, smart grids, and transportation networks. Established approaches for community detection usually depend on some user-defined criteria (e.g., minimum cut, normalized cut, modularity, etc.). These criteria-based methods usually involve some optimization procedures and need to specify some parameters, which are thus time consuming and sensitive to parameters. In this paper, inspired by the Mathew effect of human society, we view a network as a social system and design a new algorithm called CDME (community detection based on the Matthew effect). Relying on the new concept, CDME has many desirable properties. It allows uncovering high-quality communities driven by dynamic CDME is also parameter free. More importantly, since CDME works in a local way and only needs to calculate the attractiveness of neighboring nodes, which lend itself to handling large-scale networks. Experiments on both synthetic and real-world data sets have demonstrated that CDME has many benefits and outperforms many state-of-the-art algorithms.

1. Introduction

In recent years, complex network mining has attracted considerable attention, and many methods have been developed to study the function and topology of networks [1–5]. Community detection provides a promising way to analyze their structural characteristics, study their organizational function, and mine the hidden connections. In addition, community detection is widely used as a powerful tool in multiple disciplines and fields, such as computer science, bioinformatics, sociology, economic, and epidemiology [6–9]. For example, it can be used to discover organizational groups and provide personalized services in social networks [10]. In e-commerce networks, community detection can be used for intelligent recommendations and accurate marketing [11]. In criminal and counter-terrorism networks, it can be used to find criminal gangs [12]. Identifying communities in networks can also benefit many applications, such as optimizing routing tables [13], discovering the close links between proteins [14], and analyzing the cooperative relationships among authors in a citation network [15]. Therefore, it is important to study and develop efficient and accurate methods to identify the community structure. Thus far, a variety of community detection methods have been proposed from different perspectives; these include graph-partitioning-based methods [16], modularity-based methods [17], dynamics-based methods [18–20], and other methods [21]. Among these approaches, identifying communities through the inherent topology of networks with a dynamic model is an emerging and promising method owing to its simplicity, efficiency, accuracy, and data-driven nature. The main idea is to treat the network as a dynamical system and build a dynamic model to simulate the interaction among nodes. With the continuous interaction between nodes, the dynamics will converge to a steady state in which the community structure will be revealed intuitively. Although many dynamics-based methods have been proposed and some success has been achieved, most of the methods have some limitations, such as high time complexity, parameter setting requirement, and stability issues. For example, WalkTrap [18] is a dynamics-based method that uses a random walk to obtain high-quality communities. However, the time complexity is $O(mn^2)$. The Markov Cluster Algorithm (MCL) [22] is a famous dynamics-based approach and widely used in graph clustering. Nevertheless, it is sensitive to the parameter inflation [23]. Moreover, the Label Propagation Algorithm (LPA) [20] has a nearly linear time complexity. However, the results of community detection are not always steady. The Fluid Communities (FluidC) algorithm [24] is a diffusion-based method.

* Corresponding author.
E-mail address: szj@pdsu.edu.cn (Z. Sun).

https://doi.org/10.1016/j.knosys.2020.106256
0950-7051/

Similar to the LPA, FluidC does not always return the same result during each run.

In this study, a new community detection method based on the Matthew effect, called CDME, is proposed to reveal the community structures in networks. We view a network as a social system and study the dynamics over time. Inspired by the real world, we design a new dynamical model to simulate the Matthew effect. This new perspective presents a fresh approach to community detection and enables high-quality community partition. In addition, the CDME algorithm does not need parameter settings or prior knowledge of the network structure. Importantly, CDME can be used to identify community structure in large-scale networks because of its low time complexity. We will cover the CDME algorithm in detail in Section 3, but first let us understand the basic idea of this method.

1.1. Basic idea

In the real world, only a few individuals have most of the resources, and these advantages attract more people to join them. This is the Matthew effect [25], and this can be observed in nature and in many aspects of life activities. The Matthew effect is mainly used to portray the phenomenon that the rich get richer and the poor get poorer. Similarly, this phenomenon also exists in complex networks. For example, a small number of nodes have more resources in a network. In particular, in a scale-free network, a small number of nodes have a large degree, while most nodes have a small degree [26]. Based on this principle, we are curious about whether we can automatically uncover the community structure by simulating the Matthew effect of nodes, because, in the community structure, most of the edges are concentrated within the community, while the edges between the communities are very few. This coincides with the Matthew effect principle. Inspired by this, we developed a new approach based on the Matthew effect to acquire insights into the partitioning of communities, where the main idea is to consider the network as an adaptive dynamical system and study the dynamics over time. Specifically, in a social network, a small number of people have more resources, which attract and gather other individuals to join, forming initial groups. These groups are more attractive, which can further attract more individuals to join and form larger communities. Over time, different individuals are attracted to different communities according to their own interests and hobbies. We will formally define the Matthew effect model on a given network in Section 3. Here, we focus on the intuition of clustering by the Matthew effect, which consists of the following stages. First, each node of a network has its own resources and a different community. Then, the nodes with more resources attract their neighbors to join and form core groups. Next, building upon the Matthew principle, an increasing number of nodes are attracted to join the core groups and form larger communities. Finally, as time evolves, each node is attracted to different communities.

To better illustrate the basic idea, we use a toy network as an example. Fig. 1 displays the process of community detection based on the Matthew effect, which can be divided into several phases. In the beginning, each person has his or her own resources and an independent group, as shown in Fig. 1(a). In the second phase, people with more resources attract neighbors to join and form the initial core group. For example, because user U_7 has more resources, this attracts U_1, U_8, U_{10}, and U_{12} to join its group, as shown in Fig. 1(b). In the third phase, because of the Matthew effect, an increasing number of people are attracted to join the core groups. For example, user U_9 is attracted by a group whose label is L_7, as shown in Fig. 1(c). In the end, everyone is drawn to different communities, and the community structure is naturally exposed, as shown in Fig. 1(d).

1.2. Contributions

By imitating the Matthew effect, the CDME algorithm has several desirable properties for community detection in complex networks. The most important are as follows:

- **New viewpoint**: We consider the problem of community detection from a new viewpoint: the Matthew effect, which reveals the community structure in a network by a natural manner.
- **High performance**: Compared to several representative community detection methods, the CDME algorithm has proven to be more efficient and can find high-quality communities in both synthetic and real-world networks. (cf. Figs. 3 and 4 and Tables 6 and 7). This is because the CDME method not only considers the node-to-node impact, but also the community's impact on it.
- **Parameter-free**: Instead of algorithms that depend on prior knowledge and parameter settings, the CDME method does not need parameter settings, and the community structure is automatically revealed based on the Matthew principle.
- **Scalability**: Owing to the local interaction model, CDME only needs to calculate the attractiveness of neighboring nodes, which contribute to a low time complexity of $O(n \cdot k^2)$. The value of k is usually very small. Therefore, the CDME algorithm can be applied to large-scale networks.

The rest of this paper is organized as follows. In Section 2, we briefly provide an overview of related work. In Section 3, we give a description of the main idea of CDME and present the Matthew effect model and algorithm in detail. In Section 4, we compare CDME with several representative algorithms on synthetic and real-world networks. Finally, in Section 5, we provide the conclusions.

2. Related work

Many related works have been presented for detecting communities in networks during recent decades. In the following, we only briefly introduce the algorithms related to the present study. More detailed literature on community detection is afforded by the excellent reviews [27,28].

2.1. Graph-partitioning-based methods

A community detection method based on graph-partitioning divides one network into disjoint groups, thereby minimizing the number of edges between groups. Numerous graph-partitioning-based algorithms have been developed. Specifically, Kernighan and Lin [29] designed the Kernighan–Lin algorithm in 1970; this is one of the earliest approaches based on graph-partitioning a graph and is still frequently used. Wu and Leahy have developed a graph-partitioning method based on the minimum-cut criterion [30], which employs the weights of the deleted edges to calculate the minimum cut. Next, disjoint communities are obtained by iteratively computing the minimum cuts through bisecting the existing segments. One disadvantage of the minimum cut is that the partition is unbalanced as a result of the bias toward partitioning small size subgraphs. To solve this problem, researchers have proposed ratio cut [31] and normalized cut [16] methods. The normalized cut is a well-known graph partitioning approach that employs a generalized eigenvalue decomposition to optimize the normalized cut. In most cases, this class of graph-partitioning methods depending on the eigenvector decomposition of an adjacent matrix is also called spectral clustering [32]. One of its advantages is that it can recognize the sample space of any shape and converge to the most global solution. However, it requires real community partitioning as prior knowledge, which may be difficult to obtain in advance.

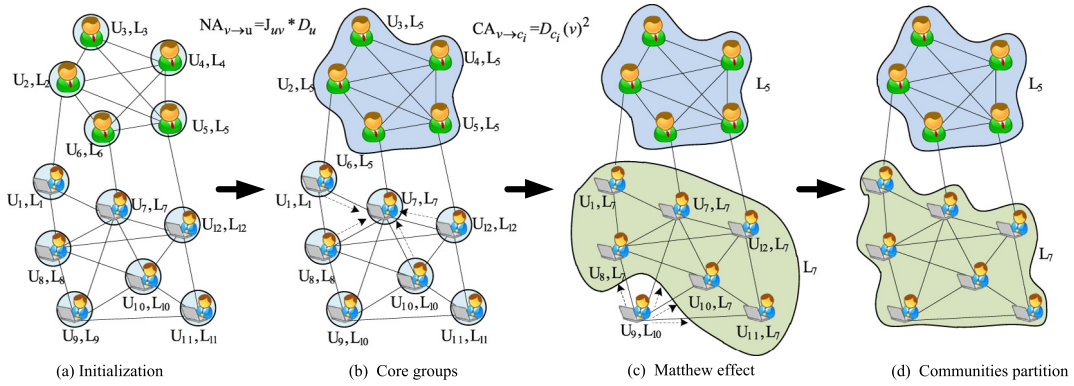

Fig. 1. Diagram of community detection based on the Matthew effect. (a) A social network where the edges represent the relationships among users, and everyone has a different label. (b) Different resource attractiveness leads to the formation of core groups. (c) The Matthew effect attracts more people to join the communities. (d) The end result is that communities are uncovered naturally.

2.2. Optimization-based methods

Optimization techniques are widely used in community detection. The goal is to find the optimal value of the function indicating the quality of clustering in all possible clustering spaces. The objective function is used to express the goodness of community partitions. Modularity is the most popular objective function proposed by Newman and Girvan [33] for estimating the quality of community detection in networks. Modularity-based methods can reveal the optimal partitions of communities by optimizing the modularity function. However, it is impossible to perform an exhaustive optimization of modularity, because modularity optimization is an NP-complete problem [23]. Fortunately, many algorithms have been developed for efficiently acquiring approximate solutions, such as Louvain (BGLL) [34], EDCD [35], Fiduccia–Mattheyses (FM) [36], and Fastgreedy (FG) [17]. However, one disadvantage of the modularity-based approaches is the "resolution limit" on real-world networks [37].

2.3. Dynamics-based methods

Dynamics-based methods regard the networks as adaptive dynamical systems, and their dynamics, such as synchronization, diffusion, and spin dynamics, get inspected over time. Numerous algorithms based on dynamics have been proposed in the past ten years. One of the most popular approaches of this style is random walk dynamics, in which it is presumed that the walker is likely to be trapped in dense clusters because of higher inner edge density and lower outer edge density. WalkTrap (WT) [18] is a classic algorithm based on a random walk; it employs random walk dynamics to measure the similarity between vertices. One disadvantage of WT is the high time complexity ($O(mn^2)$), and it cannot be used to handle large-scale networks. Another representative method based on the random walk is Infomap, proposed by Rosvall and Bergstrom [38]. It is a map equation algorithm that considers community detection as a coding problem and yields the optimal communities using the minimum description length principle. The advantage of Infomap is that it can be applied to weighted, directed, and undirected networks. The MCL [22] is a popular graph-clustering algorithm that uses Markov chains to simulate the random walk and employs two alternating processes—*expansion* and *inflation*—to detect communities in networks. The MCL can be applied to weighted and unweighted networks and yields higher clustering quality. However, it is sensitive to the parameter *inflation* [23]. The LPA [20] is a famous diffusion approach that detects the communities by label propagation. The LPA has a nearly linear time complexity and can be easily used to handle large-scale networks. However, one drawback of the LPA is that the results are not always steady. Therefore, various improved methods based on the LPA have been proposed [39,40]. FluidC [24] is also a dynamics method based on propagation; it identifies communities by imitating the expansion and contraction of fluids in an environment until equilibrium is reached. FluidC has lower time complexity and good scalability. Nevertheless, the results of community detection, similar to those of the LPA, are also unstable.

2.4. Other methods

Skrlj et al. [21] developed a community detection method based on network representation learning. It regards community detection as a network embedding clustering problem and finds the optimal partition of the Silhouette metric. Pan et al. [41] proposed an adjacent node similarity optimization combination connectivity algorithm, which finds the optimal community structure by computing the modularity of each community. Bu et al. proposed a novel graph K-means framework based on game theory for detecting community structure in social media networks [42]. Cao et al. designed a dynamic game model to find the locally Pareto-optimal prosumer community group structure in smart grids [43].

Despite the success of the many proposed community detection algorithms, existing methods still have limitations. Here, we introduce a more intuitive approach to uncover the community structure based on the Matthew effect that can not only discover high-quality communities but also handle large-scale networks.

3. Proposed method

3.1. Preliminaries

Before elaborating the CDME method, we first introduce some notation and formalize some basic definitions. All of the key symbols are listed in Table 1, and we briefly review the notation here. Let $G = (V, E)$ be the given network, which represents an undirected and an unweighted graph, where V is the set of nodes and E is the set of edges.

Definition 1 (*Jaccard Similarity Coefficient [44]*)**.** Given an undirected network $G = (V, E)$, the Jaccard similarity coefficient of nodes u and v is defined as

$$J_{uv} = \frac{|\Gamma_u \bigcap \Gamma_v|}{|\Gamma_u \bigcup \Gamma_v|}, \tag{1}$$

where $\Gamma_u = N(u) \cup \{u\}$. It is a set of neighbors of node u that comprises node u and its directly connected nodes.

Table 1
Notation summary.

Symbol	Description
n	Number of vertices ($n = \|V\|$)
m	Number of edges ($m = \|E\|$)
D_v	Degree of vertex v
$d_{c_i}(v)$	Internal degree of vertex v in a community c_i
$N(v)$	Neighbors of vertex v
J_{uv}	Jaccard similarity coefficient between vertices u and v
$NA_{v\to u}$	Degree of attraction of vertex u to vertex v
$CA_{v\to c_i}$	Degree of attraction of c_i to vertex v

In the real world, people have their own resources and hobbies. Individuals with the same hobbies are more likely to attract each other and form a community. To describe the attraction between nodes, we use the degree of nodes and the similarity between them to represent the attraction of nodes.

Definition 2 (*Node Attraction*)**.** Given an undirected network $G = (V, E)$, the attraction of node u to v is defined as

$$NA_{v\to u} = J_{uv} * D_u, \tag{2}$$

where J_{uv} represents the Jaccard similarity coefficient between nodes u and v, and D_u denotes the degree of node u.

Similarly, communities are also attractive to nodes in a network. For example, a well-known football club attracts many football players to join the team. Here, we use the connectivity of nodes within a community to characterize the attractiveness of the community to the nodes.

Definition 3 (*Community Attraction*)**.** Given the undirected network $G = (V, E)$, the attraction of one community to a node v is defined as

$$CA_{v\to c_i} = D_{c_i}(v)^2, \tag{3}$$

where $D_{c_i}(v)$ denotes the degree of connectivity from a node v to one community c_i, which portrays how close a node is to one community. Based on the number of edges from one node to different communities, there are two situations, one in which the numbers of edges are different and the other in which the numbers of edges are the same. Formally, we define $D_{c_i}(v)$ as follows:

$$D_{c_i(v)} = \begin{cases} d_{c_i}(v), & d_{c_i}(v) \neq d_{c_j}(v), \\ \sum\limits_{u\in N(v), u\in c_i} d_{c_i}(u), & d_{c_i}(v) = d_{c_j}(v), \end{cases} \tag{4}$$

where $d_{c_i}(v)$ is the internal degree of node v in a community c_i.

Fig. 2 shows an example of community attraction for a case consisting of thirteen nodes and two communities. We can use Eqs. (3) and (4) to calculate the community attraction. For node 6, $d_{c_i}(6) = 2$ and $d_{c_j}(6) = 1$, and we can obtain $d_{c_i}(6) \neq d_{c_j}(6)$, $D_{c_i(6)} = 2$, and $D_{c_j(6)} = 1$. Therefore, $CA_{6\to c_i} = 4$ and $CA_{6\to c_j} = 1$, and node 6 can join community c_i with greater community attraction. For node 13, because $d_{c_i}(13) = d_{c_j}(13)$, we need to consider the impact of indirect neighbors on node 13. According to the second part of Eq. (4), we can calculate that the community attraction of two communities to node 13 as $CA_{13\to c_i} = 16$ and $CA_{13\to c_j} = 9$, respectively. Therefore, community c_i has a greater appeal to node 13.

3.2. Matthew effect model

After introducing the key notation, we begin to construct the model of the Matthew effect, which consists of initialization, core groups, and Matthew effect stages. At first, each node acts as a separate group, and each node has its own resources. Next, some nodes are attracted by other nodes to form core groups because of common interests or hobbies. Then, an increasing number of nodes are attracted to join the core groups because of the Matthew effect. At last, every node is attracted to different communities.

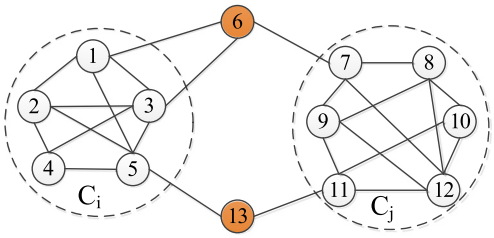

Fig. 2. Example of community attraction.

(1) **Initialization.** In the real world, the more friends one person has, the more resources he or she may have. Therefore, we use the degree of a node to describe the resources it has. In the beginning, every node has its own resources, and each one is regarded as an independent community.

(2) **Core groups.** In real life, people play different roles and have different resources in a community. The more important the person is, the more possible he or she attracts people to join the community. Similarly, each node in a network also has its own resources that attract other nodes. We use the term *node attraction* to describe the attraction between nodes (cf. Eq (2)). Because of *node attraction*, each node attracts its neighbors to join its community. In other words, each node selects the neighbor node with the strongest attraction according to

$$C_{core\_v} = \begin{cases} c_v, & D_v > \max(D_u),\ u \in N(v), \\ c_u, & D_v \leq \max(D_u),\ u \in N(v), \\ \max(NA(v\to u)) & \end{cases} \tag{5}$$

where C_{core_v} represents the core group to which node v belongs, and D_v is the degree of node v. It then joins its group. If node v has the largest degree compared with neighbor nodes, then node v still belongs to the original group c_v. In contrast, if a neighbor node has the larger degree than node v, then the node v chooses the neighbor u with the maximum node attraction $\max(NA(v \to u))$ and joins its community. Through a round of iteration, nodes with more resources are more likely to attract their neighbors to join their communities and form many core groups.

(3) **Matthew effect.** After core groupings are obtained, an increasing number of nodes are attracted to different core groupings according to

$$C_v = \underset{\max(CA_{v\to c_i})}{C_i}, \tag{6}$$

where C_i is the neighborhood community of node v, and $CA_{v\to c_i}$ represents the attraction of community C_i to node v. In the real world, the better the team is, the more attention and participation it will attract. Similarly, a node naturally chooses the most attractive community to join. When a node joins one community, the structure of this community may change, and its attractiveness may also change. Therefore, we iteratively introduce the Matthew effect. In each iteration, every node updates its community label. Finally, driven by the network topology, the communities to which the nodes belong do not change, and

the network structure reaches an equilibrium state. Then, the community structure of the network is also revealed naturally.

3.3. CDME algorithm

In this section, we introduce the Matthew effect algorithm CDME, which involves the following steps.

(1) **Initialization**. In the beginning, we let each node be an independent community. Here, we use the node number to initialize the community label for each node.
(2) **Computing the core groups**. First, we need to calculate the attraction between nodes using Eq. (2), which is related to the resources owned by nodes and the similarity between nodes. Because of node attraction, a node may attract neighboring nodes to join its community, forming the core group. Then, we use a round of iterations to compute core groups according to Eq. (5).
(3) **Simulating the Matthew effect**. Here,we use an iterative method to simulate the Matthew effect process of community attraction nodes according to Eqs. (3), (4) and (6). Building upon the Matthew effect models, a core group continuously attracts the surrounding nodes to join, forming a larger community structure. In addition, we use the normalized mutual information (NMI) indicator to assess the quality of each community partition.
(4) **community detection**. As time evolves, the network structure reaches a steady state because of topologically driven influences; then, we can get the best partition of communities. The CDME algorithm is given in Algorithm 1.

3.4. Complexity analysis

The CDME algorithm is mainly composed of three parts. In the first step, each node acts as an independent community, and the time complexity is $O(n)$. In the second step, two loops are needed to compute the core groups in a network. Consequently, the time complexity is $O(k \cdot n)$, where k is the average degree of a network. In the third step comprising the Matthew effect process, the time complexity is $O(L \cdot k^2 \cdot n)$, where L is the total number of iterations, which typically ranges between 3 and 10. Hence, the computational complexity of CDME is $O(k^2 \cdot n)$. We note that $k \ll n$, and thus the CDME can handle large-scale networks.

4. Experiments

To evaluate the performance of the CDME algorithm, we use synthetic and real-world networks to demonstrate its benefits based on comparisons with several community detection methods. Before making a detailed experimental comparison, we briefly introduce the representative algorithms.

Ncut [16] is a classic graph-clustering method. It employs the similarity matrix of the sample data to cluster the eigenvectors obtained by eigendecomposition. It then transforms the optimal problem of Ncut (normalized cut) into the eigenvector of the solution matrix.

Infomap [38] is a network-clustering algorithm based on the map equation. It treats community detection as a coding problem and yields the optimal community structure by using the minimum description length principle.

LPA [20] is a well-known graph label propagation approach. It takes the label with the largest number of labels on the neighbor nodes as its own label. It is simple and efficient, but the result of each iteration is not stable.

Algorithm 1 CDME

Input:
$G = (V, E)$
//Initialize communities of each node
for each node v in V **do**
  $C_v \leftarrow$ the node number of v
end for
//Compute the core groups
for each node v in V **do**
  for each node u in $N(v)$ **do**
    compute the Jaccard similarity coefficient using Eq. (1)
    compute the node attraction using Eq. (2)
  end for
  compute core groups C_{core} using Eq. (5)
end for
//Simulate the Matthew effect
$Flag = TRUE$, $NMI_{max} = 0$
while *Flag* **do**
  for each node v in V **do**
    for each node u in $N(v)$ **do**
      compute the community attraction using Eqs. (3) and (4)
    end for
    update core groups C_{core} using Eq. (6)
  end for
  $C = C_{core}$
  $S = NMI(C, true\_clusters)$
  if $S > NMI_{max}$ **then**
    $NMI_{max} = S$
  else
    $Flag = FALSE$
  end if
end while
// return communities C
Output: C

WT [18] is a network-clustering algorithm based on random walks. It partitions the communities by computing the similarities between nodes using random walks.

FG [17] is a hierarchical agglomeration method for detecting communities. It uncovers the community structure by greedily optimizing the modularity.

Louvain [34] is a famous community detection algorithm that employs a multi-level modularity optimization method to reveal the community structure. Louvain can be used to detect hierarchical communities.

MCL [22] is a popular clustering algorithm based on the simulation of flow in graphs. MCL is widely used in different domains, mostly in life sciences.

FluidC [24] is also a label propagation method that imitates the process of fluids expanding and contracting to uncover the community structure.

EDCD [35] is an optimal modularity algorithm that iteratively deletes edges using a restriction strategy to find strongly connected communities.

SCD [21] is a community detection algorithm based on network embedding. It reveals the community structure via optimization of the Silhouette metric.

ASOCCA [41] is an adjacent node similarity optimization combination connectivity algorithm. It utilizes the local similarity measure based on clustering coefficient to detect the communities.

For all experiments, we use the default values suggested by the authors as the parameter values of every comparison

algorithm. Because Ncut and FluidC need the number of communities as prior knowledge, we use the real number of communities as their parameters. The FG, LPA, WT, Louvain, and Infomap algorithms were implemented with igraph (http://igraph.org/) packages. Ncut employs the sklearn-cluster method (http://scikit-learn.org/stable/modules/clustering.html). MCL is available from the website http://micans.org/mcl/. FluidC is available at Github (github.com/FerranPares-/Fluid-Communities). SCD is available at Github (github. com/SkBlaz/SCD). ASOCCA is available at Github (github.com/BasdekD/community-detection-in-social-networks). For every network, each algorithm was run 20 times independently, and the results were then averaged. All experiments have been performed on a desktop computer with a 3.3 GHz CPU with an Intel Core i5 processor and 16.0 GB RAM.

4.1. Evaluation metrics

To extensively evaluate the performance of different algorithms, we employed three widely used evaluation measures for evaluating the accuracy of community detection. Before presenting the testing results, let us briefly introduce the three indexes.

One of the most used metrics is NMI [45], which is a similarity index originating from information theory. NMI measures how close the predicted communities are to the ground truth. It is defined as follows:

$$\mathrm{NMI}(X;Y)=\frac{2I(X;Y)}{H(X)+H(Y)}, \tag{7}$$

where $I(X;Y)$ is the mutual information between X (ground truth) and Y (predicted communities) and $H(X)$ represents the entropy of X. The value of NMI ranges from 0 to 1. NMI $=$ 1 indicates that the predicted communities exactly match the ground truth. In contrast, NMI $=$ 0 indicates that the predicted communities are completely independent of the ground truth.

Another popular metric is the adjusted rand index (ARI) [46], which measures the similarity between two clusters. It is defined as follows:

$$\mathrm{ARI}=\frac{\mathrm{RI}-\mathrm{ExpectedRI}}{\mathrm{MaxRI}-\mathrm{ExpectedRI}}, \tag{8}$$

where RI denotes a similarity measure between two clusters, which computes all pairs of samples. Then, the numbers of pairs that are assigned to the same or different clusters in the predicted and ground truth are counted [47]. Specifically,

$$\mathrm{ARI}=\frac{\sum_{ij}\binom{n_{ij}}{2}-\left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}\right]-\left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]/\binom{n}{2}}, \tag{9}$$

where n_{ij}, a_i, and b_j are values from the contingency table. For more details, refer to [48].

Cluster purity [49] is an external index for community detection quality. It is the percentage of the total number of samples that were classified correctly. It is defined as follows:

$$\mathrm{Purity}(\Omega,C)=\frac{1}{N}\sum_{i=1}^{k}\max_j|\omega_i\cap c_j|, \tag{10}$$

where N is the number of samples, $\Omega=\{\omega_1,\omega_2,\ldots,\omega_k\}$ is the set of predicted communities, and $C=\{c_1,c_2,\ldots,c_j\}$ is the set of ground truth.

Table 2
Description of the parameters for the LFR model.

Symbol	Definition
n	Number of nodes of a generated network
μ	Mixing parameter
k	Average degree of a generated network
$C_{\min}$	Minimum of community sizes of a generated network
$C_{\max}$	Maximum of community sizes of a generated network
τ_1	Power-law exponents for the degree sequence
τ_2	Minus exponent for the community size distribution
cc	Average clustering coefficient of a generated network

4.2. Synthetic networks

In this section, we use the famous LFR model [50] to create synthetic benchmark networks. LFR can be easily controlled to generate networks by multiple attribute parameters of real-life networks, such as average degree, community size, and clustering coefficient. The most important parameter of the LFR model is the mixing parameter μ, which is used to control the complexity of community partition. 2 describes the parameters of the LFR model.

To evaluate the performances of the comparison algorithms, we varied parameter μ from 0.1 and 0.8 to increase the difficulty of the generated networks. We fixed the other parameters to $n=1000$, $k=15$, $C_{\min}=10$, $C_{\max}=50$, $\tau_1=2$, and $\tau_2=1$. Fig. 3 shows the performances of each algorithm on the three different evaluation metrics. In the NMI metric, CDME, Infomap, Ncut, LPA, MCL, WT, Louvain, and FluidC methods yielded almost perfect clustering when the parameter μ ranges from 0.1 to 0.5. However, the performances of LPA, Infomap, and FluidC began to drop significantly as the value of μ increases. In particular, the LPA had an NMI index of substantially zero when the value of μ was increased to 0.6. In contrast, MCL and CDME algorithms were more robust than other comparison methods, and they still achieved better community partition when μ was increased to 0.8. Fig. 3(a) shows that the FG algorithm was not ideal for these networks, and the NMI result was not comparable with that of other algorithms. FG also yielded lower values in terms of ARI and Purity metrics, as shown in Fig. 3(b) and (c). This indicates that the FG method is more sensitive to the parameter μ; that is, it was sensitive to noise. Similar to their performance on the NMI metric, CDME, Infomap, Ncut, and WT still performed comparatively well on the ARI metric. However, the performance of the LPA, MCL, FG, EDCD, SCD, and ASOCCA methods decreased significantly when $\mu > 0.6$. In terms of the Purity metric, EDCD, MCL, CDME, WT, and Ncut achieved better community purity than other methods. In particular, the value of the EDCD algorithm on the Purity metric was close to 1. Although EDCD yielded a high Purity value, its NMI and ARI values were lower. This is because it divided the community too finely.

To more accurately reflect the performance of each comparison algorithm, we employed the standard deviation to evaluate the results of community detection. Specifically, we increased the parameter μ from 0.1 to 0.8, and 10 networks were generated for each change in the parameter, resulting in a total of 80 networks. Table 3 illustrates the average accuracy with standard deviation of each algorithm on these synthetic networks. For brevity, we only show the results of the NMI index. We can see that most of the methods achieved low standard deviation. For example, CDME, Infomap, and WT reached the highest value of 1, and the lowest standard deviation of 0 when the parameter μ varied from 0.1 to 0.4. Even when the parameter μ was increased to 0.8, MCL and CDME still achieved better community division. However, the average accuracy of the LPA was close to 0 as the parameter μ was increased to 0.6.

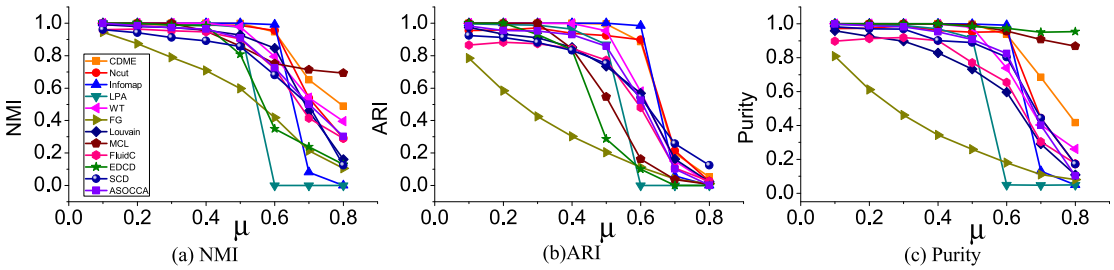

Fig. 3. Performance of each comparison algorithm with the synthetic benchmark networks when the parameter μ ranges from 0.1 to 0.8.

Table 3
Average accuracy and standard deviation (std) of each method on the NMI index when the parameter μ changes from 0.1 to 0.8.

Method	NMI (mean(std))							
	DS_μ1	DS_μ2	DS_μ3	DS_μ4	DS_μ5	DS_μ6	DS_μ7	DS_μ8
CDME	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	0.95(0.02)	0.65(0.03)	0.50(0.01)
Ncut	1.00(0.01)	1.00(0.01)	1.00(0.01)	1.00(0.01)	1.00(0.01)	0.96(0.01)	0.54(0.04)	0.29(0.02)
Infomap	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.99(0.00)**	0.08(0.24)	0.00(0.00)
LPA	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	1.00(0.01)	0.98(0.02)	0.00(0.00)	0.00(0.00)	0.00(0.00)
WT	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	0.98(0.01)	0.79(0.03)	0.54(0.05)	0.40(0.06)
FG	0.95(0.01)	0.87(0.01)	0.79(0.02)	0.71(0.02)	0.60(0.03)	0.42(0.04)	0.21(0.02)	0.11(0.01)
Louvain	0.99(0.01)	0.98(0.01)	0.97(0.01)	0.96(0.01)	0.93(0.01)	0.85(0.02)	0.46(0.06)	0.16(0.02)
MCL	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	0.95(0.02)	0.86(0.02)	0.75(0.02)	**0.71(0.01)**	**0.69(0.01)**
FluidC	0.96(0.01)	0.96(0.01)	0.95(0.01)	0.95(0.01)	0.90(0.03)	0.72(0.06)	0.41(0.03)	0.29(0.01)
EDCD	**1.00(0.00)**	**1.00(0.00)**	1.00(0.01)	0.98(0.01)	0.81(0.04)	0.35(0.03)	0.23(0.03)	0.13(0.02)
SCD	0.96(0.01)	0.94(0.01)	0.91(0.02)	0.89(0.02)	0.85(0.03)	0.68(0.04)	0.48(0.03)	0.12(0.04)
ASOCCA	**1.00(0.00)**	0.98(0.01)	0.97(0.01)	0.96(0.02)	0.90(0.02)	0.72(0.03)	0.50(0.03)	0.30(0.04)

Table 4
Average accuracy and standard deviation (std) of each method on the NMI index when the average degree parameter $\langle k\rangle$ ranges from 5 to 25.

Method	NMI (mean(std))				
	DS_k1	DS_k2	DS_k3	DS_k4	DS_k5
CDME	**0.96(0.01)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**
Ncut	0.94(0.02)	0.99(0.01)	0.98(0.01)	0.98(0.01)	0.98(0.00)
Infomap	0.95(0.01)	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**
LPA	0.94(0.01)	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**
WT	0.94(0.01)	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**
FG	0.75(0.01)	0.94(0.01)	0.95(0.01)	0.95(0.01)	0.94(0.01)
Louvain	0.93(0.01)	0.99(0.01)	0.99(0.01)	0.99(0.00)	**1.00(0.00)**
MCL	0.93(0.02)	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**
FluidC	0.86(0.01)	0.90(0.01)	0.96(0.01)	0.97(0.01)	0.98(0.01)
EDCD	0.90(0.01)	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**
SCD	0.83(0.03)	0.93(0.02)	0.96(0.01)	0.96(0.01)	0.96(0.01)
ASOCCA	0.86(0.02)	0.90(0.02)	**1.00(0.00)**	0.98(0.01)	0.98(0.01)

In addition, to evaluate the effectiveness of each comparison approach on networks with different community densities, we fixed the parameter $\mu = 0.1$ and varied the average degree parameter $\langle k\rangle$ from 5 to 25 to create synthetic benchmark networks. Fig. 4 illustrates the efficiency of each comparison algorithm on the synthetic networks with different metrics when the parameter $\langle k\rangle$ ranged from 5 to 25. We can see that CDME, Ncut, Infomap, MCL, LPA, WT, Louvain, and EDCD methods achieved good results on these synthetic networks when the average degree parameter $k \geq 10$. However, the quality of community detection by FG, SCD, ASOCCA, FluidC, and EDCD methods clearly decreased when $\langle k\rangle \leq 5$. In particular, the FG algorithm did not achieve ideal performance on these networks. The FG algorithm was sensitive to community density on these synthetic networks when $\langle k\rangle \leq 10$. This might be caused by the limit of the modularity resolution [37]. Table 4 lists the average accuracy and standard deviation for each comparison algorithm on the NMI metric as the average degree changed from 5 to 25. We notice that when $\langle k\rangle = 5$, FG and FluidC methods achieved lower NMI values than other algorithms.

Table 5
Statistical characteristics of each real-world network where $|V|$ denotes the number of nodes, $|E|$ is the number of edges, $\langle k\rangle$ represents the average degree, CC is the clustering coefficient, and $\#C$ is the number of communities.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\langle k\rangle$	CC	$\#C$
Karate	34	78	4.588	0.57	2
Football	115	613	10.661	0.40	12
Polbooks	105	441	8.4	0.49	3
Dolphin	62	159	5.13	0.30	2
Amazon	334,863	925,872	5.53	0.40	75,149
DBLP	317,080	1,049,866	6.622	0.63	13,477
LiveJournal	3,997,962	34,681,189	17.35	0.28	287,517

4.3. Real-world networks

To further evaluate each comparison algorithm, we studied their performances on real-world selected from distinct domains with different characteristics. Because these networks have ground truth, we used NMI, ARI, and Purity metrics to evaluate the quality of community detection by every algorithm. The statistics of these real-world networks are summarized in Table 5. These datasets are available at http://snap.stanford.edu/data/, http://networkrepository.com/networks.php, and https://networkdata.ics.uci.edu/index.php. Before the specific evaluation, let us briefly describe these network datasets.

Zachary's karate network: This network originates from the USA-based Zachary's karate club dataset. It is a friendship network with 34 members. The network is divided into two groups because of differences of opinion among leaders.

Football network: This famous network derives from American college football games. It contains 115 college football teams representing nodes and 613 edges denoting regular-season games. The teams are divided into 12 communities as the ground truth.

Politics books network: This network comes from the American politics books dataset. These books are sold by online sellers on Amazon.com. The network includes 105 nodes denoting books and 441 edges representing the frequent co-purchasing of books by the same buyers.

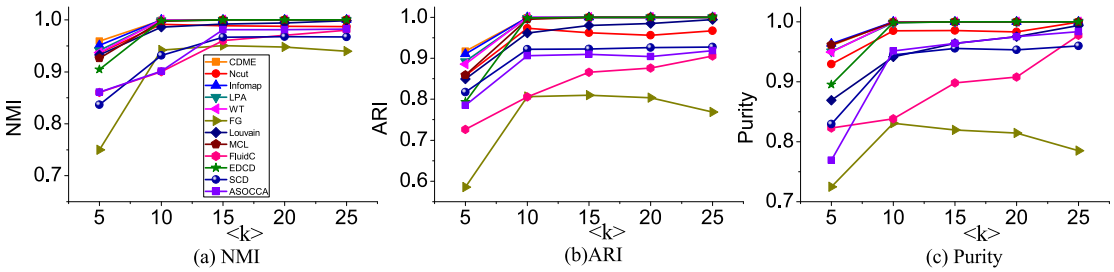

Fig. 4. Effectiveness of each comparison algorithm with the synthetic benchmark networks when the average degree $\langle k \rangle$ ranges from 5 to 25.

Dolphin network: This network is formed by the frequency of bottlenose dolphins playing together. It consists of 62 nodes and 159 edges, grouped into two communities.

Amazon network: This network is a product co-purchasing network collected by the crawling Amazon website. It contains 334,863 nodes representing products and 925,872 edges denoting the relationship among frequent co-purchasing products. The categories to which the products belong are considered ground-truth communities.

DBLP collaboration network: This collaboration network is provided by the DBLP computer science bibliography. It consists of 317,080 nodes and 1,049,866 edges. The authors are denoted as nodes, and the co-authorships among authors are represented as edges; that is, two authors are connected if they publish one paper together. Authors who published in a certain journal or conference form a community.

LiveJournal network: LiveJournal is a social network deriving from a free on-line blogging community. It contains 3,997,962 nodes and 34,681,189 edges. In this network, users form a group denoted as a ground-truth community, and each user is represented as a node.

Tables 6 and 7 list the results of community detection by comparison algorithms on real-world networks. $\#C$ represents the number of communities obtained by each method. Ncut and FluidC algorithms employed the real community number as their prior knowledge. Therefore, we do not display the number of communities detected by these two algorithms. We can see that the CDME algorithm performed very well on these real-world networks. For Zachary's karate club network, the CDME method successfully unclosed the communities with the highest metric scores (NMI = 1, ARI = 1, and Purity = 1) and outperformed other comparison algorithms. Fig. 5 is a visualization of the result of community detection by CDME, which identified two communities that were exactly the same as the ground truth. Some algorithms cannot classify the tenth node into the correct community because it is closely related to the two communities. However, CDME still recognized it correctly, because our approach not only considers the attraction of the nodes but also the appeal of the communities. The EDCD, LPA, MCL, FluidC, and Ncut methods also had good performance, with most nodes being correctly divided. However, some nodes were wrongly clustered by WT and SCD algorithms, which led to a lower NMI value. The performances of the comparison algorithms are given in Table 6. On the college football network, many comparison methods achieved good clustering results because of the higher average degree ($\langle k \rangle$ = 10.66). These further validated the test results of these algorithms on the synthetic benchmark networks (cf. Fig. 4). Specifically, the CDME method achieved the highest NMI scores. The Ncut, MCL, Infomap, FluidC, and EDCD algorithms also achieved good performances. Fig. 6 shows plots of the communities detected by CDME. It is interesting to observe that CDME automatically detected 12 communities with high quality (NMI = 0.93). In terms of the ARI and Purity metrics, the MCL, Infomap,

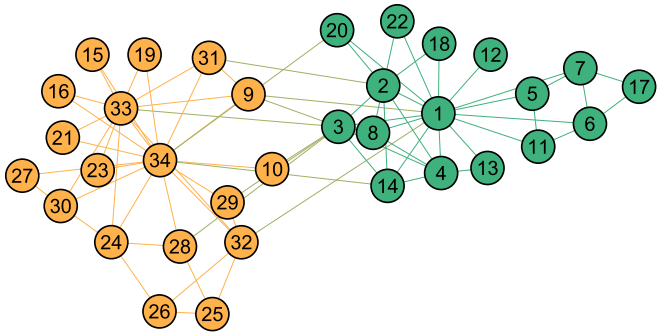

Fig. 5. CDME used on the karate club network. Two communities were detected.

and FluidC achieved the best performance. However, the FG, SCD, and ASOCCA methods performed not ideally and yielded lower metric scores. For the politics books network, CDME achieved good clustering results and had the highest NMI and ARI scores (NMI = 0.58 and ARI = 0.67). Fig. 7 shows a visualization of communities detected by CDME, and three communities were found. Although the MCL yielded the highest Purity value (= 0.96), it achieved lower NMI and ARI scores (NMI = 0.48 and ARI = 0.38). For the Dolphin network, CDME achieved good results and performed better than other comparison algorithms (Fig. 8 and Table 6). Most of the methods yielded comparable results. However, Infomap and EDCD performed not ideally. In particular, EDCD achieved the lowest NMI value, indicating that most nodes were clustered incorrectly.

To evaluate the performance of the comparison algorithm on large-scale networks, we used the large datasets, namely, Amazon, DBLP, and LiveJournal. To more effectively distinguish the community detection effects of each algorithm, we evaluated the performances of all methods on NMI, ARI, and Purity metrics with the top-5000 highest-quality true communities. In the Amazon network, the CDME, WT and Infomap algorithms had the best performances, yielding very high NMI and Purity scores (NMI = 0.96 and Purity = 0.99). In terms of the ARI, Infomap achieved the highest value (ARI = 0.94). Ncut cannot handle this network because of its high time complexities. Fig. 9 shows a visualization of communities detected by CDME. Since this network contains a large number of communities, in order to show a clearer visualization, we only display the top 10 largest communities detected. For the DBLP collaboration network, the CDME, LPA, and FluidC methods achieved good efficiency, with the highest NMI score (NMI = 0.75). CDME also yielded the highest Purity score (Purity = 0.92). However, some nodes were wrongly clustered by the Louvain and FG methods, resulting in lower NMI scores. On the LiveJournal social network, CDME achieved the best community quality, with high scores (NMI = 0.93, ARI = 0.49, and Purity = 0.98). For the Ncut, WT, FG, Infomap, MCL, EDCD, SCD, and ASOCCA algorithms, we still did

Table 6
Performances of each algorithm on real-world networks.

	Karate				Football				Polbooks				Dolphin			
	NMI	ARI	Purity	#C	NMI	ARI	Purity	#C	NMI	ARI	Purity	#C	NMI	ARI	Purity	#C
CDME	**1**	**1**	**1**	2	**0.93**	0.89	0.92	12	**0.58**	**0.67**	0.84	3	**0.70**	0.58	0.98	3
Ncut	0.83	0.88	0.97	–	**0.92**	0.89	**0.93**	–	0.57	**0.67**	0.84	–	0.53	0.57	0.89	–
WT	0.55	0.33	0.94	5	0.89	0.82	0.87	10	0.54	0.65	0.85	4	0.57	0.42	0.95	4
LPA	0.84	0.88	0.97	3	0.9	0.81	0.88	14	0.53	0.64	0.83	4	0.58	0.35	0.98	5
FG	0.71	0.68	0.97	3	0.76	0.47	0.58	6	0.53	0.64	0.84	4	0.65	0.49	0.98	4
MCL	0.84	0.88	0.97	2	**0.92**	**0.9**	**0.93**	12	0.48	0.38	**0.96**	5	0.54	0.2	**1**	12
Louvain	0.62	0.46	0.97	4	0.86	0.71	0.8	10	0.51	0.55	0.85	4	0.65	0.39	0.99	5
Infomap	0.71	0.7	0.97	3	**0.92**	**0.9**	**0.93**	12	0.5	0.54	0.85	6	0.43	0.01	0.84	6
FluidC	0.84	0.88	0.97	–	**0.92**	**0.9**	**0.93**	–	0.55	0.55	0.85	–	0.64	**0.59**	0.76	–
EDCD	0.93	0.95	0.97	3	**0.92**	**0.9**	0.92	11	0.54	0.64	0.78	4	0.37	0.13	0.38	16
SCD	0.50	0.32	0.82	8	0.70	0.50	0.61	16	0.50	0.52	0.83	5	0.51	0.40	0.80	10
ASOCCA	0.83	0.88	0.97	2	0.75	0.51	0.62	7	0.56	0.68	0.84	4	0.58	0.39	0.98	5

Table 7
Performances of each algorithm on large-scale networks.

	Amazon				DBLP				LiveJournal			
	NMI	ARI	Purity	#C	NMI	ARI	Purity	#C	NMI	ARI	Purity	#C
CDME	**0.96**	0.62	**0.99**	16,716	**0.75**	0.02	**0.92**	93,432	**0.93**	**0.49**	**0.98**	27,263
Ncut	–	–	–	–	–	–	–	–	–	–	–	–
WT	**0.96**	0.62	0.98	14,905	0.73	0.03	0.73	1,3201	–	–	–	–
LPA	0.91	0.53	**0.99**	22,504	**0.75**	0.01	0.78	21,076	0.88	0.05	0.2	29,193
FG	0.91	0.5	0.61	1,525	0.5	0.06	0.24	3,141	–	–	–	–
MCL	0.9	0.49	**0.99**	46,557	0.63	0.01	0.49	37,964	–	–	–	–
Louvain	0.87	0.41	0.46	249	0.54	0.1	0.25	565	0.75	0.06	0.26	7,322
Infomap	**0.96**	**0.94**	**0.99**	17,296	0.74	0.01	0.76	16,976	–	–	–	–
FluidC	0.90	0.18	**0.99**	–	**0.75**	0.02	0.91	–	0.87	0.09	0.89	–
EDCD	–	–	–	–	–	–	–	–	–	–	–	–
SCD	–	–	–	–	–	–	–	–	–	–	–	–
ASOCCA	–	–	–	–	–	–	–	–	–	–	–	–

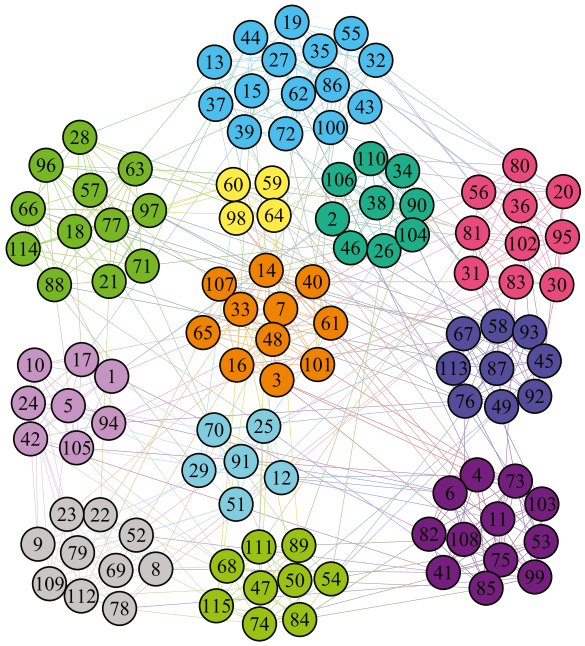

Fig. 6. CDME used on the American football network. Twelve communities were detected.

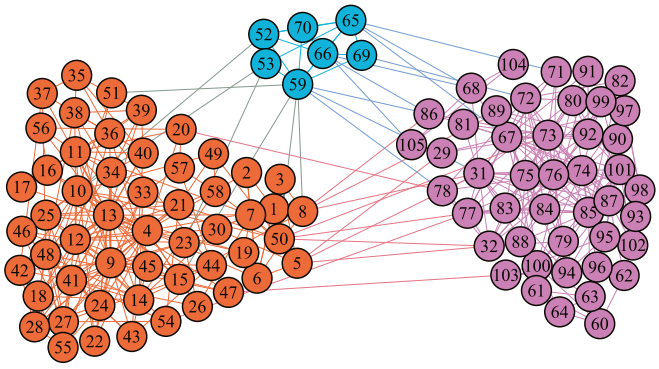

Fig. 7. CDME used on the American politics books network. Three communities were detected.

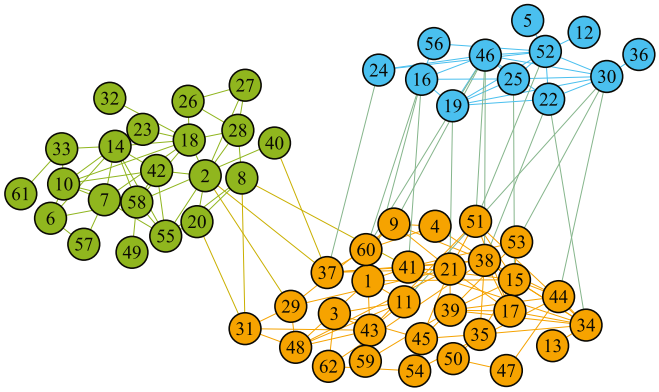

Fig. 8. CDME used on the dolphin network. Three communities were detected.

not get community partition results after running the experiment for longer than 10 days. The LPA, Louvain algorithm, and FluidC algorithm achieved comparable results.

In summary, Infomap and CDME performed very well on the synthetic benchmark networks, where they achieved most of the highest metric scores. On real-world networks, CDME also achieved the best community partition. This is because CDME not

Fig. 9. The top 10 largest communities detected by CDME in Amazon network.

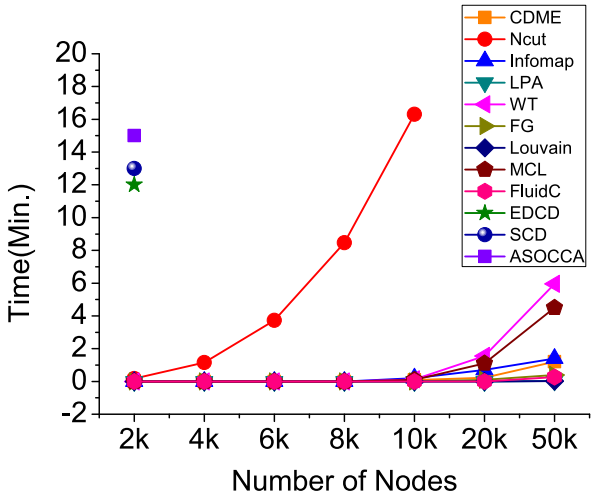

Fig. 10. Runtime of each comparison algorithm on the synthetic benchmark networks with nodes ranging from 2000 to 50,000.

only considers the node-to-node influence, but also the community's influence on it, so CDME can detect high-quality communities. In addition, since CDME works in a local way and only needs to calculate the attractiveness of neighboring nodes, which lend itself to handling large-scale networks.

4.4. Runtime

To evaluate the scalability of CDME with respect to network size, we employed the LFR model to create benchmark networks with different sizes. The number of nodes was varied from 1000 to 1,000,000 and we fixed the average degree to $k = 15$. Figs. 10 and 11 illustrate the running time of each method. We can observe that CDME was faster than Ncut, EDCD, WT, FG, MCL, SCD, and ASOCCA because its time complexity was $O(k^2 \cdot n)$, where the value of k was usually small. Therefore, the CDME algorithm can be used to handle large-scale networks. However, CDME was

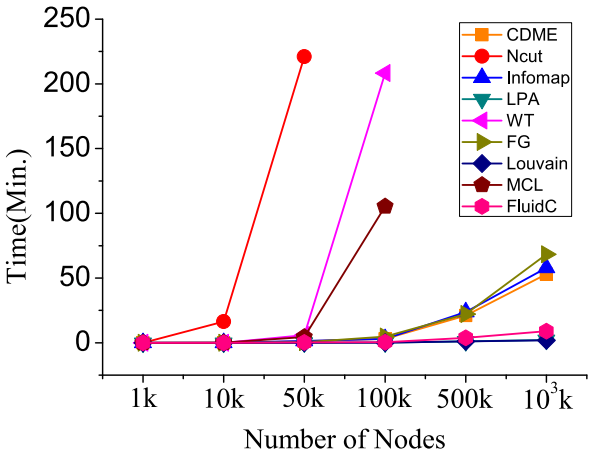

Fig. 11. Runtime of the different algorithms on the LFR benchmark with nodes ranging from 1000 to 1,000,000.

slower than LPA, Louvain, and FluidC. Although these algorithms were much faster than CDME, the quality of communities detected was poor. In addition, LPA and FluidC methods suffered from stability problems.

5. Conclusions

In this study, we proposed a novel and powerful algorithm called CDME to automatically uncover community structure in complex networks. First, We provide a generalized definition of node attraction, which can be used to capture the core groups. Then, we carefully study the dynamic model between nodes in a network, and propose a novel Matthew effect model, which treats the network as a social system and simulates the interaction among nodes. Further, we design a community detection algorithm based on Matthew effect model to simulate the process. Finally, We compared CDME with several representative community detection methods on synthetic and real-world networks. Extensive experiments demonstrated that CDME performed well at detecting communities in small to large networks with high quality, and better than the comparison algorithms considered in the evaluation. On the direction of future work, we plan to extend CDME on dynamical networks and extremely large networks.

Source code of our algorithm is available online at: https://github.com/sunwww168/CDME.

CRediT authorship contribution statement

Zejun Sun: Conceptualization, Methodology, Software, Investigation, Writing - original draft. **Yanan Sun:** Investigation, Formal analysis, Visualization. **Xinfeng Chang:** Validation, Formal analysis. **Qiming Wang:** Supervision, Data curation. **Xuetong Yan:** Software. **Zhongqiang Pan:** Writing review & editing. **Zongping Li:** Project administration, Resources.

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.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61403062 and 41601025), the

Science-Technology Foundation for Young Scientist of Sichuan Province (2016JQ0007), the Science and Technology Research Project of the Science and Technology Department of Henan Province (182102210472), and the Research and Practice Project of "Internet Plus Education" Special Topic in Pingdingshan University (PX-920408).

References

[1] J. Xiang, Y. Zhang, J.-M. Li, H.-J. Li, M. Li, Identifying multi-scale communities in networks by asymptotic surprise, J. Stat. Mech. Theory Exp. 2019 (3) (2019) 033403.
[2] Z. Bu, H. Li, J. Cao, Z. Wu, L. Zhang, Game theory based emotional evolution analysis for chinese online reviews, Knowl.-Based Syst. 103 (Supplement C) (2016) 60–72.
[3] J. Shao, Z. Zhang, Z. Yu, J. Wang, Y. Zhao, Q. Yang, Community detection and link prediction via cluster-driven low-rank matrix completion, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI, 2019, pp. 3382–3388.
[4] Z. Sun, B. Wang, J. Sheng, Z. Yu, R. Zhou, J. Shao, Community detection based on information dynamics, Neurocomputing 359 (2019) 341–352.
[5] J. Shao, C. Böhm, Q. Yang, C. Plant, Synchronization based outlier detection, in: Machine Learning and Knowledge Discovery in Databases, Springer Nature, Berlin, Heidelberg, 2010, pp. 245–260.
[6] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[7] M.E. Newman, Community detection in networks: Modularity optimization and maximum likelihood are equivalent, 2016, arXiv preprint arXiv:1606.02319.
[8] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, C. Faloutsos, Epidemic thresholds in real networks, ACM Trans. Inf. Syst. Secur. 10 (4) (2008) 1:1–1:26.
[9] J. Shao, X. Wang, Q. Yang, C. Plant, C. Böhm, Synchronization-based scalable subspace clustering of high-dimensional data, Knowl. Inf. Syst. 52 (1) (2017) 83–111.
[10] J. Zheng, S. Wang, D. Li, B. Zhang, Personalized recommendation based on hierarchical interest overlapping community, Inform. Sci. 479 (2019) 55–75.
[11] A. Alsini, A. Datta, D.Q. Huynh, J. Li, Community aware personalized hashtag recommendation in social networks, in: Data Mining, Springer Singapore, Singapore, 2019, pp. 216–227.
[12] A. Bahulkar, B.K. Szymanski, N.O. Baycik, T.C. Sharkey, Community detection with edge augmentation in criminal networks, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, IEEE, 2018, pp. 1168–1175.
[13] V. Rosset, M.A. Paulo, J.G. Cespedes, M.C. Nascimento, Enhancing the reliability on data delivery and energy efficiency by combining swarm intelligence and community detection in large-scale WSNs, Expert Syst. Appl. 78 (2017) 89–102.
[14] J. Chen, K. Li, K. Bilal, A.A. Metwally, K. Li, P. Yu, Parallel protein community detection in large-scale PPI networks based on multi-source learning, IEEE/ACM Trans. Comput. Biol. Bioinform. (2018) 1.
[15] Z. Huang, Y. Qiu, A multiple-perspective approach to constructing and aggregating Citation Semantic Link Network, Future Gener. Comput. Syst. 26 (3) (2010) 400–407.
[16] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[17] A. Clauset, M.E.J. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (2004) 066111.
[18] P. P., L. M., Computing communities in large networks using random walks, J. Graph Algorithms Appl. 10 (2) (2006) 191–218.
[19] J. Shao, Z. Han, Q. Yang, T. Zhou, Community detection based on distance dynamics, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, 2015, pp. 1075–1084.
[20] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (2007) 036106.
[21] B. Skrlj, J. Kralj, N. Lavrac, Embedding-based silhouette community detection, 2019, arXiv:Social and Information Networks.
[22] S. Van Dongen, Graph Clustering by Flow Simulation (Ph.D. thesis), University of Utrecht, 2000.
[23] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174.
[24] F. Parés, D.G. Gasulla, A. Vilalta, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés, T. Suzumura, Fluid communities: A competitive, scalable and diverse community detection algorithm, in: Complex Networks and their Applications, Vol. VI, Springer International Publishing, Cham, 2018, pp. 229–240.
[25] R.K. Merton, The matthew effect in science, Science 159 (3810) (1968) 56–63.
[26] A.-L. Barabási, R. Albert, H. Jeong, Mean-field theory for scale-free random networks, Physica A 272 (1) (1999) 173–187.
[27] B.S. Khan, M.A. Niazi, Network community detection: A review and visual survey, 2017, arXiv preprint arXiv:1708.00977.
[28] S. Fortunato, D. Hric, Community detection in networks: A user guide, Phys. Rep. 659 (Supplement C) (2016) 1–44.
[29] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Syst. Tech. J. 49 (2) (1970) 291–307.
[30] Z. Wu, R. Leahy, An optimal graph theoretic approach to data clustering: theory and its application to image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 15 (11) (1993) 1101–1113.
[31] L. Hagen, A.B. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 11 (9) (1992) 1074–1085.
[32] S. White, P. Smyth, A spectral clustering approach to finding communities in graphs, in: Proceedings of the 2005 SIAM International Conference on Data Mining, 2005, pp. 274–285.
[33] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113.
[34] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008) P10008.
[35] X. Chen, J. Li, Community detection in complex networks using edge-deleting with restrictions, Physica A 519 (2019) 181–194.
[36] C.M. Fiduccia, R.M. Mattheyses, A linear-time heuristic for improving network partitions, in: Papers on Twenty-Five Years of Electronic Design Automation, in: 25 years of DAC, ACM, New York, NY, USA, 1988, pp. 241–247.
[37] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. 104 (1) (2007) 36–41.
[38] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Natl. Acad. Sci. 105 (4) (2008) 1118–1123.
[39] W. Hu, Finding statistically significant communities in networks with weighted label propagation, Soc. Netw. 2 (03) (2013) 138–146.
[40] J. Xie, B.K. Szymanski, Community detection using a neighborhood strength driven Label Propagation Algorithm, in: 2011 IEEE Network Science Workshop, 2011, pp. 188–195.
[41] X. Pan, G. Xu, B. Wang, T. Zhang, A novel community detection algorithm based on local similarity of clustering coefficient in social networks, IEEE Access 7 (2019) 121586–121598.
[42] Z. Bu, H. Li, C. Zhang, J. Cao, A. Li, Y. Shi, Graph K-means based on leader identification, dynamic game, and opinion dynamics, IEEE Trans. Knowl. Data Eng. 32 (7) (2020) 1348–1361.
[43] J. Cao, Z. Bu, Y. Wang, H. Yang, J. Jiang, H. Li, Detecting prosumer-community groups in smart grids from the multiagent perspective, IEEE Trans. Syst. Man Cybern. Syst. 49 (8) (2019) 1652–1664.
[44] C. Hennig, B. Hausdorf, Design of dissimilarity measures: A new dissimilarity between species distribution areas, in: Data Science and Classification, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 29–37.
[45] A. Strehl, J. Ghosh, Cluster ensembles - A knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.
[46] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Amer. Statist. Assoc. 66 (336) (1971) 846–850.
[47] L. Hubert, P. Arabie, Comparing partitions, J. Classification 2 (1) (1985) 193–218.
[48] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (1) (2010) 2837–2854.
[49] Y. Zhao, G. Karypis, Criterion Functions for Document Clustering: Experiments and Analysis, Tech. Rep., 2002.
[50] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.