Computing (2023) 105:417–453
https://doi.org/10.1007/s00607-022-01131-z

REGULAR PAPER

Community detection algorithms for recommendation systems: techniques and metrics

Chaitali Choudhary[1] · Inder Singh[1] · Manoj Kumar[2]

Received: 6 April 2022 / Accepted: 29 October 2022 / Published online: 12 November 2022

Abstract

In the early days of social networking, a community was typically viewed as a collection of user pro-files sharing common interests and likes. This community continued to grow by searching for, pro-posing, and adding new members who shared similar characteristics but were unlikely to cause conflict with existing members. Today, things have shifted dramatically. Social networking platforms are not limited to creating identical user profiles: the enormous amounts of data acquired each day have enabled the prediction and recommendation of relationships, habits, and everyday activities such as shopping, dining, and holiday destinations. A long time ago, the topic of community detection drew considerable attention, and communities are shown as clusters within a more extensive network. Community detection has also become a broader problem as graph databases have grown in popularity, as has the diversity of personal information and dynamic behavior. Community detection is a beneficial method for maintaining graph data sets effectively. Community detection has become critical as online communication has grown in popularity, personalization drifts have varied, and internet users' activities have gotten more unpredictable. This article examined all community detection techniques and the parametric data needed to evaluate them.

Keywords Community detection techniques · Metrics · Agglomerative · Divisive · Similarity index · Stream data · Recommendation systems · Principal component analysis (PCA)

✉ Manoj Kumar
wss.manojkumar@gmail.com

Chaitali Choudhary
chaitali.choudhary@gmail.com

Inder Singh
inderddn@gmail.com

[1] School of Computer Science, University of Petroleum and Energy Studies Dehradun, Bidholi, Dehradun, Uttarakhand 248007, India

[2] Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, Dubai Knowledge Park, Dubai, United Arab Emirates

1 Introduction

A community is a natural phenomenon in biological networks, chemical bonds, neural networks, social sites, marketing, etc. Identifying a community is a crucial task and can help in various essential decision-making processes, including which network structure they follow, how they interact with other nodes and their interactions. There are multiple algorithms for the same thing. Still, none of them is the best, guaranteeing the best community detection on various metrics, like the similarity index, inter and intra-cluster density, Modularity, embeddedness, connectedness, etc. Any algorithm must satisfy all the criteria, leading to a highly cohesive group of nodes and having minimal coupling with other nodes or clusters. In this paper, we discussed the community detection algorithm and techniques. Agglomerative and differential methods are the two main types of community detection methods. Edges are added one by one to a network that only comprises nodes using agglomeration methods. From the more potent edge to the weaker edge, edges are added. In differential methods, edges of the whole graph are eliminated one by one. Community discovery is a process that identifies clusters of interacting vertices (i.e., nodes) in a network based on their structural qualities [1, 2]. Numerous community identification methods have been created, including approaches and tools from diverse fields like biology, physics, social sciences, applied mathematics, and computer sciences [3]. However, a single community recognition method would fail to recognize all types of networks [4, 5] due to the enormous diversity of complex networks formed by various activities. Algorithmic biases enhance performance on one kind of network while reducing performance on another network, a natural trade-off. Because networks may be static or dynamic in nature, community identification techniques are highly dependent on their topology. Community finding is much easier in a fixed network than in a dynamic network.

There are several approaches to community detection in static networks [6–9], the majority of which are optimization-based algorithms that seek the optimal solution for the defined objective function [10–13]. In addition to optimization-based techniques, a bottom-up ap-proach based on clustering utilizing correlation coefficients and random walk similarity exist [14–16]. Modularity maximization [17] and spectral clustering [18] are the workhorses for community identification in static networks, respectively. However, the majority of real-world networks are dynamic in nature, and several research has been conducted to get a better understanding of their evolutionary behavior [19–22]. A recent study has also concentrated on the analysis of dynamic communities' time-varying features [23, 24]. Historically, spectral clustering, modularity maximization, and other statistical techniques were employed to detect distinct communities in networks. However, real-world networks, such as social and biological networks, are characterized by multiple community memberships, in which a node is connected to numerous distinct groups concurrently [25, 26]. Considering the constraint of a node belonging to innumerable communities, overlapping community identification techniques are the answer to this issue [27]. Additionally, many methods have been suggested for detecting disjoint and overlapping communities inside a network [27]. Various researchers also worked on community detection with deep learning techniques In order to address experimentation settings, Su et al. [28] summarised the well-known benchmark data sets, evaluation measures, and open-source implementations. Additionally, they point to implementation possibilities in their work and explore the practical applications of community detection in diverse areas. Finally, they provide an overview of future prospects by posing difficult questions in the quickly expanding deep learning field. To describe the state-of-the-art in the area of community detection, Jin et al. [29] built and presented a unified architecture of network community identification techniques. We specifically provide a thorough analysis of the available community identification techniques and propose a novel taxonomy that classifies the techniques into two groups: probabilistic graphical models and deep learning. Huang et al. [30] provided a thorough grasp of community discovery techniques in multilayer networks by comparing earlier research and examining a number of typical approaches. The use of parallel computing, shared memory, and distributed memory on the current community identification approaches was comprehensively covered by Naik et al. [31]. More particularly, the systematic literature review that was done to compile pertinent papers from various digital libraries and grey literature is the main contribution of their study. A community discovery approach based on a deep sparse autoencoder was suggested by Li et al. [32] Prior to obtaining the route weight matrix for the node's second-order neighbours, they first determined the nodes' second-order neighbours. Then, based on the unsupervised deep learning approach, a deep sparse autoencoder may be built to extract the feature matrix that has a higher capacity to describe the network's features. The low-dimensional feature matrix was clustered in order to generate the community structure, and the K-means technique was used.

There can be any number of communities in a particular network, and they can be of various sizes. These qualities make community detection incredibly challenging. However, in the domain of com-munity detection, numerous distinct strategies have been proposed. They can differ in two ways: the procedure that leads to community structure estimation and the estimated communities. This creates a dilemma about how various algorithms should be compared theoretically and practically. The rest of this paper is organized as follows. Section 2 gives the overview of various Metrics used in community detection like Intra and Inter cluster density, Embeddedness, Community size etc. Section 3 presents a technical overview of formal concept analysis for communities. Section 4 overviews the preliminaries and categorization of existing community detection approaches, Sect. 5 discusses application of community detection in recommender system and conclude in Sect. 6.

Systems that provide recommendations to users are crucial in providing them with pertinent information. Social interactions within the community provide a new depth to suggestions. Utilizing communities discovered by analysis of extensive social networks is a difficult procedure that involves many processing phases. The tie recognition step uses this data to formalise the links between individuals and items after gathering enough information to indicate social relationships, interests, preferences, and actions of the users. This concept examines these relationships to find user groups with comparable preferences. This approach is conceptually and practically hard since combining community identification methods in the recommendation process requires many phases. For community-based recommender systems to advance, a number of pressing situations and research issues must be resolved, such as:

- It is advantageous for recommender systems to include more data from supplementary sources or even from inside the social network. For instance, if there is a link between the two domains, an e-commerce firm may use ratings from the book shop to enhance the accuracy of suggestion in the movie store. By comparing user behaviours and input from two or more communities, this method of cross-domain recommendation may be strengthened in order to deduce patterns that are beneficial for the suggestion [33].
- The accuracy of the recommendations is impacted by potential interest shifts, changes in peer dynamics, and the cyclical popularity of things that are shared. For instance, Nepal et al. [34, 35] and Yin et al. [36] weight each contribution in accordance with the premise that more recent interactions are significant than those that happened in the distant past. Dynamic community recognition methods [37] or evolving social graph clustering methods that take into consideration these temporal aspects have largely gone unnoticed up to this point.
- Due to the dearth of substantial datasets including explicit and implicit social signals, it is challenging to acquire realistic and accurate lab-based assessments for evaluating suggestion accuracy. As a result, there is no comparison analysis for determining the accuracy of various community-based RS using the same input information.
- Recent developments in deep learning systems provide new methods for assigning concise representation of structured data from complicated graphs (e.g., graph2vec [38]). Additionally, session-based recommendations and, more generally, taking into consideration the temporal elements of user preferences, are now often achieved using recurrent neural networks [39]. Future community-based recommender systems should logically make use of these technologies to increase the accuracy and comprehension of the data gathered.
- Although session-based recommender systems are a developing research area in the field of recommendations [40], none of the community-based recommender systems surveyed explicitly mention short-term transactional patterns or sessions, identifying significant patterns and the user preference shift from one transaction to the next.
- Users’ aggregate activity is taken advantage of by recommenders to offer suggestions that are very tailored. Statistical biases, such as those caused by the sampling procedure, validity, completeness, noise, or spam, are not taken into account in empirical assessments, which often base their conclusions on samples of the whole population. In certain circumstances, recommenders may magnify these biases, resulting in the phenomenon known as bias disparity. While there have recently been a number of attempts to lessen this problem, research on community-based recommender systems built with societal principles like justice, accountability, and openness in mind is still lacking.

2 Ontological characteristics for communities

2.1 Intra-cluster density

Intra-cluster density is defined as the number of points in the clusters' representation points' nearby [28]. For well-separated clusters, the intra-cluster density is much higher. The intra-cluster density can be defined as follows:

$$intraden(c) = 1/c \sum_{i=1}^{c} \sum_{j=1}^{r_i} density(v_{ij}) where, c > 1 \tag{1}$$

The term density (v_{ij}) is defined as $\sum_{l=1}^{n_i} f(x_l, v_{ij})$ where x_i belongs to ith cluster, and $f(x_l, v_{ij})$ is defined by

$$f(x_l, v_{ij}) = 1 \; if \; ||x_l - v_{ij}|| \leq stdev \; else \; 0 \tag{2}$$

2.2 Inter-cluster density

The term "inter-cluster density" refers to the thickness in the areas between clusters. The density in the space between clusters is meant to be extremely low. The following is a definition:

$$inter_d en\ (c) = \sum_{i=1}^{c} \sum_{j=1}^{c} \frac{||close_r ep\ (i) - close_r ep\ (j)||}{||std_d ev\ (i)|| + ||std_d ev\ (j)||} * density\left(u_{ij}\right) \tag{3}$$

Where close $rep(i)$ and close $rep(j)$ are the nearest pair of representations of the ith and jth clusters, u_{ij} is the middle point between the pair points close $rep(i)$ and close $rep(j)$.

2.3 Embeddedness

The embeddedness metric quantifies how many of a node's near neighbours are its community members. It is defined as the ratio of the targeted node's internal degree k_{int} to its overall degree k [29, 30]. The following equation defines the embeddedness metric:

$$Embeddedness = \frac{k_{int}}{k} \tag{4}$$

The highest value of embeddedness is 1 when a node and all of its neighbors are members of the same community (k = k_{int}) (the core node), and 0 when all of a node's neighbors are members of separate communities ($k_{int} = 0$).

2.4 Community size

An essential feature of community structure is the community size distribution. It has been extensively researched in real-world networks and appears to follow a power law with an exponent β of between 1 and 2. This results in a diverse distribution of community sizes, with many small communities and just a few massive ones. The smallest community size in real-world networks is two, while the maximum community size varies significantly depending on the class and granularity of the simulated system [31, 32].

2.5 Fraction of correctly classified nodes (FCC)

It is adequately categorized when at least half of the nodes in a node's reference community are also present in the same estimated community, according to this criterion. Furthermore, if the estimated community is a fusion of multiple reference communities, all of the affected nodes are deemed to have been misclassified by the algorithm. The total number of adequately categorized nodes is divided by n to normalize the measure, resulting in an expected value between 0 and 1.

2.6 Rand index (RI)

The Rand index (RI) measures the percentage of node pairings for which both the estimated and reference community topologies are consistent with one another. Agreement exists for a given pair of nodes when both nodes belong to the same community or belong to separate communities for both community topologies. The result is that there is a dispute when the nodes belong to the same community in one community structure but belong to two distinct communities in another. The Rand index ranges between 0 and 1, indicating whether the algorithm successfully calculates the community structure. The adjusted Rand index (ARI) is a chance-corrected variant of the Rand index that ranges from -1 (less than chance agreement) to 1 (perfect Agreement) (complete Agreement). A pure random consensus is represented by the number zero.

2.7 The normalised mutual information (NMI)

The Normalised Mutual Information (NMI) was introduced in the context of classical clustering [41] and is used to compare two independent partitions of a same data set by calculating how much information they share. It was developed by Danon et al. [42] for the purpose of evaluating the effectiveness of community identification methods, and has subsequently been adopted by a number of additional writers [43]. The score is 1 if the projected communities perfectly match the reference communities, and 0 if they are fully independent.

2.8 Internal transitivity

The internal transitivity is computed by averaging the traditional local transitivity across its nodes. The connectedness of a node's immediate neighbors dictates its local transitivity. It is defined as the number of connections between neighbors divided by the total number of relationships that would exist if all ties were connected. In other words, it indicates the ratio of existing to potential connections in the neighborhood of a node. Internal transitivity is officially defined as

$$T\,(c) = \frac{1}{n_c}\sum_{i\epsilon c}\frac{2 * l(i)}{k_{int}i[k_{int}i - 1]} \tag{5}$$

Here, i denotes a node, n_c denotes the number of nodes in community $c, l(i)$ is the number of connections between node i's neighbors who are also members of the same community and denotes the internal degree of a node i (as defined previously for the embeddedness). Internal transitivity distributions vary significantly with community size in real-world networks. It grows with the use of the Internet and other communication networks. It grows until it reaches a maximum value in biological and social networks, at which point it begins to decrease.

2.9 Scaled density

The density of a community c is defined as the ratio of the number of connections it has, represented by m_C, to the number of links it might have if all its nodes were linked. In the case of an undirected network, the latter is $n_c(n_c - \mathbf{1})/\mathbf{2}$, where n_c is the community's node count, yielding $\rho = \mathbf{2}m_c/n_c\,(n_c - \mathbf{1})$. In comparison to the network's general density, the community density provides an assessment of the community's cohesion: a community should, by definition, be denser than the network to which it belongs. Scaled density is a variation produced by multiplying the density by the size of the community.

$$\tilde{\rho}\,(c) = \rho(c)n_c = 2m/\,(n_c - 1) \tag{6}$$

If the community under consideration is a tree with $m_c = n_C - 1$ linkages and $\tilde{\rho}\,(c) = 2$. If it is a clique (a fully linked sub-network), then $m_c = n_C\,(n_C - 1)\,/2$ and $\tilde{\rho}\,(c) = n_C$. As a result of the scaled density, it is possible to describe the community's structure. Specific real-world networks, such as the Internet or communication networks, contain fundamentally tree-like communities. On the other hand, for different kinds of networks, such as social and information networks, the scaled density rises in proportion to the size of the community. Finally, biological networks show a hybrid behavior, with small communities being tree-like and larger communities being denser and more like cliques.

2.10 Average density

Between two nodes, the distance is equal to the length of their shortest path. When averaged over all node pairings in a community, it enables evaluating the community's cohesiveness. Small communities (allegedly small-world in real-world networks) imply that the average distance between communities should rise logarithmically with community size. For more prominent communities, the average length grows gradually or even stabilizes for some real-world networks, such as communication networks. A short average distance can be explained by a high density (social), the availability of hubs (communication, Internet), or a combination of the two (biological, information).

2.11 Hub dominance

In terms of community structure, a hub is a node linked to a large number of other nodes within the same community. The hub dominance ratio may be used to assess the existence of a central hub in a community C, which equals:

$$h\left(c\right)=\overset{max}{\underset{c}{}}(k_{int})/(n_c-1) \tag{7}$$

The numerator represents the highest internal degree discovered in C, whereas the denominator represents the highest degree theoretically conceivable given the community size. When at least one node is linked to all other nodes in the community, the hub dominance value equals one. It can be zero only if no nodes are connected, which is improbable in a community. This property's behavior in real-world networks is context-dependent. It is nearly the maximum for all community sizes in communication networks, implying that hubs exist in all communities. Other classes have fewer hubs in their significant communities, which explains why their hub dominance usually declines as community size increases [44].

2.12 Edge betweenness centrality (EBC)

When it comes to networks, the edge betweenness centrality (EBC) may be defined as the number of shortest routes that travel through a particular edge in the network. Each edge is assigned an EBC score based on the fastest ways of connecting all of the graph's nodes. In graphs and networks, the shortest route is the one that covers the least amount of distance between any two nodes. As an example, consider this situation to learn how EBC scores are computed. Take a look at the graph below:

We will now attempt to get the EBC scores for all of the edges in this graph. As this is an iterative process, we have provided an overview of it here: Lists are easy to create:

- One node at a time, we will display the shortest routes between it and the other nodes until we cover every node of the graph.
- We will calculate the EBC scores for each edge based on the shortest routes taken by the edges.

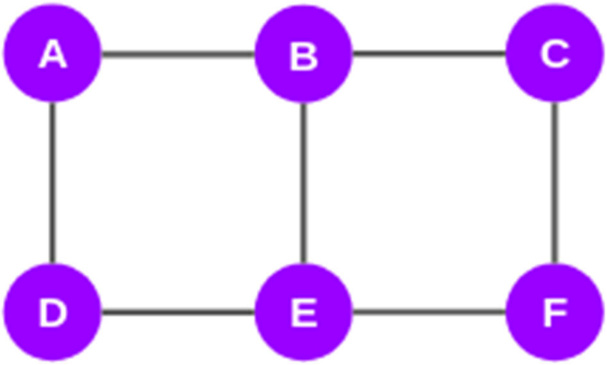

Fig. 1 Example graph for calculating Edge betweenness

- This procedure must be repeated for every node in the graph. As you can see in the graph above, there are six nodes in all. As a result, there will be six iterations of this procedure in all.
- This implies that every edge will get six points. These scores will be tallied together edge-by-edge.
- At the end of the process, the total score of each edge will be divided by two to get the EBC score.

3 Formal concept analysis for community detection

Let G = (N,E) be a graph representing a network where N would be a set of nodes and E is the set of social links between nodes

A clique in an undirected graph G = (V,E) is a subset of the vertex set c V , such that for every two vertices in c, there exists an edge that connects the two. A maximal clique (MC) would be an unextendable one including one more adjacent vertex, that is to say, a clique which does not exist exclusively within the vertex set of a larger clique.

let F = now (V, C, I), the formal context connecting any actor V who is a member of the set C of maximum cliques. I is the binary connection that connects V and C. To locate communities in social networks, formal concept assessment methods are using the Galois lattice based on the context F = (V, C, I).

To locate communities in social networks, formal concept assessment methods are using the Galois lattice based on the context F = (V, C, I).

Let's look at a network with 15 nodes and 32 edges in G(N,E), which produces 4 maximum cliques as an example (a,b,c,d). Calculations based on the formal context K=(G, M, I), where G is a collection of objects, M is a set of characteristics, and I is a binary relation between sets G and M, I G M, may be made. Two subsets A M and B G are defined as sets of qualities shared by the objects in A and B, respectively, for the two sets A G and B M.

The derivation is defined as ’ and can can formally calculated as:

$$A' := a\epsilon M|oIa\forall o\epsilon A(intension) \tag{8}$$

$$B' := o\epsilon G|oIa\forall a\epsilon B(extension) \tag{9}$$

All the subsets of G and all the subsets of M are defined as having a pair of correspondence (’,’), which is a clique that only meets when they do not overlap, by Ali et al. [45]. The overlap is transitive, symmetric, and reflexive. Thus, the overlap represents an equivalence connection. If we concentrate on the diagram, we can see that two nodes of level k overlap if they cross at level k + 1.

4 Community detection algorithms

4.1 For disjoint communities

Identifying communities is a fundamental need to understand the structure and functionality of a social network. The prevalent method of community discovery is to divide the network into discrete groups of members who communicate extensively inside. This is called disjoint community detection, and this method disregards the notion that a person can belong to more than one group.

4.1.1 Traditional methods

Graph partitioning
The graph partitioning problem involves dividing the vertices into g groups of a specific size in order to reduce the number of connections between the groups. The cut size refers to the number of edges that link two clusters. The most well-known method for this kind of graph partitioning is METIS, and there is a decent Python wrapper for the optimized C version (which must be built/installed separate-ly). It accepts input in the form of networkX graphs or essential adjacency lists. Graph partitioning, finite element mesh partitioning, and the creation of fill reduction orderings for sparse matrices are all performed using the serial programmes in METIS. The multilevel recursive-bisection, multilevel k-way, and multi-constraint partitioning schemes created in our lab served as the foundation for the algorithms used in METIS. The three steps of the METIS multilevel method each have a number of algorithms:

- By creating a series of graphs G0, G1,..., GN, where G0 is the original graph and Gi has more vertices than Gj for each value of 0 to N, you may coarsen the graph.
- Create a GN division.
- As you refine the partition with regard to each graph, project it back through the series in the order of GN,..., G0. The refined partition projected onto G0, the last partition calculated during the third phase, is a partition of the initial graph

METIS agrees with a graph with weighted vertices and edges and produces an array of partitions up to the specified number while reducing the weight of the edges being cut. You will still need to decide how many segments to divide your graph. Here in

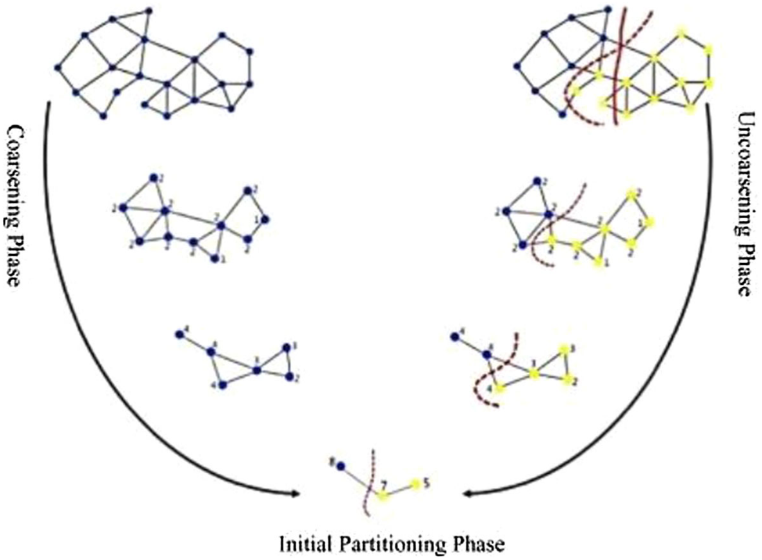

Fig. 2 METIS

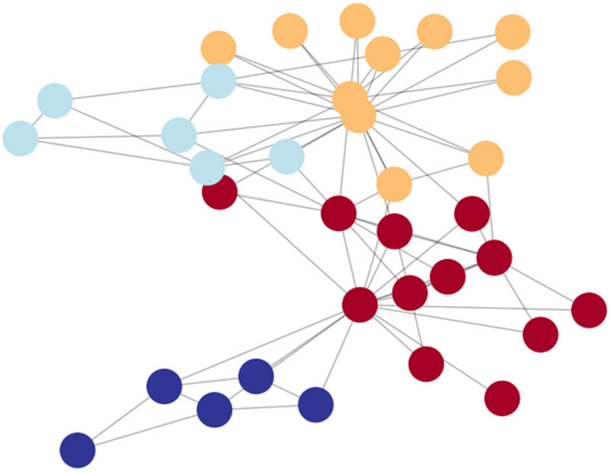

Fig. 3 Graph partitioning example

this paper, we have taken an example of graph partitioning using METIS which can be shown in Fig. 2 below:

Hierarchical clustering
In general, no one knows how a graph's community structure is formed. It is unusual to find information on the number of clusters into which the network is split and other information about the vertices' membership. Clustering techniques, such as graph partitioning, are ineffective in certain in-stances. To begin, appropriate assumptions

about the number and size of clusters must be made, often incorrect. On the other hand, the network may have a hierarchical structure, implying many layers of vertices grouping, with tiny clusters included inside more significant clusters, which are themselves contained within larger clusters, and so on. For example, social networks usually show-case hierarchical structures. In such situations, hierarchical clustering methods [46] may be used, which are clustering approaches that expose the graph's multilevel structure. Hierarchical clustering is widely used in various fields, including social network research, biology, engineering, and market-ing. The definition of a similarity measure between vertices is the first step in every hierarchical clustering method. Following the selection of a measure, the similarity between each pair of vertices, whether connected or not, is calculated. A new n*n matrix X, the similarity matrix, is produced due to this process. The same can also be applied using link measures. Specifically, the clustering procedures are ex-plained in the three stages below:

- Determine the degree of each node in the link network.
- To begin, assign each node to a cluster; then, using the single linkage function, merge the clusters iteratively according to the degrees of the nodes.
- When all nodes belong to a single cluster, stop merging.

The clustering process is then recorded as a dendrogram, including all the hierarchical module organ-ization's information. The similarity value at which the two clusters merge is referred to as the module's strength. It is encoded as the height of the relevant dendrogram branch to offer extra information.

Partitional clustering

Partitional clustering (or partitioning clustering) is a kind of clustering technique that divides observations within a data set into several groups depending on their similarity. The algorithms need the analyst to define the desired number of clusters. The following are some of the most often used partitional clustering algorithms:

- In K-means clustering [27], each cluster is represented by its center or a subset of its data points. K-means is very sensitive to outliers and abnormal data points.
- Clustering of K-medoids, or PAM (Partitioning Around Medoids, Kaufman et al. [47, 48]), in which each cluster is represented by one of the cluster's objects. When opposed to k-means, PAM is less prone to outliers.
- CLARA (Clustering Big Applications) is an extensive data set adaptation of the PAM method.

Spectral clustering

Spectral clustering is an exploratory data analysis(EDA) method for partitioning large multidimensional data sets into clusters of comparable data in lower dimensions. The overarching goal is to classify all unstructured data pieces into various categories based on their distinctiveness. "Spectral clustering is a widely used technique for multivariate statistical analysis." 'Spectral Clustering employs a connectivity-based method to clustering,' which identifies communities of nodes (i.e., data points) that are linked or directly next to one another in a graph. After that, the nodes are mapped to a low-dimensional space readily segmented into clusters. Spectral clustering extracts

Table 1 Comparitive analysis of traditional methods

Algorithm	Advantage	Disadvantage
Graph partitioning	Successful for mesh-based graphs and excellent control for communication	More expensive and edge cut model for communication volume
Hierarchical clustering	Easy to implement, good visualization and no need to specify clusters in advance	Slower, decision regarding merge and split points needs to be taken earlier. Also doesn't work with noise and outlier
Partitional clustering	Simple, scalable and suitable for clearly defined clusters	Doesn't perform in high dimensional values and need to define cluster parameters in advance
Spectral clustering	Good mathematical foundation and works good in clusters with relations (similarity)	Cant work with noisy data and expensive for large datasets

information about the network or data set from specific matrices' eigenvalues (spectrum) (i.e., Affinity Matrix, Degree Matrix, and Laplacian Matrix). Spectral clustering techniques are visually appealing, simple to apply, and relatively quick, particularly for a few thousand rows of sparse data sets. Spectral clustering approaches data clustering as a graph partitioning issue, making no assumptions about the shape [46].

4.1.2 Modularity based

Extremal optimization
The approach is prompted by recent breakthroughs in our understanding of out-of-equilibrium occurrences as seen through the lens of self-organized criticality, a concept designed to account for emergent complexity in physical systems. Extremal optimization is a strategy that gradually replaces very uncomfortable variables in an unsatisfactory solution with new, random ones. Due to these dynamics, large, avalanche-like oscillations in the cost function n self-organize, effectively scaling barriers and allowing for the investigation of local optima in distant parts of the configuration space without the need for parameter adjustment. Extremal optimization combines approximation techniques influenced by equilibrium statistical physics, such as simulated annealing, by using models to mimic the dynamics of granular media, evolution, or geology. This is just one instance of systematically bringing new insights into non-equilibrium processes to complex optimization issues. This technique is broadly applicable and is comparable with - and even superior to - more sophisticated general-purpose heuristics on testbeds of restricted optimization problems involving up to 105 variables, including bi-partitioning, coloring, and satisfiability. Analyzing an appropriate model accurately will only forecast the method's only free parameter in line with all experimental findings [49–51]. Although extreme optimization is primarily used to find non-overlapping communities, Jing et al. [52] developed an EO-based technique for optimizing the modularity function of networks and detecting their overlapping communities. A modified modularity function is designed to address the issue of res-

olution limits. Secondly, they defined the local fitness functions of nodes, which can be linearly concatenated to produce the modularity function. Thirdly, a new mutation operator is developed to efficiently and effectively explore the solution space.

Spectral optimization
Examining the eigenvalues and eigenvectors of a specific matrix may help enhance Modularity which can be calculated by the formula below:

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m} \tag{10}$$

Where A is the adjacency matrix, m is the total number of edges in the graph, and $k_i,\ k_j$ represents the degree of vertices. Consider the vector s, which represents any partitions of the graph that fall into two clusters $\mathcal{A}$ and $\mathcal{B}$: $s_i = +1$ if vertex i falls in cluster $\mathcal{A}$, $s_i = -1$ if vertex i fall in cluster $\mathcal{B}$. Modularity can be evaluated as:

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta\left(c_i c_j\right) = \frac{1}{4m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})(s_i s_j + 1) \tag{11}$$

$$= \frac{1}{4m}\sum_{ij} B_{ij} S_i S_j = \frac{1}{4m} s^T B s \tag{12}$$

Standard matrix products were specified by the final expression. The eigenvectors of the modularity matrix B may be used to decompose the vector S. So, the final equation derived is as follows:

$$Q = \frac{1}{4m}\sum_{i} a_i u_i^T B \sum_{j} a_j u_j = \frac{1}{4m}\sum_{i=1}^{n}(u_i^T \,.\, s)^2 \beta_i \tag{13}$$

Where β_i is an eigenvalue of B corresponding to eigenvector u_i. This implies that by substituting the modularity matrix for the Laplacian matrix on bipartitions, one may maximize modularity [72].

Greedy techniques
Newman's greedy approach was the first to optimize Modularity [73]. It is a hierarchical agglomerative clustering technique in which groups of vertices are progressively merged to create bigger communities, resulting in enhanced modularity. The procedure begins with n clusters, each consisting of a single vertex. The process does not begin with any edges; they are added one by one as the method proceeds. However, since we are looking for the maximum Modularity on the space of the complete graph's partitions, the Modularity of the partitions investigated throughout the procedure is always computed using the graph's entire topology. The number of groups in the graph is decreased from n to n – 1 by adding the first edge to the collection of disconnected vertices, resulting in a new network partition. Compared to the previous arrangement, the edge is chosen such that this partition maximizes (or reduces) Modularity. In the same way, the additional edges are added. If the addition of an edge has no effect on the

Table 2 Survey on various community detection techniques used

Paper details	Techniques used	Outcomes	Shortcomings/future scope	Remarks/datasets
Maity et al. [53]	Clique percolation method (CPM)	Unveiled overlapping communities and can distinguish random graphs from the graphs having community structure	Only suitable when subgraphs are fully connected in a given graph of a network Communities are detected for networks	using the proposed approach for different values of k the clique size
Arab et al. [54]	Hybrid merging of sub-communities	Proposed a bottom-up community detection method. Starting with fine-grained communities and merging small preliminary communities is done in a hybrid way to maximize two quality functions	Its low cost and accuracy need to be tested on possibly extensive networks	We conducted our experiment on three different sets of real and artificial networks (http://www.cc.gatech.edu/dimacs10/archive/clustering.shtml; YanandGregory 2012; Lancichinetti etal. 2008)
Chintalapudi et al. [45]	Fast Greedy, Label Propagation, Multilevel, Optimal Modularity, Spinglass, Walktrap, Infomap	Reviews disjoint community detection algorithms in undirected and directed social networks	Algorithms for overlapping community detection in real-world networks were not focused	Three undirected networks with 100, 1000, and 10,000 nodes are generated using Erdos-Renyi random network generator with 500, 5000, 50,000 edges, respectively

Table 2 continued

Paper details	Techniques used	Outcomes	Shortcomings/future scope	Remarks/datasets
Li et al. [55]	Principal component analysis	Overcomes time complexity problem of NMF techniques and has less time complexity than classical algorithms	Data can get arbitrarily re-scaled if principal components are chosen using a correlation matrix	
Dolghik et al. [56]	Graph-based partitioning on SAN	They carried out a user-centric evaluation to estimate the user-perceived quality and usefulness of the recommendations	Improvement of the novelty of recommendations as well as the enhancement of the user-perceived quality and to compare our approach with other systems (e.g., Last.fm) via A/B testing	The evaluation process is based on several artificial metrics (mean absolute error, root mean square error etc.)
Ferreira et al. [57]	Graph based method	Proposed a technique for time series clustering using community detection in complex networks	The same idea can be extended to multivariate time series clustering	Applied network science to temporal data mining using e 45-time series data sets from the UCR repository
Yoldemir et al. [58]	stable overlapping replicator dynamics (SORD)	SORD identifies overlapping subnetworks from weighted graphs with statistical control over false node inclusion	The proposed methodology can be applied to real-time datasets for accurate performance measures	They are applied on synthetic datasets spanning a wide variety of network configurations

Table 2 continued

Paper details	Techniques used	Outcomes	Shortcomings/future scope	Remarks/datasets
Ghorbani et al. [59] 7 Statistical Inference based methods (SBM) and Bayesian Inference	Used to model and analyze accurate graphs such as social networks. DBOCD overcomes the drawbacks of BNMTF	It can be applied to detect overlapping communities	The experimental results on the synthetic and natural data showed that DBOCD could outperform other popular methods in different types of community evolution	
Liu et al. [60]	NMFGR	Better clustering effect than NMF	It does not ensure sparseness of factors	Combined these two similarity metrics to a unified metric which may contain some complementary topological information of community structure of the network
Nguyen et al. [61]	Louvain method	Proposed CD-Clustering which builds an unweighted graph and detects highly cohesive groups of nodes precision and recall in most the cases	It can be used to reduce the complexity of CD-Clustering for better scalability	real categorical datasets were used to show that their method outperforms the existing initialization methods for k-nodes in terms of accuracy, using a fast community detection technique

Table 2 continued

Paper details	Techniques used	Outcomes	Shortcomings/future scope	Remarks/datasets
Zhang et al. [62]	Clique Percolation Method	They proposed a weak-CPM for overlapping community detection in large-scale networks	To investigate extended versions of W-CPM by considering the W-CPM as a framework	Results demonstrated the performance of the proposed method over six popular overlapping community detection algorithms
Hajiabadi et al. [63]	Internal and External association to Detect Communities (IEDC)	IEDC algorithm based on a primary node-based criterion via each node's internal and external association degree computation to detect communities	To perform a theoretical analysis of the proposed approach via the probabilistic method and to study the other properties of the networks under study, such as core-periphery based on the IEDC	Tested the proposed method on extensive simulation experiments with two artificial generative procedures, the LFR, and the MMSB procedures
Guerrero et al. [64]	Genetic Algorithm	Suggests using competitive genetic algorithms to rapidly detect any number of community structures in complex grid networks	To study how partitions obtained with community detection algorithms could provide helpful contingency analysis and islanding strategies	Shows good performance of genetic algorithms to detect communities in power grids

Table 2 continued

Paper details	Techniques used	Outcomes	Shortcomings/future scope	Remarks/datasets
Bello-Orgaz et al. [65]	Edge encoding	Uses a phnotype-type encoding based on the edge information, and a new fitness function to optimize two classical objectives in CDP	To design a self-adaptive version of the algorithms for selecting the optimization criteria used as fitness function during the own evolving process	Dataset collection of real-world networks providing the ground truth and publicly widely used networks to compare the algorithm's performance against alternative community detection approaches
Hui et al. [55]	HICODE (HIdden COmmunity DEtection)	HICODE uses an existing community detection algorithm as the 'base' method. Iteratively weakens the structure of detected layers to reveal the hidden structure and then applies a refinement procedure	It is also possible to apply our ideas for other network mining tasks such as link prediction or maximum influence propagation	Evaluated different variants of HICODE on other reducing methods and different base algorithms and show that ReduceWeight performs the best in general
Moscato et al. [66]	Game Theory	Proposed an algorithm based on Game Theory for Community Detection in Online Social Networks	It can be extended for multimedia features	The evaluation was carried out on Microsoft Azure cloud platform using a Virtual Machine in which they used a technological stack based on graph-tool

Table 2 continued

Paper details	Techniques used	Outcomes	Shortcomings/future scope	Remarks/datasets
Cruz-Lovera et al. [67]	Partitional Clustering	With these techniques, it is possible to determine the main areas of research activity and identify the structures of the collaboration network in the field of renewable energy. The work results show that the literature in this field has substantially increased during the last ten years	The study of the implementation of energy policies in each country, the energy savings achieved, and the relationship with its scientific production on the subject, quantitatively and qualitatively, would be the subject of future work	The dataset was of world scientific publications of renewable energy indexed in the Scopus database to study the most critical areas of investigation in energy saving
Wu et al. [68]	Deep learning	Obtained higher Modularity than other deep learning methods	Not applied for dynamic community detection, so the time-series models are required to extract the Spatio-temporal features for arbitrary social networks	Can effectively detect high-quality communities for each practical dataset, especially for the graph with obscure subspace structure
Jain et al. [69]	Louvain method	Communities are determined using the Louvain method, and choices made by an expert in a particular community are served as recommendations to others	Not considering user-specific characteristics. Instead, taking the choices made by an expert in a particular community	The dataset used in this paper was collected from the music streaming service Deezer in November 2017

Table 2 continued

Paper details	Techniques used	Outcomes	Shortcomings/future scope	Remarks/datasets
Malhotra et al. [70]	Label propagation algorithm (LPA)	They proposed four variants based on LPA to overcome the unexpected community allocation problem generating better clusters by utilizing link strength and node attribute information to enhance the quality of detected communities	Future work could extend the existing paradigm to detect overlapping communities where each node can associate itself with more than one community like in real-world systems	Proposed variants require no parameter for categorical attributes and only one parameter for continuous features. Finally, the best results are obtained by Variant I having a maximum Normalized Mutual Information score of 0.86, 0.88 on the two synthetic networks
Souravlas et al. [71]	deep learning techniques for community detection	Provided a survey on the most recent advances in community detection using deep learning strategies	Deep learning techniques can be used with various other methods for community detection	

partition, and the edge is located inside one of the previously defined clusters, Modularity is unaffected. The technique identifies n divisions, each with a unique number of clusters ranging from n to 1. In this subgroup of partitions, the most outstanding value of Modularity comes close to the algorithm's modularity maximum. The variation Q of Modularity indicated by merging any two communities in the running partition must be calculated at each iteration step to choose the best merger. Since merging communities with no edges can never raise Q, only pairs of communities connected by edges must be considered, which cannot exceed m. This part of the calculation requires a time O(m) since each Q may be computed in constant time. After deciding which communities to merge, the running partition's matrix e_{ij}, which represents the percentage of edges between clusters I and j, must be up-dated (required to compute Q), which may be done in the worst-case time O(n). Because the method takes n-1 rounds (community mergers) to complete, its complexity is O((m + n)*n), or O(n^2) on a sparse graph, allowing for clustering analysis on considerably bigger networks than the Girvan and Newman algorithm (up to an order of 100000 vertices with current computers). Clauset et al. pointed out in a subsequent article [74] that Newman's algorithm's updating of the matrix e_{ij} includes a significant number of pointless operations due to the adjacency matrix's sparsity. This process may be done more efficiently by using sparse matrix data structures such as max-heaps, which reorganize the data into binary trees. The sparse matrix of modularity variations Q_{ij}, a max-heap containing the most significant components in each row of the matrix Q_{ij}, as well as the labels of the associated communities, and a simple array whose elements are the sums of the elements in each row of the old matrix e_{ij} were all preserved by Clauset et al. These three data structures, which update considerably quicker than Newman's approach, may be used to achieve modularity optimization. The approach has an O(m*log n) complexity, where d is the depth of the dendrogram reflecting the successive divisions found during the algorithm's execution, which grows exponentially with log n for networks with a strong hierarchical structure. The technique then runs in O(n*log n) time for such networks, allowing for analyzing the community structure of extensive graphs with up to 106 vertices. Their greedy's optimization is presently one of the few methods to estimate maximum modularity on such huge networks. This greedy optimization of modularity results in the rapid formation of big communities at the cost of tiny ones, often resulting in low modularity maximum values. Danon et al. [42] proposed that the modularity variation Q caused by the merging of two communities be normalized by the percentage of edges incident on one of the two communities to favour tiny clusters. This technique produces greater modularity optima than Newman's original formula, mainly when communities are highly dissimilar in size. Wakita and Tsurumi [75] observed that, as a result of the bias toward large communities, Clauset et al. quick technique is inefficient since it results in severely imbalanced dendrograms for which the relation d log n does not hold. As a consequence, the technique often executes in its most complicated state. To address this problem, they proposed a modification in which the community merger that results in the highest value of the product of the modularity variation Q times a factor (consolidation ratio), which peaks for communities of comparable size, is sought at each step. Thus, a trade-off exists between increased modularity and community balance, in return for a large speed advantage that enables the analysis of systems with up to 107 vertices. Interestingly, this adjustment often leads in greater mode maxima than those seen with the Clauset et al. versions on large social networks. Girvan and Newman [72, 73, 76] offer one of the first methods for small network community discovery. It is a divisive (or top-down) technique for identifying and gradually removing edges from a network through the edge betweenness score. It is equal to the shortest routes between vertex pairs that go along the edge in question. Following a greedy paradigm, the algorithm repeatedly evaluates the betweenness of all edges in the network and deletes the one with the most significant score. Its fundamental concept advocates for the elimination of boundaries between communities.

Genetic algorithms
Genetic algorithms were utilized to improve modularity. A conventional genetic algorithm generates a collection of alternative solutions to a problem, numerically represented as chromosomes, and an objective function for optimizing the space of possible solutions. The objective function determines the chromosomes' biological fitness. Typically, one begins with a randomly generated collection of candidate solutions and gradually adjusts them using approaches inspired by natural chromosomal processes, such as point mutations (unexpected changes in specific regions of the chromosome) and crossing over (generating new chromosomes by merging parts existing chromosomes). The fitness of the most recent pool of candidates is then determined, with the most fit chromosomes having the highest chance of survival in the following generation. After several iterations, only solutions with a high fitness score survive. Tasgin et al. [77] use partitions to represent chromosomes and Modularity to represent the fitness function. Tasgin et al. obtained findings of comparable quality using a greedy modularity optimization on Zachary's karate club [78], the college football network [79], and the Girvan and Newman benchmark. Liu et al. [80] also used genetic algorithms. In this example, the most remarkable modularity partition is produced by executing sequential bipartitions of the graph, with each bipartition selected using a genetic algorithm on each subgraph (starting with the original graph itself) that is considered isolated from the rest of the graph. A bipartition is acceptable only if it increases the overall modularity of the graph.

Simulated annealing
Simulated annealing is a stochastic method for locating 'low-cost' configurations without being stuck in 'high-cost' local minima. This is accomplished via computed temperature T. When T is large, the system may explore arrangements with a high cost, while when T is small, the procedure only explores areas with a low price. The system steadily falls into deep minima by beginning with a large T and progressively lowering it, ultimately overcoming minor cost barriers. The goal of finding communities inside a network is to maximize the Q, where Q is Modularity which is defined as a scale value between -1 and 1 that measures the density of edges inside communities to edges outside communities. Larger values of Q indicating stronger community structure. The traditional simulated annealing technique (SA) [81] randomly selects a node and moves it to a chosen community at each stage. The community may be any of the existing communities, including the one in which the node currently resides or a brand-new community with no nodes.Numerous approaches based on SA for

Table 3 Comparitive analysis of Modularity based algorithms

Algorithm	Advantage	Disadvantage
Extremal optimization	Less memory usage with usage of nature inspired optimization techniques. Parallel speedup, High load balancing, less execution time with a smaller number of migrations.	Low scalability with high response time
Greedy techniques	Easier to implement and require less computation	Don't reach global optimization level
Genetic algorithms	Easy to write, understand and implement and easy identification of errors	Time consuming, and difficult to put in algorithm
Simulated annealing	Cant find exact optimal solution and escapes local maxima	Very sensitive to input parameters

detecting communities have been proposed in recent years [82, 83]. However, the SA approach suffers from three drawbacks that contribute to its lengthy run time. To begin, it completely optimizes Q. Second, it begins at a rapid rate. Third, only one node is moved at a time. On this basis, Hu et al. proposes a fast simulated annealing (FSA) approach for detecting communities. He et al. [84] first partitioned the community according to a similarity measure and then began optimizing the Q at a low temperature. Each step involves the transfer of a component which contains set of links and nodes, not simply a single node.

4.1.3 Dynamic algorithms

Random walk
A random walk is a technique that may be used to discover communities inside a network; in other words, when a random walk is employed, it scans the nodes in a series of steps; it starts with the initial node and proceeds to surrounding nodes using a random procedure. A random walk is a technique for discovering communities inside a network; in other words, it scans the nodes in phases; it begins with the starting node and goes to nearby nodes using a random process. The fundamental concept of [85] is to conduct brief random walks, assuming that the nodes visited during the same walk belong to the same community. During a walk, the next node to visit is one of the visited node's neighbors, which is selected at random. To begin, a similarity matrix S is constructed to aggregate the walks; each item S[i][j] represents the degree of similarity between nodes I and j, with all entries set to zero. Each node in the network is used as a starting point for a random walk once. A user-specified number of steps are taken from that node throughout the network. The next node is selected probabilistically from all neighbors (a node may be visited any number of times during a walk). As evidence of community membership, the nodes visited during the journey are recorded in set C. The entries in S are increased after each walk to match the nodes in C. The number of steps may be determined by a graph-theoretic feature (for example, diameter or number of nodes) or by the user. When all the walks have been completed, each item in the

matrix represents the frequency with which two nodes appeared on the same route. A higher number indicates a greater likelihood of belonging to the same community.

Algorithm 1 Random Walk Algorithm [87]

```
Require: Length of random walks
Ensure: Community graph
for $i = 1....n$ and $j = 1....n$ do
    S[i][j]=0
end for
for each node startnode=1........n do
    i=startnode
    C=startnode
    for number of steps h=1,.......num-steps do
        Randomly select nextnode from neighbors(i)
        C=C U nextnode
        i = nextnode
    end for
    for each node $i \in C$ and each node $j \in C$, $i \neq j$ do
        S[i][j] +=1
    end for
end for
```

Narges et al. [87] proposed an algorithm that aims to find communities so that modularity factor increases; for this goal, random walks with random local search agents are combined.

Instant optimal

In this method, the communities identified at time t must be the most relevant communities in light of the network's condition at that moment. The two-stage technique is the most often used way for resolving this issue: Lists are easy to create:

- Detect static communities on a snapshot-by-snapshot basis.
- For each snapshot, compare the identified communities to the communities discovered in the preceding one.

The earliest methods proposed were using this approach. However, it is losing popularity due to in-stability problems. This approach is notably used by Hopcroft et al. [88], Palla et al. [89], Wang et al. [90], Rosvall and Bergstrom [91], Chen et al. [92], Greene et al. [93], Dhouioui and Akaichi [94]

Temporal trade-off

Communities identified at time t represent a trade-off between the best communities associated with the network at that moment and the history of communities discovered by this approach. There are two types of such algorithms: those that update the communities existing at t-1 in response to net-work changes between t and t-1 (implicit trade-off), and those that define a quality function explicitly as a trade-off between a quality function for communities at t (e.g., Modularity) and similarity be-tween partitions at t and t-1 (e.g. Normalized Mutual Information NMI). The technique is typically comprised of three steps: Numbered (ordered) lists are easy to create:

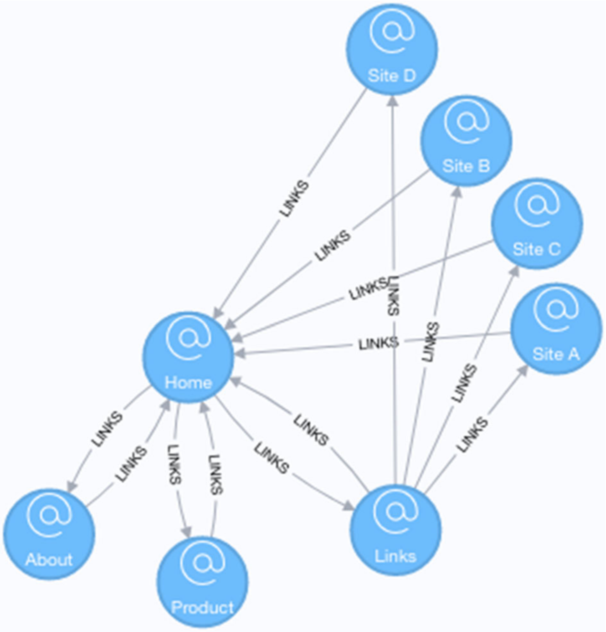

Fig. 4 Random Walk algorithm, using a simple graph [86]

1. On the first snapshot, detect static communities.
2. The network at t + 1 and the communities at t, detect communities on snapshot t + 1.
3. Return to step 2 if not all snapshots have been processed.

Cross-time communities
This method examines all stages of evolution simultaneously. Single community detection is performed, taking into account all of the network's periods in a single pass and giving a single decomposition. For the snapshot example, this procedure is shown in Fig. 4.

Girvan Newman
This algorithm is mainly concerned with divisive techniques. These techniques have received little attention in the prior literature, whether in social network theory or elsewhere, but they seem to have many potentials. This technique starts with the network of interest, identifies the least identical linked pairs of vertices, and then delete the edges divisively. Then splits the network into smaller and smaller components frequently, and may halt the process at any point and consider the features to be the network communities. The process may be seen as a dendrogram, which depicts the network's progressive splits into smaller and smaller clusters. This method is similar to the cross-time communities approach, but it takes a different philosophical perspective. Rather than searching for the most weakly linked vertex pairs and seeks for the network edges that are most "between" other vertices, implying that the edge is responsible for

connecting many pairs of others in some way. In terms of similarity, such edges do not have to be weak.

4.1.4 For overlapping community detection

Clique percolation method
The most well-known method for identifying overlapping communities in networks is clique percolation. The communities are constructed using k-cliques, complete subgraphs with k vertices. It is based on the notion that cliques are more likely to develop from densely linked interior edges than sparsely connected exterior edges. It is based on the idea that cliques are more prone to develop because of the high density of people on the internal borders of society. On the other side, it is improbable that intercommunity edges form cliques: this notion was previously included in Radicchi et al. [95] divisive's approach. Palla et al. [31]defined a complete network with k vertices as a k-clique. Take note that a k-clique is distinct from the n-clique often employed in social research. If a clique could travel through a graph in any manner, it would very certainly get stuck inside its initial community since it would be unable to pass through the bottleneck created by the intercommunity edges. Palla et al. developed a variety of ideas to carry out this notion. If two cliques share k 1 nodes, they are neighboring. A k-clique chain is formed by joining neighboring k-cliques. If two k-cliques are part of a k-clique chain, they are linked. Finally, a k-clique community is the biggest connected subgraph produced by joining a k-clique with all other k-cliques. The k-clique community is a massive component of all neighboring k-cliques linked by a k-clique series. Additionally, top graph clusters (Macropol et al. [96]), SVINET (Gopalan et al. [97]), and label propagation algorithms fall within this group (Raghavan et al. [98]). This method suffers from the same flaw as Radicchi et al. [43], in that it expects a high number of cliques in the graph. Consequently, it may fail to produce meaningful covers for graphs with just a few cliques, such as technical networks and specialized social networks. However, if many cliques are present, the technique may create a trivial community structure, such as a cover consisting of the whole graph as a single cluster in this instance. A more fundamental issue is that the approach looks for subgraphs that "contain" numerous cliques, which may be very different objects than communities, rather than natural communities, which would be compatible with the widely accepted concept of dense subgraphs (for example, they could be "chains" of cliques with low internal edge density, as opposed to communities). Another major problem is that, similar to leaves on trees, many vertices in current networks are excluded from communities. It's conceivable to envision some kind of post-processing method that would enable them to be included in communities, but this would need additional criteria outside of the framework that inspired the approach. Furthermore, it is not immediately clear which value of k should be utilized to identify critical structural relationships. Finally, the parameters for determining the weighted graph threshold and the definition of directed k-cliques were selected somewhat haphazardly.
Link graph and Link partitioning
By dividing a network's connections into communities, one may discover overlapping communities for the nodes by noting that a node belongs to the communities of the

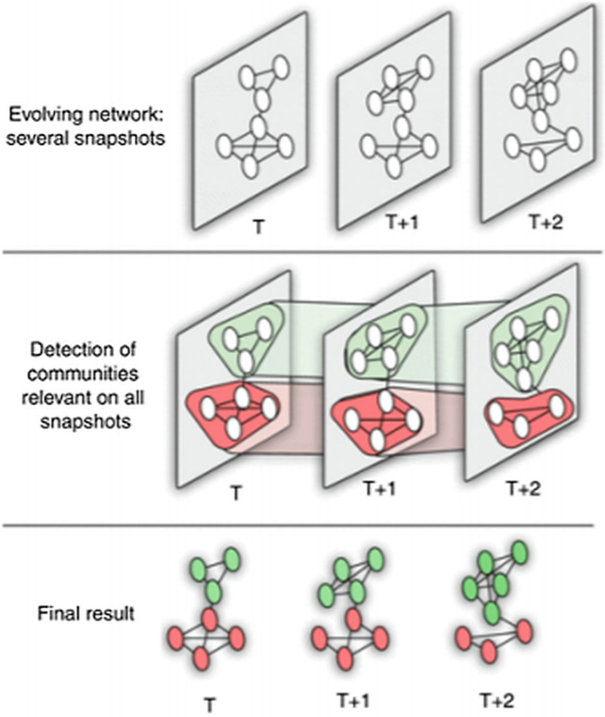
Evolving network: several snapshots
T
T+1
T+2
Detection of communities relevant on all snapshots
T
T+1
T+2
Final result
T
T+1
T+2

Fig. 5 Cross-time dynamic community detection method [30]

Table 4 Comparative analysis of dynamic algorithms

Algorithm	Advantage	Disadvantage
Random walk	Small load balancing and attains local load balancing	Variable performance
Instant optimal	Simple and easy to modify	Sensitive to outliers
Temporal tradeoff	allow to cope with the instability problem that affects Instant-optimal ones	Not easily parallelizable for the community detection
Cross time communities	This class of algorithms does not suffer from problems of instability and community drift that affect previous ones	not able to handle on-the-fly/real-time community detection
Girvan Newman	Appropriate speed and the ability to solve problems with high dimensions	Can only give hierarchical decomposition

links it is connected. Using this toy example in Fig. 6, a meaningful partition is defined as dividing the connections into two groups of equal importance (straight blue lines and the dashed red lines). Since it is located at the interface between both link communities, the central node is considered to be a member of both communities in this scenario.

Few other methods have been suggested to identify overlapping communities of nodes, such as [31, 43, 100], which are described below. In other words, communities are defined as a division of the connections rather than as a partition of the nodes [99, 101]. Therefore, a node may have links to multiple communities and, as a result, it may be considered a member of several communities. The center node is a straightforward example in a Bow Tie network, as seen in Fig. 5. Suppose various kinds of connections link the nodes of a network. In that case, this link partition method should be particularly effective when the nodes are heterogeneous, and the links are highly homogeneous, as is the point in this study. In the instance of the social network described above, this would occur when an individual's friendship network and work network only had a very tiny amount of overlap.

Fuzzy detection
Luo et al. [102] analyzed The network structure from the composition of fuzzy relations. A new method based on fuzzy relations is presented for non-overlapping community identification, dubbed CDFR (Community Detection by Fuzzy Relations). The basic concept of CDFR is to identify the NGC node (Nearest node with Greater Centrality) for each node and then calculate the fuzzy connection between them using this information. The community to which a node belongs is therefore dependent on the NGC node that it is connected to. In addition, a decision graph will be created to assist the identification of communities of interest. Experiments on artificial and real-world networks confirmed the efficiency and superiority of the CDFR algorithm developed by the researchers. Sun et al. [103] used fuzzy transitive rules to reveal community structure in complex networks. By varying the criteria, their approach can ultimately partition the network into several communities with varying resolutions. The findings indicated that their system based on Rule I performs better when the similarity between nodes belonging to the same group is higher than the similarity between nodes belonging to different groups, which is precisely the reverse of Rule II. When mu is near 0.8, our approach outperforms specific state-of-the-art algorithms. This technique shed fresh light on network partitioning and community discovery. A popular method for community identification in Social Network Analysis (SNA) repeatedly employs three phases: spectral mapping, clustering (using either the Fuzzy C-Means or the K-Means algorithms), and modularity calculation. Regardless of its efficacy, this technique is inefficient. Utilizing Graphics Processing Units is a viable solution to this issue. Additionally, since this algorithm is iterative, the new dynamic parallelism technology lends itself to a beautiful solution. Al-Ayyoub et al. [43] presented different novel GPU implementations of both versions of the algorithm: Hybrid CPU-GPU, Dynamic Parallel, and Hybrid Nested Parallel. These new implementations vary in their reliance on the CPU and their use of dynamic parallelism. We conduct a comprehensive series of experiments to compare these implementations in various con-texts. The findings indicate that the Hybrid Nested Parallel implementation speeds up about two or-ders of magnitude.

Statistical inference-based methods
Dao proposed a stochastic block model (SBM), a math-ematical model composed of random blocks Riolo et al. [104]. Holland et al. [105] first described the SBM (1983). They used Riolo et al. implementation, employing a Monte Carlo sampling method to optimize a Bayesian posterior probability distribution across potential network com-

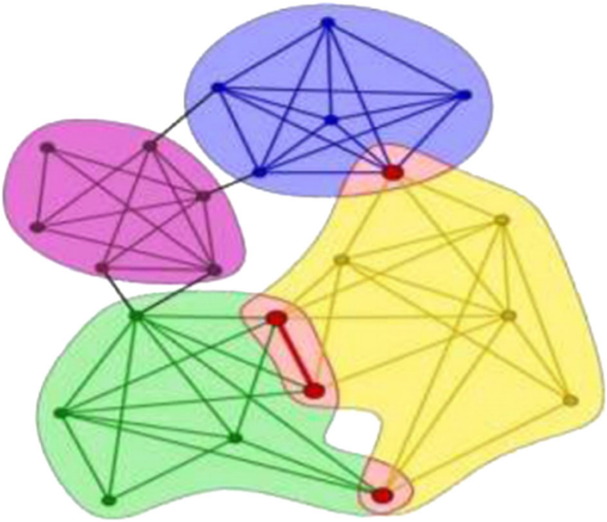

Fig. 6 Clique perlocation method for community detection [32]

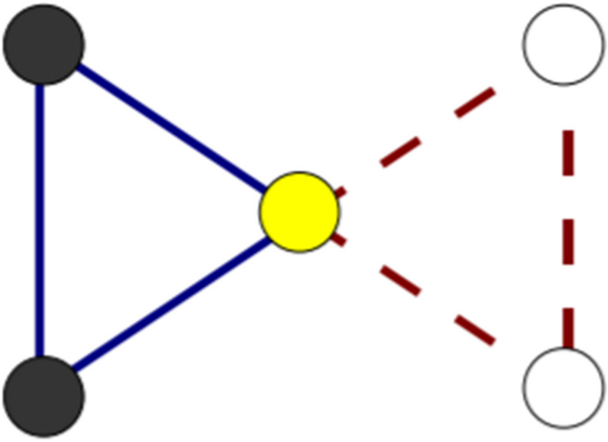

Fig. 7 Example of link graph and link partitioning [99]

munity splits. This probability denotes the likelihood of fitting an anticipated network model to the observed net-work data. The authors compute posterior probability in this block model variation using a novel prior on the number of communities based on a queueing-type process. In the following sections, we examine both conventional SBM and its degree-corrected variant DCSBM, which has been shown to perform better in practice. Local optimization of ordering statistics Lancichinetti et al. [32] estimate a community's statistical significance by calculating the likelihood of discovering a comparable one in a null scenario. Nodes are progressively aggregated into communi-

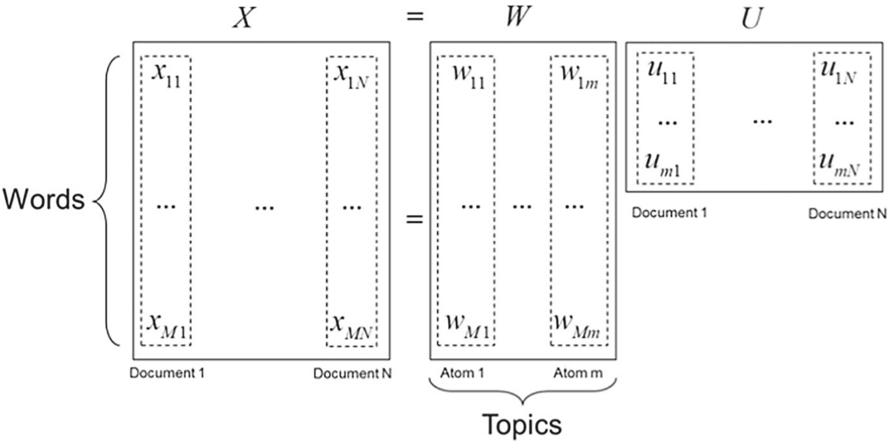

Fig. 8 Dictionary-based community forming using NMF [106]

ties to identify essential communities using this approach. Then, nodes are exchanged across communities to enhance the degree of importance.

NMF based methods
Non-Negative Matrix Factorization Methodologies (NMF) Many algorithms, especially those based on spectrum methods, identify communities based on eigenvalues, which have a physical meaning that is difficult to explain in terms of real-world applications. NMF has shown to be a valuable technique for data analysis that improves interpret-ability. This is a brand-new technique for assessing a dynamic network's structural and functional properties made up of overlapping communities.

It is a machine learning method that decomposes a given feature matrix to reveal the characteristics of a particular structure [2, 107, 108]. This approach has the advantage of eliminating negative feature vector components. According to the conventional decomposition, X = W U, as shown in Fig. 7. Where X is the input data matrix of size M*N and W, U denote the estimated factors of the input data matrix of dimensions M*m and m*N, respectively, with m denoting the factorization rank. It is selected so that (M + N) m < MN. The objective is to reduce:

$$f(W, U) = ||X - WU||_F^2 \tag{14}$$

Principal component based analysis (PCA) methods
Newman et al. [72] developed a Principal Component Based Analysis(PCA) approach for detecting overlap-ping communities in 2014. This method used PCA to determine the optimal number of eigenvectors and then mapped the nodes to a low-dimensional subspace using a Laplacian matrix. It uses fuzzy C-means (FCM) to uncover overlapping structures inside a network. Each vertex in FCM has a sealed membership degree indicating its participation in various clusters. PCA is used to extract principal components from provided nodes by examining the dispersion of adjacent eigenvalues to determine the optimum number of eigenvectors, which is critical for the

efficiency of spectral clustering meth-ods. When PCA is complete, spectral analysis reveals the overlapping community structure. In 2016, Li et al. used a PCA method to enhance a traditional community identification algorithm based on k-means. Three steps comprise the algorithm. To begin, it determines the distance between the network's nodes. The distance between nodes belonging to the same community is much less than be-tween nodes belonging to different communities. The nodes are then transferred to p-dimensional space. Finally, the k-means method determines the K number of communities included inside a net-work. Tao et al. [109] presented another method for detecting overlapping communities based on PCA and membership index (MI), dubbed PCA-MI. PCA extracts relevant characteristics from the complicated network, and then MI is used to categorize nodes into distinct communities.

5 Conclusion

This study aimed to identify and analyze various community detection techniques and their evaluation metrics. They were employed in fraud detection research disseminated in data mining, and this study makes significant contributions in theory and practice. The findings of this study indicate that community identification techniques have been around for a long time and have applications in various areas such as physics, biology, and graph networks. Every area has its own set of requirements, and we must select algorithms carefully to meet those requirements. The identified gaps are needed for new empirical study by researchers working on this topic. Additionally, this article provides practitioners with a road map for appreciating the relationship between the nature of their community, various sorts of difficulties, and the right graph-based solutions for their requirements and application areas. This assessment highlights four broad categories into which community identification approaches may be classified: traditional methods, modularity-based techniques, dynamic techniques, and overlapping community-based strategies. Current events have a strong resemblance to groups that overlap, and researchers may use this categorization to pick from various metrics that are suitable to their topic. Although comprehensive, this systematic review may have missed some pertinent research due to the limits of the particular domains, psychological effects used in community detection, and time-scale used for this study. Thus, future work may include applying these strategies to a single domain and evaluating their performance over various parameters. Because this survey covers a wide variety of domains, it may be limited down according to the domains to which it applies.

References

1. Lei X, Qian X (2016) Rating prediction based on social sentiment from textual reviews. IEEE Trans Multimed 18(9):1910–1921. https://doi.org/10.1109/TMM.2016.2575738
2. Yang J, Leskovec J (2013) Overlapping community detection at scale: a nonnegative matrix factorization approach. In: WSDM 2013—proceedings of 6th ACM international conference on web search data min, pp 587–596. https://doi.org/10.1145/2433396.2433471

3. Sánchez-Oro J, Duarte A (2018) Iterated Greedy algorithm for performing community detection in social networks. Futur Gener Comput Syst 88:785–791. https://doi.org/10.1016/j.future.2018.06.010
4. Plantié M, Crampes M (2012) Social media retrieval. Comput Commun Netw. https://doi.org/10.1007/978-1-4471-4555-4
5. Yang Z, Algesheimer R, Tessone CJ (2016) A comparative analysis of community detection algorithms on artificial networks. Sci Rep. https://doi.org/10.1038/srep30750
6. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174. https://doi.org/10.1016/j.physrep.2009.11.002
7. Flake GW, Tarjan RE, Tsioutsiouliklis K (2004) Graph clustering and minimum cut trees. Internet Math 1(4):385–408. https://doi.org/10.1080/15427951.2004.10129093
8. Gregory S (2009) Finding overlapping communities using disjoint community detection algorithms. In: Complex networks. Springer, pp 47–61
9. Gao C, Ma Z, Zhang AY, Zhou HH (2018) Community detection in degree-corrected block models. Ann Stat 46(5):2153–2185. https://doi.org/10.1214/17-AOS1615
10. Shi C, Yu PS, Yan Z, Huang Y, Wang B (2014) Comparison and selection of objective functions in multiobjective community detection. Comput Intell 30(3):562–582. https://doi.org/10.1111/coin.12007
11. Hagen L, Kahng AB (1992) New spectral methods for ratio cut partitioning and clustering. IEEE Trans Comput Des Integr Circuits Syst 11(9):1074–1085. https://doi.org/10.1109/43.159993
12. Schaub MT, Delvenne JC, Rosvall M, Lambiotte R (2017) The many facets of community detection in complex networks. Appl Netw Sci. https://doi.org/10.1007/s41109-017-0023-6
13. Pothen A, Simon HD, Liou KP (1989) Partitioning alex sparse pothen matrices with eigenvectors of graphs, p 27
14. Pons Pacal LM (2005) Computing communities in large networks using random walks. Download.Springer.Com, pp 284–293
15. Newman MEJ (2004) Fast algorithm for detecting community structure in networks. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Top 69(6):5. https://doi.org/10.1103/PhysRevE.69.066133
16. Von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416. https://doi.org/10.1007/s11222-007-9033-z
17. Newman MEJJ, Girvan M, Newman MEJJ (2004) Fast algorithm for detecting community structure in networks. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Top. 69(23):5. https://doi.org/10.1073/pnas.0601602103
18. Cheng Z, Ding Y, Zhu L, Kankanhalli M (2018) Aspect-aware latent factor model: rating prediction with ratings and reviews, pp 639–648. https://doi.org/10.1145/3178876.3186145
19. Asur S, Parthasarathy S, Ucar D (2009) An event-based framework for characterizing the evolutionary behavior of interaction graphs. ACM Trans Knowl Discov Data 10(1145/1631162):1631164
20. Backstrom L, Huttenlocher D, Kleinberg J, Lan X (2006) Group formation in large social networks: membership, growth, and evolution. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 44–54. https://doi.org/10.1145/1150402.1150412
21. Dunlavy DM, Kolda TG, Acar E (2011) Temporal link prediction using matrix and tensor factorizations. ACM Trans Knowl Discov Data. doi 10(1145/1921632):1921636
22. Fu W, Song L, Xing EP (2009) Dynamic mixed membership blockmodel for evolving networks. In: Proceedings of the 26th annual international conference on machine learning, pp 329–336. https://doi.org/10.1145/1553374.1553416
23. Xu KS, Hero AO (2014) Dynamic stochastic blockmodels for time-evolving social networks. IEEE J Sel Top Signal Process 8(4):552–562. https://doi.org/10.1109/JSTSP.2014.2310294
24. Mankad S, Michailidis G (2013) Structural and functional discovery in dynamic networks with non-negative matrix factorization. Phys Rev E 88(4):42812. https://doi.org/10.1103/PhysRevE.88.042812
25. Kelley S (2009) Overlapping communities in large-scale, vol 2009
26. Stephen Kelley MM-I, Goldberg M, Konstantin MAW (2012) Defining and discovering communities in social networks. Optim Appl 57:463–505. https://doi.org/10.1007/978-1-4614-0754-6
27. Chakraborty T, Chakraborty A (2013) OverCite: finding overlapping communities in citation network. In: Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining, pp 1124–1131. https://doi.org/10.1145/2492517.2500255
28. Su X et al (2022) A comprehensive survey on community detection with deep learning. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2021.3137396

29. Jin D et al (2021) A survey of community detection approaches: from statistical modeling to deep learning. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/tkde.2021.3104155
30. Huang X, Chen D, Ren T, Wang D (2021) A survey of community detection methods in multilayer networks, vol 35. Springer, p 1
31. Naik D, Ramesh D, Gandomi AH, Babu Gorojanam N (2022) Parallel and distributed paradigms for community detection in social networks: a methodological review. Expert Syst Appl 187:115956. https://doi.org/10.1016/j.eswa.2021.115956
32. Li S, Jiang L, Wu X, Han W, Zhao D, Wang Z (2021) A weighted network community detection algorithm based on deep learning. Appl Math Comput 401:126012. https://doi.org/10.1016/j.amc.2021.126012
33. Holland PW, Laskey KB, Leinhardt S (1983) Stochastic blockmodels: first steps. Social Netw 5(2):109–37. https://doi.org/10.1016/0378-8733(83)90021-7
34. Nepal S, Paris C, Bista SK (2012) Srec: a social behaviour based recommender for online communities. In: Herder E, Yacef K, Chen L, Weibelzahl S (eds) Workshop and poster proceedings of 20th conference on user modeling, adaptation, and personalization, Montreal, Canada, July 16–20, 2012, volume 872 of CEUR workshop proceedings. CEUR-WS.org
35. Nepal S, Paris C, Pour PA, Freyne J, Bista SK (2015) Interaction based recommendations for online communities. ACM Trans Internet Technol 15(2):6:1-6:21
36. Cazabet R, Amblard F (2014) Dynamic community detection. In: Alhajj R, Rokne J (eds) Encyclopedia of social network analysis and mining. Springer, New York, pp 404–414
37. Narayanan A, Chandramohan M, Venkatesan R, Chen L, Liu Y, Jaiswal S (2017) graph2vec: learning distributed representations of graph. CoRR, abs/1707.0. arXiv:1707.05005
38. Devooght R, Bersini H (2017) Long and short-term recommendations with recurrent neural networks. In: Proceedings of the 25th conference on user modeling, adaptation and personalization, association for computing machinery, pp 13–21. https://doi.org/10.1145/3079628.3079670
39. Jannach D, Ludewig M, Lerche L (2017) Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. User Model User Adapt Interact 27(3–5):351–392. https://doi.org/10.1007/s11257-017-9194-1
40. Edizel B, Bonchi F, Hajian S, Panisson A, Tassa T (2020) FaiRecSys: mitigating algorithmic bias in recommender systems. Int J Data Sci Anal 9(2):197–213. https://doi.org/10.1007/s41060-019-00181-5
41. Wu S, Chow TWS (2004) Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. Pattern Recognit. 37(2):175–188. https://doi.org/10.1016/S0031-3203(03)00237-1
42. Albert R, Barabási AL (2002) Statistical mechanics of complex networks. Rev Mod Phys 74(1):47–97. https://doi.org/10.1103/RevModPhys.74.47
43. Berahmand K, Bouyer A, Vasighi M (2018) Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. IEEE Trans Comput Soc Syst 5(4):1021–1033. https://doi.org/10.1109/TCSS.2018.2879494
44. Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043):814–818. https://doi.org/10.1038/nature03607
45. Ali SS, Bentayeb F, Missaoui R, Boussaid O (2014) An efficient method for community detection. https://doi.org/10.1007/978-3-319-08326-1
46. Lancichinetti A, Saramäki J, Kivelä M, Fortunato S (2010) Characterizing the community structure of complex networks. PLoS One. https://doi.org/10.1371/journal.pone.0011976
47. Fred ALN, Jain AK (2003) Robust data clustering. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2:5. https://doi.org/10.1109/cvpr.2003.1211462
48. Danon L, Díaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. J Stat Mech Theory Exp 09008(9):219–228. https://doi.org/10.1088/1742-5468/2005/09/P09008
49. Moscato V, Picariello A, Sperlí G (2019) Community detection based on game theory. Eng Appl Artif Intell 85(July):773–782. https://doi.org/10.1016/j.engappai.2019.08.003
50. De Cruz-lovera C, De L (2019) Analysis of research topics and scientific collaborations in energy saving using bibliometric techniques and community detection
51. Wu L, Zhang Q, Chen CH, Guo K, Wang D (2020) Deep learning techniques for community detection in social networks. IEEE Access 8:96016–96026. https://doi.org/10.1109/ACCESS.2020.2996001
52. Rodrigues F, Ferreira B (2016) Product recommendation based on shared customer's behaviour. Proc Comput Sci 100:136–146. https://doi.org/10.1016/j.procs.2016.09.133

53. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E Stat Nonlinear Soft Matter Phys 78(4):1–5. https://doi.org/10.1103/PhysRevE.78.046110
54. Getting Started with Community Detection in Graphs and Networks. https://www.analyticsvidhya.com/blog/2020/04/community-detection-graphs-networks/
55. Hastie T, Tibshirani R, Friedman J (2009) Model assessment and selection
56. Kaufman L, Rousseeuw PJ (2009) Finding groups in data: an introduction to cluster analysis, vol 344. Wiley
57. Rousseeuw PJ, Kaufman L, Trauwaert E (1996) Fuzzy clustering using scatter matrices. Comput Stat Data Anal 23(1):135–151. https://doi.org/10.1016/S0167-9473(96)00026-6
58. Maity S, Rath SK (2014) Extended Clique percolation method to detect overlapping community structure. In: Proceedings of 2014 International conference advanced computer communication and informatics, ICACCI 2014, pp 31–37. https://doi.org/10.1109/ICACCI.2014.6968420
59. Arab M, Afsharchi M (2014) Community detection in social networks using hybrid merging of sub-communities. J Netw Comput Appl 40(1):73–84. https://doi.org/10.1016/j.jnca.2013.08.008
60. Prasad C (2015) A survey on community detection algorithms in large scale real world networks. In: 2nd International conference on computing for sustainable global development (INDIACom). IEEE, pp 1323–1327
61. Li H, Deng K, Cui J, Dong Z, Ma J, Huang J (2018) Hidden community identification in location-based social network via probabilistic venue sequences. Inf Sci (Ny) 422:188–203. https://doi.org/10.1016/j.ins.2017.09.019
62. Dolgikh D, Jelínek I (2015) Graph-based music recommendation approach using social network analysis and community detection method. ACM Int Conf Proc Ser 1008:221–227. https://doi.org/10.1145/2812428.2812453
63. Ferreira LN, Zhao L (2016) Time series clustering via community detection in networks. Inf Sci (Ny) 326:227–242. https://doi.org/10.1016/j.ins.2015.07.046
64. Yoldemir B, Ng B, Abugharbieh R (2016) Stable overlapping replicator dynamics for brain community detection. IEEE Trans Med Imaging 35(2):529–538. https://doi.org/10.1109/TMI.2015.2480864
65. Ghorbani M, Rabiee HR, Khodadadi A (2016) Bayesian overlapping community detection in dynamic networks, vol 3, no 2, pp 1–10. arXiv:1605.02288
66. Liu X, Wei YM, Wang J, Wang WJ, He DX, Song ZJ (2016) Community detection enhancement using non-negative matrix factorization with graph regularization. Int J Mod Phys B 30(20):1–18. https://doi.org/10.1142/S0217979216501307
67. Nguyen HH (2017) Clustering categorical data using community detection techniques. Comput Intell Neurosci. https://doi.org/10.1155/2017/8986360
68. Zhang QZYLW (2020) A real-time recommender system design based on spark streaming
69. Hajiabadi M, Zare H, Bobarshad H (2017) IEDC: an integrated approach for overlapping and non-overlapping community detection. Knowl Based Syst 123:188–199. https://doi.org/10.1016/j.knosys.2017.02.018
70. Guerrero M, Montoya FG, Baños R, Alcayde A, Gil C (2018) Community detection in national-scale high voltage transmission networks using genetic algorithms. Adv Eng Inform 38(July):232–241. https://doi.org/10.1016/j.aei.2018.07.001
71. Bello-Orgaz G, Salcedo-Sanz S, Camacho D (2018) A multi-objective genetic algorithm for overlapping community detection based on edge encoding. Inf Sci (Ny) 462:290–314. https://doi.org/10.1016/j.ins.2018.06.015
72. Malhotra D, Chug A (2021) A modified label propagation algorithm for community detection in attributed networks. Int J Inf Manag Data Insights 1(2):100030. https://doi.org/10.1016/j.jjimei.2021.100030
73. Souravlas S, Anastasiadou S, Katsavounis S (2021) A survey on the recent advances of deep community detection. Appl Sci 11(16):1–20. https://doi.org/10.3390/app11167179
74. Chen MR, Chen BP, Zeng GQ, Di L, Chu P (2020) An adaptive fractional-order BP neural network based on extremal optimization for handwritten digits recognition. Neurocomputing 391:260–272. https://doi.org/10.1016/j.neucom.2018.10.090
75. Boettcher S, Perçus AG (2003) Extremal optimization: an evolutionary local-search algorithm. Oper Res Comput Sci Interfaces Ser 21:61–77. https://doi.org/10.1007/978-1-4615-1043-7
76. Duch J, Arenas A (2005) Community detection in complex networks using extremal optimization. Phys Rev E Stat Nonlinear Soft Matter Phys 72(2):1–4. https://doi.org/10.1103/PhysRevE.72.027104

77. Ding J, Azizbek S, Sun Y, Tan P, Wang F (2021) Detecting overlapping communities in networks with extremal optimization. Int J Innov Comput Inf Control 17(1):355–368. https://doi.org/10.24507/ijicic.17.01.355
78. Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582. https://doi.org/10.1073/pnas.0601602103
79. Newman MEJ (2004) Fast algorithm for detecting community structure in networks. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics 69(6):5. https://doi.org/10.1103/PhysRevE.69.066133
80. Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Top 70(6):6. https://doi.org/10.1103/PhysRevE.70.066111
81. Wakita K, Toshiyuki T (2007) Finding community structure in mega-scale social networks: [Extended Abstract]. 16th International world wide web conference, WWW2007, pp 1275–76. https://doi.org/10.1145/1242572.1242805
82. Tasgin M, Bingol H (2007) Community detection in complex networks using genetic algorithms. ArXiv Preprint ArXiv:0711.0491
83. Zachary WW (2009) Flow model for conflict an information fission in small groups. Small 33(4):452–73
84. Newman MEJ (2004) Detecting community structure in networks. Eur Phys J B 38(2):321–330
85. Liu X et al (2007) Effective algorithm for detecting community structure in complex networks based on GA and clustering. In: Yong S et al (eds) Computational science—ICCS. Springer, Berlin, pp 657–64
86. Manuscript A (2008) Nihms35572, 433(7028), pp 895–900
87. Hu F, Liu Y (2015) A novel algorithm infomap-SA of detecting communities in complex networks. J Commun 10(7):503–511. https://doi.org/10.12720/jcm.10.7.503-511
88. Sartakhti JS, Afrabandpey H, Saraee M (2017) Simulated annealing least squares twin support vector machine (SA-LSTSVM) for pattern classification. Soft Comput 21(15):4361–4373. https://doi.org/10.1007/s00500-016-2067-4
89. He J, Chen D, Sun C (2016) A fast simulated annealing strategy for community detection in complex networks. IEEE Int Conf Comput Commun 5(27):2380–2384
90. Xin Y, Xie Z-Q, Yang J (2016) An adaptive random walk sampling method on dynamic community detection. Expert Syst Appl 58:10–19. https://doi.org/10.1016/j.eswa.2016.03.033
91. The Random Walk algorithm. https://neo4j.com/docs/graph-algorithms/current/labs-algorithms/random-walk/
92. Azizifard N (2014) Social network clustering. Int J Inf Technol Comput Sci 6(1):76–81. https://doi.org/10.5815/ijitcs.2014.01.09
93. Hopcroft J, Khan O, Kulis B, Selman B (2004) Tracking evolving communities in large linked networks. Proc Natl Acad Sci U S A 101(SUPPL. 1):5249–5253. https://doi.org/10.1073/pnas.0307750100
94. Palla G, Barabási AL, Vicsek T (2007) Quantifying social group evolution. Nature 446(7136):664–667. https://doi.org/10.1038/nature05670
95. Wang Y, Wu B, Du N (2008) Community evolution of social network: feature, algorithm and model, 60402011 [Online]. Available: arXiv:0804.4356
96. Rosvall M, Bergstrom CT (2010) Mapping change in large networks. PLoS One. https://doi.org/10.1371/journal.pone.0008694
97. Chen W, Liu Z, Sun X, Wang Y (2010) A game-theoretic framework to identify overlapping communities in social networks. Data Min Knowl Discov 21(2):224–240. https://doi.org/10.1007/s10618-010-0186-6
98. Wang Z, Li Z, Yuan G, Sun Y, Rui X, Xiang X (2018) Tracking the evolution of overlapping communities in dynamic social networks. Knowl Based Syst 157:81–97. https://doi.org/10.1016/j.knosys.2018.05.026
99. Dhouioui Z (2014) Tracking dynamic community evolution in social.pdf, no. Asonam, pp 764–770
100. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E Stat Nonlinear Soft Matter Phys 69(22):1–16. https://doi.org/10.1103/PhysRevE.69.026113
101. Gopalan PK, Blei DM (2013) Efficient discovery of overlapping communities in massive networks. Proc Natl Acad Sci U S A 110(36):14534–14539. https://doi.org/10.1073/pnas.1221839110
102. Macropol K, Singh A (2010) Scalable discovery of best clusters on large graphs. Proc VLDB Endow 3(1):693–702. https://doi.org/10.14778/1920841.1920930

103. Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E Stat Nonlinear Soft Matter Phys 76(3):1–12. https://doi.org/10.1103/PhysRevE.76.036106
104. Evans TS, Lambiotte R (2009) Line graphs, link partitions, and overlapping communities. Phys Rev E Stat Nonlinear Soft Matter Phys 80(1):1–9. https://doi.org/10.1103/PhysRevE.80.016105
105. Nicosia V, Mangioni G, Carchiolo V, Malgeri M (2009) Extending the definition of modularity to directed graphs with overlapping communities. J Stat Mech Theory Exp 3:2009. https://doi.org/10.1088/1742-5468/2009/03/P03024
106. Sun PG (2015) Community detection by fuzzy clustering. Phys A Stat Mech Appl 419:408–416. https://doi.org/10.1016/j.physa.2014.10.009
107. Porter MA, Onnela J-P, Mucha PJ (2009) Communities in networks, pp 0–26 [Online]. Available: arXiv:0902.3788
108. Luo W, Yan Z, Bu C, Zhang D (2020) Community detection by fuzzy relations. IEEE Trans Emerg Top Comput 8(2):478–492. https://doi.org/10.1109/TETC.2017.2751101
109. Riolo MA, Cantwell GT, Reinert G, Newman MEJ (2017) Efficient method for estimating the number of communities in a network. Phys Rev E 96(3):1–13. https://doi.org/10.1103/PhysRevE.96.032310

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