Advanced Review

Community detection in social networks

Punam Bedi* and Chhavi Sharma

The expansion of the web and emergence of a large number of social networking sites (SNS) have empowered users to easily interconnect on a shared platform. A *social network* can be represented by a graph consisting of a set of *nodes* and *edges* connecting these nodes. The nodes represent the individuals/entities, and the edges correspond to the interactions among them. The tendency of people with similar tastes, choices, and preferences to get associated in a social network leads to the formation of virtual clusters or communities. Detection of these communities can be beneficial for numerous applications such as finding a common research area in collaboration networks, finding a set of likeminded users for marketing and recommendations, and finding protein interaction networks in biological networks. A large number of community-detection algorithms have been proposed and applied to several domains in the literature. This paper presents a survey of the existing algorithms and approaches for the detection of communities in social networks. We also discuss some of the applications of community detection.

How to cite this article:
WIREs Data Mining Knowl Discov 2016, 6:115–135. doi: 10.1002/widm.1178

INTRODUCTION

A social network for an individual is created with his/her interactions and personal relationships with other members in the society. Social networks represent and model the social ties among individuals. With the rapid expansion of the web, there is a tremendous growth in online interaction of users. Many social networking sites, e.g., Facebook, Twitter etc., have also come up to facilitate user interaction. As the number of interactions have increased manifold, it is becoming difficult to keep track of these communications. Human beings tend to get associated with people of similar likings and tastes. The easy-to-use social media allows people to extend their social life in unprecedented ways as it is difficult to meet friends in the physical world but much easier to find friends with similar interests online. These real-world social networks have interesting patterns and properties, which may be analyzed for numerous useful purposes.

Social networks have a characteristic property to exhibit a community structure. If the vertices of the network can be partitioned into either disjoint or overlapping sets of vertices such that the number of edges within a set exceeds the number of edges between any two sets by some reasonable amount, we say that the network displays a community structure. Networks displaying a community structure may often exhibit a hierarchical community structure as well.[1]

The process of discovering the cohesive groups or clusters in the network is known as community detection. It forms one of the key tasks of *social network analysis*.[2] The detection of communities in social networks can be useful in many applications where group decisions are taken, e.g., multicasting a message of interest to a community instead of sending it to each one in the group or recommending a set of products to a community. The applications of community detection have been highlighted toward the end of the article.

State-of-the-art in community-detection research for social networks is presented in this article. The

*Correspondence to: pbedi@cs.du.ac.in

Department of Computer Science, University of Delhi, Delhi, India

Conflict of interest: The authors have declared no conflicts of interest for this article.

paper begins with the basic concepts of social networks and communities. Various methods for community detection are categorized and discussed in the next section followed by a list of standard datasets used for analysis in community-detection research along with the links for download if available online. Some potential applications of community detection in social networks are briefly described in the next section. The Discussion section argues the advantages of using a method with respect to another, the kind of community structure they obtain, etc., and the Conclusion section concludes the paper.

BASIC CONCEPTS

Social Network

A social network is depicted by a social network graph G consisting of n number of nodes denoting n individuals or the participants in the network. The connection between node i and node j is represented by the edge e_{ij} of the graph. A directed or an undirected graph may illustrate these connections between the participants of the network. The graph can be represented by an adjacency matrix A in which $A_{ij} = 1$ in case there is an edge between i and j, else $A_{ij} = 0$. Social networks follow the properties of complex networks.[3,4]

Some real-life examples[1] of social networks include friends-based, telephone, email, and collaboration networks. These networks can be represented as graphs, and it is feasible to study and analyze them to find interesting patterns amongst the entities. These appealing prototypes can be utilized in various useful applications.

Community

A community can be defined as a group of entities closer to each other in comparison to other entities of the dataset. A community is formed by individuals such that those within a group interact with each other more frequently than with those outside the group. The closeness between entities of a group can be measured via similarity or distance measures between entities. McPherson et al.[5] stated that '*similarity breeds connection*.' They discussed various social factors that lead to similar behavior or homophily in networks. The communities in social networks are analogous to clusters in networks. An individual represented by a node in a graph may not be part of just a community or a group; it may be an element of many closely associated or different groups existing in the network. For example, a person may concurrently belong to college, school, friends, and family groups. All such communities that have common nodes are called *overlapping communities*.

Identification and analysis of the community structure has been done by many researchers applying methodologies from numerous forms of sciences. The quality of clustering in networks is normally judged by clustering coefficient, which is a measure of how much the vertices of a network tend to cluster together. The global clustering coefficient[6] and the local clustering coefficient[7] are two types of clustering coefficients discussed in literature.

Methods for Grouping Similar Items

Communities are those parts of the graph that have denser connections inside and few connections with the rest of the graph.[8] The aim of *unsupervised learning* is to group together similar objects without any prior knowledge about them. In case of networks, the clustering problem refers to the grouping of nodes according to their similarity computed based on topological features and/or other characteristics of the graph. Network partitioning and clustering are two commonly used methods in literature to find the groups in the social network graph. These methods are briefly described in the next subsections.

Graph Partitioning

Graph partitioning is the process of partitioning a graph into a predefined number of smaller components with specific properties. A common property to be minimized is called cut size. A cut is a partition of the vertex set of a graph into two disjoint subsets, and the size of the cut is the number of edges between the components. A multicut is a set of edges whose removal divides the graph into two or more components. It is necessary to specify the number of components one wishes to get in case of graph partitioning. The size of the components must also be specified as otherwise, a likely but not meaningful solution would be to put the minimum degree vertex into one component and the rest of the vertices into another. As the number of communities is usually not known in advance, graph partitioning methods are not suitable to detect communities in such cases.

Clustering

Clustering is the process of grouping a set of similar items together in structures known as *clusters*. Clustering the social network graph may give a lot of information about the underlying hidden attributes, relationships, and properties of the participants as

well as the interactions among them. The hierarchical clustering and partitioning method of clustering are the commonly used clustering techniques that have been discussed in the literature.

In hierarchical clustering, a hierarchy of clusters is formed. The process of hierarchy creation or levelling can be agglomerative or divisive. In agglomerative clustering methods, a bottom-up approach to clustering is followed. A particular node is clubbed or agglomerated with similar nodes to form a cluster or a community. This aggregation is based on similarity. In divisive clustering approaches, a large cluster is repeatedly divided into smaller clusters.

Partitioning methods begin with an initial partition amidst the number of clusters preset and the relocation of instances by moving them across clusters, e.g., K-means clustering. An exhaustive evaluation of all possible partitions is required to achieve global optimality in partitioned-based clustering. This is time consuming and sometimes infeasible; hence, researchers use greedy heuristics for iterative optimization in partitioning methods of clustering. The next section categorizes and discusses major algorithms for community detection.

ALGORITHMS FOR COMMUNITY DETECTION

A number of community-detection algorithms and methods have been proposed and deployed for the identification of communities in literature. There have also been modifications and revisions to many methods and algorithms already proposed. A comprehensive survey of community detection in graphs has been done by Fortunato[8] in the year 2010. Other reviews available in the literature are by Coscia et al.[9] in 2011, Fortunato and Castellano[10] in 2012, Porter et al.[11] in 2009, Danon et al.[12] in 2005, and Plantié and Crampes[13] in 2013. The presented work reviews the algorithms available till 2015 to the best of our knowledge, including the algorithms given in the earlier surveys. Papers based on new approaches and techniques, like big data, not discussed by previous authors have been incorporated in our article. The algorithms for community detection are categorized into approaches based on graph partitioning, clustering, genetic algorithms, label propagation-based, semantics-based, methods for overlapping community detection (clique-based and non-clique-based methods), and community detection for dynamic networks. Algorithms under each of these categories are described below.

Graph Partitioning-based Community Detection

Graph partitioning-based methods have been used in the literature to divide the graph into components such that there are few connections between the components. The Kernighan–Lin[14] algorithm for graph partitioning was amongst the earliest techniques to divide a graph. It partitions the nodes of the graph into subsets of given sizes so as to minimize the sum of costs on all edges cut. A major disadvantage of this algorithm, however, is that the number of groups has to be predefined. The algorithm, however, is quite fast, with a worst-case running time of $O(n^2)$. Newman[15] reduces the widely studied maximum likelihood method for community detection to a search through a group of candidate solutions, each of which is itself a solution to a minimum-cut graph-partitioning problem. The paper shows that the two most essential community inference methods based on the stochastic block model or its degree-corrected variant[16] can be mapped onto versions of the familiar minimum-cut graph-partitioning problem. This has been illustrated by adapting the Laplacian spectral partitioning method[17,18] to perform community inference.

Clustering-based Community Detection

The main concern of community detection is to detect clusters, groups, or cohesive subgroups. The basis of a large number of community-detection algorithms is clustering. Amongst the innovators of community-detection methods, Girvan and Newman[19] had a main role. They proposed a divisive algorithm based on *edge-betweenness* for a graph with undirected and unweighted edges. The algorithm focused on edges that are most 'between' the communities, and communities are constructed progressively by removing these edges from the original graph. Three different measures of calculation of edge-betweenness in vertices of a graph were proposed by Newman and Girvan.[20] The worst-case time complexity of the edge-betweenness algorithm is $O(m^2n)$ and is $O(n^3)$ for sparse graphs, where m denotes the number of edges, and n is the number of vertices.

The Girvan Newman (GN) algorithm has been enhanced by many authors and applied to various networks[21–28]. Rattigan et al.[21] proposed the indexing methods to reduce the computational complexity of the GN algorithm significantly. Chen et al.[22] extended the GN algorithm to partition weighted graphs and used it to identify functional modules in the yeast proteome network. Pinney et al.[24] also built

an algorithm that uses the GN algorithm for the decomposition of networks based on the graph theoretical concept of betweenness centrality. Their paper inspected the utility of betweenness centrality to decompose such networks in diverse ways.

Radicchi[29] et al. also proposed an algorithm based on the GN algorithm, introducing a new definition of community. They defined 'strong' and 'weak' communities. The algorithm uses an edge-clustering coefficient to perform the divisive edge removal step of GN and has a running time of $O(m^4/n^2)$ and $O(n^2)$ for sparse graphs. Moon et al.[30] have proposed and implemented the parallel version of the GN algorithm to handle large-scale data. They have used the MapReduce model (Apache Hadoop) and GraphChi.

Newman and Girvan first defined a measure known as 'modularity' to judge the quality of partitions or communities formed.[20] The modularity measure proposed by them has been widely accepted and used by researchers to gauge the goodness of the modules obtained from the community detection algorithms with high modularity corresponding to a better community structure. Modularity was defined as $Q = \sum_i e_{ii} - a_i^2$, where e_{ii} denotes the fraction of the edges that connect vertices in community i; e_{ij} denotes the fraction of the edges connecting vertices in two different communities, i and j, while $a_i = \sum_j e_{ij}$ is the fraction of edges that connect to vertices in community i. The value $Q = 1$ indicates a network with strong community structure.

The optimization of modularity function has received great attention in the literature. Table 1 lists clustering-based community-detection methods, including algorithms that use modularity and modularity optimization.

Newman[31] has worked to maximize modularity so that the process of aggregating nodes to form communities leads to maximum modularity gain. This change in modularity on joining two communities defined as $\Delta Q = e_{ij} + e_{ji} - 2a_ia_j = 2(e_{ij} - a_ia_j)$ can be calculated in constant time and, hence, is faster to execute in comparison to the GN algorithm. The run time of the algorithm is $O(n^2)$ for sparse graphs and $O((m + n)\, n)$ for others. In a recent article, a scalable version of this algorithm was implemented using MapReduce by Chen et al.[45]. Newman[32] generalized the betweenness algorithm for weighted networks. The modularity was now represented as $Q = \frac{1}{2}m\sum_{ij}\left[A_{ij} - \frac{k_ik_j}{2m}\right]\delta(c_i, c_j)$, where $m = \frac{1}{2}\sum_{ij} A_{ij}$ represents the number of edges in the graph, while k_i, k_j are degrees of vertices i and j, and $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise. Newman[33], in yet another approach, characterized the modularity matrix in terms of eigenvectors. The equation for modularity was changed to $Q = \frac{1}{4m}s^{\mathrm{T}}Bs$, where the modularity matrix was given as $B_{ij} = A_{ij} - \frac{k_ik_j}{2m}$, and modularity was defined using eigenvectors of the modularity matrix. The algorithm runs in $O(n^2 \log n)$ time, where $\log n$ represents the average depth of the dendrogram.

Clauset et al.[34] used greedy optimization of modularity to detect communities for large networks. For a network structure with m edges and n vertices, the algorithm has a running time of $(md \log n)$, where 'd' denotes the depth of the dendrogram. For sparse real-world networks, the running time is $(n \log^2 n)$.

Blondel et al.[35] designed an iterative two-phase algorithm known as the *Louvain method*. In the first phase, all nodes are placed into different communities, and then, the modularity gain of moving a node from one community to another is found. In case this modularity gain is positive, the node is shifted to a new community. In the second phase, all the communities found in the earlier phase are treated as nodes, and the weight of links is found. The algorithm improves the time complexity of the GN algorithm. It has a linear run time of $O(m)$. Guimera et al.[36] used simulated annealing for modularity optimization and showed that computing the modularity of a network is similar to determining the ground-state energy of a spin system. Additionally, the authors showed that the stochastic network models give rise to modular networks due to fluctuations. Zhou et al.[37] attempted to improve modularity using simulated annealing, introducing the idea of inter edges and intra edges. The authors modified the modularity equation to include inter and intra edges as:

$$Q = \frac{1}{2m}\sum_{ij}^{n}\Bigg[\underbrace{\left(A_{ij} - \frac{k_ik_j}{2m}\right)\delta(C_iC_j)}_{\text{Intra factor}} - \underbrace{\beta\left(A_{ij} - \frac{k_ik_j}{2m}\right)^{\alpha}\left(1 - \delta(C_i, C_j)\right)}_{\text{Inter factor}}.\Bigg]$$

Here, α and β are undetermined parameters and affect the value of the inter factor. The value of β is increased and α is reduced when large communities are expected. Duch et al.[38] proposed a heuristic search-based approach for the optimization of modularity function using an extremal optimization technique, which has a complexity of $O(n^2 \log^2 n)$. The

TABLE 1 | Clustering-based Community Detection

Author (Algorithm)	Approach	Parameters	Code Availability
Newman and Girvan[20]	Divisive clustering (using '*modularity*' as a quality metric)	Edge-betweenness	https://github.com/kjahan/community
Newman[31–33]	Modularity maximization	Refs 31,32: Modularity, Ref 33: eigenvector and eigenvalue	Ref 31: http://web.ist.utl.pt/aplf/code/gcf-003.html, Ref 32: http://deim.urv.cat/~sergio.gomez/radatools.php#download Ref 33: http://deim.urv.cat/~sergio.gomez/radatools.php#download
Clauset e al[34]	Greedy optimization of modularity	Edges, vertices, modularity	http://www.cs.unm.edu/~aaron/research/fastmodularity.htm
Blondel et al. (Louvain Method)[35]	Hierarchical clustering	Nodes, edges, modularity	https://perso.uclouvain.be/vincent.blondel/research/louvain.html
Guimera et al.[36], Zhou et al.[37]	Modularity optimization using simulated annealing	Ref 36: No. of links, linking probability, no. of modules, no. of partitions, modularity Ref 37: No. of edges, inter factor and intra factor, modularity	No
Duch et al.[38]	Modularity optimization using extremal optimization	No. of nodes, links, degree, modularity	http://deim.urv.cat/~sergio.gomez/radatools.php#description
Ye et al. (AdClust)[39]	Agglomerative clustering	Vertices, force, modularity	No
Wahl and Sheppard[40]	Hierarchical fuzzy spectral clustering	Fuzzy modularity, Jaccard similarity	No
Falkowski et al. (DENGRAPH)[41]	Density-based clustering	Distance Function	No
Dongen et al. (MCL)[42]	Markovian clustering	Number of nodes	http://www.micans.org/mcl/#source
Nikolaev et al.[43]	Entropy centrality-based clustering	Transition probability matrix for Markov process	No
Steinhauser et al.[44]	Consensus clustering, random walk	Similarity matrix , length of random walks	No

AdClust method[39] can extract modules from complex networks with significant precision and strength. Each node in the network is assumed to act as a self-directed agent representing flocking behavior. The vertices of the network travel toward the desirable adjoining groups.

Wahl and Sheppard[40] proposed a hierarchical fuzzy spectral clustering-based approach. They argued that determining the sub-communities and their hierarchies are as important as determining communities within a network. The Density based graph clustering (DENGRAPH)[41] algorithm uses the idea of density-based incremental clustering of spatial data and is intended to work for large dynamic datasets with noise. The Markov Clustering Algorithm (MCL)[42] is a graph flow simulation algorithm that can be used to detect clusters in a graph and is analogous to the detection of communities in the networks. This algorithm consists of two alternate processes of 'expansion' and 'inflation.' Markov chains are employed to perform a random walk through a graph. The method has a worst-case run time of $O(nk^2)$, where n represents the number of nodes, and k is the number of resources. Nikolaev et al.[43] used an 'entropy centrality measure' based on the Markovian process to iteratively detect communities. A random walk through the nodes is performed to find the communities existing in the network structure. For a graph, the transition probability matrix for a Markov chain is created. A locality t is selected, and those edges, for which the average entropy centrality for the nodes over the graph is reduced, are selected and removed. The algorithm proposed by Steinhaeuser et al.[44] performs many short random walks and interprets visited nodes during the same walk as similar nodes, which gives an indication that they belong to the same community. The similar nodes are aggregated, and community structure is created using consensus clustering. It has a run time of $O(n^2 \log n)$.

Genetic Algorithms (GA)-based Community Detection

Genetic algorithms (GA) are adaptive heuristic search algorithms whose aim is to find the best solution under the given circumstances. A GA starts with a set of solutions known as chromosomes, and fitness function is calculated for these chromosomes. If a solution with a maximum fitness is obtained, search is terminated else crossover and mutation operators are applied to the current set of solutions with some probability to obtain the new set of solutions. Community detection can be viewed as an optimization problem in which an objective function that captures the intuition of a community with better internal connectivity than external connectivity is chosen to be optimized. GA have been applied to the process of community discovery and analysis in a few recent research works. These are described briefly in this section. Table 2 enlists the algorithms available in the literature for community detection based on GA.

Pizzuti[46] proposed the GA-Net algorithm, which uses a locus-based graph representation of the

TABLE 2 | Genetic Algorithms-based Community Detection

Author (Algorithm)	Approach	Parameters	Code Availability
Pizzuti(GA-Net)[46]	Community score as fitness function	Community score	http://staff.icar.cnr.it/pizzuti/codes.html
Pizzuti(MOGA-Net)[47]	Multiobjective optimization	Community score Community fitness	http://staff.icar.cnr.it/pizzuti/codes.html
Hafez et al.[48]	Single objective, multiobjective optimization	Number of genes, mutation Crossover operators	No
Mazur et al.[49]	Community score and modularity as fitness functions	Fitness functions	No
Liu et al.[50]	Genetic algorithm and clustering	Size of population, maximal generation number, maximum no. of generations for unimproved fittest chromosome fraction of mined hubs, no. of communities	No
Tasgin et al.[51]	Modularity optimization	Modularity, population size, number of chromosomes	No
Zadeh [52]	Multipopulation cultural algorithm	Belief Space based average and normative parameters	No

network. The nodes of the social network are depicted by genes and alleles. The algorithm introduces and optimizes the community score to measure the quality of partitioning. All the dense communities present in the network structure are obtained at the end of the algorithm by selectively exploring the search space without the need to know the exact number of groups in advance. Another GA-based approach, MOGA-Net,[47] proposed by Pizzuti optimizes two objective functions, i.e., the community score and community fitness. The higher the community score, the denser the clustering obtained. The community fitness is the sum of fitness of nodes belonging to a module. When this sum reaches its maximum, the number of external links is minimized. MOGA-Net generates a set of communities at different hierarchical levels in which solutions at deeper levels, consisting of a higher number of modules, are contained in solutions having a lower number of communities. Hafez et al.[48] have performed both single-objective and multi-objective optimization for community-detection problems. The former optimization was done using roulette selection-based GA, while the nondominated sorting genetic algorithm II (NSGA-II) was used for the latter process. Mazur et al.[49] have used modularity as the fitness function in addition to the community score. The authors worked on undirected graphs, and their algorithm can also discover single-node communities. Liu et al.[50] used GA in addition to clustering to find the community structures in a network. The authors have used a strategy of repeated divisions. The graph is initially divided into two parts; then, the subgraphs are further divided, and a nested GA is applied to them. Tasgin et al.[51] have also optimized the network modularity using GA. A multicultural algorithm[52] for community detection employs the fitness function defined by Pizzuti[46] in GA-Net. The belief space, which is a state space for the network and contains a set of individuals that have a better fitness value, has been used in this work to guide the search direction by determining a range of possible states for individuals. A GA for the optimization of modularity, proposed by Nicosia et al.[53], has been explained in the overlapping communities section later.

Label Propagation-based Community Detection

Label propagation in a network is the propagation of a label to various nodes existing in the network. Each node attains the label possessed by a maximum number of the neighboring nodes. This section discusses some label propagation-based algorithms for discovering communities. Table 3 contains a listing of these algorithms, discussed in detail later in the section.

Label Propagation Algorithm (LPA) was proposed by Raghavan et al.[54] in which initially, each node tries to achieve a label from the maximum number of labels possessed by its neighbors. The stopping criterion for the process was also the same, i.e., when each node achieves a label that a maximum number of its neighboring nodes have. Each iteration of the algorithm takes $O(m)$ time, where m is the number of edges. Speaker listener label propagation algorithm[55] (SLPA) is an extension to LPA that could analyze different kinds of communities such as disjoint communities, overlapping communities, and hierarchical communities in both unipartite and bipartite networks. The algorithm has a linear run time of $O(Tm)$, where T is the user-defined maximum number of iterations, and m is the number of edges. Based on the SLPA algorithm, Hu[56] proposed a weighted label propagation algorithm (WLPA). It uses the similarity between any two of the vertices in a network based on the labels of the vertices achieved in label propagation. The similarity of these vertices is then used as a weight of the edge in label propagation.

LPA was further improved by Gregory[57] in his algorithm COPRA (Community Overlap Propagation Algorithm). It was the first label propagation-based procedure that could also detect overlapping communities. The run time per iteration is $O(vm \log(vm/n$, where n is the number of nodes; m is the edges; and v is the maximum number of communities per vertex. LabelRank Algorithm[58] uses the LPA and Markov clustering algorithm (MCL). The node identifiers are used as labels. Each node receives a number of labels from the neighboring nodes. A community is formed for the nodes having the same highest probability label. Four operators are applied, namely, propagation that propagates the label to neighbors; inflation, i.e., the inflation operator of the MCL algorithm; cut-off operator that removes the labels below a threshold; and an explicit conditional update operator responsible for a conditional update. The algorithm runs in $O(m)$ time, where m is the number of edges. The LabelRank algorithm was modified to LabelRankT algorithm by Xie et al.[60] This algorithm included both the edge weights and the edge directions in the detection of communities. This algorithm works for *dynamic* networks as well and is also able to detect evolving communities.

Wu et al.[59] proposed a balanced multi label propagation algorithm (BMPLA) for the detection of overlapping communities. Using this algorithm,

TABLE 3 | Label Propagation-based Community Detection

Author (Algorithm)	Approach	Applications/ Improvements	Parameters	Code Availability
Raghavan et al. (LPA)[54]	Iterative label propagation	SLPA[55] WLPA[56] COPRA[57] LabelRank[58] BMPLA[59]	Ref 55: nodes, labels Ref 56: labels, threshold Ref 57: label, similarity Ref 59: nodes	Ref 54 : http://igraph.wikidot.com/community-detection-in-r Ref 55 : https://sites.google.com/site/communitydetectionslpa/ Ref 57: http://www.cs.bris.ac.uk/~steve/networks/software/copra.html
Xie et al. (LabelRank)[58]	(1) Propagation, (2) Inflation, (3) cut-off, (4) conditional update.	LabelRankT[60]	Ref 58: belongingness coefficient, threshold Ref 60: nodes	Refs 58,60: No
Wu et al. (BMLPA)[59]	Label propagation, overlapping Communities	–	Number of vertices, labels to which vertices belong, average degree	No

BMLPA, Balanced Multi-Label Propagation Algorithm; LPA, Label Propagation Algorithm.

vertices can belong to any number of communities without having a global maximum limit on the largest number of community memberships required by COPRA.[57] Each iteration of the algorithm takes $O(n \log n)$ time to execute, where n is the number of nodes.

Semantics-based Community Detection

Semantic content and edge relationships in a semantic network may be additionally used to partition the nodes into communities. The context as well as the relationship of the nodes are taken into consideration in the process of semantic community detection. Latent dirichlet allocation[61] (LDA) is used in several semantic community-based community-detection approaches. A clustering algorithm based on the link-field-topic (LFT) model is put forward by Xin et al.[62] to overcome the limitation of defining the number of communities beforehand. The study forms the semantic link weight (SLW) based on the investigation of LFT to evaluate the semantic weight of links for each sampling field. The proposed clustering algorithm is based on the SLW, which could separate the semantic social network into clustering units. In another work, the authors[63] have used ARTs (Author-Recipient-Topics) model and divided the process into two phases, namely, LDA sampling and community detection. In the former process, multiple sampling ARTs have been designed. A community-clustering algorithm has also been proposed. The procedure could detect the overlapping communities. Xia et al.[64] constructed a semantic network using information from the comment content extracted from the initial HTML source files. An average score is obtained for two users for each link, assuming comments to be implicit links between people. An analytical method for taking out comment content is proposed to build the semantic network, for example, the terms and phrases in data are counted in comments as supportive or opposing. Each phrase is given an associated numerical trust value. On this semantic network, the classical community-detection algorithm is applied henceforth. Ding[65] has considered the impact of topological as well as topical elements in community detection. Topology-based approaches are based on the idea that the real-world networks can be modelled as graphs where the nodes depict the entities, while the interactions between them are shown by the edges of the graph. On the other hand, topic-based community detection has a basis that the more words two objects share, the more similar they are. The author performs systematic analysis with topology-based and topic-based community-detection methodologies

on the co-authorship networks. The paper puts forward the argument that to detect communities, one should take into account the topical and topological features of networks together. A community-detection algorithm, SemTagP (Semantic Tag propagation), has been proposed by Ereteo et al.[66] that takes yield of the semantic data captured while organizing the RDF (Resource Description Framework) graphs of social networks. It basically is an extension of the LPA[54] algorithm that performs the semantic propagation of tags. The algorithm detects and, moreover, labels communities using the tags used by the group during the social labelling process and the semantic associations derived between tags. In a study by Zhao et al.,[67] a topic-oriented approach consisting of an amalgam of social objects clustering and link analysis has been used. First, a modified form of k-means clustering, named 'Entropy Weighting K-Means (EWKM) algorithm,' has been used to cluster the social objects. A subspace-clustering algorithm is applied to cluster all the social objects into topics. On the clusters obtained in this process, topical community detection or link analysis is performed using a modularity optimization algorithm. The members of the objects are separated into topical clusters having unique topics. A link analysis is performed on each topical cluster to discover the topical communities. The end result of the entire method is topical communities. A community-extraction approach is given by Abdelbary et al.,[68] which integrates the content published within the social network with its semantic features. Community discovery is performed using the two-layer generative Restricted Boltzmann Machines model. The model presumes that members of a community communicate over matters of common concern. The model permits associate members to belong to multiple communities.

Latent semantic analysis (LSA)[69] and latent dirichlet allocation (LDA)[61] are the two techniques extensively employed in the process to detect topical communities. Nyugen et al.[70] have used LDA to find hyper groups in the blog content, and then, sentiment analysis is done to further find the meta-groups in these units. A link-content model is proposed by Natarajan et al.[71] to discover topic-based communities in social networks. Community has been modelled as a distribution employing Gibbs sampling. This paper uses links and content to extract communities in a content-sharing network, Twitter.

Methods to Detect Overlapping Communities

A recent survey by Amelio et al.[72] gives a comprehensive review of major overlapping community-detection algorithms, and in their work, they have also included a category of dynamic networks-based overlapping community detection. There exists another exhaustive review of methods for discovering overlapping communities conducted by Xie et al.[73] The following section discusses some of the methods used to detect overlapping communities. A few recent works in this area that have been covered in the section have not been covered in the previous surveys. Tables 4 and 5 enlist the methods discussed in this section.

TABLE 4 | Clique-based Methods for Overlapping Community Detection

Author (Algorithm)	Approach	Parameters	Code Availability
Palla et al. (CPM)[74]	Clique percolation method	Nodes , threshold weight	http://igraph.wikidot.com/community-detection-in-r, http://www.cfinder.org/
Lancichinetti et al. [75]	Fitness function	Fitness function	No
Du et al. (ComTector)[76]	Kernels-based clustering	Set of all kernels	No
Shen at al (EAGLE)[77]	Agglomerative hierarchical clustering	Similarity between two communities	No
Evans et al. [78–80]	Line graph, clique graph	Links, partition	No
Lee et al. (GCE)[81]	Cliques-based expansion	Fitness function	https://sites.google.com/site/greedycliqueexpansion/
Gregory et al. (CONGA[25], CONGO[82] Peacock algorithm[83])	Split betweenness	Ref 25: vertex, split betweenness, Ref. 82: Local betweenness, short paths Ref 83: ratio of max. edge betweenness and max. split betweenness	Refs 25,82,83: http://www.cs.bris.ac.uk/~steve/networks/

TABLE 5 | Non-clique Methods for Overlapping Community Detection

Author (Algorithm)	Approach	Parameters	Code Availability
Nicosia et al. [53]	Modularity for overlapping communities genetic algorithm approach	In degree, out degree, belongingness coefficient	No
Pizzuti (GA-NET+)[84]	GA based	Community score	http://staff.icar.cnr.it/pizzuti/codes.html
Lancichinetti et al. (OSLOM)[85]	Edge direction, weights, hierarchy	N vertices, E edges, degree of subgraph, internal and external degree of subgraph	http://www.oslom.org/
Baumes et al.[86]	Clusters of overlapping vertices	Internal edge intensity external edge intensity, internal edge probability, edge ratio intensity ratio	No
Chen et al[87]	Game theory based	Set of communities Gain function, Loss function	No
Alvari et al.[88,89]	Game theory based	Set of snapshots, with V vertices and E edges	https://github.com/hamidalvari/D-GT
Shi et al. (GaoCD)[90]	Objective function: partition density	Size of population, running generation ratio of crossover, ratio of mutation	No
Xing et al. (OCDLCE)[91]	Community detection, merging and refining	Nodes, edges, neighbors of node	No
Bhat et al. (OCMiner)[92]	Density based	Threshold ⊠	No
Zhang et al.[93]	Preference-based non-negative matrix factorization	Number of nodes, edges, communities	No
Kozdoba et al.[94]	Cluster aggregation	Probability measures, number of components, threshold parameter	No
Whang et al.[95]	Seed expansion	Nodes, edges, number of seeds, Pagerank link following parameter, biconnnected cores	https://www.cs.utexas.edu/~joyce/codes/cikm2013/nise.html
Rees et al.[96]	Egonets, Friendship groups	Number of nodes	No

Clique-based Methods for Overlapping Community Detection

A community can be interpreted as a union of smaller, complete (fully connected) subgraphs that share nodes. A k-clique is a fully connected subgraph consisting of k nodes. A k-clique community can be defined as a union of all k-cliques that can be reached from each other through a series of adjacent k-cliques. Many researchers have used cliques to detect overlapping communities. Important contributions using cliques for overlapping community detection are summarized in Table 4.

The clique percolation method (CPM) was proposed by Palla et al.[74] to detect overlapping communities. The method first finds all cliques of the network and uses the algorithm of Everett et al.[97] to identify communities by component analysis of a clique–clique overlap matrix. CPM has a run time of $O(\exp(n))$. The CPM proposed by Palla et al.[74] could not discover the hierarchical structure along with the overlapping attribute. This limitation was overcome through the method proposed by Lancichinetti et al.[75] It performs a local exploration in order to find the community for each of the nodes. In this process, the nodes may be revisited any number of times. The main objective was to find local maxima based on a fitness function. CFinder[98] software was developed using CPM for overlapping community detection. Du et al.[76] proposed ComTector (Community DeTector) for detection of overlapping communities using maximal cliques. Initially, all maximal cliques in the network that form the kernels of a potential community are found. Then, the agglomerative technique is iteratively used to add the vertices left to their closest kernels. The obtained clusters are adjusted by merging a pair of fractional communities in order to optimize the modularity of the network. The run time of the algorithm is $(C * T^2)$, where the communities detected are denoted by C, and T is the number of triangles in the network. EAGLE, an agglomerativE hierarchicAl clusterinG based on maximaL cliquE algorithm has been proposed by Shen et al.[77] In the first step, maximal cliques are discovered, and those smaller than a threshold are discarded. Subordinate maximal cliques are neglected, and the remaining give the initial communities (also the subordinate vertices). The similarity is found between these communities, and communities are repeatedly merged together on the basis of this similarity. This is repeated till one community remains at the end. Evans et al.[78] proposed that by partitioning the links of a network, the overlapping communities may be discovered. In an extension to this work, Evans et al.[79] used weighted line graphs. In another work, Evans[80] used clique graphs to detect the overlapping communities in real-world social networks. Greedy clique expansion (GCE)[81] first identifies cliques in a network. These cliques act as seeds for expansion along with the greedy optimization of a fitness function. A community is created by expanding the selected seed and performing its greedy optimization via the fitness function proposed by Lancichinetti et al.[75] Cluster-overlap Newman Girvan algorithm (CONGA) was proposed by Gregory[25]. This method was based on the split- betweenness algorithm of Girvan–Newman. The run time of the method is $O(m^3)$. In another work, CONGO[82] (CONGA Optimized) algorithm was proposed, which used a local betweenness measure, leading to an improved complexity, $O(n \log n)$. A two-phase Peacock algorithm for the detection of overlapping communities is proposed in Gregory[83] using disjoint community-detection approaches. In the first phase, the network transformation was performed using the split betweenness concept proposed earlier by the author. In the second phase, the transformed network is processed by a disjoint community-detection algorithm, and the detected communities were converted back to overlapping communities of the original network.

Non-Clique Methods for Overlapping Community Detection

Some other non-clique methods to discover overlapping communities are given in Table 5. These methods have been briefly explained in this section.

An extension of Newman's modularity for directed graphs and overlapping communities was performed by Nicosia et al.[53], and modularity was given by $Q_{ov} = \frac{1}{m}\sum_{c\in C}\sum_{i,j\in V}\left[\beta_{1(i,j),C}A_{ij}\frac{\beta^{out}_{1(i,j),c}k^{out}_i\beta^{in}_{1(i,j),c}k^{in}_j}{}\right]$. The authors defined a belongingness coefficient $\beta_{1,c}$ of an edge 1 connecting nodes i and j for a particular community c and which is given by $\beta_{l,c} = \mathcal{F}(\alpha_{i,c}, \alpha_{j,c})$ where the definition for $\mathcal{F}(\alpha_{i,c}, \alpha_{j,c})$ is taken as arbitrary, e.g., it can be taken as a product of the belonging coefficients of the nodes involved or as $\max(\alpha_{i,c}, \alpha_{j,c})$. $\beta^{out}_{l(i,j),c} = \frac{\sum_{j\in V}\mathcal{F}(\alpha_{i,c},\alpha_{j,c})}{|V|}$, and $\beta^{in}_{l(i,j),c} = \frac{\sum_{i\in V}\mathcal{F}(\alpha_{i,c},\alpha_{j,c})}{|V|}$. A genetic approach has been used in this work for the optimization of modularity function. Another work that uses the genetic approach to overlapping community detection is GA-NET+ by Pizzuti.[84] GA-NET+ could detect overlapping communities using community score. The GA has been run on the line graph L(G) of the graph G.

The order statistics local optimization method (OSLOM)[85] detects clusters in networks and can handle various kinds of graph properties like edge direction, edge weights, overlapping communities, hierarchy, and network dynamics. It is based on the local optimization of a fitness function expressing the statistical significance of clusters with respect to random fluctuations, which is estimated with tools of extreme and order statistics. Baumes et al.[86] considered a community as a subset of nodes that induces a locally optimal subgraph with respect to a density function. Two different subsets with significant overlap can be locally optimal, which forms the basis to find overlapping communities.

Chen et al.[87] used a game-theoretic approach to address the issue of overlapping communities. Each node is assumed to be an agent trying to improve the utility by joining or leaving the community. The community of the nodes in Nash equilibrium is assumed to form the output of the algorithm. Utility of an agent is formulated as the combination of a gain and a loss function. To capture the idea of overlapping communities, each agent is permitted to select multiple communities. In another game-theoretic approach, Alvari et al.[88] proposed an algorithm consisting of two methods, PSGAME based on the Pearson correlation and NGGAME centered on the neighborhood similarity measure. Alvari et al.[89] proposed the dynamic game theory method (D-GT), which treated nodes as rational agents. These agents perform actions in an iterative and game-theoretic manner so as to maximize the total utility.

A link clustering-based GA, GaoCD, proposed by Shi et al.[90] detects overlapping communities. It determines clusters of links with the same features as links usually characterize distinctive relations amongst the nodes. Therefore, nodes fit into multiple communities. The procedure applies genetic operation to cluster links using partition density as an objective function. The run time for the method was calculated to be $O(gs(m + n))$, where g represents generation number; s is size of the population; m represents the number of edges; and n is the number of nodes. The OCDLCE (Ovelapping Community Detection by Local Community Expansion) algorithm[91] for overlapping communities was based on community expansion. The procedure has three stages, namely, community detection, community merging, and community refining. Bhat et al.[92] proposed a new density-based community-detection technique, OCMiner. It does not need the neighborhood threshold parameter to be fixed by the users, which makes it different from other density-based methods as computing a value for a threshold parameter is a major task for density-based methods. It automatically finds the neighborhood-threshold parameter for each node locally from the original network. Zhang et al.[93] have incorporated the idea of implicit link preferences in their model. This is performed using preference-based non-negative matrix factorization (PNMF). They have considered the fact that for any node, preferences of the friends' (neighbor) nodes are higher than any other non-neighbor nodes. Stochastic gradient-based descent procedure has been used in order to learn the parameters. They have performed experimental runs on several datasets to prove that their model leads to better modularity and F1 score values. In a clustering-based approach, Kozdoba et al.[94] presented an algorithm CLAGO (Cluster Aggregation for Overlapping Communities) where the process to find overlapping communities is divided into two parts. In the first part, the disjoint communities are found, and then, a random walk through them helps discover the overlapping communities. Whang et al.[95] have proposed neighborhood-inflated seed expansion (NISE), a seed-based expansion approach to detect communities. To find the effective seeds, two approaches, namely 'Graclus centers' and 'Spread hubs' have been proposed and used. For expansion of a seed, the algorithm of PageRank Clustering is deployed. Rees et al.[96] have used the idea of egonets and friendship groups in their work. Egonet is the viewpoint from any node. It consists of the vertices that are adjacent to the central ego node and the edges between those nodes. From the egonets, the friendship groups are found. The communities are thereafter discovered from these friendship groups. The run time of their algorithm was calculated to be $O(n(\log n)^2 + n^2 \log n)$.

Community Detection for Dynamic Networks

Dynamic networks are the networks in which the membership of the nodes of communities evolve or change over time. The task of community identification for dynamic networks has received relatively less attention than the static networks. Table 6 gives a summary of these methods, discussed later in the section.

The methods have been categorized into two classes by Bansal et al.[99] one designed for data that is evolving in real time, known as incremental or online community detection, and the other for data where all the changes of the network evolution are known *a priori*, known as offline community detection. Wolf et al.[100] proposed mathematical and computational

TABLE 6 | Community Detection for Dynamic Networks

Author(Algorithm)	Approach	Parameters	Static Algorithm Used	Code Availability
Bansal et al.[99]	Greedy agglomerative	Nodes, edges	Ref 34: Clauset et al.	No
Wolf et al.[100]	Mathematical framework	Some metagroup statistics	_	No
Tantipathananandh[101]	Graph coloring problem, heuristics	Individual cost, group cost, c-cost	_	No
Lin et al. (FacetNet)[102]	Iterative algorithm	Snapshot cost and temporal cost	_	http://www.yurulin.com/download/code/facetnet.html
Palla et al.[103]	Joint graphs	Auto-correlation, stationary parameter	Ref 73: CPM (Palla et al.)	No
Greene et al.[104]	Step communities	Time step t	_	No
He at al[105]	Dynamicity in the Louvain algorithm	Time t	Ref 35: Blondel et al. (Louvain method)	No
Dinh et al.[106]	Modularity maximization for dynamic networks	$\Delta G^{(t)}$: change in graph snapshot, $\Delta G^{(t)}$: community structure at time t, degree	_	No
Nguyen et al.[107]	QCA (Quick community adaptation)	Nodes, edges	_	No
Takaffoli[108]	Events: split, survive, dissolve, merge, and form	Community similarity	_	No
Kim et al.[109]	Nano-communities, quasi-clique-by-clique	Temporal cost, snapshot cost	_	No
Chi et al.[110]	Evolutionary spectral clustering	Temporal cost, snapshot cost	_	No
Folino et al. (DYNMOGA)[111]	Multiobjective genetic algorithm	Community score, NMI	_	http://staff.icar.cnr.it/pizzuti/codes.html
Kim et al. (CHRONICLE)[112]	Two-stage clustering	Cosine similarity general similarity(GS)	_	No

CPM, Clique Percolation Method.

formulations for the analysis of dynamic communities on the basis of social interactions occurring in the network. Tantipathananandh et al.[101] made assumptions about the individual behavior and group membership. They framed the objective as an optimization problem by formulating three cost functions, namely, i-cost, g-cost, and c-cost. Graph coloring and heuristics-based approaches were deployed. FacetNet, proposed by Lin et al.,[102] is a unified framework to study the dynamic evolutions of communities. The community structure at any time includes the network data as well as the previous history of the evolution. They have used a cost function and proposed an iterative algorithm that converges to an optimal solution. Palla et al.[103] conducted experiments on two diverse datasets of phone call network and collaboration network to find time dependence. After building joint graphs for two time steps, the CPM algorithm[74] was applied. They have used an auto-correlation function to find overlap among two states of a community and a stationarity parameter that denotes the average correlation of various states. Greene et al.[104] proposed a heuristic technique for the identification of dynamic communities in the network data. They represented the dynamic network graph as an aggregation of time step graphs. Step communities represent the dynamic communities at a particular time. The algorithm begins with the application of a static community-detection algorithm on the graph. In the subsequent steps, dynamic communities are created for each step, and Jaccard similarity is calculated. They have also generated a benchmark dataset for experimental work. The algorithm by Bansal et al.[99] involves the addition or deletion of edges in the network. The algorithm is built on the greedy agglomerative technique of the modularity-based method earlier proposed in the work of Clauset et al.[34]. He et al.[105] improvised the Louvain method[35] to include the concept of dynamicity in the formation of communities. A key point in their algorithm is to make use of previously detected communities at time $t-1$ to identify the communities at time. Dinh et al.[106] proposed A^3 CS, an adaptive approximation algorithm for community detection that uses the power-law distribution and achieves approximation guarantees for the NP-hard (non-deterministic polynomial-time hard) modularity maximization problem, particularly on dynamic networks.

Nguyen et al.[107] have attempted to identify disjoint community structure in dynamic social networks. An adaptive modularity-based framework, Quick Community Adaptation (QCA), is proposed. The method finds and traces the progress of network communities in dynamic online social networks. Takaffoli et al.[108] have proposed a two-step approach to community detection. In the first step, the communities extracted at different time instances are compared using weighted bipartite matching. Next, a 'meta' community is constructed, which is defined as a series of similar communities at various time instances. The five events to capture the changes to community are split, survive, dissolve, merge, and form. A similarity function is used to calculate the similarity between two communities, and a community matching algorithm has been employed thereafter. Kim et al.[109] proposed a particle-and-density-based evolutionary clustering method for the discovery of communities in dynamic networks. Their approach is grounded on the assumption that a network is built of a number of particles termed as nano-communities, where each community is further made up of particles termed as quasi-clique-by-clique (l-KK). The density-based clustering method uses a cost-embedding technique and optimal modularity method to ensure temporal smoothness even when the number of cluster varies. They have used an information theory-based mapping technique to recognize the stages of the community, i.e., evolving, forming, or dissolving. Their method improves accuracy and is time efficient as compared to the FacetNet method proposed earlier. In another approach proposed by Chi et al.,[110] two frameworks for evolutionary spectral clustering have been proposed, namely, preserving cluster quality (PCQ) and preserving cluster membership (PCM). In this work, the temporal smoothness is ensured by some terms in the clustering cost functions. These two frameworks combine the processes of community extraction and community evolution. They use a cost function that consists of the snapshot and temporal cost. The clustering quality of any partition determines the snapshot cost, while the temporal cost definition varies for each of the frameworks. For the PCQ framework, the temporal cost is decided by the cluster quality when the current partition is applied to the historic data. In PCM, the difference between the current and the historic partition gives the temporal cost. Both the frameworks proposed can tackle the change in the number of clusters. In their work, dynamic multi-objective genetic algorithm (DYNMOGA), Folino et al.[111] have used a GA-based approach to dynamic community detection. They attempt to achieve temporal smoothness by multi-objective optimization, i.e., the maximization of snapshot quality (community score is used) and minimization of temporal cost (here, normalized mutual information NMI is used).

Kim et al.,[112] in their method CHRONICLE, have performed two-stage clustering, and the method

can detect clusters of path group type in addition to the single path type clusters. In the first stage of the algorithm, called as $CHRONICLE_{1st}$, the cosine similarity measure is used. In the second stage of the algorithm, the measure proposed and used is general similarity (GS). It is a combination of the two measures, structural affinity and weight affinity.

STANDARD DATASETS FOR COMMUNITY DETECTION

The datasets most frequently employed for experimental studies in community-detection research in the literature can be divided into real and artificial (generated) datasets, which are also the benchmark datasets as given in Table 7.

Real-time datasets like the Karate club network by Zachary[113] represent the relationships between 34 members of a karate club over a period of 2 years. Dolphin Social Network depicts the social interactions and behavior of bottlenose dolphins for a period of 7 years as studied by Lusseau et al.[114] The American College Football Network[19] dataset consists of the football teams in America. There are also other realtime datasets like the Southern women dataset[115] etc. Amongst the artificial datasets, one has been created by Girvan Newman,[19] where 128 vertices lead to four communities. An algorithm to generate benchmark datasets was proposed by Lancichinetti et al., known as the LFR benchmark.[116] A number of datasets are also available at the web pages, www-personal.umich.edu/~mejn/netdata[117] and https://snap.stanford.edu/data/.[118]

SOME POTENTIAL APPLICATIONS OF COMMUNITY DETECTION

With the enormous growth of the SNS and their users, the graphs representing these sites are becoming very complex and, hence, are difficult to visualize and understand. Communities can be considered a summary of the whole network, thus making the network easy to comprehend. The discovery of these communities in social networks can be useful in various applications. Some of the applications where community detection is useful are briefly described below.

Trend Analysis in Citation Networks

In academia, citation networks are constructed by citation relationships between papers and researchers. Communities in a citation network represent related papers on a single topic or researchers working on the same topic. Here, the network is grouped into communities where a community is represented as papers on that topic or researchers working on that topic. Detecting these communities in citation networks can provide information about various core topics in which papers are published as well about the researchers working io these topics.

Improving Recommender Systems with Community Detection

Recommender systems (RS) use data of similar users or similar items to generate recommendations. This is analogous to the identification of groups or similar nodes in a graph. Hence, community detection holds an immense potential for recommendation algorithms. Cao et al.[119] have used a community detection-based approach to improve the traditional collaborative filtering process of RS. The process starts with the mapping of user-item matrix to user similarity structure. On this matrix, a discrete particle swarm optimization (PSO) algorithm is applied to detect communities. The items are then recommended to the user based on the discovered communities.

Evolution of Communities in Social Media

With the increase in the number of SNS, the focus and scope of sites are expanding and diversifying. In addition to common sites like Facebook, Twitter, MySpace, and Bebo, other sites like Flickr for photo sharing have also come up. The analysis of the tweet–retweet and the follower–followee network on Twitter provides an insight into the community structure existing in the Twitter network. Sentiment analysis of the tweets may be performed as an intermediary step to find the general nature of the tweets, and then, community-detection algorithms may be applied to help deduce the structure of communities. Zalmout et al.[120] applied the community-detection algorithm to the UK political tweets dataset. Community question answering (CQA) has been used by Zhang et al.[121] to discover overlapping communities in dynamic networks based on user interactions.

DISCUSSION

A number of community-detection algorithms have been discussed in this article. The method to be used at any point may be decided on the basis of the structure of the network and kind of communities. An

TABLE 7 | Standard Datasets for Community Detection

Type	Dataset Name	Link for Download
Real dataset	Zachary karate club	www.personal.umich.edu/~mejn/netdata
	Dolphin dataset	www.personal.umich.edu/~mejn/netdata
	American college football network	www.personal.umich.edu/~mejn/netdata
	Southern women dataset	http://networkdata.ics.uci.edu/netdata/html/davis.html
Benchmark dataset	Girvan and Newman[19]	Not available
	Lancichinetti et al.[116]	https://sites.google.com/site/andrealancichinetti/software

algorithm that may be time and space efficient and hence suitable for a particular network type may prove expensive in another case. Depending upon various parameters and network topological characteristics, the output of a method may differ in different scenarios. For example, a method designed for a directed network may not run efficiently on an undirected network. The paper discussed the categories of algorithms, the kind of input they require, and the type of communities detected. The earlier methods of Newman and Girvan emphasized the basic property of edge-betweenness and modularity. A large number of algorithms optimized the modularity factor and could create a sufficient drop in the run time. However, a majority of the algorithms did not consider the overlapping and dynamicity property of the nodes in a network, which was covered later. Most of the GA-based approaches do not require the exact number of groups in advance. The algorithms based on GA try to optimize the fitness function, e.g., modularity giving the best community structure, but computationally, they may be expensive. Graph partitioning-based methods also divide the graph in a manner such that there are few or no connections between the partitions. On the contrary, the entities generally belong to multiple communities in a real-world social network; hence, there occurs vertices overlap. In that case, the overlapping communities-based algorithms result in better partitions than the disjoint community-detection methods. Taking the time factor into account, the label propagation-based methods are time efficient. Most of these procedures are based on the network structure of the graphs and do not require the parameters like number of communities or their sizes.

The latest buzz in recent years has been large-scale networks and big data. These factors have been taken into account in relatively recent works that cater to these kinds of network structures. Some traditional algorithms have been extended to work for big data as well. The evolving membership of the communities (or nodes) has been considered in the dynamic network-based approaches. The standard community-detection approaches do not make use of the semantic content in graphs; however, in some cases like topic-based division, this information is valuable. The methods based on semantic community detection may be used in these situations. The field of community detection is evolving, multifaceted, and versatile in nature and, hence, may be and has been applied to myriad fields.

CONCLUSION

The area of community detection holds a vast potential for the discovery of communities in today's exponentially growing social networks. The basic concepts of social networks, community structure, and methods for grouping similar items are presented in this paper. A category-wise compiled review of the state-of-the-art algorithms for community detection in social networks is presented. Application of the algorithms to detect communities in actual networks of Facebook, Twitter, and LinkedIn etc. can provide a substantial amount of information for myriad purposes. The discovery and analysis of communities is used in biology, sociology ,and many other branches of science. Such information may prove to be useful for commercial, educational, or developmental purposes. Details about various datasets used by the existing algorithms in the literature along with some potential applications of community detection for social networks are also included in the paper.

ACKNOWLEDGMENTS

The authors duly acknowledge the UGC MRP Grant No. [42-139/2013 (SR)] by University Grants Commission (UGC) of India for partially supporting this research work. The authors also acknowledge the INSPIRE Fellowship No. IF131087 by Department of Science & Technology (DST), India.

REFERENCES

1. Özturk K. Community detection in social networks. Msc. Thesis. Graduate School of Natural and Applied Sciences, Middle East Technical University, 2014.
2. Tang L, Liu H. *Community Detection and Mining in Social Media, Synthesis Lectures on Data Mining and Knowlegde Discovery*. California: Morgan and Claypool; 2010.
3. Fasmer EE. Community detection in social networks. Master Thesis. Department of Informatics, University of Bergen, 2015.
4. Barabási A-L, Albert R. Emergence of scaling in random networks. *Science* 1999, 286:509–512. doi:10.1126/science.286.5439.509.
5. McPherson M, Lovin LS, Cook JM. Birds of a feather: homophily in social networks. *Annu Rev Sociol* 2001, 27:415–444. doi:10.1146/annurev.soc.27.1.415.
6. Luce RD, Perry AD. A method of matrix analysis of group structure. *Psychometrika* 1949, 14:95–116. doi:10.1007/BF02289146.
7. Watts DJ, Strogatz SH. Collective dynamics of 'small-world'networks. *Nature* 1998, 393:440–442. doi:10.1038/30918.
8. Fortunato S. Community detection in graphs. *Phys Rep* 2010, 486:75–174. doi:10.1016/j.physrep.2009.11.002.
9. Coscia M, Giannotti F, Pedreschi D. A classification for community discovery methods in complex networks. *Stat Anal Data Min* 2011, 4:512–546. doi:10.1002/sam.10133.
10. Fortunato S, Castellano C. Community structure in graphs. In: *Computational Complexity*. New York: Springer; 2012, 490–512. doi:10.1007/978-1-4614-1800-9_33.
11. Porter MA, Onnela J-P, Mucha PJ. Communities in networks. *Notices Amer Math Soc* 2009, 56:1082–1097.
12. Danon L, Diaz-Guilera A, Duch J, Arenas A. Comparing community structure identification. *J Stat Mech Theory Exp* 2005, 09:P09008. doi:10.1088/1742-5468/2005/09/P09008.
13. Plantié M, Crampes M. Survey on social community detection. In: *Social Media Retrieval Computer Communications and Networks*. London: Springer-Verlag; 2013, 65–85. doi:10.1007/978-1-4471-4555-4_4.
14. Kernighan BW, Lin S. An efficient heuristic procedure for partitioning graphs. *Bell Syst Tech J* 1970, 49:291–307. doi:10.1002/j.1538-7305.1970.tb01770.x.
15. Newman M. Community detection and graph partitioning. *Europhys Lett* 2013, 103:28003. doi:10.1209/0295-5075/103/28003.
16. Karrer B, Newman M. Stochastic blockmodels and community structure in networks. *Phys Rev E* 2011, 83:016107. doi:10.1103/PhysRevE.83.016107.
17. Fiedler M. Algebraic connectivity of graphs. *Czechoslov Math J* 1973, 23:298–305.
18. Pothen A, Simon HD, Liou K-P. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J Matrix Anal Appl* 1990, 11:430–452. doi:10.1137/0611030.
19. Girvan M, Newman M. Community structure in social and biological networks. *Proc Natl Acad Sci* 2002, 99:7821–7826. doi:10.1073/pnas.122653799.
20. Newman M, Girvan M. Finding and evaluating community structure in networks. *Phys Rev E* 2004, 69:026113. doi:10.1103/PhysRevE.69.026113.
21. Rattigan MJ, Maier M, Jensen D. Graph clustering with network structure indices. In: *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007, 783–790. ACM, doi: 10.1145/1273496.1273595.
22. Chen J, Yuan B. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics* 2006, 22:2283–2290. doi:10.1093/bioinformatics/btl370.
23. Holme P, Huss M, Jeong H. Subnetwork hierarchies of biochemical pathways. *Bioinformatics* 2003, 19:532–538.
24. Pinney JW, Westhead DR. Betweenness-based decomposition methods for social and biological networks. In: *Interdisciplinary Statistics and Bioinformatics*. UK: Leeds University Press; 2006, 87–90.
25. Gregory S. An algorithm to find overlapping community structure in networks. In: *Knowledge Discovery in Databases*. PKDD Berlin Heidelberg: Springer-Verlag; 2007, LNAI 4702:91–102. doi:10.1007/978-3-540-74976-9_12.

26. Guimera R, Danon L, Diaz-Guilera A, Giralt F, Arenas A. Self-similar community structure in a network of human interactions. *Phys Rev E* 2003, 68:065103. doi:10.1103/PhysRevE.68.065103.
27. Arenas A, Danon L, Diaz-Guilera A, Gleiser PM, Guimera R. Community analysis in social networks. *Eur Phys J B* 2004, 38:373–380. doi:10.1140/epjb/e2004-00130-1.
28. Tyler JR, Wilkinson DM, Huberman BA. E-mail as spectroscopy: automated discovery of community structure within organizations. *Inf Soc* 2005, 21:143–153.
29. Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. *Proc Natl Acad Sci U S A* 2004, 101:2658–2663. doi:10.1073/pnas.0400054101.
30. Moon S, Lee J-G, Kang M, Choy M, Lee J-w. Parallel community detection on large graphs with MapReduce and GraphChi. *Data Knowl Eng* 2015, In Press. doi:10.1016/j.datak.2015.05.001.
31. Newman M. Fast algorithm for detecting community structure in networks. *Phys Rev E* 2004, 69:066133. doi:10.1103/PhysRevE.69.066133.
32. Newman M. Analysis of weighted networks. *Phys Rev E* 2004, 70:056131. doi:10.1103/PhysRevE.70.056131.
33. Newman M. Modularity and community structure in networks. *Proc Natl Acad Sci* 2006, 103:8577–8582. doi:10.1073/pnas.0601602103.
34. Clauset A, Newman ME, Moore C. Finding community structure in very large networks. *Phys Rev E* 2004, 70:066111. doi:10.1103/PhysRevE.70.066111.
35. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *J Stat Mech Theory Exp* 2008, 2008:10008. doi:10.1088/1742-5468/2008/10/P10008.
36. Guimera R, Sales-Pardo M, Amaral LAN. Modularity from fluctuations in random graphs and complex networks. *Phys Rev E* 2004, 70:025101. doi:10.1103/PhysRevE.70.025101.
37. Zhou Z, Wang W, Wang L. Community detection based on an improved modularity. *Pattern Recognition* 2012, CCIS 321:638–645. doi:10.1007/978-3-642-33506-8_78.
38. Duch J, Arenas A. Community detection in complex networks using extremal optimization. *Phys Rev E* 2005, 72:027104. doi:10.1103/PhysRevE.72.027104.
39. Ye Z, Hu S, Yu J. Adaptive clustering algorithm for community detection in complex networks. *Phys Rev E* 2008, 78:046115. doi:10.1103/PhysRevE.78.046115.
40. Wahl S, Sheppard J. Hierarchical fuzzy spectral clustering in social networks using spectral characterization. In: *The Twenty-Eighth International Flairs Conference*, 2015, 305–310.
41. Falkowski T, Barth A, Spiliopoulou M. DENGRAPH: A density-based community detection algorithm. In: *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Fremont, CA, 2007, 112–115. doi:10.1109/WI.2007.74.
42. Dongen SV. Graph clustering by flow simulation. PhD thesis, University of Utrecht, 2000.
43. Nikolaev AG, Razib R, Kucheriya A. On efficient use of entropy centrality for social network analysis and community detection. *Soc Networks* 2015, 40:154–162. doi:10.1016/j.socnet.2014.10.002.
44. Steinhaeuser K, Chawla NV. Identifying and evaluating community structure in complex networks. *Pattern Recogn Lett* 2010, 31:413–421. doi:10.1016/j.patrec.2009.11.001.
45. Chen Y, Huang C, Zhai K. Scalable community detection algorithm with MapReduce. *Commun ACM* 2009, 53:359–366. doi:10.1147/JRD.2013.2251982.
46. Pizzuti C. GA-Net: a genetic algorithm for community detection in social networks. In: *Parallel Problem Solving from Nature–PPSN X*. Berlin Heidelberg: Springer-Verlag; 2008, LNCS 5199:1081–1090. doi:10.1007/978-3-540-87700-4_107.
47. Pizzuti C. A multiobjective genetic algorithm to find communities in complex networks. *IEEE Trans Evol Comput* 2012, 16:418–430. doi:10.1109/TEVC.2011.2161090.
48. Hafez AI, Ghali NI, Hassanien AE, Fahmy AA. Genetic algorithms for community detection in social networks. In: *12th International Conference on Intelligent Systems Design and Applications (ISDA)*, IEEE, 2012, 460–465.doi:10.1109/ISDA.2012.6416582.
49. Mazur P, Zmarzlowski K, Orlowski AJ. A genetic algorithms approach to community detection. *Acta Phys Pol A* 2010, 117:703–705.
50. Liu X, Li D, Wang S, Tao Z. Effective algorithm for detecting community structure in complex networks based on GA and clustering. In: *International Conference on Computational Science (ICCS 07)*, Springer, 2007, 657–664. doi: 10.1007/978-3-540-72586-2_95.
51. Tasgin M, Herdagdelen A, Bingol H. Community detection in complex networks using genetic algorithms. 2007, arXiv preprint arXiv: 0711.0491.
52. Zadeh PM, Kobti Z. A multi-population cultural algorithm for community detection in social networks. *Procedia Comput Sci* 2015, 52:342–349. doi:10.1016/j.procs.2015.05.105.
53. Nicosia V, Mangioni G, Carchiolo V, Malgeri M. Extending the definition of modularity to directed graphs with overlapping communities. *J Stat Mech Theory Exp* 2009, 3:P03024. doi:10.1088/1742-5468/2009/03/P03024.

54. Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Phys Rev E* 2007, 76:036106. doi:10.1103/PhysRevE.76.036106.

55. Xie J, Szymanski BK. Towards linear time overlapping community detection in social networks. In: *Advances in Knowledge Discovery and Data Mining*. Berlin Heidelberg: Springer-Verlag; 2012, LNAI 7302:25–36.

56. Hu W. Finding statistically significant communities in networks with weighted label propagation. *Soc Netw* 2013, 2:138–146. doi:10.4236/sn.2013.23012.

57. Gregory S. Finding overlapping communities in networks by label propagation. *New J Phys* 2010, 12:103018. doi:10.1088/1367-2630/12/10/103018.

58. Xie J, Szymanski BK. Labelrank: a stabilized label propagation algorithm for community detection in networks. In: *IEEE Network Science Workshop (NSW)*, 2013, 138–143.

59. Wu Z-H, Lin Y-F, Gregory S, Wan H-Y, Tian S-F. Balanced multi-label propagation for overlapping community detection in social networks. *J Comput Sci Technol* 2012, 27:468–479. doi:10.1007/s11390-012-1236-x.

60. Xie J, Chen M, Szymanski BK. LabelrankT: incremental community detection in dynamic networks via label propagation. In: *Proceedings of the Workshop on Dynamic Networks Management and Mining(DyNetMM)*, 2013, ACM, 25–32, doi:10.1145/2489247.2489249.

61. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. *J Mach Learn Res* 2003, 3:993–1022.

62. Xin Y, Yang J, Xie Z-Q. A semantic overlapping community detection algorithm based on field sampling. *Expert Syst Appl* 2015, 42:366–375. doi:10.1016/j.eswa.2014.07.009.

63. Xin Y, Yang J, Xie Z-Q, Zhang J-P. An overlapping semantic community detection algorithm base on the ARTs multiple sampling models. *Expert Syst Appl* 2015, 42:3420–3432. doi:10.1016/j.eswa.2014.11.029.

64. Xia Z, Bu Z. Community detection based on a semantic network. *Knowl-Based Syst* 2012, 26:30–39. doi:10.1016/j.knosys.2011.06.014.

65. Ding Y. Community detection: topological vs. topical. *J Informetr* 2011, 5:498–514. doi:10.1016/j.joi.2011.02.006.

66. Erétéo G, Gandon F, Buffa M. Semtagp: Semantic community detection in folksonomies. In: *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, IEEE Computer Society, 2011, 324–331, doi:10.1109/WI-IAT.2011.98.

67. Zhao Z, Feng S, Wang Q, Huang JZ, Williams GJ, Fan J. Topic oriented community detection through social objects and link analysis in social networks. *Knowl-Based Syst* 2012, 26:164–173. doi:10.1016/j.knosys.2011.07.017.

68. Abdelbary HA, El-Korany A. Semantic topics modeling approach for community detection. *Int J Comput Appl* 2013, 81:50–58. doi:10.5120/14020-2177.

69. Deerwester SC, Dumais ST, Landauer TK, Furnas GW, Harshman RA. Indexing by latent semantic analysis. *J Am Soc Inform Sci* 1990, 41:391–407.

70. Nguyen T, Phung D, Adams B, Tran T, Venkatesh S. Hyper-community detection in the blogosphere. In: *Proceedings of second ACM SIGMM workshop on Social media, WSM*, ACM, 2010, 21–26. doi:10.1145/1878151.1878159.

71. Natarajan N, Sen P, Chaoji V. Community detection in content-sharing social networks. In: *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining(ASONAM)*, ACM, 2013, 82–89. doi:10.1145/2492517.2492546.

72. Amelio A, Pizzuti C. Overlapping community discovery methods: a survey. In: *Social Networks: Analysis and Case Studies*. Weinheim: Springer-Verlag; 2014, Lecture Notes in Social Networks: 105–125. doi:10.1007/978-3-7091-1797-2_6.

73. Xie J, Kelley S, Szymanski BK. Overlapping community detection in networks: the state-of-the-art and comparative study. *ACM Comput Surv* 2013, 45:1–35. doi:10.1145/2501654.2501657.

74. Palla G, Derenyi I, Farhas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 2005, 435:814–818. doi:10.1038/nature03607.

75. Lancichinetti A, Fortunato S, Kertész J. Detecting the overlapping and hierarchical community structure in complex networks. *New J Phys* 2009, 11:033015. doi:10.1088/1367-2630/11/3/033015.

76. Du N, Wu B, Pei X, Wang B, Xu L. Community detection in large-scale social networks. In: *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, ACM, 2007, 16–25. doi:10.1145/1348549.1348552.

77. Shen H, Cheng X, Cai K, Hu M-B. Detect overlapping and hierarchical community structure in networks. *Phys A Stat Mech Appl* 2009, 388:1706–1712. doi:10.1016/j.physa.2008.12.021.

78. Evans T, Lambiotte R. Line graphs, link partitions, and overlapping communities. *Phys Rev E* 2009, 80:016105. doi:10.1103/PhysRevE.80.016105.

79. Evans T, Lambiotte R. Line graphs of weighted networks for overlapping communities. *Euro Phys J B* 2010, 77:265–272. doi:10.1140/epjb/e2010-00261-8.

80. Evans TS. Clique graph and overlapping communities. *J Stat Mech Theory Exp* 2010, 12:12037. doi:10.1088/1742-5468/2010/12/P12037.
81. Lee C, Reid F, McDaid A, Hurley N. Detecting highly overlapping community structure by greedy clique expansion. 2010, arXiv preprint arXiv:1002.1827.
82. Gregory S. A fast algorithm to find overlapping communities in networks. In: *ECML PKDD : European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, Springer, 2008, 408–423. doi:10.1007/978-3-540-87479-9_45.
83. Gregory S. Finding overlapping communities using disjoint community detection algorithms. In: *Complex Networks*. Berlin Heidelberg: Springer-Verlag; 2009, SCI 207:47–61. doi:10.1007/978-3-642-01206-8_5.
84. Pizzuti C. Overlapped community detection in complex networks. In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, ACM, 2009, 859-866. doi:10.1145/1569901.1570019.
85. Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S. Finding statistically significant communities in networks. *PLoS One* 2011, 6:e18961. doi:10.1371/journal.pone.0018961.
86. Baumes J, Goldberg MK, Krishnamoorthy MS, Magdon-Ismail M, Preston N. Finding communities by clustering a graph into overlapping subgraphs. In: *IADIS International Conference on Applied Computing*, 2005, 97–104.
87. Chen W, Liu Z, Sun X, Wang Y. A game-theoretic framework to identify overlapping communities in social networks. *Data Min Knowl Disc* 2010, 21:224–240. doi:10.1007/s10618-010-0186-6.
88. Alvari H, Hashemi S, Hamzeh A. Detecting overlapping communities in social networks by game theory and structural equivalence concept. In: *Artificial Intelligence and Computational Intelligence*. Berlin Heidelberg: Springer-Verlag; 2011, LNAI 7003:620–630. doi:10.1007/978-3-642-23887-1_79.
89. Alvari H, Hajibagheri A, Sukthankar G. Community detection in dynamic social networks: A game-theoretic approach. In: *Proceedings of Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, 2014, 101-107.
90. Shi C, Cai Y, Fu D, Dong Y, Wu B. A link clustering based overlapping community detection algorithm. *Data Knowl Eng* 2013, 87:394–404. doi:10.1016/j.datak.2013.05.004.
91. Xing Y, Meng F, Zhou Y, Zhou R. Overlapping community detection by local community expansion. *J Inform Sci Eng* 2015, 31:1213–1232.
92. Bhat SY, Abulaish M. OCMiner: a density-based overlapping community detection method for social networks. *Intelli Data Anal* 2015, 19:1–31. doi:10.3233/IDA-150751.
93. Zhang H, King I, Lyu MR. Incorporating implicit link preference into overlapping community detection. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, 396–402.
94. Kozdoba M, Mannor S. Overlapping community detection by online cluster aggregation. 2015, arXiv preprint arXiv:1504.06798 .
95. Whang JJ, Gleich DF, Dhillon IS. Overlapping community detection using seed set expansion. In: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management(CIKM)*, ACM, San Francisco, CA, Oct 27-Nov 1 2013, 2099–2108. doi:10.1145/2505515.2505535.
96. Rees BS, Gallagher KB. Overlapping community detection by collective friendship group inference. In: *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, 2010, 375-379, doi:10.1109/ASONAM.2010.28.
97. Everett MG, Borgatti SP. Analyzing clique overlap. *Connnections* 1998, 21:49–61.
98. Adamcsek B, Palla G, Farkas IJ, Dere'nyi I, Vicsek T. CFinder: locating cliques and overlapping modules in biological networks. *Bioinformatics* 2006, 22:1021–1023. doi:10.1093/bioinformatics/btl039.
99. Bansal S, Bhowmick S, Paymal P. Fast community detection for dynamic complex networks. In: *Complex Networks*. Berlin Heidelberg: Springer-Verlag; 2011, CCIS 116:196–207. doi:10.1007/978-3-642-25501-4_20.
100. Berger-Wolf TY, Saia J. A framework for analysis of dynamic social networks. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2006, 523–528, doi:10.1145/1150402.1150462.
101. Tantipathananandh C, Berger-Wolf T, Kempe D. A framework for community identification in dynamic social networks. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2007, 717–726. doi:10.1145/1281192.1281269.
102. Lin Y-R, Chi Y, Zhu S, Sundaram H, Tseng BL. FacetNet: a framework for analyzing communities and their evolutions in dynamic networks. In: *Proceedings of the 17th International Conference on World Wide Web*, ACM, 2008, 685–694. doi:10.1145/1367497.1367590.
103. Palla G, Barabási A-L, Vicsek T. Quantifying social group evolution. *Nature* 2007, 446:664–667. doi:10.1038/nature05670.
104. Greene D, Doyle D, Cunningham P. Tracking the evolution of communities in dynamic social networks. In: *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, 2010, 176–183. doi:10.1109/ASONAM.2010.17.

105. He J, Chen D. A fast algorithm for community detection in temporal network. *Phys A Stat Mech Appl* 2015, 429:87–94. doi:10.1016/j.physa.2015.02.069.

106. Dinh TN, Nguyen NP, Thai MT. An adaptive approximation algorithm for community detection in dynamic scale-free networks. In: *32th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, IEEE Press, 2013, 55–59, doi:10.1109/INFCOM.2013.6566734.

107. Nguyen NP, Dinh TN, Shen Y, Thai MT. Dynamic social community detection and its applications. *PLoS One* 2014, 9:e91431. doi:10.1371/journal.pone.0091431.

108. Takaffoli M, Sangi F, Fagnan J, Zäıane OR. Community evolution mining in dynamic social networks. *Procedia Soc Behav Sci* 2011, 22:49–58.

109. Kim M-S, Han J. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc VLDB Endow* 2009, 2:622–633. doi:10.14778/1687627.1687698.

110. Chi Y, Song X, Zhou D, Hino K, Tseng BL. On evolutionary spectral clustering. *ACM Trans Knowl Discov Data* 2009, 3:17. doi:10.1145/1631162.1631165.

111. Folino F, Pizzuti C. An evolutionary multiobjective approach for community discovery in dynamic networks. *IEEE Trans Knowl Data Eng* 2014, 26:1838–1852.

112. Kim M-S, Han J. CHRONICLE: A two-stage density-based clustering algorithm for dynamic networks. In: *Discovery Science,12th International Conference*, DS Springer, 2009, 152–167. doi:10.1007/978-3-642-04747-3_14.

113. Zachary WW. An information flow model for conflict and fission in small groups. *J Anthropol Res* 1977, 33:452–473.

114. Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behav Ecol Sociobiol* 2003, 54:396–405. doi:10.1007/s00265-003-0651-y.

115. Davis A, Gardner BB, Gardner MR. *Deep South: A Social Anthropological Study of Caste and Class*. Columbia: University of South Carolina Press; 2009.

116. Lancichinetti A, Fortunato S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys Rev E* 2009, 80:016118. doi:10.1103/PhysRevE.80.016118.

117. Newman M, Network Data, http://www-personal.umich.edu/~mejn/netdata/. (Accessed Sept. 23, 2015).

118. Leskovec J, Krevl A. SNAP datasets: Stanford Large Network Dataset Collection, https://snap.stanford.edu/data/. (Accessed Sept. 23, 2015).

119. Cao C, Ni Q, Zhai Y. An improved collaborative filtering recommendation algorithm based on community detection in social networks. In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ACM, 2015, 1–8, doi:10.1145/2739480.2754670.

120. Zalmout N, Ghanem M. Multidimensional community detection in Twitter. In: *8th International Conference on Internet Technology and Secured Transactions (ICITST)*, 2013, IEEE, 83–88. doi:10.1109/ICITST.2013.6750167.

121. Zhang Z, Li Q, Zeng D, Gao H. Extracting evolutionary communities in community question answering. *J Assoc Inform Sci Tech* 2014, 65:1170–1186. doi:10.1002/asi.23003.

