Journal of Network and Computer Applications 108 (2018) 87–111

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Review

# Community detection in networks: A multidisciplinary review

Muhammad Aqib Javed [a,*], Muhammad Shahzad Younis [a], Siddique Latif [a,b], Junaid Qadir [b], Adeel Baig [a,c]

[a] National University of Sciences and Technology (NUST), Islamabad, Pakistan
[b] Information Technology University (ITU), Punjab, Lahore, Pakistan
[c] College of Computer and Information Systems (CCIS), Al Yamamah University, Riyadh, Saudi Arabia

ARTICLE INFO

*Keywords:*
Community detection
Clustering algorithms
Modularity
Anomaly detection
Online social networks

ABSTRACT

The modern science of networks has made significant advancement in the modeling of complex real-world systems. One of the most important features in these networks is the existence of community structure. In recent years, many community detection algorithms have been proposed to unveil the structural properties and dynamic behaviors of networks. In this study, we attempt a contemporary survey on the methods of community detection and its applications in the various domains of real life. Besides highlighting the strengths and weaknesses of each community detection approach, different aspects of algorithmic performance comparison and their testing on standard benchmarks are discussed. The challenges faced by community detection algorithms, open issues and future trends related to community detection are also postulated. The main goal of this paper is to put forth a review of prevailing community detection algorithms that range from traditional algorithms to state of the art algorithms for overlapping community detection. Algorithms based on dimensionality reduction techniques such as non-negative matrix factorization (NMF) and principal component analysis (PCA) are also focused. This study will serve as an up-to-date report on the evolution of community detection and its potential applications in various domains from real world networks.

## 1. Introduction

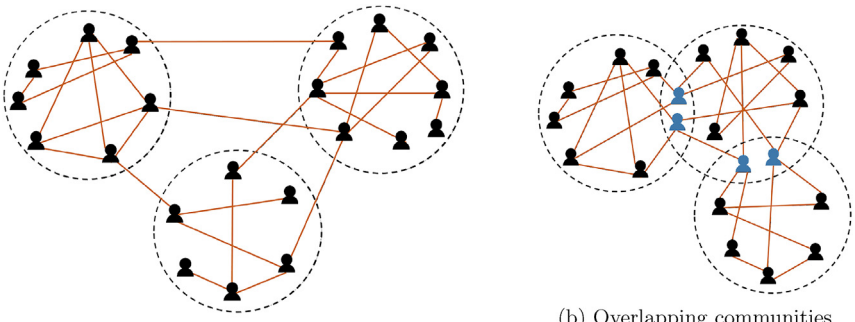
Network science is the study of networks using mathematical theory, which focuses on the analysis and characterization of network behavior. The study of networks has witnessed significant research, with the focus to understand and evaluate the statistical properties of large-scale networks (Newman, 2003). Networks are typically represented by graphs in which a group of nodes (vertices) have connections between them (edges). Graph theory is the mathematics of networks which is being used for the modeling of graphs (Stam, 2014). Erdös and Rényi (1959) introduced random graphs in which the edge probability between two nodes is same for any other pair. But real-world networks are not random graphs because they reveal a good order of patterns. One of the pertinent characteristics of real-world networks is that they have community structures, which can be excellently modeled by using graphs (Fortunato, 2010; Singh, 2014). Generally, community or cluster is defined as *a group of nodes having similar affiliations different to rest of the network* (Yang et al., 2010). Identifying community structures is a step towards the understanding of different structures of networks (Newman and Girvan, 2004) with applications in a number of fields such as online social networks and all of physical and life sciences (Lewis, 2011).

Community detection refers to the procedure of identifying groups of interacting vertices (i.e., nodes) in a network depending upon their structural properties (Yang et al., 2013; Kelley et al., 2012). Many algorithms for community detection have been developed, using techniques and tools from different disciplines such as biology, physics, social sciences, applied mathematics and computer sciences (Lancichinetti and Fortunato, 2009). However, a single community detection algorithm fails to perform in all kind of networks (Plantié and Crampes, 2013; Yang et al., 2016) because there is a wide variety of complex networks which are generated from different processes. The algorithmic biases improve performance on one type of network and reduce on another type of networks which is a natural trade-off.

Community detection algorithms strongly depend on the topology of the networks since the networks may be static or dynamic. In a static network, community discovery is an easy task as compared to the dynamic network. There exist a number of community detection

* Corresponding author.
*E-mail address:* 14mseemjaved@seecs.edu.pk (M.A. Javed).

https://doi.org/10.1016/j.jnca.2018.02.011
Received 16 September 2017; Received in revised form 7 January 2018; Accepted 19 February 2018
Available online 24 February 2018
1084-8045/



(a) Disjoint communities (b) Overlapping communities

**Fig. 1.** The left subfigure represents disjoint network community structures while the right subfigure shows (three) overlapping communities. Adapted from: (Fortunato and Hric, 2016).

approaches in static networks (Fortunato, 2010; Flake et al., 2000; Gregory, 2007; Gao et al., 2010), which are mostly optimization based algorithms that seek an optimal solution according to the defined objective function (Shi and Malik, 2000; Hagen and Kahng, 1992; Pothen et al., 1990). In line with optimization based algorithms, there exists a bottom-up approach based on clustering using the correlation coefficients and random walk similarities (Pons and Latapy, 2005; Newman, 2006; Von Luxburg, 2007). Modularity maximization (Newman, 2006) and spectral clustering (Von Luxburg, 2007; Ding, 2004) respectively, are considered as the workhorses for community identification in static networks. But most of the real-world networks are dynamic in nature and there also exist various studies to understand the evolutionary behavior of these dynamic networks (Asur et al., 2009; Backstrom et al., 2006; Dunlavy et al., 2011; Fu et al., 2009). Some recent work also focused on the tracking of time-varying properties of dynamic communities (Xu and Hero, 2014; Mankad and Michailidis, 2013). Previously, spectral clustering, modularity maximization and other statistical mechanics focused on identifying disjoint communities in networks. But real-world networks such as social networks and biological networks are characterized by multiple community memberships where a node has several connections simultaneously with multiple different groups (Kelley et al., 2011; Reid et al., 2013). Keeping in view, the constraint about multiple community memberships of a node, overlapping community detection algorithms are the solution to this problem (Chakraborty and Chakraborty, 2013; Yang and Leskovec, 2013; Maity and Rath, 2014; Ghorbani et al., 2016). Moreover, some algorithms that can detect both disjoint and overlapping communities in a network have also been proposed (Liu et al., 2014).

Based on structural properties of networks, communities can be classified as disjoint and overlapping communities. In disjoint communities, nodes belong to a single community. Fig. 1a is an example of disjoint communities whereas in the overlapping communities, one node can have links to more than one community. Fig. 1b shows three overlapping communities in which red nodes belong to more than one community (Fortunato and Hric, 2016).

In addition to listing the prevailing community detection algorithms, it is also important to present an evaluation of these algorithms. The best way to classify the relative performance of these algorithms is to test them on the standard synthetic benchmarks of complex networks with known community structure. There are different measures of performance on which the estimated and reference communities are compared with each other. Aldecoa et al. (Aldecoa and Marín, 2013a) explored the limits of community detection algorithms by checking their performance on different benchmarks of networks. Harenberg et al. (2014) put forth a comprehensive survey about the evaluation of disjoint and overlapping community detection algorithms in large-scale networks. Besides the testing of algorithms on standard benchmarks, there are also some other aspects to compare the relative performance of the algorithms, i.e., time complexity of the algorithm and the memory space required by each algorithm.

Detecting communities in large-scale networks is extremely useful, and there is a wide verity of applications that use community detection algorithms to expose the hidden structures of networks. For instance, it is very interesting to find users having similar behavior in social networks and cluster customers in e-commerce via their shopping habits. The applications of dynamic community detection are very diverse, e.g., grouping the social media subscribers for a good advertising and facilitating recommendations to the readers, etc. Similarly, community detection plays an important role in designing network protocols in delay tolerant networks, and worm containment in online social networks (Lu et al., 2015). In data networks, it is helpful to identify malicious user communities (Moghaddam, 2011). Based on the attributes of nodes and their link, community detection is being used to detect or predict future links in complex networks (Tan et al., 2014; Valverde-Rebaza and de Andrade Lopes, 2012). Further, in online shopping, product recommendation system is a big candidate to use the updated clustering techniques. It also identifies the clients having the same inclinations towards purchasing some product (Krishnamurthy and Wang, 2000; Reddy et al., 2002; Ju and Xu, 2013).

### 1.1. Our contributions and comparison with existing surveys

The domain of community detection is evolving rapidly and playing a tremendous role in solving various real-life complex problems. There are various studies, review articles, as well as different books on the applications and methods of community detection. Most of the review papers study community detection algorithms for the specific type of networks such as social networks (Wang et al., 2015a, 2015b; Bedi and Sharma, 2016), delay tolerant networks (Hui et al., 2007), and mobile phone networks (Botta and del Genio, 2017). None of the survey papers have reviewed community detection algorithms for disjoint as well as for overlapping communities along with their multidisciplinary applications in a comprehensive manner as presented in our paper. We provide a detailed comparison of our survey paper with previous review papers in Table 1. Among these ten surveys, only Xie et al. (2013) briefly shed light on the overlapping community detection methods. We are also first to include techniques like NMF and PCA-based methods for community detection along with a detailed discussion on performance evaluation of algorithms, provision of application-driven perspective, challenges, and open issues for community detection algorithms.

### 1.2. Goal, scope and overview of this survey

We have performed a multidisciplinary search to find relevant literature on the algorithms for community detection and their applications by using the keywords “community detection”, “algorithms for disjoint and overlapping community detection”, “open issues and challenges to community detection algorithms” and domain-wise real-life “applications”. Relevant papers and studies were then selected from the initial search, on the criteria that papers were published in peer-reviewed journals or conferences and have focused on community detection algorithms and their applications. The goal of this paper is to present a discrete and state-of-the-art review on the methods of community detection and their potential benefits along with the recent work

**Table 1**
Comparison of our paper with existing survey/review papers.

| Survey paper | Year | Areas focused upon | Areas not focused upon |
|---|---|---|---|
| Danon et al. (2005) | 2005 | Compared the prevailing algorithms of community detection at that time in terms of computational cost, revisited modularity measure for community detection. | Overlapping community detection methods were not compared, anomaly aware models were not included. |
| Pons (2007) | 2007 | Discussed algorithms for dynamic community detection using random walk type algorithms. | Only graph partitioning approaches were surveyed. |
| Gulbahce and Lehmann (2008) | 2008 | Summarized future trends related to community detection using hierarchical clustering. | Only hierarchical approaches were focused leaving behind all other classes of community detection algorithms. |
| Porter et al. (2009) | 2009 | Addressed unresolved issues of community discovery domain, solved examples related to large social networks | Only covered graph partitioning approaches in the survey. |
| Yang et al. (2010) | 2010 | Summarized network community mining problem on optimization and heuristic basis, extended mining problem to dynamic networks. | Did not cover outlier aware models and evaluation criteria of stated algorithms |
| Fortunato (2010) | 2010 | Discussed critical issues like the importance of clustering, a procedure to test methods for comparison of static and overlapping community detection methods. | Reported very few applications related to real-world networks, Outlier aware models were not included in the survey. |
| Papadopoulos et al. (2012) | 2012 | Discussed scalability of community detection algorithms to real-world data sets, focused on the complexity and memory requirements of the existing community detection methods. | Outlier aware methods and applications of community detection algorithms were not surveyed. |
| Xie et al. (2013) | 2013 | Reviewed state of the art overlapping community detection algorithms including some NMF techniques. | PCA and its variants based overlapping community detection algorithms were missing, No discussion on applications. |
| Malliaros and Vazirgiannis (2013) | 2013 | Topology based division of community detection algorithms for static, dynamic and overlapping networks. | Did not cover the outlier aware models for community detection, stated only a few real-world applications |
| Chintalapudi and Prasad (2015) | 2015 | Reviews disjoint community detection algorithms in undirected and directed social networks. | Algorithms for overlapping community detection in real-world networks were not focused. |
| This paper | 2017 | Reviews prevailing algorithms for disjoint and overlapping communities. Comprises of detailed discussion regarding performance testing, potential applications, open issues and future directions as well. | Empirical comparison of algorithms is outside this review's scope and is thus not covered. |

that has not been reviewed previously. We hope this paper will help researchers and scientists for the analysis and evaluation of this domain.

The rest of this review paper is organized as follows. In Section 2, we present a unified and precise detail of the algorithms used for community detection problems. Benchmarking and testing of these algorithms on synthetic as well as artificial networks is described in Section 3. The potential applications of community detection in real world networks are stated in Section 4. In Section 5, we present the current challenges to community detection. Section 6 highlights the open issues and recent trends in community detection and finally this paper is concluded in Section 7.

## 2. Community detection algorithms

Community detection is widely studied and various algorithms have been proposed so far. Researchers have classified community detection algorithms in several ways depending upon the dimensions of their work (Pons and Latapy, 2005; Yang and Leskovec, 2013; Papadopoulos et al., 2012). Fortunato (2010) has extensively studied the community detection algorithms by accommodating all approaches discussed in other surveys. But the most recent proposed techniques for overlapping communities still need to be covered. Therefore, in this review, we covered all community detection algorithms reported in the prevailing state-of-the-art surveys and additionally review NMF and PCA-based methods for overlapping communities.

To facilitate understanding for the readers, we have also summarized our findings about community detection algorithms in Fig. 2 and Table 2. More specifically, we provide a taxonomy of community detection algorithms in Fig. 2 and a comparison of these techniques in terms of their advantages and disadvantages is shown in Table 2.

### 2.1. Algorithms for disjoint communities

Many algorithms have been proposed to detect community structure in networks. Most of these algorithms designed to discover disjoint communities, which are discussed below.

#### 2.1.1. Traditional algorithms

The research on community detection using graphs was started in early 1970s (Fortunato, 2010; Wang et al., 2015a) and many algorithms based on clustering were proposed, which are called traditional algorithms. Here are some famous traditional algorithms which introduced substantial concepts of community detection and paved path for future advancements.

*2.1.1.1. Partitional clustering.* It separates the network nodes into $K$ (assumed) clusters by maximizing or minimizing a loss function based on the distance between them. Some partitional clustering techniques along with the frequently used loss functions are listed below:

- *Minimum K-clustering*: Here the cost function is cluster's diameter which is the maximum distance between two points of a cluster.
- *K-clustering sum*: It is similar to minimum $K$-clustering except that the average distance between pairs of cluster points replaces the cluster's diameter in the cost function.
- *K-center*: In this method, a centroid is defined for each cluster. The maximum distance (diameter) of every node from centroid is calculated as the cost function. Clusters and centroids are selected in order to minimize the largest value of diameter.
- *K-median*: it is same as $K$-center except that the cost function comprises the average distance rather than the maximum distance.

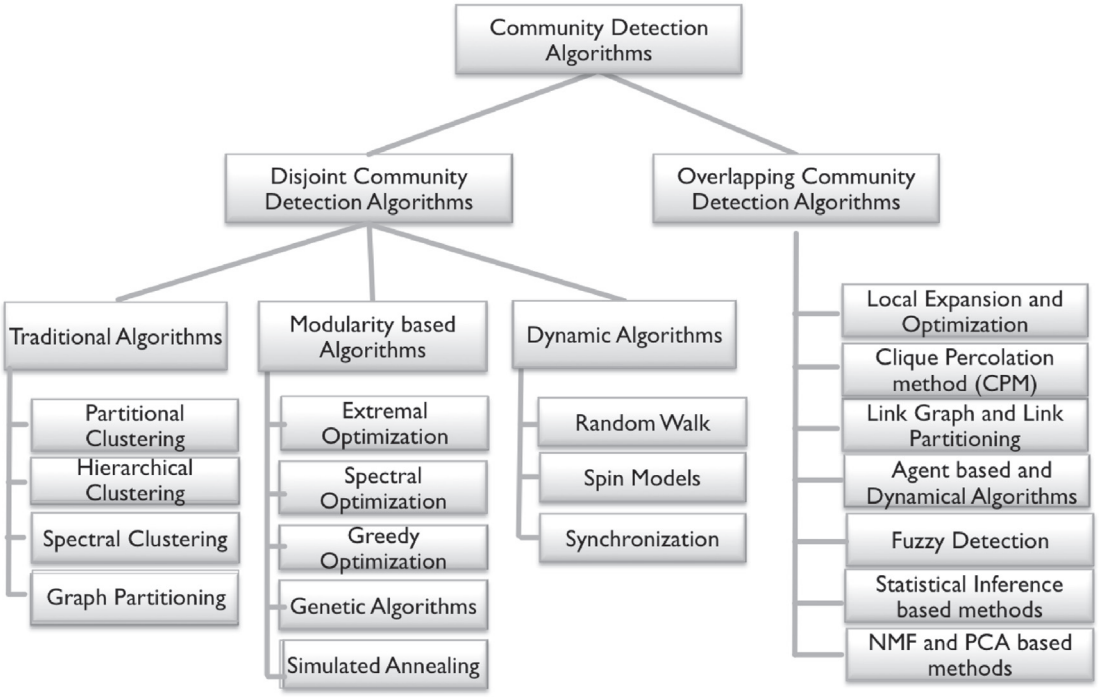



**Fig. 2.** Classification breakdown of algorithms for community detection (synthesized from Fortunato and Hric, 2016; Wang et al., 2015a; Xie et al., 2013).

The most popular approach for partitional clustering is the *k-means clustering* (McQueen, 1967) which minimizes the squared loss function of intra-cluster distance. This *k-means* problem can be easily solved by Lloyd's algorithm (Lloyd, 1982). In line with *k-means clustering*, another popular technique is *fuzzy k-means clustering* (Dunn, 1973; Bezdek, 2013). This technique accommodates the feature of a node belonging to two or more clusters at the same time and has applications in pattern recognition. Specifying the number of clusters at the beginning is the limitation of this algorithm.

*2.1.1.2. Hierarchical clustering.* It is very rare to know about the number of clusters in which a graph is divided and indications about nodes links in advance. In such cases the strategies like partitional clustering may not work up to the mark because of unjustified assumptions about size and number of clusters about the given graph. In contrary, the network graph may have a hierarchical structure where several levels of node clusters are present. In such cases hierarchical clustering plays an important role. Hierarchical clustering techniques are suitable for the graphs having hierarchical structures (Zhang et al., 2014). The idea behind these algorithms is to develop a binary tree that merges similar clusters based on the similarity between vertices. In hierarchical clustering, there is no need to specify the number of clusters beforehand as compared to partitional clustering. Following are some popular approaches of hierarchical clustering.

(i) *Agglomerative Algorithms:* In the agglomerative algorithms, clusters are combined iteratively (bottom-up) if they have high similarity index (or similarity score) (Maqbool and Babri, 2004). It starts with an assumption of having a single object in a cluster and merges the closest pair of clusters based on similarity score. The maximal clique[1] and hierarchical link-based clustering are the examples of agglomerative hierarchical clustering algorithms (Shen et al., 2009). The main advantage of these algorithms is the generation of small clusters which may be helpful in community discovery, whereas their limitation is the lack of provision for object's relocation if merged incorrectly at an early stage. Another problem with this clustering strategy is that the nodes with one neighbor are often classified as an independent cluster which does not make sense in certain cases. Agglomerative hierarchical clustering can not scale well when the points are embedded in space and distance is used as a dissimilarity measure in clustering process. If the distance is not trivially defined, complexity becomes heavier which is $O(n^2)$ for single linkage and $O(n^2 \log n)$ for complete average linkage.

(ii) *Divisive Algorithms:* Divisive algorithms are top-down hierarchical clustering approach (Roux, 2015). It starts with the whole samples in one cluster and then partitioning using flat clustering algorithm is performed that removes the edges which connect low similarity vertices and highest edge betweenness (Morvan et al., 2017). After a dendrogram[2] is made, communities are discovered by cutting of the tree. Cutting position and merging positions need to be appropriate or otherwise results will be of low quality. Girvan and Newman proposed a number of algorithms for community detection in the class of divisive algorithms (Newman and Girvan, 2004; Newman, 2006; Girvan and Newman, 2002).

One of their methods is based on edge betweenness centrality [3] in which edges having the largest centrality are removed. Then the metric is recalculated and edge removal step is repeated until no edge remains. One of the pluses of this Girvan-Newman (GN) (Newman and Girvan, 2004; Girvan and Newman, 2002) method is that there is no need of prior knowledge about the number of clusters. But this algorithm was costly in terms of computational time i.e., $O(m^2n)$ which is not suitable for large scale networks. Tyler et al. (2005) brought a novelty in GN algorithm and instead of calculating exact edge betweenness of all the links, they approximated it using Monte Carlo method of estimation. The result was an improvement in speed and computational cost with

[1] It is subgraphs in which each node is connected to every other node in the clique.

[2] A dendrogram is a tree in which each level denote the partition of graph nodes. Level 0 represents the first partition with smallest communities. Higher level has bigger communities.

[3] It is the number of the shortest paths that pass through an edge.

**Table 2**
Pros and cons of different community detection methods.

| Class | Algorithms | Advantages | Disadvantages | References |
|---|---|---|---|---|
| Traditional Algorithms | Hierarchical Clustering | No need to provide number of clusters in advance | If merging is not good, results may be bad | (Fortunato, 2010; Shen et al., 2009; Ahn et al., 2010; Barabási and Albert, 1999; Friedman et al., 2001) |
| | Girvan-Newman (GN) Algorithms | No need to provide number of clusters in advance | It is not scalable to more than few hundred nodes | (Newman and Girvan, 2004; Newm3an, 2006; Girvan and Newman, 2002; Newman, 2004a) |
| | Tyler and Wilkinson Algorithm | Faster than classical GN Algorithms | Ambiguity in edge removal step for dense subgraphs | (Tyler et al., 2005; Wilkinson and Huberman, 2004) |
| | Rattigen Algorithm (fast variant of GN) | It is fast due to use of a novel metric instead of edge betweenness | Works only for disjoint communities | (Rattigan et al., 2007) |
| | Spectral Clustering | Very good results in complex shapes | Does not work well on outliers, difficult to find best objective to be used | (Shi and Malik, 2000; Von Luxburg, 2007; Ding, 2004; Donath and Hoffman, 1973; Fiedler, 1973; Barnard et al., 1995; Meila and Shi, 2001; Ng et al., 2002; Pothen, 1997) |
| | Graph Partitioning (Kernighan-Lin Algorithm etc.) | It is robust as compared to other traditional algorithms | Cluster size should be provided in advance, results may be random due to random initialization of partition | (Pothen, 1997; Kernighan and Lin, 1970) |
| Modularity-based Algorithms | Simulated annealing (Guimera and Amaral Algorithm) | Good performance due to simulated annealing based optimal solution | Dependent on convergence speed of simulated annealing | (Guimera and Amaral, 2005) |
| | Fast GN Algorithm | Faster than previous GN algorithm | It is not scalable to more than few hundred nodes, not comparable to greedy techniques | (Newman, 2004a) |
| | Clauset Algorithm | Efficiently quickens the GN algorithm by using data structures of sparse matrix | Needs cluster size in advances | (Clauset et al., 2004) |
| | Extremal Optimization | Significant gain in computing time as compared to simulated annealing, Performs well on the GN benchmark as well | Significant gain in computing time as compared to simulated annealing. Performs well on the GN benchmark as well | (Guimera and Amaral, 2005; Newman, 2004b; Boettcher and Percus, 2001a; Liu and Liu, 2010) |
| | Spectral Optimization | In contrast to graph partitioning, there is a distinct stopping criteria related to variation in modularity score | It is only fine for spectral bisectioning and fails when there are more than two clusters. | (Newman and Girvan, 2004; Newman, 2006) |
| | Genetic Algorithms | Does not require number of communities in advance, can solve multi-objective optimization problem of community detection | May fail in random networks having quite high modularity values | (Liu et al., 2014; Tasgin et al., 2007; Pizzuti, 2008; Gong et al., 2011; Gong et al., 2012; Zeng and Liu, 2015) |
| Dynamic Algorithms | Random Walk | Can be used in conjunction with agglomerative methods to detect communities | Expensive (e.g., slower than the greedy techniques) | (Pons and Latapy, 2005; Hughes, 1996; Zhou, 2003; Zhou and Lipowsky, 2004) |
| | Synchronization | High precision and low time complexity | Unreliable in variable size communities | (Boccaletti et al., 2007; Arenas et al., 2006) |
| | Spin Models | Have tunable parameters regarding size of communities | Very slow as compared to greedy Clauset algorithm, only works for small scale networks | (Papadopoulos et al., 2012; Reichardt and Bornholdt, 2004) |
| Algorithms for Overlapping Communities | Fuzzy Detection (Label Propagation) | Low Time complexity and can detect overlapping community | Can detect only one community | (Gregory, 2007; Nepusz et al., 2008) |
| | Clique Percolation Method (CPM) | Can unveil overlapping communities, Can distinguish random graphs from the graphs having community structure | Only suitable when sub graphs are fully connected in given graph of a network | (Chakraborty and Chakraborty, 2013; Maity and Rath, 2014; Palla et al., 2005; Farkas et al., 2007; Kumpula et al., 2008) |
| | Non Negative Matrix Factorization (NMF) with varying input matrix form | Modern era algorithm for overlapping community detection | Costly in computational time and poor scalability | (Mankad and Michailidis, 2013; Yang and Leskovec, 2013; Rossi et al., 2013; Zhang et al., 2007; Zarei et al., 2009) |
| | Symmetric Non Negative Matrix Factorization | Removed redundant constraints in estimation | Only for undirected network graph | (Wang et al., 2011; Zhao et al., 2010) |
| | Bayesian Nonnegative Matrix Factorization (BNMF) and Initialized-BNMF (IBNMF) | Overcomes drawback of Modularity maximization method, computationally more efficient than NMF | BNMF works only for static networks | (Psorakis et al., 2011) |
| | NMFGR | Better clustering effect than NMF | Does not ensure sparseness of factors | (Liu et al., 2016a) |

(*continued on next page*)

**Table 2** *(continued)*

| Class | Algorithms | Advantages | Disadvantages | References |
|---|---|---|---|---|
| | Bounded NMTF (BNMTF) | Outperforms empirical NMF methods | Number of communities should be set beforehand | (Zhang and Yeung, 2012) |
| | Principal component analysis | Overcomes time complexity problem of NMF techniques, has less time complexity than classical algorithms | Data can get arbitrarily re-scaled if principle components are chosen using correlation matrix | (Li et al., 2016; Yeung and Ruzzo, 2001) |
| | Statistical Inference based methods (SBM) and Bayesian Inference | Used to model and analyze real graphs such as social network. DBOCD overcomes drawbacks of BNMTF, it can detect overlapping communities as well | Classical SBM fails to detect overlapping communities | (Ghorbani et al., 2016; Holland and Leinhardt, 1976; Fienberg et al., 1985; Hoff et al., 2002; Wasserman and Pattison, 1996; Wasserman and Faust, 1994; Doreian et al., 2005; Yang et al., 2011) |
| | Line Graph and Link Partitioning | A novel method which partition the links instead of nodes which can detect even overlapping communities as well | Quality of community detection is not so good as in node partitioning methods | (Fortunato, 2010; Ahn et al., 2010; Kim and Jeong, 2011; Evans, 2010) |

reduction in accuracy. Wilkinson et al. (Wilkinson and Huberman, 2004) improved the speed of calculation of the GN algorithms for gene co-occurrence network that was too large to be analyzed by GN algorithms.

A fast variant of the GN algorithms was also suggested by Rattigan et al. (2007), in which betweenness was approximated by performing searches between the sampled pairs of nodes (Rattigan et al., 2006). He showed that the complexity of the algorithm can be reduced to $O(m)$ by ensuring accuracy in estimation of edge betweenness. As computing of the edge betweenness is time-consuming, Radicchi et al. (2004) introduced link clustering coefficient. It is the number of loops that a link goes through. In this method, the low values of the edge clustering coefficient correspond to the inter-community edges. Iterative cut off technique is adopted for the links having a lower value of this parameter. The time required to compute edge clustering coefficient is $O(m^4/n^2)$ which is lower than computing time of edge betweenness in GN algorithm. It makes the algorithm by Radicchi et al. on average faster than GN algorithm. Fortunato et al. (2004) devised a method similar to GN algorithm, where edges are removed with decreasing value of information centrality that is a substitute of edge centrality. It is based on the concept of efficiency that figures out how easily an information flows through a graph using shortest paths between nodes. The vertex with a large value of the information centrality is removed and the process is repeated till there are no more edges. This algorithm performs better when communities have a high degree of interconnectedness.

*2.1.1.3. Spectral clustering.* It includes all algorithms that divide a graph into clusters using the eigenvectors of the input data matrix (Auffarth). It transforms a given set of objects into a set of points in multidimensional space, whose coordinates are the eigenvector elements. This transformation reveals implicit properties of the initial data set, and spectral clustering can be used to cluster data that cannot be successfully done by directly applying $k$-means. In the domain of spectral clustering, the first contribution was proposed by Donath and Hoffman (1973), in which they used eigenvectors of adjacency matrix and eigenvalues of similarity matrix for graph partitioning. In the same year, Fielder (Fiedler, 1973) obtained the bipartition of a graph using the second smallest eigenvalue of Laplacian matrix ($L$)—that is the difference of degree matrix ($D$) and adjacency matrix ($A$) of a graph. Spectral clustering was extended to community detection techniques (Pothen et al., 1990; Barnard et al., 1995) and also used for machine learning problems by Meila et al. (Meila and Shi, 2001) and Ding (2004). In another work, Spielmat and Teng (1996) have written a comprehensive survey on spectral clustering techniques, which can be consulted for more details.

There are three popular methods for spectral clustering, one of which is unnormalized while the remaining two are normalized methods (Shi and Malik, 2000; Ng et al., 2002). More specifically, unnormalized spectral clustering uses unnormalized Laplacian matrix $L$, while the other two uses the normalized Laplacian matrix. Equations (1) and (2) show two normalized Laplacians of a graph. Both are linked to each other:

$$L_{sym} := D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2} \tag{1}$$

$$L_{rw} := D^{-1}L = I - D^{-1}W \tag{2}$$

$L_{sym}$ is the symmetric matrix and $L_{rw}$ is associated to random walk. A standard reference for normalized Laplacian can be seen in (Chung, 1997). In contrast to $k$-means, spectral clustering can work successfully on large data sets with a constraint of sparse similarity graph. As similarity graph opts, one goes for the solution of a linear problem regardless of clinging in search of local minima. However, one should keep in mind that picking a proper similarity graph is a non-trivial task that can make spectral clustering unstable.

Spectral clustering efficiency is reduced in large scale data environment because it requires much time and space to store full similarity matrix and needs eigen decomposition as well. The time complexity for eigen decomposition and space complexity for storing similarity matrix is around $O(n^3)$ and $O(n^2)$ respectively which is unacceptable in large scale data processing (Semertzidis et al., 2015). In the current research scenario, Nyström extension technique proves to be very effective for the case of large scale data clustering because it reduces the described problem of computational cost by using approximation methods (Jia et al., 2017). Initially, the design of Nyström technique was oriented for the solution of integral equations but later on this method was deployed in kernel machine learning problems to speed up the eigen decomposition process (Williams and Seeger, 2001). Clustering performance using Nyström method is directly dependent on the selection of sampling points, i.e., more number of samples lead to better approximations. But increasing the sampling rate is also not favorable in case of big data problems. So, multiple extensions in Nyström techniques were proposed to overcome the described issue-the details of which can be seen in (Jia et al., 2017) and references therein. Jia et al. (2017) used Nyström extension technique to approximate eigenvectors by sampling a few number of data points using a dynamic incremental sampling method. The incremental sampling method samples the data according to different probability distributions. Here the sampling error is reduced by increasing the sampling time. Usually, accuracy of clustering is traded off to get efficiency of the algorithm but this algorithm improved the performance of clustering and ensured a good balance between algorithms' efficiency and sampling quality.

*2.1.1.4. Graph partitioning.* The algorithms in this class divide the vertices into $g$ groups whose size is predefined in order to get the minimum number of links between identified groups (Pothen, 1997). The number of vertices running among the clusters is termed as cut size. If one does not give the number of clusters in advance and inflict a partition with minimum cut size, a trivial solution will be the output. The Kernighan-Lin (KL) algorithm (Kernighan and Lin, 1970) is a heuristic approach for graphs partitioning and still applied in combination with other techniques. It aims at minimization of an evaluation function which is the difference of the intra-community and inter-community links. The motivation behind this approach was the problem of segmenting electronic circuits on boards, in which the vertices in different boards were supposed to be linked with a minimum number of interconnections. Actually, it is an optimization of the benefit function $Q$. This function can be defined as the difference between the number of edges inside the module and the number of edges lying between them.

In addition to the bisection approach in graph partitioning, there is another fast and popular approach, called spectral bisection method, which is based on properties of Laplacian matrix spectrum (Spielmat and Teng, 1996). There are some other popular methods for graph partitioning such as level structure partitioning, multilevel algorithms, and geometric algorithms, whose details can be seen in (Pothen, 1997).

### 2.1.2. Modularity-based algorithms

Modularity-based methods try to maximize modularity whose definition can be seen from equation 3

$$Q = \sum_{i} (e_{ij} - a_i^2) \tag{3}$$

where $e_{ij}$ is number of links with one end in group $i$ and other in group $j$ while $a_i = \sum_i e_{ij}$ is the number of links with one end in group $i$ (Newman, 2004b). These algorithms try to find the best value of $Q$ in NP-hard community detection problems mostly using some heuristics.

*2.1.2.1. Extremal optimization (EO).* This is a heuristic search method proposed by Boettcher and Percus (2001a, 2001b) for approximating solutions to hard optimization problems. It is based on the dynamics of non-equilibrium processes of systems exhibiting self-organized criticality (SOC) (Bak et al., 1987), where more proper solutions come forth actively without tuning of parameters. It optimizes modularity function $Q$ by using fitness value of genetic algorithms. A fitness measure of a vertex is the ratio between vertex modularity and its degree. EO is equivalent to simulated annealing in performance but is more efficient computationally. The algorithmic steps for EO are as follows, split the given network into two clusters having an equal number of nodes. In every iteration, shift the node having lowest fitness value into other clusters. As the partition changes so recalculate the fitness value of vertices. This process continues till $Q$ function stops improving. Another method named as pairwise constrained structure-enhanced extremal optimization-based semi-supervised algorithm (PCSEO-SS algorithm) is proposed by Li et al. (2014), which can solve the problem of false connections in addition to communities detection precisely. It can work efficiently in case of limited prior information.

*2.1.2.2. Spectral optimization.* It is a modularity based optimization technique using spectral information of given data matrix in contrast to traditional spectral clustering algorithms which have no optimization function to be minimized. Eigenvalues and eigenvectors of the modularity matrix can be used to optimize modularity (Chen et al., 2014a). For example, if the eigenvectors with two largest eigenvalues are taken, a split of the graph can be obtained in a network of three clusters. Modularity on bisections can be optimized through spectral bisection by substituting modularity matrix in place of Laplacian matrix (Newman and Girvan, 2004; Newman, 2006). The result may be improved further if vertices are shifted from one to another community in order to get the highest increase or lowest decrease in modularity value. This technique also finds applications in greedy algorithms and extremal optimization.

*2.1.2.3. Greedy optimization.* Newman proposed a greedy method to maximize the modularity (Newman, 2004a). It is an agglomerative hierarchical clustering algorithm, in which edges are joined to shape larger communities such that modularity is increased. The greedy optimization of modularity runs to form large communities with a poor value of modularity maximas. Clauset et al. (2004) indicated that the matrix $e_{ij}$ used by Newman (matrix $e_{ij}$ shows the fraction of edges between clusters $i$ and $j$ of the running partition) to compute modularity $Q$, has a large number of useless operations due to sparse adjacency matrix. They proposed the use of max-heaps—a data structure designed for sparse matrices that works by rearranging the data in the form of a binary tree—to obtain efficiency in algorithmic performance. The update, in this case, was proved to be much quicker that the Newman's greedy technique. The computational cost of the algorithm was $O(n \log^2 N)$ This algorithm can be used to find modularity maxima of large network graph and its code[4] is freely available. Similarly, a better approach for optimum modularity as compared to the Newman greedy optimization is proposed by Denon et al. (Danon et al., 2006), which normalizes the modularity variation $\Delta Q$ by the fraction of links incident to one of the two groups to favor small communities. This idea works better than Newman's recipe when community sizes are largely different. Blondel et al. (2008) proposed Louvain algorithm, a heuristic method based on modularity optimization for the community detection in networks of unprecedented size. It outperformed Newman and Clauset's algorithm in term of computational time which is linear with number of edges in graph, i.e., $O(m)$.

*2.1.2.4. Simulated annealing.* It is a probabilistic method for global optimization, which detects the communities in a complex network by maximizing the modularity. Liu et al. (Liu and Liu, 2010) have used simulated annealing with $k$-means algorithms. This method not only detects community in a complex network but also indicates the central node of each community. The algorithm proposed by Guimera and Amaral (2005) is also an optimization based method that used simulated annealing (SA) for the regulation of local search process. Due to SA, the algorithm proposed by Guimera et al. has a good performance in finding a global optimal solution. Another plus of this algorithm is that it does not require prior knowledge about the number of communities.

*2.1.2.5. Genetic algorithms.* Genetic algorithms (GAs) are optimization techniques which are inspired by biological evolution. GAs are also employed to optimize the network modularity $Q$ in order to detect community structure of a network. Previously GAs were used to partition a graph (Bui and Moon, 1996). Tasgin et al. (2007) used GA to detect community structure in the complex networks based on optimization of modularity for the very first time. These algorithms do not require the number of communities beforehand. Pizzuti (2008) presented a GA called GA-NET which employed a concept of *community score* to show the quality of partitioning of social networks. Community score is the maximization of internal links in a community structure. It efficiently reduced the invalid search when only the actual correlations of all nodes were considered in each operator. Gong et al. (2011) proposed a memetic algorithm named as Meme-Net by optimizing the modularity density. In this algorithm, GA was combined with a local search climbing strategy to improve the performance of traditional GAs. So far all the optimization algorithms employed single optimization criteria, Gong et al. (2012) introduced an evolutionary algorithm based on optimization of two contradictory objectives named as negative ratio association and ratio cut. Similarly, Liu et al. (2014) and Zeng et al.

[4] http://cs.unm.edu/~aaron/research/fastmodularity.htm.

(Zeng and Liu, 2015) proposed multi-objective evolutionary algorithms for detection of communities in signed social networks. The algorithm by Liu et al. (2014) is based on similarity and capable of detecting communities in both separated and overlapping communities. Li et al. (Li and Liu, 2016) proposed an algorithm named MAGA-Net which is a multi-agent genetic algorithm for modularity optimization. It outperformed both GA-NET and Meme-Net in terms of accuracy and stability.

#### 2.1.3. Dynamic algorithms

We will discuss here, three dynamic algorithms (regarding the processes running on the graphs), i.e., random walk, spin models, and synchronization.

*2.1.3.1. Random walk.* A random walk can be employed to detect clusters in a graph by passing over the nodes randomly in order to merge different groups using a bottom-up approach. All random walk algorithms can also be extended to apply on weighted graphs (Hughes, 1996). There are various studies for community detection using the random walk. In 2003, Zhou (2003) defined the distance between a pair of edges using the random walk. The distance between two vertices is the average number of edges that a random walker travels to reach from one node to the other. Close edges are likely to belong to the same community or cluster. In 2004, Zhou and Lipowsky (2004) used biased random walkers. These walkers usually move towards the vertices which have maximum neighbors with the starting node in graphs. The authors used the Brownian movement and proposed a procedure called "Netwalk" that detects communities in this biased random walk. Netwalk is an agglomerative hierarchical clustering method, where the similarity between vertices is showed by their nearness. In 2005, Pons et al. (Pons and Latapy, 2005) proposed Walktrap algorithm that uses modularity value to cut dendrogram while making use of random-walk based similarity between nodes and between clusters. This method is similar to greedy techniques but it becomes expensive regarding computational time which is of the order of $O(n^2 \log N)$ for the case of dense internal edges.

*2.1.3.2. Spin models.* Spin models have been used in statistical mechanics. Potts (Wu, 1982) model is the popular approach in this domain. Inspired by the idea of superparamagnetic clustering of data (Blatt et al., 1996), Reichardt and Bornholdt (2004) proposed a community detection method that mapped network graph onto a zero temperature $q$ Potts model with interactions of nearest neighbors. Potts spin variables are assigned to the nodes with community pattern. Later on, Reichardt (Reichardt and Bornholdt, 2006) introduces spin glass techniques in which every single vertex is assumed to be in a spin state. In spite of being non-deterministic, this algorithm has tunable parameters regarding the size of the community.

*2.1.3.3. Synchronization.* Synchronization occurs in systems of interacting units and it can also be applied to clustering problems (Wang and Chen, 2002). If oscillators are placed at vertices with random phase, they synchronize earlier with the community in which they are present as compared to other communities. If evolution time is allowed, clusters may be recognized at full synchronization in the graph (Pikovsky et al., 2003). It was first shown by Arenas et al. (2006), who detected that the structural scales exposed by synchronization technique, represent groups of eigenvalues of the Laplacian matrix of the graph that aids in graph clustering. Based on the principle of synchronization, Boccaletti et al. advocated a community detection method (Boccaletti et al., 2007). Here the dynamics are the variations of Kuramoto's model that devised opinion changing rate model (Pluchino et al., 2005). This algorithm has a time complexity of $O(mn)$ and it shows good results on Girvan Newman benchmark. The main drawback of synchronization based algorithms is the unreliability of this approach in the case of varying size communities.

### 2.2. Algorithms for overlapping communities

The methods discussed so far aim at disjoint community detection. Here we will discuss algorithms for overlapping communities that can also be used for disjoint communities. Gregory (2007) proposed Cluster-Overlap Newman Girvan algorithm (CONGA) for overlapping community detection. The proposed algorithm is a variant of Girvan and Newman traditional hierarchical divisive clustering approach (Newman and Girvan, 2004; Girvan and Newman, 2002) which is extended with a novelty in the vertex splitting procedure. This algorithm inherits the high computational complexity from GN algorithm and has worst case cost $O(m^3)$ on a sparse graph. In a refined version, Gregory used local betweenness to optimized speed and proposed an algorithm named as CONGO (Gregory, 2008). Following are some popular community detection methods for the overlapping communities.

#### 2.2.1. Local expansion and optimization

This line of search is based on maximization of a local benefit function which features the quality of densely connected nodes. Baumes et al. (2005) proposed an algorithm for overlapping community detection based on the iterative scan (IS) and rank removal (RaRe). RaRe ranks the nodes according to a certain criteria and highly ranked nodes are removed till small disjoint clusters cores are formed. These cores are also termed as seed communities for the IS process which executes a greedy optimization and expands these seed communities until a density function cannot be improved. The optimization density function can be stated as:

$$f(c) = \frac{w_{in}^c}{w_{in}^c + w_{out}^c} \tag{4}$$

where $w_{in}$ and $w_{out}$ are the internal and external weights of community $c$. The time complexity of this algorithm in the worst case is $O(n^2)$. The quality of community detection is dependent on the quality of seeds. Due to nodes removal during expansion, IS may produce disconnected components.

Kelley (2009) modified IS and named it CIS by checking the connectedness at each iteration. The fitness function was also improved with the addition of edge probability measure. To control the algorithm in sparse networks, a controlling factor $\lambda$ was also introduced. The fitness function of CIS can be stated as:

$$f(c) = \frac{w_{in}^c}{w_{in}^c + w_{out}^c} + \lambda e_p \tag{5}$$

where $e_p$ is the edge probability.

LFM proposed by Lancichinetti et al. (2009) used degree of a community as input to fitness function instead of weights of communities.

$$f(c) = \frac{k_{in}^c}{(k_{in}^c + k_{out}^c)^\alpha} \tag{6}$$

while $k_{in}^c$ and $k_{out}^c$ are total internal and external degree of community $c$ and $\alpha$ is the resolution parameter to control the size of communities. The time complexity of LFM is $O(n_c s^2)$ where $n_c$ is the number of communities and $s$ is the average size of communities. Lancichinetti et al. (2011) introduced Order Statistics Local Optimization Method (OSLOM) algorithm but it resulted in a significant number of singleton communities.

#### 2.2.2. Clique percolation method (CPM)

CPM is based on cliques formation concept by the subgraphs where every node is connected to other nodes in a clique. Palla et al. (2005) proposed that the internal edges with high-density tend to form cliques while inter-community edges do not form cliques. The term $K$-clique is used to express a complete graph with $K$-vertices. It is presumed that the network comprises of $k$ adjacent cliques sharing $k-1$ nodes with each other. Each clique is in one community but cliques in different

communities can share nodes in common. This heuristic is helpful in discovering overlap within the node clusters. CPM extension was targeted for the analysis of the weighted, bipartite and directed graphs by Farkes et al. (Farkas et al., 2007). They proposed a threshold for clique weights as the geometric mean of all edges' weights. The threshold value is chosen marginally greater than a critical value for which $K$-clique community emerges. It helps to obtain the richest kind of clusters.

A fast implementation of CPM is developed in (Kumpula et al., 2008) called sequential clique percolation algorithm (SCP). It starts from an empty graph and detects $K$-clique communities by sequentially inserting the edges of the graph under study. The time complexity of this method is linearly dependent on the number of cliques but it is still faster than the original CPM. A big plus of this method is that it can recover community structures of a graph in a single run by inserting edges in descending order of weights and by storing each detected community after addition of every single edge. Sometimes CPM algorithms do not cover the whole network, i.e., some nodes may not belong to any cluster irrespective of having connectivity with other nodes. Maity et al. (Maity and Rath, 2014) proposed a novel algorithm and extended CPM in order to make sure that every node belongs to at least one community. Initially, communities are detected using CPM and then expanded for remaining nodes on basis of belonging coefficient of every node. Belonging coefficient shows how strongly a node is attached to a particular community (Nicosia et al., 2009). Evaluation of this extended version proved its superiority over classical CPM. Finally, Chakraborty et al. (Chakraborty and Chakraborty, 2013) used CPM and devised an algorithm named OverCite which can detect overlapping communities in citations network that contains information about authors, papers, and venues.

In spite of the conceptual simplicity, CPM looks like pattern matching algorithm instead of community detecting algorithm because they tend to find specific and localized structure in network (Xie et al., 2013).

#### 2.2.3. Line graph and link partitioning

Instead of partitioning the nodes in the network model, partitioning the links in that network has also got attention in community detection domain. Here a node is called as overlapping if the links connected to it belong to more than one cluster. Ahn et al. (2010) partitioned the links using the hierarchical clustering of edge similarity. The time complexity of this algorithm is $O(nk_{\max}^2)$ where $k_{\max}$ is the maximum node degree in the given network. Kim et al. (Kim and Jeong, 2011) extended infomap (Rosvall and Bergstrom, 2008) to line graph using minimum description length principle and encoded minimum path of the random walk on line network. Evans (2010) extended line graphs to clique graphs for overlapping community detection where cliques act as nodes of the weighted graph. There is no guarantee of high-quality community detection using link partitioning technique because these algorithms rely on an ambiguous definition of the community (Fortunato, 2010).

#### 2.2.4. Agent based and dynamical algorithms

In Label Propagation Algorithm (LPA) (Raghavan et al., 2007), nodes having the same label tend to form a community. Each vertex is visited and assigned a label based on voting of its neighbors. This vertex iteration continues till convergence. LPA are important due to their simple concept and computational efficiency that is linear with number of edges in graph i.e., $O(m)$. LPA can be extended to detect overlapping communities. Gregory (2010) proposed community overlap propagation algorithm (COPRA) which was an extension of LPA. In this algorithm, each node receives coefficients from its neighbors and averages them in order to update its belonging coefficient at each time step. For input matrix of size $m \times n$, the time complexity of this algorithm is $O(vm\log(\frac{vm}{n}))$ per iteration while $v$ is the parameter to control a node's association with multiple communities. Xie et al. (2011) extended LPA to speaker-listener label propagation algorithm (SLPA) to identify overlapping communities and overlapping nodes. It is a speaker-listener based information propagation process. In contrast to LPA where a node forgets the label in the previous iteration, SLPA provides each node with a memory to store the labels of the previous iteration. A good plus of SLPA is that it does not require information about the number of communities beforehand. Its computational complexity is $O(tm)$ while $t$ is the number of predefined iterations and $m$ is the number of edges. SLPA can be extended to weighted and directed networks by including the interaction rules known as SLPAw.

Chen et al. (2010) introduced a game-theoretic framework for overlapping community detection. Community formation dynamics are taken as a strategic game called community formation game in which nodes are taken as selfish actors joining or leaving a group based on her utility measurements. Nash equilibrium is used in community detection while the time taken to reach that equilibrium is $O(m^2)$ where $m$ is the number of edges.

#### 2.2.5. Fuzzy detection

These are label propagation methods but also able to find overlapping communities in the network by computing belonging factor or a soft membership vector for every vertex (Gregory, 2007). The dimension of this factor should be provided beforehand, which is the major drawback of this method. Nepusz et al. (2008) used the fuzzy community method to solve the overlapping community detection problem as a nonlinear constrained optimization. The method used in this approach was simulated annealing. They minimized the following objective function

$$f = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(\widetilde{s_{ij}} - s_{ij})^2$$
$$s_{ij} = \sum_{c} a_{ic}a_{jc} \tag{7}$$

$w_{ij}$ are predefined weights, $\widetilde{s_{ij}}$ is the prior similarity of node $i$ and $j$, $s_{ij}$ is similarity measure and $a_{ic}$ is the fuzzy membership of vertex $i$ in community $c$. Wang et al. (2009) used a disjoint community detection method with a local optimization algorithm to unveil the overlapping nodes in the network.

#### 2.2.6. Statistical inference based methods

Graph clustering is a type of statistical inference problem. A brief description of some of the inference based methods for overlapping community detection in graphs is given below.

*2.2.6.1. Stochastic block model (SBM).* In statistics, SBM is a probabilistic model that not only finds its application in overlapping community detection but it also describes how communities are formed. It was proposed by Holland and Leinhardt (1976) and modified by Fienberg et al. (1985), Hoff et al. (2002) and Wesserman et al. (Wasserman and Pattison, 1996; Wasserman and Faust, 1994). Block model is very common approach regarding network analysis, e.g., social network analysis. It breaks down a graph into the class of edges having common attributes. A thorough discussion of SBM is beyond this review's scope, but details can be found in (Doreian et al., 2005).

*2.2.6.2. Bayesian inference based methods.* In 2011, Yang et al. (2011) proposed a Dynamic Stochastic Block Model (DSBM) to model communities and their evolution. The authors also suggested Bayesian treatment for the estimation of parameters. Two versions have been proposed, one is the online inference which modifies the model with time while other is the offline inference that learns the model with data obtained in all time steps in contrast with the network analysis that deals with online inference. Ghorbani et al. (2016) proposed a Dynamic

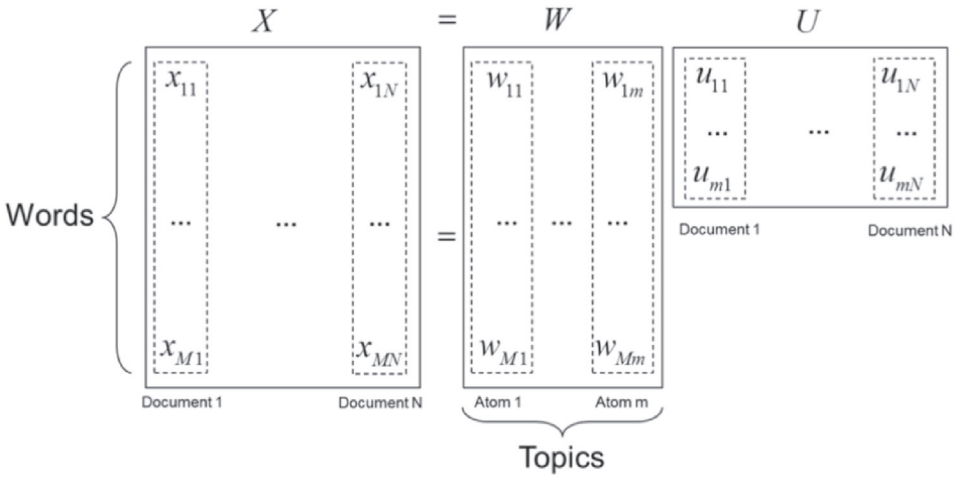



**Fig. 3.** Dictionary decomposition in NMF.

Bayesian Overlapping Community Detector (DBOCD). It assumes that in every temporal snap of the network, overlapping components of communities are the thick sections utilizing link communities instead of general node communities. This algorithm extracts overlapping communities while conserving the communities' consistency over time. It performs better than the recent dynamic methods for overlapping community detection.

*2.2.7. Non negative matrix factorization (NMF) approaches*

Many algorithms especially the spectral methods identify communities from eigenvalues whose physical meaning is very hard to explain with real-world applications. NMF has emerged as a handy tool for data analysis with improved interpretability (Lee and Seung, 1999). This is particularly a recent approach for the discovery of structural and functional properties of a dynamic network with overlapping communities. It is a machine learning algorithm that decomposes a given feature matrix in order to uncover the features of a given structure (Mankad and Michailidis, 2013; Yang and Leskovec, 2013; Rossi et al., 2013). The property of this technique is to avoid the feature vectors from negative components. In standard decomposition $X = WU$ as shown in Fig. 3. Where $X$ is the input data matrix with dimension $M \times N$ and $W, U$ are the approximated factors of the input data matrix having dimensions $M \times m$ and $m \times N$ respectively with $m$ as the rank of factorization. It is chosen so that $(M + N)m < MN$. The goal is to minimize

$$f(W, U) = \|X - WU\|_F^2 \quad \text{s.t } W, U \geq 0 \tag{8}$$

Other formulation of NMF as an optimization problem can be stated as

$$\min D(X \| WU) \quad \text{s.t } W, U \geq 0 \tag{9}$$

Both these formulations are convex in $W$ only or $U$ only. If we assume both variables as unknown together, the optimization problem is non-convex. Following are some variants of NMF.

*2.2.7.1. Kernel NMF.* Zhang et al. (2007) replaced the input feature matrix with diffusion kernel that is Laplacian of given network. This method can detect fuzzy communities. The uniqueness of this method is to provide the information about how much a node belongs to a certain community. For an undirected graph $G = (V, E)$, the Laplacian may be expressed as

$$L = \begin{cases} 1 & \text{for } i \sim j \\ -d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

$i \sim j$ means that node $i$ and $j$ have an edge between them and $d_i$ shows the degree of vertex $i$. The exponential of this Laplacian is denoted by $K$, which is a symmetric matrix.

$$K = \exp(\beta L) = \lim_{n \to \infty} (1 + \frac{\beta L}{n})^n \tag{11}$$

$\beta$ controls the degree of diffusion. The new feature matrix $B$ (analogous to $X$ given in the basic NMF model above) can be obtained as follows

$$B_{ij} = \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}} \tag{12}$$

Since $B$ is symmetric so $M = N$ in this case. Here all column and rows show the similarity measure between nodes. The root-mean-square error (RMSE) shown by $\|X - WU\|_2$ is minimized by iterative updates of $W$ and $U$. The error is actually between the real feature vector given as $X$ and the reduced dimension set of $WU$. The algorithm runs for a fixed number of iteration or until RMS error is reduced to a certain threshold.

*2.2.7.2. NMF with correlation matrix.* Zarei et al. (2009) replaced input feature matrix $X$ with the correlation matrix of column vectors of Laplacian matrix. So the new graph matrix $C$ can be written as

$$X = C_{ij} := \frac{(cov(L)_{ij})}{\sqrt{((cov(L)_{ii}(cov(L))_{jj}))}} \tag{13}$$

The covariance $cov(L)_{ij}$ is equal to $cov(l_i, l_j)$ while $l_i$ being the $i_{th}$ column of Laplacian matrix. The matrix $C$ shows the correlation of the Laplacian matrix $L$. High value of $C_{ij}$ means higher correlation. A relation between spectral clustering and NMF method can be seen in (Ding et al., 2005) where the feature matrix $X$ can be decomposed as $X = UU^T$.

*2.2.7.3. Bayesian NMF.* In 2011, Psorakis et al. (2011) proposed a hybrid approach that deployed Bayesian NMF model to detect overlapping communities in a network. The Bayesian approach has been proved to be computationally efficient quantifying how much influence has a node in each group. The hybrid scheme overcomes the drawbacks of the modularity maximization methods for community detection. The algorithm works only for a single snapshot of time evolving networks, e.g., static network. Similarly, Mankad et al. (Mankad and Michailidis, 2013) used NMF for the dynamic community detection by introducing smoothing constraints on the resultant factors of NMF.

*2.2.7.4. Bounded non-negative matrix tri-factorization (BNMTF).* BNMTF was proposed by Zhang et al. (Zhang and Yeung, 2012) and defined by the following expression.

$$f(U, B) = \|G - UBU^T\|_F^2 \tag{14}$$

Alternatively

$$\min L(G,U,B) + \lambda U \quad \text{s.t } B \geq 0, 0 \leq U \leq 1 \text{ and } \lambda > 0 \tag{15}$$

Here $G$ denotes the adjacency matrix, $U$ shows the membership of each node in each community and $B$ is the interaction between communities. BNMTF uses two loss functions named as KL-divergence and squared loss to compute error. Loss functions may be expressed as

$$L_{sq}(G,U,B) = \|G - \hat{G}\|_F^2 L_{kl}(G,U,B) = \sum_{ij}(g_{ij} \ln \frac{g_{ij}}{\hat{g}_{ij}} - g_{ij} + \hat{g}_{ij}) \tag{16}$$

As described above, $G$ is sparse adjacency matrix having $\hat{g}_{ij}$, the $(i,j)_{th}$ entry of $\hat{G}$. Besides modularity, another performance metric was used to evaluate the performance of the both strategies (squared loss and KL divergence) that is area under the curve (AUC) score based on modularity. BnMTF$_{kl}$ was proved better than BNMTF$_{sq}$ in both scores. This algorithm was further modified to deal with sparsity, e.g., s-BNMTF$_{kl}$ and s-BNMTF$_{sq}$, and it outperformed its predecessor when these algorithms were evaluated on modularity and AUC score. The drawback of this algorithm is that it requires the number of communities to be set in beforehand and this method performed for community detection in static networks leaving a roam to introduce the same for the dynamic networks.

*2.2.7.5. Symmetric, asymmetric, and joint NMF (SNMF, ANMF and JNMF).* These three algorithms were proposed by Wang et al. (2011) to discover hidden clusters in undirected, directed and compound networks respectively. The factorization is done using the model $G = XSX^T$, in which $G$ shows the adjacency matrix, $X$ is the cluster membership and $S$ is the diagonal matrix whose entries show the connectivity within each cluster. It seems to be analogous to the eigenvalue decomposition of $G$ but in real-world networks, the adjacency matrix may have non-zero off-diagonal entries, e.g., the case of overlapping clusters, and columns of $X$ will not be orthogonal to each other anymore. In the case described above, the tri-factorization is no more equivalent to eigenvalue decomposition.

In SNMF, $S$ can be absorbed into $X$, i.e., $\hat{X} = XS^{\frac{1}{2}}$ and the problem reduces to

$$\min \|G - \hat{X}\hat{X}^T\|_F^2 \tag{17}$$

where $X$ is solved by multiplicative update rule.

SNMF works for undirected networks and successfully clusters overlapping communities. Zhao et al. (2010) removed redundant constraints in estimation and modified the NMF problem to symmetrical NMF (sNMF).

ANMF works for directed networks. In which the adjacency matrix $A$ is asymmetric so as the matrix $S$. Here normalization constraints on $X$ are not enforced instead it is further normalized by inserting a diagonal matrix between $X$ and $S$ which can be stated as $XSX^T = (XD^{-1})(DSD^T)(XD^{-1})^T$ with the following objective function

$$\min \|A - XSX^T\|_F^2 \tag{18}$$

JNMF works for compound networks that are beyond directed and undirected graphs—e.g., in a movie recommendation system, the analyst is provided with three networks, i.e., user-user, movie-movie, and user-movie network. Let $U$ be considered as the user-user matrix, $D$ denotes the movie-movie matrix and $M$ shows user-movie matrix. So three objective functions are required to be minimized simultaneously $\|M - X\|$, $\|U - XX_T\|$ and $\|D - X^TX\|$. JNMF solves the problem by minimizing $l(X,M,U,D)$ while,

$$l(X,M,U,D) = \|M - X\|^2 + \alpha|U - XX_T\|^2 + \beta\|D - X^TX\|^2 \text{ s.t } \alpha, \beta > 0 \tag{19}$$

Here, $\alpha$ and $\beta$ are constants to trade off between different parts. All the three algorithms strongly depend on the quality of provided data (adjacency matrix $G$ for SNMF, $A$ for ANMF and $U$, $M$ and $D$ for the JNMF).

*2.2.7.6. Initialized Bayesian nonnegative matrix factorization (IBNMF).* Tang et al. (2014) claimed that most of the existing NMF algorithms converged slowly. These algorithms initialize NMF factors ($U$, $V$) randomly that create different results for different initialization. They used a famous approximation as an initialization step in NMF algorithm. They used NNDSVD (Nonnegative Double Singular Value Decomposition) to initialize the BNMF algorithm and called it IBNMF. The advantage of this initialization was achieved in the form of more stable and computationally efficient algorithm with high accuracy in grouping the network entities. It also overcomes the drawback of modularity based methods.

*2.2.7.7. NMF with graph regularization (NMFGR).* In 2016, Liu et al. (2016a) proposed graph regularizer based NMF model named as NMFGR. They used similarity metric for preprocessing of the network along with the regularization term. NMF formulation for community detection can be shown as minimizing $\min_{U\leq 0, V\leq 0} f(X, UV^T)$, where $f(A,B)$ is the loss function and $X$ is the adjacency matrix which is approximated by $UV^T$. Graph regularizer is shown as

$$R = Tr(V^TLV) \tag{20}$$

The unified model for graph regularized NMF is shown as

$$\min_{U\leq 0, V\leq 0} \quad f(X, UV^T) + \gamma Tr(V^TLV) \tag{21}$$

$\gamma$ controls the balance between NMF and regularization term.

*2.2.7.8. NMF with iterative bipartition (NMFIB).* This algorithm was proposed by Dongxiao et al. (He et al., 2014). They suggested a generative model for communities discovery and solved it just like an optimization problem by adopting the NMF technique and then extended it by introducing an iterative bipartition technique. NMFIB identified protein-protein interaction (PPI) community structure better than the algorithm of Ahn et al. (2010).

*2.2.7.9. Orthogonal NMF (ONMF).* In 2014, Pompili et al. (2014) proved that the orthogonality constraint improves the clustering results in comparison to the standard NMF. ONMF can be described as follows

$$\min_{U\in R^{mxk}, V\in R^{kxn}} \|M - UV\|_F^2, \quad \text{s.t } U, V \geq 0 \text{ and } VV^T = I_k \tag{22}$$

ONMF was solved using two new methods. The first method was derived after proving a mathematical equivalence between ONMF and weighted spherical $k$-means. Following are the two conditions in which ONMF is preferable than spherical $k$-means: (1) when scaling does not affect the cluster assignment, and (2) when larger norm data points are given importance The other is augmented Lagrangian based method that can be seen in Equation (23).

$$L_\rho(U,V,\Lambda) = \frac{1}{2}\|M - UV\|_F^2 + < \Lambda, V > + \frac{\rho}{2}\|\min(V,0\|_F^2 \tag{23}$$

$M$ is input data matrix having $U$ and $V$ as non-negative factors and $\rho$ is the quadratic penalty.

*2.2.8. PCA-based methods*

In 2014, Lin et al. (2014) proposed a PCA-based method for overlapping community detection. This algorithm uses PCA in order to select the best number of eigenvectors and then uses Laplacian matrix to map the nodes in low dimensional subspace. It uses fuzzy C-means (FCM) to reveal overlapping structure in a network. In FCM every vertex has a sealed membership degree that depicts its belonging to different clusters. PCA is used to pull out principle components of given nodes by

checking the spread of neighboring eigenvalues in order to choose the optimal number of eigenvectors which play a vital role in the performance of spectral clustering algorithms. When PCA completes its job, overlapping community structure is unveiled using spectral analysis.

In 2016, Li et al. (2016) improved classic community detection algorithm based on *k*-means using PCA approach. The algorithm runs in three stages. First, it calculates the distance between the nodes present in the network. The distance between nodes belonging to the same community is small than with other groups. Second, the nodes are mapped into $p$ dimensional space. Finally, *k*-means algorithm is applied to unveil the $K$ number of communities in a network.

Yuan et al. (2016) proposed another approach based on PCA and membership index (MI) named as PCA-MI for the detection of overlapping communities. PCA draws out the pertinent features of the complex network and then MI is applied to classify nodes belonging to different communities.

### 2.3. Decentralized implementation of community detection algorithms

'Big Data' has substantial impact in different domains such as artificial intelligence, marketing, finance, computational biology, network analysis etc. Due to extremely large dataset, high dimensionality and distributed storage, scalable and rich enough algorithms are needed to tackle the above mentioned challenges of complex and huge networks. In this regard, Boyd et al. (2011) put forth a 'toolbox' based on convex optimization in order to derive and implement the distributed algorithms. Implementation of the distributed algorithms is discussed in cloud computing environment as well. For a broader insight of parallel and distributed computing, one can look into a book by Bertsekas et al. (Bertsekas and Tsitsiklis, 1989).

Mostly, the community tracking algorithms assume that the connectivity data, i.e., the adjacency matrix is acquired and processed in centralized manner while in many cases, network data is stored across many computing agents which may be spatially distributed across different geographical sites. For example, it is impracticable to get the complete web structure in order to obtain its community structure and making a single node responsible to work as a centralized node. For that reason, it is essential to distribute the whole network into multiple processing nodes to make the community detection more scalable by executing several parallel community detection processes.

Motivated by this scenario, Baingana et al. (Baingana and Giannakis, 2015) performed community and anomaly tracking in a decentralized manner using the stochastic gradient descent iterations. It made this approach memory efficient, as compared to the batch recovery approach that was impractical due to huge memory cost in big data context. Here, alternating-direction method of multipliers (ADMM) was used to solve the decentralized optimization problem. The main idea was that each node worked on a subset of data in addition to sharing its solution with neighbors till consensus was achieved. Hu et al. (2014) formulated a decentralized and privacy preserving multi-objective optimization problem for community detection in social networks. The work was aimed to solve the problem of secure friend discovery on decentralized social networks. Similarly, Huang et al. (2013) presented a multi agent decentralized implementation of the said algorithm in which a group of independent agents work in the form a group in order to mine a network by a self-aggregation strategy. Hui et al. (2007) proposed a distributed community detection strategy which approximated its corresponding centralized procedure up to 90% accuracy.

## 3. Benchmarking and comparison

In section 2, different prevailing algorithms for community detection have been stated categorically with their working principles and implementations. Pros and cons of each algorithm of a particular class have been summarized in Table 2. In this section, important ingredients regarding performance testing of algorithms on standard benchmark networks and different aspects of algorithms' comparison have been described.

To facilitate the understanding of readers, a comparison of different algorithms based on the measurement of important quality measures such as normalized mutual information, etc., has been summed in Table 4. Different studies regarding the comparative analysis of algorithms using different synthetic and artificial benchmarks has been summarized in Table 5. Further, another important aspect regarding the algorithmic performance, i.e., computational cost, of all the algorithms is put forth in Table 6 to select a cost friendly solution.

### 3.1. Benchmarks

The best way to classify the relative performance of these algorithms is to test them on the standard synthetic benchmarks of complex networks with known community structure. There are two benchmarks for the said purpose called open and closed (Aldecoa and Marín, 2013b).

Open benchmarks start with a network having known community structure which is progressively degraded with random rewiring in such a way that the links between nodes of different communities increase and leave the network as an open-ended unknown network structure. In this benchmark, algorithm performance can be evaluated by comparison of detected communities in start with the detected communities at the end due to increasingly difficult structure with random rewiring of nodes and links between nodes. The first open benchmark was formulated by Girvan and Newman known as GN benchmark (Girvan and Newman, 2002). It has 128 nodes with each nodes having 16 links on average and split into four equal sized communities. Arenas et al. (2006) extended the GN benchmark graph with an embedded hierarchical structure. The network contains 256 nodes with two hierarchical levels: A total of 16 groups each with 16 nodes (micro-communities) in it, and 4 larger groups with 64 nodes in each group (macro-communities).

Most of the algorithms performed well on the GN benchmark (Lancichinetti and Fortunato, 2009; Danon et al., 2005), which pointed out towards the need to develop more complex benchmarks. In addition, the node degree has poisson distribution in GN benchmark in contrast to real networks where degree distribution shows a fat tail. So, Lancichinetti, Fortunato and Radicchi proposed a new benchmark named as LFR benchmark (Lancichinetti et al., 2008) having evident advantages over the GN benchmark. In LFR benchmark node degree follows power law with ease of exponents chosen by the user which makes it realistic to build many communities. In LFR, many algorithms performed poorly which had optimum performance in GN benchmark (Lancichinetti and Fortunato, 2009; Lancichinetti et al., 2011; Aldecoa and Marín, 2013b). GN and LFR both make a network having communities of equal sizes so it led to proposal of a third benchmark based on relaxed caveman (RC) structure where isolated cliques form the network with each one corresponding to a community and getting interconnected by rewiring process. The choice of selecting the initial clique makes RC benchmark ideal for making skewed distribution of clusters (Aldecoa and Marín, 2012, 2013b).

Closed benchmarks also start with a network of known community structure but in contrast to open benchmarks, here the rewiring of the links is not random (Aldecoa and Marín, 2012). These benchmarks have the same community structure in their initial and final network but with nodes reassigned randomly among the communities. Rewiring in the described process is called Conversion (C) which ranges from 0% to 100%. $C = 100\%$ means that the final structure of the rewired network has been obtained. Further details about the closed benchmarks can be seen in (Aldecoa and Marín, 2012). Main advantage of using closed benchmarks is the information about the optimality of algorithm under test based on a parameter called variation of information (VI) (Meilă, 2007).

All the benchmarks described above, are synthetic networks, i.e., computer generated networks. There are also real-world community

**Table 3**
Real-network data sets from different social networks having known community structure with $n$ number of nodes and $k$ degree (Xie et al., 2013).

| Network | $n$ | $k$ | Network | $n$ | $k$ |
|---|---|---|---|---|---|
| karate | 34 | 4.5 | PGP | 10680 | 4.5 |
| football | 115 | 10.6 | Email | 33696 | 10.7 |
| lesmins | 77 | 6.6 | P2P | 62561 | 2.4 |
| dolphins | 62 | 5.1 | Epinions | 75877 | 10.6 |
| CA-GrQc | 4730 | 5.6 | Amazon | 262111 | 6.8 |

networks which can be used to test the community detection algorithms. Table 3 shows some real world data sets [5] of different networks having known community structure.

Obtaining real-world networks with which a ground truth can be associated, is not only difficult, but computationally expensive as well. In addition, it is not possible to control many feature of real-world network, e.g., degree, degree distribution, size of communities, etc., (Yang and Leskovec, 2013).

### 3.2. Measures for comparison

Estimated communities by the algorithms are compared with the reference communities on the basis of different measures which are widespread and used throughout in the literature. A brief description of some of those measures is as follows.

The centrality metrics includes three measures which are defined as follows: *Betweenness Centrality* (BC) is the measure of tendency of a node that can be found out along the shortest path between two other nodes. This measure tells that a nodes with high BC can act as an important route for the information flow in the network and removal of that node may collapse the network. *Closeness centrality* (CC) sums all the shortest paths between that node and other nodes. This measure shows, how much a node is closer to other nodes in the given network graph. *Degree centrality* (DC) is about the directly connected nodes in the network. This centrality metric containing BC, CC and DC, determines which node belongs to which community and how much influential a node can be in a network (Girvan and Newman, 2002).

The *Fraction of Correctly Classified nodes* (FCC) was first used by Girvan and Newman (2002). According to this measure of performance, a node is well classified if its estimated community is the same than for the majority of nodes present in its reference community. In order to normalize this measure, total number of nodes which are correctly classified are divided by the total number of nodes $n$ in the network.

The *Rand Index* (RI) is the proportion of node pairs for which both communities agree. If RI value is 0, it means that the algorithm has failed to estimate the community structure. Perfect estimation shows RI value of 1. *Adjusted RI* (ARI) is also a corrected for chance version of RI ranging from −1(less than chance) to 1 (complete agreement) (Rand, 1971).

The *Normalized Mutual Information measure* (NMI) (Ana and Jain, 2003) was defined in the context of classical clustering (Pothen et al., 1990) to compare two different partitions of one data set, by measuring how much information they have in common. Danon et al. (2006) evaluated the performance of algorithms for community detection using this measure. NMI was also used by other authors such as Lancichinetti et al. (2008). NMI index has a value 1 if the estimated community matches the reference community and 0 in the opposite case.

Orman et al. (2012) conducted a comparative survey to evaluate community detection algorithms on LFR benchmark. The finding are summarized in Table 4 for the ease of readers.

[5] http://snap.stanford.edu/data/index.html#socnets.

### 3.3. Aspects of performance comparison

There are two aspects to assess the performance of algorithms: (a) the computational cost (b) requirement of the main memory to implement the particular method. The first aspect for the performance comparison of algorithms i.e., computational cost of each algorithm, is stated in the relevant text of that algorithm in Section 2. The other bottleneck in community detection is the memory consumption. The minimum memory consumption of a community detection method is scaled linearly with the graph size, i.e., $(2m + 2n)$. There are many implementations of community detection algorithms which use the matrix of the graph (adjacency matrix). So it should be present in the memory and it takes $2m + 3n$ of memory for the unweighted graph. Different strategies of clustering require different data structures to speed up their community detection. For example, Clauset et al. (2004) used $\Delta Q$ matrix instead of adjacency matrix. This new matrix required $n$ binary trees and $n$ max heaps which increased the requirement of memory space. Similarly, the modularity maximization approach proposed by Newman et al. (Newman, 2006) requires modularity matrix to be present in the memory and take additional memory space up to $n^2$.

On contrary, there are some algorithms which perform well even under low memory constraints such as LPA proposed by Raghavan et al. (2007). It can work with a memory space of $n + k_{\max}$ while $n$ is the community assignment vector and $k_{\max}$ is the vector of large vertex neighborhood (Papadopoulos et al., 2012). Spectral approaches need a set $C$ of eigenvector thus increasing the required space by $Cn$.

### 3.4. Comparing algorithms

First of all, Danon et al. (2005) proposed the use of standard network benchmarks for testing of community detection algorithms. A detailed comparison of algorithms in the prospective of computational cost was performed when the algorithms were subjected to test under an ad-hoc network of 128 nodes and 4 communities (GN benchmark).

After that comparative study, A. Lancichinetti and S. Fortunato (Lancichinetti and Fortunato, 2009) tested the algorithms on LFR benchmark (LFR benchmark (Lancichinetti et al., 2008) was also proposed by the same authors) by pointing out the shortcomings of GN benchmark which are about the same degree of all nodes and same size of all communities. It has already been described that LFR benchmark is more complex than GN benchmark. The algorithms are tested under GN, LFR and random graphs and a comparison is put forth about the leading performer algorithms in different network benchmarks.

Aldecoa et al. (Aldecoa and Marín, 2013a) performed a detailed performance evaluation of different community detection algorithms using complex closed benchmarks to check the optimality of those algorithms. Instead of using modularity to check the community quality of the algorithm, a new global quality measure called Surprise (S) was adopted. Surprise evaluates the probability of finding a particular number of intra-community links in the partitions of given network while assuming that those links appeared randomly. In that paper, a detailed and extensive analysis of 17 different algorithms is performed on open and closed benchmarks (see (Aldecoa and Marín, 2013b) for the detailed performance evaluation of 17 algorithms).

Orman et al. (2012) conducted a comprehensive study to compare famous community detection algorithms on their artificial network generated using LFR benchmark. The findings of that study have already been summarized in Table 4.

Similarly, Harenberg et al. (2014) presented an advanced review about the empirical evaluation of the disjoint and overlapping community detection algorithms. In that review, 13 algorithms were compared using goodness metrics and performance metrics. The goodness metrics evaluates an algorithm on the basis of structural properties of the identified communities by that particular algorithm. The performance metrics evaluates the algorithms according to their performance on ground truth communities. The details about the performance of those several

**Table 4**
Table showing traditional performance measure of selective algorithms on an artificial network based on LFR benchmark (Orman et al., 2012).

| Algorithm | Ref. | FFC value | RI value | ARI value | NMI value |
|---|---|---|---|---|---|
| Copra | Gregory (2010) | 0.090 | 0.068 | 0.002 | 0.070 |
| Fast Greedy | Newman (2004a) | 0.080 | 0.919 | 0.272 | 0.588 |
| InfoMap | Rosvall and Bergstrom, (2008) | 0.862 | 0.997 | 0.930 | 0.930 |
| InfoMod | Rosvall and Bergstrom, (2007) | 0.255 | 0.971 | 0.256 | 0.620 |
| Louvain | Blondel et al. (2008) | 0.425 | 0.982 | 0.692 | 0.735 |
| Oslom | Lancichinetti et al. (2011) | 0.415 | 0.932 | 0.337 | 0.685 |
| WalkTrap | Pons and Latapy (2005) | 0.818 | 0.979 | 0.614 | 0.865 |

**Table 5**
Different studies about the comparative evaluation of community detection algorithm using different benchmark networks.

| Authors | Ref. | Benchmarks used for testing algorithms |
|---|---|---|
| Danon et al. | (Danon et al., 2005) | GN |
| Lancichinetti et al. | (Lancichinetti and Fortunato, 2009) | GN, LFR, random graphs |
| Aldecoa et al. | (Aldecoa and Marín, 2013a; Aldecoa and Marín, 2013b) | LRF, RC |
| Orman et al. | (Orman et al., 2012) | LFR |
| Harenberg et al. | (Harenberg et al., 2014) | artificial synthetic network based on CPM |
| Xie et al. | (Xie et al., 2013) | LFR, real-world social networks of Table 3 |
| Yang et al. | (Yang et al., 2016) | LFR |

disjoint and overlapping algorithms can be seen in (Harenberg et al., 2014) and the references therein.

Xie et al. (2013) put forth a state of the art comparative study about the overlapping community detection algorithms. An in-depth comparison of multiple algorithms (a total of 14) is presented under LFR benchmark and real-world social networks. A framework to detect overlapping nodes is also proposed which helps to measure over-detection[6] and under-detection.[7]

Yang et al. (2016) performed a comparative analysis of the most widely used community detection algorithms on LFR benchmark. Accuracy[8] and computing time were the main aspects of comparison between the algorithms. The reliability of the algorithms is checked by the variations in mixing parameters and the number of nodes while generating the benchmark graphs.

Table 5 sums up the studies regarding testing of algorithms on standard benchmarks as described above. Table 6 shows computational cost of community detection algorithms of Section 2 on LFR benchmark network.

## 4. Applications of community detection

The goal of community detection algorithms is to deduce the properties and relationship of nodes in a graphical network which is not possible from direct measurements/observations. This section gives a flavor of what can be done by using community detection algorithms. There are many networks such as social networks, technological networks, informational networks, and biological networks where community detection is playing an important role. Some prominent application areas can be seen in Fig. 4. Table 7 can be seen to get quick references about the application of community detection algorithms in a particular network domain. Below we present the potential applications of community detection algorithms in different domains.

### 4.1. Applications in online social networks

An online social network is the interaction of people with each other through the web. Online social networking services—such as those offered by Facebook, Twitter, Skype, Instagram, WhatsApp, Flickr—are extremely popular among personal and business users (Ellisonet al, 2007). Yang et al. (2010) discussed applications of community mining algorithms in social networks that range from network reduction for large scale network analysis to community mining in distributed and dynamic networks. In that chapter, different algorithms are applied to mine scientific collaboration network from these social networks as well.

Social media analytics has been shown to be very effective in *Information and Communication Technology for Human Development* (ICTD) research for diverse tasks such as epidemic control, crisis response, and predictive policing (Ali et al., 2016) (Qadir et al., 2016). The community detection in these networks helps to infer the offline as well as on-line relations of individuals, finding a network of friendship between students of different universities and analysis of networks regarding scientific collaborations (Bedi and Sharma, 2016).

Twitter is a popular social networking service that is being widely used in fields as diverse as education, broadcasting, and corporate marketing. Various studies have utilized different community detection techniques on Twitter to analyze public emotional reaction, visual expression (Diakopoulos and Shamma, 2010) and visualizing user relationships and network characteristics (De Choudhury et al., 2010). Ozer et al. (2016) used non negative matrix factorization framework to cluster the politically inclined users on Twitter network into political communities. Hughes and Palen (2009) analyzed the reactions of Twitter users on the social and political issues. They showed that how these big events can attract new users to Twitter. In another work, Java et al. (2007) studied topological and geographical properties of Twitter's social network and presented the observations of the microblogging process. The users intentions and how the users with same intentions come close to each other is also studied.

Ferrara et al. (Ferrara, 2012) have studied large-scale community structures by using more than 500 million Facebook users. They identified the communities with a high degree of similarity and also showed the emergence of well-defined communities inside the Facebook.

Tyler (Tyler et al., 2005) applied modified GN algorithm to study emails network between the people working at HP labs. The proposed method can measure the degree of membership of each node of the

[6] when the count of claimed overlapping nodes is more than existing number of overlapping nodes.

[7] when the count of claimed overlapping nodes is less than existing number of overlapping nodes.

[8] It is a measure of similarity between the structure generated by LFR benchmark and the algorithm under test.

**Table 6**
Table showing how the computational cost of different approaches scales with $n$ number of nodes, $m$ number of links, $c$ number of communities, $t$ number of predefined iterations with size of community $s$ and average degree $k$ of LFR benchmark graphs. (Lancichinetti and Fortunato, 2009; Yang et al., 2016; Fortunato and Hric, 2016).

| Algorithm | Order of Cost | Algorithm | Order of Cost |
|---|---|---|---|
| Girvan and Newman (2002) | $O(m^2n)$ | Fortunato et al. (2004) | $O(m^3n)$ |
| Newman and Girvan (2004) | $O(m^2n)$ | Newman (2004a) | $O(n \log^2 n)$ |
| Radicchi et al. (2004) | $O(m^4/n^2)$ | Palla et al. (2005) | $O(\exp(n))$ |
| Clauset et al. (2004) | $O(n \log^2 N)$ | Raghavan et al. (2007) | $O(m)$ |
| Blondel et al. (2008) | $O(m)$ | Reichardt and Bornholdt (2006) | $O(n^{3.2})$[45] |
| Pons and Latapy (2005) | $O(n^2m)$ | Baumes et al. (2005) (RaRe) | $O(n^2)$ |
| Pons and Latapy (2005) (For sparse networks) | $O(n^2 \log N)$ | Lancichinetti et al. (2009) (LFM) | $O(cs^2)$ |
| Rosvall and Bergstrom (2008) | $O(m)$ | Girvan and Newman, (2002) | $O(m^2n)^3$ |
| Ahn et al. (2010) | $O(nk^2_{max})$ | Xie et al. (2011) (SLPA) | $O(tm)$ |
| Chen et al. (2010) | $O(m^2)$ | Boettcher and Percus, (2001b) | $O(n^2 \log n)$ |
| Guimera and Amaral (2005) | parameter dependent | Reichardt and Bornholdt, (2004) | parameter dependent |

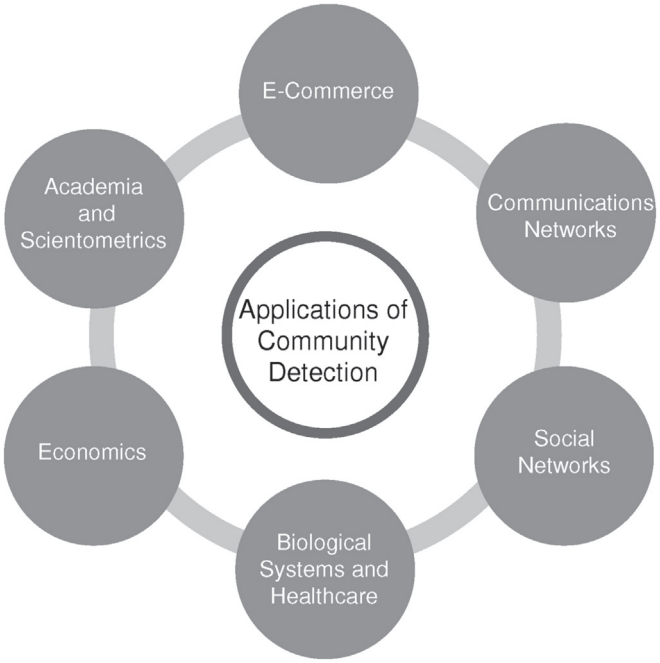



**Fig. 4.** Some prominent applications in which community detection is useful.

community and can also be applied in case of overlapping community structure.

Due to the huge increase in photo sharing on social media such as Flickr, it is needed to cluster photos in order to automatically organize the content. Papadopoulos et al. (2010) presented a graph based clustering algorithm to extract groups of related images within large image collections. It helped the users to navigate through a large collection of photos very efficiently. Moellic et al. (Moëllic et al., 2008) used the shared nearest neighbor approach on two graphs of photos in which photos served as nodes and the tags served as the links between photos. The proposed approach was compared with the classical $k$-means clustering algorithm. The results proved that the algorithm shows robust and representative clusters as compared to $k$-means approach.

Furthermore, in social texts streams, clustering finds application in detecting and tracking the events (Sayyadi et al., 2009). As the events and stories can be characterized on the basis of keywords, documents describing the same event will contain similar set of keywords. H. Sayyadi (Sayyadi et al., 2009) proposed and developed an algorithm about event detection and co-occurrence of keywords by using community detection algorithms used for social network analysis.

Due to large-scale popularity of social networks, some security issues come in front, i.e., social network worms characterized by the high concealment and difficult to eradicate. Wang et al. (2017) proposed an inhibition strategy against these worms by using community detection method based on link clustering.

Community discovery methods can also be applied to detect terrorist groups in social networks. The social media pages or groups are controlled by the owners to authenticate the unwanted access. Therefore, a terrorist group can be identified by the friend of friend activities using betweenness centrality measure and community detection algorithms. Waskiewicz (2012) has provided a good analysis of terrorist group activities on the social network websites like Facebook and Twitter. Techniques ranging from ego network analysis, betweenness centrality, and community detection algorithms, are applied to identify these friend of a friend relationships.

### 4.2. Applications in communication networks

The presence of community structure in the communication networks has enabled a wide variety of applications such as efficient routing in Mobile Ad Hoc Networks (MANETs) and worm containment in Online Social Networks (Nguyen et al., 2014). The social-aware algorithms for routing in MANETs are of great interest, because they have the properties like the social networks (Hui and Crowcroft, 2007; Daly and Haahr, 2007; Chaintreau et al., 2007). The community detections based routing algorithms have provided substantial improvements over the conventional routing methods (Chaintreau et al., 2007; Hui et al., 2011). However, these algorithms are not applicable to dynamic MANETs since they require significant processing time and computational costs for re-computation of the network structure at every change in the network topology (Nguyen et al., 2014). Blondel et al. (2008) studied a mobile phone communication network and applied a fast hierarchical modularity optimization algorithm proposed by the authors themselves.

Community detection algorithms are also being used in search engines designing and optimization, where similar objects are separated using different clustering algorithms (Ahmed and Bansal, 2013). World Wide Web has a graphical structure in which web pages act as nodes and links between them act as edges (Scharnhorst, 2003). Getting the searched item on the front page is directly proportional to the efficiency of the clustering algorithms applied to group the objects on the search engine (Liu et al., 2007), (Girolami and He, 2003). Lipai (2008), applied $k$-means clustering algorithm with some modifications for web page clustering process. The efficiency of the algorithm is tested on criteria of easy retrieval in term of speed and relevance. Soliman et al. (2015) presented a novel approach for search engine results based on the semantics of retrieved documents instead of words in the documents. This approach is a variant of hierarchical clustering and helps to cluster documents on the semantic basis.

**Table 7**
Reference papers for applications of community detection algorithms in various network domains.

| Network Domains | References |
|---|---|
| Social Networks | (Yang et al., 2010; Bedi and Sharma, 2016; Tyler et al., 2005; Ellisonet al, 2007; Qadir et al., 2016; Ali et al., 2016; Diakopoulos and Shamma, 2010; Ozer et al., 2016; De Choudhury et al., 2010; Hughes and Palen, 2009; Java et al., 2007; Ferrara, 2012; Papadopoulos et al., 2010; Moëllic et al., 2008; Wang et al., 2017; Sayyadi et al., 2009; Waskiewicz, 2012) |
| Communications Networks | (Blondel et al., 2008; Nguyen et al., 2014; Hui and Crowcroft, 2007; Chaintreau et al., 2007; Daly and Haahr, 2007; Hui et al., 2011; Ahmed and Bansal, 2013; Scharnhorst, 2003; Soliman et al., 2015) |
| E-Commerce | (Adomavicius and Tuzhilin, 2005; Zanin et al., 2008; Ricci et al., 2011; Ríos and Videla-Cavieres, 2014; Reichardt and Bornholdt, 2007; Golbeck, 2005; Beigi et al., 2014) |
| Academia and Scientometrics | (Chakraborty and Chakraborty, 2013; Nguyen et al., 2014; Oyelade et al., 2010; Wang et al., 2016; Xin et al., 2013; Ji Jinet al, 2016) |
| Biological Systems and Healthcare | (Ravasz et al., 2002; Maslov and Sneppen, 2002; Krause et al., 2003; Guimerà et al., 2010; Shen-Orr et al., 2002; Schwarz et al., 2008; Viana et al., 2009; Yanrui et al., 2015; Chen et al., 2014b; Rives and Galitski, 2003; Zhang, 2017; Sporns, 2013) |
| Economics | (ŞERBAN et al., 2013; Brauksa, 2013; Gui et al., 2014; Wu et al., 2015) |

### 4.3. Applications in E-Commerce

Detecting communities in smart marketing has an interesting application such as enhancing on-line shoppers, products recommendation, and targeted advertising. In Large shopping networks, community detection can be used to classify customers and give recommendations about future purchases depending upon their purchase history (Adomavicius and Tuzhilin, 2005). In this context, Zanin (Zanin et al., 2008) proposed an algorithm for product recommendation systems which recommends the products to customers based on their interest. Community detection based social recommendation systems yield better recommendations by identifying the geographical location of users, their social relationships, and preferences of their friends (Reddy et al., 2002; Ricci et al., 2011). An interesting approach based on data mining techniques has been proposed in (Ríos and Videla-Cavieres, 2014) that generates different communities of products. The product communities can be used to group customers, enhance store layout and provide personalized recommendations as well as it also allows to complete the billions of transactions within a reasonable time.

Reichardt et al. (Reichardt and Bornholdt, 2007) applied community detection algorithm proposed by the authors themselves on a bidding data network of Germen version of e-Bay which is the most popular auction site. In that network bidders act as nodes and an edge exists if the two bidders have interest for the same item. Clusters are detected using a multiresolution modularity optimization method.

Golbeck (2005) was first to work on user's trust prediction and defined properties of trust such as composability, asymmetry, and transitivity. He also introduced different algorithms to predict binary and weighted trust values based on a specific trust propagation model. Beigi et al. (2014) proposed the community-based model for trust propagation between users, even they are not closely connected. This approach leverages the available network of trust relations and rating similarities by assuming that the customer's trust values are strongly correlated with other customers in the same community.

### 4.4. Applications in academia and scientometrics

Community detection plays a vital role in academics, i.e., it is very important to monitor and predict the performance of students. Oyelade et al. (2010) used *k*-means clustering for the prediction of students' performance. Similarly, academic search engines are crucial for science research activities. A study exploits community detection methods and proposes two algorithms, which comprehensively analyze several metrics like author similarity, influence textural similarity, and closeness for efficient paper recommendation systems in (Wang et al., 2016).

In academic library services, books recommendation is an important task for constructing learning environment. Xin et al. (2013) proposed a novel method for books recommendation based on influential entities and user's social behavior. The authors worked on the identification of communities having similar interests and proposed a top-*N* recommendation system to recommend books to the readers. The system was based on the books borrowing record which was very helpful in developing reader-reader similarity network.

Citation network analysis is a well-known research area, used for ranking of authors and the publication places of research papers. Due to a huge number of publications every year, it is very challenging for users to grab relevant materials. Due to interdisciplinary publications, it is mandatory to make efficient overlapping community detection algorithms (Chakraborty and Chakraborty, 2013). Ji et al. (Ji Jinet al, 2016) have put a good effort in order to summarize the co-authorship and citations network data for statisticians. It is termed as the first effort of its own type by providing the community with data set for self-study. They have performed several analyses to unveil many interesting patterns. In another study, an adaptive modularity-based approach, Quick Community Adaptation (QCA) is proposed and tested on the arXiv[9] e-print citation network (Nguyen et al., 2014). It is shown that the results obtained by QCA are promising and higher than those obtained by MIEN and OSLOM (Order Statistics Local Optimization Method).

### 4.5. Applications in biological systems and healthcare

The existence of community structures is widely accepted in biological networks such as metabolic network (Ravasz et al., 2002), protein interaction networks (Maslov and Sneppen, 2002), food-web structure (Krause et al., 2003; Guimerà et al., 2010), gene regulatory networks (Shen-Orr et al., 2002), and pollination networks (Olesen et al., 2007). There are a variety of community detection algorithms developed for social networks that can also be successfully extended to the biological networks (Schwarz et al., 2008; Viana et al., 2009). For example, component definition and GN algorithm are used in (Yanrui et al., 2015) to detect communities in thermophiles metabolic networks and mesophiles metabolic networks. Liu et al. (Liu and Chen, 2013) proposed an algorithm named as Disease-Gene Network detecting algorithm based on Principal Component Analysis (DGN-PCA) in order to unveil the communities in a diseasome ("disease-gene association" (Goh et al., 2007)) bipartite network. The algorithm was aimed for disease prevention and medical diagnosis.

Proteins are related to two cellular modules, i.e., functional modules and protein complexes (see (Chen et al., 2014b) for more details). These two moduli of proteins have been explored in various studies, such as the modular organization of interacting proteins and protein complexes is studied in (Rives and Galitski, 2003); protein complexes and functional modules in yeast is explored in (Spirin and Mirny, 2003) and

[9] http://www.cs.cornell.edu/projects/kddcup/datasets.html.

functional modules in protein-protein interaction networks is detected by using the standard GN algorithm by Dunn et al. (2005).

Community detection also finds its applications in healthcare. It is helpful in lungs cancer detection by analyzing the rampant growth of cells in lung tissues (Bechtel et al., 2005), neurosciences to study brain networks (Zhang, 2017; Sporns, 2013), and flow of disease detection by studying contact networks (Zhao and Gao, 2007; Chu et al., 2009). To detect early stages of breast cancer, Wang et al. (Wang and Garibaldi, 2005) applied *k*-means (non-fuzzy) and fuzzy *C* means (FCM) to cluster the IR spectra collected from a lymph node tissue section which is helpful in cancer diagnosis.

### 4.6. Applications in economics

In economics and financial data analysis, it is important to categorize the companies on basis of their economic value, net income, current sales, equity, and stock price. Serban et al. (ŞERBAN et al., 2013) conducted a study to cluster companies based on their economic and financial indicators using ascending hierarchical clustering, partitional and *k*-means clustering. Similarly, exploration of economic development in different regions (country wise or part of the country) can also be highlighted using clustering techniques. Brauksa (2013) used *k*-means clustering to discuss similarities and differences of economic and social development in municipalities of Latvia.

In the stock market, each stock is represented by a vertex and edge represents the correlations of stock values in the market. This portfolio management helps the investors in distributing or concentrating their investments. Gui et al. (2014) stated the way to construct the network of stock market and detect communities in it. They revealed community structure by using the modularity metric *Q*. The empirical data used in this analysis was based on the Hong Kong stock market.

Wu et al. (2015) investigated community structure of components stock of Shanghai's stock market SSE180-index. They built a stock correlation network taking stocks as nodes and edges as the correlation coefficient of logarithm returns of the stock price. Community structure is revealed using GN algorithm.

To facilitate the understanding of readers, Table 8 shows different applications domains in which community detection algorithms have been applied to serve different purposes.

### 4.7. Others

Besides having applications in above-mentioned domains, community detection has various general applications to identify the important hidden network's community structure.

#### 4.7.1. Fraud detection

Detection of fraud is an exciting topic that requires inputs from other domains like community detection along with data mining techniques. Frauds detection has enormous applications in real-life networks like telecommunication network, social networks, and healthcare.

Pinheiro (2012) detected fraud events in a telecommunication network with a two-step algorithm. In the first step, communities were detected via data analysis of a telecommunication network and in second step some basic graph properties such as degree and betweenness were used as a measure to detect the abnormal nodes or outliers. Similarly, Gangopadhyay et al. (Gangopadhyay and Chen, 2016) developed two methods to analyze different types of relationships especially small but exclusive relationships for the detection of fraud in healthcare. These algorithms are very efficient for the evaluation of such suspicious communities and also applicable to large scale healthcare dataset. In another study, an interesting approach based on graph analytics is adopted for the detection of fraud and abuse in healthcare (Liu et al., 2016b).

#### 4.7.2. Link prediction

Link prediction tells about the change in association pattern between nodes and factors affecting the association of nodes. In complex networks, it predicts future links between nodes by finding attributes of nodes and links between nodes (Tan et al., 2014). Valverde-Rebaza and de Andrade Lopes (2012) presented such approach for link prediction based on the community structures in the dynamic complex networks like social networks. Link prediction in social networks has many potential applications such as recommending new items to users, friendship suggestion and discovering spurious connections (Jalili et al., 2017). In bio-informatics, link prediction technique is the key to determine protein-protein interaction (Lei and Ruan, 2013). Similarly, in commerce, it is the key technique of recommender systems (Cao et al., 2010) and it is at the back of site hyperlink prediction over Internet (Liben-Nowell and Kleinberg, 2007).

#### 4.7.3. Refactoring software packages

Refactoring is an effective way to improve the design of the existing code without changing its working (Fowler and Beck, 1999; Pan et al., 2013). Community detection is also being employed for refactoring of software packages. For example, Pan et al. (2013) presented a novel method to refactor object-oriented software package structure by using classes and their dependencies in software networks. Maruyama and Shima (1999) proposed an automatic refactoring mechanism by using weighted dependence graphs depending upon the modification histories. Kanemitsu et al. (2011) presented a program dependency graph visualization approach to identify refactoring opportunities in the source code based on the strength of data connection between sentences. Some other representative works in this domain are reported in (Atkinson and King, 2005; Tsantalis and Chatzigeorgiou, 2011).

#### 4.7.4. Anomaly detection in networks

The anomaly can be described as the observation that deviates from other observed data and makes it suspicious (Hawkins, 1980). In some cases, anomalies or outliers may be due to fault in measurements, bad data or due to data corruption. Some prominent approaches that are highly helpful in determining the anomalies are clustering based (Nieves and Jiao, 2009; Amer and Goldstein, 2012; Portnoy et al., 2001), classification based (Yang et al., 2006; Steinwart et al., 2005; Mukkamala et al., 2002), graph based (Noble and Cook, 2003; Padmanabhan et al., 2013), knowledge-based and soft computing based (Bhuyan et al., 2014). Detailed discussion on anomaly detection can be seen in (Bhuyan et al., 2014; Akoglu et al., 2015). Anomaly detection is a crucial task in some of the very high impact applications such as finance, healthcare and law enforcement and security (Akoglu et al., 2015).

M.Araujo (2017) proposed algorithms for communities detection and anomalies discovery in temporal networks. The algorithm has been tested on various networks such as phone calls, flights, co-authorship, "purchased together" items, movies ratings etc. The study has a broad impact on recommendation systems, fraud detection, contextual forecasting and graph understanding.

Chen et al. (2012) proposed an approach to identify community-based anomalies in evolutionary networks. They formulated a parameter-free and scalable algorithm to identify all possible types of community-based anomalies (grown, split, merged, born, shrunken and vanished communities).

Baingana et al. (Baingana and Giannakis, 2016) proposed a novel approach for jointly tracking communities and anomalies in time-varying networks. The proposed algorithm tracks temporal and overlapping communities while compensating for anomalies. In social networks, community detection methods can also be used to detect nodes' anomalous behavior (such as unnecessary friend requests on Facebook or spam emails) (Ranshous et al., 2015).

**Table 8**
Table shows different applications domains in which community detection algorithms have been applied to serve different purposes.

| Application Domain | Reference | Network Under Test | Clustering Technique | Community Detection Task |
|---|---|---|---|---|
| Social Networks | Ozer et al. (2016) | Twitter | NMF | To cluster the politically motivated users from huge content categories of Twitter network |
| | Waskiewicz (2012) | Facebook | Hierarchical | Terrorist group identification using friend of a friend network |
| | Tyler et al. (2005) | Emails network | GN | Identification of communities from email logs |
| | Wang et al. (2017) | Networks from Table 3 | GN, LPA | Identification and prevention of social network worms |
| Communication Networks | Blondel et al. (2008) | Web pages network | Modularity Optimization | Unveiling community structure of large and complex networks such as unravelling of modular structure of internet network |
| | Ahmed and Bansal 2013) | Web pages network | *K*-means | Finding the relevant web page from huge number of web pages, retrieving relevant websites from larger collection of websites, mining the databases of unlabelled documents |
| | Soliman et al. (2015) | Search engine network | Hierarchical | Retrieval of documents from search engines based on semantics instead of words, i.e., semantic clustering |
| | Hui et al. (2011) | MANETs | CPM | To enhance the delivery performance and forwarding performance of social-based forwarding algorithms |
| E–Commerce | Ríos and Videla-Cavieres 2014) | Transactions record from a retail chain in Chile | COPRA, SLPA | To enhance store layout, generate customers groups, personalized recommendations among others |
| | Reichardt and Bornholdt 2007) | E–Bay | GN | To reveal coherent interest profiles of the bidders in a bidding network for market segmentation of e-bay |
| | Beigi et al. (2014) | Data from Epinions and Ciao(product review sites) | Game-theoretic | Bootstrapping trust prediction on product review websites |
| Academia and Scientometrics | Oyelade et al. (2010) | Synthetic data network of a private school | *k*-means, Hierarchical | Modeling a benchmark clustering algorithm to monitor the progression of students in order to enhance the decision making performance of academic planners in higher institution |
| | Wang et al. (2016) | Microsoft Academic Graph | Random walk | To reveal implicit relevance between papers |
| | Ji Jinet al (2016) | Citations network | Spectral | Identification of highly cited authors, provision of dataset for future research in this domain |
| Biological Systems and Healthcare | Wang and Garibaldi (2005) | Synthetic dataset from consenting patients | *k*-means, FCM | Clustering of lymph node tissues diagnosed by spectroscopic techniques for breast cancer detection |
| | Liu and Chen (2013) | Dataset from patients | PCA | Disease prevention and medical diagnosis |
| | Dunn et al. (2005) | PPI Network | GN | The method helps to quickly screen the small to medium size protein interaction datasets |
| Economics | Gui et al. (2014) | Stock market network based on stock prices | Greedy Optimization | For decision making in the stock market to predict and control financial activities |
| | ŞERBAN et al. (2013) | Synthetic network of a stock market | Hierarchical | Clustering of companies based on economic value added, net income, current sales, equity and stock price in New York Stock Exchange (NYSE) |
| | Wu et al. (2015) | Undirected stock correlation network | GN | To look into the community structure of components stock of Shanghai Stock Market SSE 180-index |

## 5. Challenges faced by community detection algorithms

### 5.1. Clustering of mixed-type data

Despite the excessive studies on the algorithms for community detection, identifying structures in real-world networks still face different hurdles. One prominent hurdle faced by traditional clustering algorithms is dealing with mixed-type data[10]. Mixed-type categorical and numerical data are ubiquitous in real-world networks. In these networks, one has to face not only the numerical data but the data with categorical attributes such as color, gender and type of disease etc. as well (Du et al., 2017; Ding et al., 2017).

Traditional clustering algorithms can only handle numerical values that makes them unsuitable for clustering of mixed-type data. One possible technique to handle the mixed-type data is pre-processing i.e., to convert the categorical attributes to new forms e.g., the binary strings, and then applying all the aforementioned algorithms for numerical data such as *K-means*. Binary encoding is a common pre-processing technique which translates each categorical attribute to a binary attribute such as Ralambondrainy's algorithm in (Ralambondrainy, 1995). But these pre-processing techniques destruct the original structure of attributes. As *k-means* paradigm is an iterative process, the algorithms for mix-type data may get trapped in local optima. Rodriguez et al. (Rodriguez and Laio, 2014) proposed a density peak algorithm (DPC) which detects non-spherical clusters and does not need to iterate as well. But DPC cannot find number of clusters. Efforts are still underway to overcome the shortcomings of DPC algorithm.

Liang et al. (Liang and Chen, 2016) proposed the 3DC clustering to improve DPC algorithm. This algorithm adapts divide and conquer strategy with density-reachable concept to correctly find the number of clusters and most appropriate cluster centers. Du et al. (2016) proposed DPC based on k nearest neighbors (DPC-KNN) that also tries to overcome the algorithmic limitations of DPC by offering another option for computing local density. Du et al. (2017) proposed a novel algorithm for clustering of the mixed-type data called DPC-MD in which DCP is improved to deal with numerical, categorical and mixed data by using a new similarity criteria. Ding et al. (2017) proposed an entropy based DPC algorithm for mixed type of data employing fuzzy neighborhood (DP-MD-FN). Here DPC is integrated with entropy based technique and fuzzy neighborhood relation is used to redefine measure of local density. The results show that this algorithm is robust against the noise in categorical attributes.

### 5.2. Validation issues

The evaluation of the quality of identified communities by different algorithms is also problematic. Currently, the state of the art is to design artificial networks with desired structural properties and run the being tested algorithm on such structures. But this process does not validate the performance of an algorithm for real network. Ideally, it is required to compare the performance of different community detection algorithms on real networks. It is possible to take the real network and add (or remove) links in it in order to check the accuracy of the algorithm in the discovery of robust community structure (Karrer et al., 2008).

### 5.3. Incorporating bipartite information

A bipartite network contains two different types of nodes. An edge only exists between the nodes of different type leaving the same type of nodes detached. For example, gene regulatory network, drug-target network, and gene-disease network, contains two different types of nodes and links (Gulbahce and Lehmann, 2008). To analyze bipartite networks, they are projected onto some other networks, e.g., disease-disease network. However, it has been proved that the projection of network discards crucial network information (Lehmann et al., 2008). Incorporating the bipartite information can result in a refined unveiling of community information in the network under consideration.

### 5.4. Availability of live dataset

In addition, both the collection and processing of large-scale real data are very difficult. Some of the challenges involved include pre-processing of data, ensuring data privacy, and satisfying ethical concerns. In particular, the raw data need to be parsed and pre-processed before it is possible to create a network from the dataset by using the observed relations. As the paradigm has shifted to dynamic community detection from the static one, the live or dynamic dataset is a dire need to carry on the research in the dynamically evolving networks. But in various cases, obtaining ground-truth data to evaluate the results is impossible or non-trivial.

### 5.5. Others

Many of the existing community detection algorithms implicitly assume that knowledge about the entire structures of the network is available and known. Since such global information might not be known for large and complex networks, therefore local community detection algorithms are getting more focus (Yang and Leskovec, 2015). These local algorithms start from given a number of seed nodes and expand them into possible overlapping structures by analyzing a small part of a large network. This can lead to various redundant community structures, which is computationally expensive. The selection of a good number of seeds is very crucial by using a seeding algorithms. If seeding algorithm fails to choose sufficient number of seeds, then communities might only encompass a subset or part of the network. Therefore, the selection of enough seeds distributed over the whole network is challenging.

In the same way, most of the recent community detection algorithms assume the networks as the variation of a random graph, while the real-world networks are far from this assumption. Therefore, we need such algorithms that can directly include the known pattern in the statistical model (Danon et al., 2005; Gulbahce and Lehmann, 2008).

## 6. Open issues and future trends

Community discovery is still an active research topic with multiple open problems that relate to theoretical and empirical prospects. The concept of community in the networks is qualitatively intuitive. But analyzing network structure requires the unambiguous and quantitative specification of community structure. Once the community is defined, it is possible to find all subgraphs in a given network that fulfill the definition. In practice, this task is computationally out of reach even for small systems. In some cases artificial networks are considered, that is far from the objective of community detection. Below we discuss some open issue and recent trends.

### 6.1. Transient community detection

When temporal information related to a community (i.e., transient community) is considered, most of the existing community detection methods fail, because of the aggregation of contact information into a single network which is either weighted or unweighted. Detection of transient communities plays an important role in all delay tolerant networks. Recently, a contact-burst based clustering method (CCM) has been proposed by Zhang et al. (Zhang and Cao, 2017) in order to reveal transient communities by tapping the pairwise contact process.

[10] A data set which contains both numerical and categorical attributes is termed as mixed-type data (Du et al., 2017; Ding et al., 2017).

### 6.2. Estimation of number of communities

As community detection is a widely studied domain in network analysis and there exist numerous algorithms to detect communities in several types of networks which range from artificial to real-world network. But it should be noted that most of the algorithms require the number of communities beforehand as an input parameter to these algorithms which decreases the utility of community detection as an analytical tool. There exist two approaches for the estimation of the number of communities in a network. One method is based on modularity maximization which itself can also be used for community detection (Newman and Girvan, 2004; Newman, 2004a) and the other one is based on statistical inference. Newman et al. (Newman and Reinert, 2016) presented a rigorous algorithm based on statistical inference for the problem described above. Le et al. (Le and Levina, 2015) proposed a method for estimation of number of communities by using spectral properties of certain graph operators.

### 6.3. Temporal analysis

Real world networks evolve over time, and the communities in such networks also change accordingly. Therefore, exploiting temporal characteristics are crucial to learn deep insights about the communities. This problem has studied for single layer graphs but almost no work is done for multi-layer graphs (Kim and Lee, 2015). The evolution modeling of multi-layer graphs is extremely complex and still an open issue to solve.

### 6.4. Selfish nodes in network

Community detection algorithms face problems in terms of handling selfish nodes which use the resources (e.g., routing paths) only for their own purpose and hesitate to share the resources with their neighbors. This behavior seriously affects the performance of the networks such as wireless sensor networks, mobile social networks, and mobile ad hoc networks. For such networks, algorithms are required that can detect such nodes and also punish this nodal behavior (Sen and Goswami, 2010).

### 6.5. Differential privacy

Communities in complex networks capture a valuable information about the evolution of networks and even about the organization. In the last decade, a great number of community detection algorithms have been proposed but the privacy factors like differential privacy are rarely considered (Nguyen et al., 2016). By differential privacy, the interactions between the nodes are made hidden in output clustering. This problem is quite new and needs special attention to develop algorithms to accommodate the privacy factor.

### 6.6. Outlier aware algorithms

Many community detection algorithms for the overlapping communities have overlooked the anomalous behavior of nodes. Recently, Baingana et al. (Baingana and Giannakis, 2016) have proposed a new approach for tracking communities as well as anomalies in dynamic networks. Unlike prior methods, their algorithm compensates for the anomalies and capitalizes on sparsity. There is a need to make the outlier aware efficient algorithms whose convergence is not disturbed by the anomalous nodes in the network.

### 6.7. Scalability of community detection algorithms

The dimensionality reduction in case of big data is the key concern in modern algorithms which tend to bring the data to low dimensional space before discovering patterns. In this context, PCA-based algorithms are of much importance. Online clustering and streaming clustering are the two recent advances in the algorithms which can tackle the big data problems. But there is still need to make algorithms more efficient, versatile, scalable and distributed in order to reduce the overhead by improving the accuracy and speed related performance parameters.

### 6.8. Heterogeneity of real-world networks

Community detection algorithms are developed using the multidisciplinary techniques that range from statistical physics to applied mathematics and computer science. These techniques emphasize on the improvement of the community detection while keeping the computational complexity as low as possible. Further, it is a ground reality that there is no unique definition of community, so it is very difficult that a single algorithm is stamped as a reliable algorithm for community detection in all type of networks. For example, finding communities in a social network is totally different than detecting cluster in a cellular network. This heterogeneity of real-world networks poses a discrete challenge for community detection algorithms (Gulbahce and Lehmann, 2008). Similarly, the growth of real-world networks is inevitable and efficient and scalable algorithms are required to handle the growing size of networks.

## 7. Conclusion

In this paper, we have presented a comprehensive and up-to-date literature review on the community detection algorithms in networks including their positives and negatives, with special intentions on the modern inclinations in this field. The criteria for performance comparison of these algorithms are also highlighted in addition to tests of these algorithms on standard benchmarks. Unlike previous surveys on community detection, we also reviewed the multidisciplinary applications of community detection in various domains like sociology, biology, education, and technology. Further, we also highlighted the open issues and future trends in this field. We hope that this paper will serve as an inclusive and descriptive study on community detection and its applications that will help the researchers and young scientists to quickly become familiar with the most important aspect of this domain.

## References

Adomavicius, G., Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. 17 (6), 734–749.
Ahmed, M.E., Bansal, P., 2013. Clustering technique on search engine dataset using data mining tool. In: Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference. IEEE, pp. 86–89.
Ahn, Y.-Y., Bagrow, J.P., Lehmann, S., 2010. Link communities reveal multiscale complexity in networks. Nature 466 (7307), 761–764.
Akoglu, L., Tong, H., Koutra, D., 2015. Graph based anomaly detection and description: a survey. Data Min. Knowl. Discov. 29 (3), 626–688.
Aldecoa, R., Marín, I., 2012. Closed benchmarks for network community structure characterization. Phys. Rev. E 85 (2), 026109.
Aldecoa, R., Marín, I., 2013. Exploring the limits of community detection strategies in complex networks. Sci. Rep. 3.
Aldecoa, R., Marín, I., 2013. Surprise maximization reveals the community structure of complex networks. Sci. Rep. 3.
Ali, A., Qadir, J., ur Rasool, R., Sathiaseelan, A., Zwitter, A., Crowcroft, J., 2016. Big data for development: applications and techniques. Big Data Anal. 1 (1), 2.
Amer, M., Goldstein, M., 2012. Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer. In: Proc. of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012), pp. 1–12.
Ana, L., Jain, A.K., 2003. Robust data clustering. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference. vol. 2. IEEE. pp. II–II.
Araujo, M., 2017. Communities and Anomaly Detection in Large Edge-labeled Graphs.
Arenas, A., Diaz-Guilera, A., Pérez-Vicente, C.J., 2006. Synchronization reveals topological scales in complex networks. Phys. Rev. Lett. 96 (11), 114102.
Asur, S., Parthasarathy, S., Ucar, D., 2009. An event-based framework for characterizing the evolutionary behavior of interaction graphs. ACM Trans. Knowl. Discov. Data TKDD 3 (4), 16.
Atkinson, D.C., King, T., 2005. Lightweight detection of program refactorings. In: Software Engineering Conference, 2005. APSEC'05. 12th Asia-Pacific. IEEE. pp. 8–pp.

B. Auffarth, Spectral Graph Clustering, Universitat de Barcelona, course report for Technicas Avanzadas de Aprendizaj, at Universitat Politecnica de Catalunya.
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. ACM, pp. 44–54.
Baingana, B., Giannakis, G.B., 2015. Dynamic and decentralized learning of overlapping network communities. In: Computational Advances in Multi-sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop. IEEE, pp. 97–100.
Baingana, B., Giannakis, G.B., 2016. Joint community and anomaly tracking in dynamic networks. IEEE Trans. Signal Process. 64 (8), 2013–2025.
Bak, P., Tang, C., Wiesenfeld, K., 1987. Self-organized criticality: an explanation of the 1/f noise. Phys. Rev. Lett. 59 (4), 381.
Barabási, A.-L., Albert, R., 1999. Emergence of scaling in random networks. Science 286 (5439), 509–512.
Barnard, S.T., Pothen, A., Simon, H., 1995. A spectral algorithm for envelope reduction of sparse matrices. Numer. Lin. Algebra Appl. 2 (4), 317–334.
Baumes, J., Goldberg, M.K., Krishnamoorthy, M.S., Magdon-Ismail, M., Preston, N., 2005. Finding communities by clustering a graph into overlapping subgraphs. IADIS AC 5, 97–104.
Bechtel, J.J., Kelley, W.A., Coons, T.A., Klein, M.G., Slagel, D.D., Petty, T.L., 2005. Lung cancer detection in patients with airflow obstruction identified in a primary care outpatient practice. CHEST J. 127 (4), 1140–1145.
Bedi, P., Sharma, C., 2016. Community detection in social networks. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 6 (3), 115–135.
Beigi, G., Jalili, M., Alvari, H., Sukthankar, G., 2014. Leveraging community detection for accurate trust prediction. In: ASE International Conference on Social Computing.
Bertsekas, D.P., Tsitsiklis, J.N., 1989. Parallel and Distributed Computation: Numerical Methods, vol. 23. Prentice Hall, Englewood Cliffs, NJ.
Bezdek, J.C., 2013. Pattern Recognition with Fuzzy Objective Function Algorithms. Springer Science & Business Media.
Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K., 2014. Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. 16 (1), 303–336.
Blatt, M., Wiseman, S., Domany, E., 1996. Superparamagnetic clustering of data. Phys. Rev. Lett. 76 (18), 3251.
Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. J. Stat. Mech. Theor. Exp. 2008 (10), P10008.
Boccaletti, S., Ivanchenko, M., Latora, V., Pluchino, A., Rapisarda, A., 2007. Detecting complex network modularity by dynamical clustering. Phys. Rev. E 75 (4), 045102.
Boettcher, S., Percus, A.G., 2001. Extremal optimization for graph partitioning. Phys. Rev. E 64 (2), 026114.
Boettcher, S., Percus, A.G., 2001. Optimization with extremal dynamics. Phys. Rev. Lett. 86 (23), 5211.
Botta, F., del Genio, C.I., 2017. Analysis of the communities of an urban mobile phone network. PLoS One 12 (3), e0174198.
Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. 3 (1), 1–122.
Brauksa, I., 2013. Use of cluster analysis in exploring economic indicator. differences among regions: the case of Latvia. J. Econ. Bus. Manag. 1 (1), 42–45.
Bui, T.N., Moon, B.R., 1996. Genetic algorithm and graph partitioning. IEEE Trans. Comput. 45 (7), 841–855.
Cao, B., Liu, N.N., Yang, Q., 2010. Transfer learning for collective link prediction in multiple heterogenous domains. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 159–166.
Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J., 2007. Impact of human mobility on opportunistic forwarding algorithms. IEEE Trans. Mobile Comput. 6 (6).
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. ACM, pp. 1124–1131.
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.
Chen, Z., Hendrix, W., Samatova, N.F., 2012. Community-based anomaly detection in evolutionary networks. J. Intell. Inf. Syst. 39 (1), 59–85.
Chen, M., Kuzmin, K., Szymanski, B.K., 2014. Community detection via maximization of modularity and its variants. IEEE Trans. Comput. Soc. Syst. 1 (1), 46–65.
Chen, B., Fan, W., Liu, J., Wu, F.-X., 2014. Identifying protein complexes and functional modules—from static ppi networks to dynamic ppi networks. Briefings Bioinf. 15 (2), 177–194.
Chintalapudi, S.R., Prasad, M.K., 2015. A survey on community detection algorithms in large scale real world networks. In: Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference. IEEE, pp. 1323–1327.
Chu, X., Guan, J., Zhang, Z., Zhou, S., 2009. Epidemic spreading in weighted scale-free networks with community structure. J. Stat. Mech. Theor. Exp. 2009 (07), P07043.
Chung, F.R., 1997. Spectral Graph Theory, vol. 92. American Mathematical Soc.
Clauset, A., Newman, M.E., Moore, C., 2004. Finding community structure in very large networks. Phys. Rev. E 70 (6), 066111.
Daly, E.M., Haahr, M., 2007. Social network analysis for routing in disconnected delay-tolerant manets. In: Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, pp. 32–40.
Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A., 2005. Comparing community structure identification. J. Stat. Mech. Theor. Exp. 2005 (09), P09008.
Danon, L., Díaz-Guilera, A., Arenas, A., 2006. The effect of size heterogeneity on community identification in complex networks. J. Stat. Mech. Theor. Exp. 2006 (11), P11010.
De Choudhury, M., Lin, Y.-R., Sundaram, H., Candan, K.S., Xie, L., Kelliher, A., et al., 2010. How does the data sampling strategy impact the discovery of information diffusion in social media? ICWSM 10, 34–41.
Diakopoulos, N.A., Shamma, D.A., 2010. Characterizing debate performance via aggregated twitter sentiment. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 1195–1198.
Ding, C., 2004. A tutorial on spectral clustering. In: Talk Presented at ICML Slides available at: http://crd.lbl.gov/~cding/Spectral/.
Ding, C.H., He, X., Simon, H.D., 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. SDM, vol. 5. SIAM, pp. 606–610.
Ding, S., Du, M., Sun, T., Xu, X., Xue, Y., 2017. An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood. Knowl. Base Syst. 133, 294–313.
Donath, W.E., Hoffman, A.J., 1973. Lower bounds for the partitioning of graphs. IBM J. Res. Dev. 17 (5), 420–425.
Doreian, P., Batagelj, V., Ferligoj, A., 2005. Generalized Blockmodeling, vol. 25. Cambridge University Press.
Du, M., Ding, S., Jia, H., 2016. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. Knowl. Base Syst. 99, 135–145.
Du, M., Ding, S., Xue, Y., 2017. A novel density peaks clustering algorithm for mixed data. Pattern Recogn. Lett. 97, 46–53.
Dunlavy, D.M., Kolda, T.G., Acar, E., 2011. Temporal link prediction using matrix and tensor factorizations. ACM Trans. Knowl. Discov. Dataactions Knowl. Discov. Data TKDD 5 (2), 10.
Dunn, J.C., 1973. A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-separated Clusters. Taylor & Francis.
Dunn, R., Dudbridge, F., Sanderson, C.M., 2005. The use of edge-betweenness clustering to investigate biological function in protein interaction networks. BMC Bioinf. 6 (1), 39.
Ellison, N.B., et al., 2007. Social network sites: definition, history, and scholarship. J. Computer-Mediated Commun. 13 (1), 210–230.
Erdös, P., Rényi, A., 1959. On random graphs, i, Publicationes Mathematicae (Debrecen). 6, 290–297.
Evans, T.S., 2010. Clique graphs and overlapping communities. J. Stat. Mech. Theor. Exp. 2010 (12), P12037.
Farkas, I., Ábel, D., Palla, G., Vicsek, T., 2007. Weighted network modules. N. J. Phys. 9 (6), 180.
Ferrara, E., 2012. Community structure discovery in facebook. Int. J. Soc. Netw. Min. 1 (1), 67–90.
Fiedler, M., 1973. Algebraic connectivity of graphs. Czech. Math. J. 23 (2), 298–305.
Fienberg, S.E., Meyer, M.M., Wasserman, S.S., 1985. Statistical analysis of multiple sociometric relations. J. Am. Stat. Assoc. 80 (389), 51–67.
Flake, G.W., Lawrence, S., Giles, C.L., 2000. Efficient identification of web communities. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 150–160.
Fortunato, S., 2010. Community detection in graphs. Phys. Rep. 486 (3), 75–174.
Fortunato, S., Hric, D., 2016. Community detection in networks: a user guide. Phys. Rep. 659, 1–44.
Fortunato, S., Latora, V., Marchiori, M., 2004. Method to find community structures based on information centrality. Phys. Rev. E 70 (5), 056104.
Fowler, M., Beck, K., 1999. Refactoring: Improving the Design of Existing Code. Addison-Wesley Professional.
Friedman, J., Hastie, T., Tibshirani, R., 2001. The Elements of Statistical Learning, vol. 1. Springer series in statistics Springer, Berlin.
Fu, W., Song, L., Xing, E.P., 2009. Dynamic mixed membership blockmodel for evolving networks. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, pp. 329–336.
Gangopadhyay, A., Chen, S., 2016. Health care fraud detection with community detection algorithms. In: Smart Computing (SMARTCOMP), 2016 IEEE International Conference. IEEE, pp. 1–5.
Gao, L., Yang, J., Wang, H., Zhang, H., 2010. A measure of growth of user community in osns. In: Quality of Service (IWQoS), 2010 18th International Workshop. IEEE, pp. 1–2.
Ghorbani, M., Rabiee, H.R., Khodadadi, A., 2016. Bayesian Overlapping Community Detection in Dynamic Networks. arXiv preprint arXiv:1605.02288.
Girolami, M., He, C., 2003. Probability density estimation from optimally condensed data samples. IEEE Trans. Pattern Anal. Mach. Intell. 25 (10), 1253–1264.
Girvan, M., Newman, M.E., 2002. Community structure in social and biological networks. Proc. Natl. Acad. Sci. U.S.A. 99 (12), 7821–7826.
Goh, K.-I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., Barabási, A.-L., 2007. The human disease network. Proc. Natl. Acad. Sci. U.S.A. 104 (21), 8685–8690.
Golbeck, J.A., 2005. Computing and Applying Trust in Web-based Social Networks Ph.D. thesis.
Gong, M., Fu, B., Jiao, L., Du, H., 2011. Memetic algorithm for community detection in networks. Phys. Rev. E 84 (5), 056101.
Gong, M., Ma, L., Zhang, Q., Jiao, L., 2012. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. Phys. Stat. Mech. Appl. 391 (15), 4050–4060.
Gregory, S., 2007. An algorithm to find overlapping community structure in networks. In: European Conference on Principles of Data Mining and Knowledge Discovery. Springer, pp. 91–102.
Gregory, S., 2008. A fast algorithm to find overlapping communities in networks. In: Machine Learning and Knowledge Discovery in Databases, pp. 408–423.
Gregory, S., 2010. Finding overlapping communities in networks by label propagation. N. J. Phys. 12 (10), 103018.

Gui, X., Li, L., Cao, J., Li, L., 2014. Dynamic communities in stock market. Abstract and Applied Analysis, vol. 2014. Hindawi Publishing Corporation.
Guimera, R., Amaral, L.A.N., 2005. Functional cartography of complex metabolic networks. Nature 433 (7028), 895–900.
Guimerà, R., Stouffer, D., Sales-Pardo, M., Leicht, E., Newman, M., Amaral, L.A., 2010. Origin of compartmentalization in food webs. Ecology 91 (10), 2941–2951.
Gulbahce, N., Lehmann, S., 2008. The art of community detection. BioEssays 30 (10), 934–938.
Hagen, L., Kahng, A.B., 1992. New spectral methods for ratio cut partitioning and clustering. IEEE Trans. Comput. Aided Des. Integrated Circ. Syst. 11 (9), 1074–1085.
Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., Padmanabhan, K., Samatova, N., 2014. Community detection in large-scale networks: a survey and empirical evaluation. Wiley Interdiscip. Rev. Comput. Stat. 6 (6), 426–439.
Hawkins, D.M., 1980. Identification of Outliers, vol. 11. Springer.
He, D., Jin, D., Baquero, C., Liu, D., 2014. Link community detection using generative model and nonnegative matrix factorization. PLoS One 9 (1), e86899.
Hoff, P.D., Raftery, A.E., Handcock, M.S., 2002. Latent space approaches to social network analysis. J. Am. Stat. Assoc. 97 (460), 1090–1098.
Holland, P.W., Leinhardt, S., 1976. Local structure in social networks. Socio. Meth. 7 (1), 1–45.
Hu, P., Chow, S.S., Lau, W.C., 2014. Secure Friend Discovery via Privacy-preserving and Decentralized Community Detection. arXiv preprint arXiv:1405.4951.
Huang, J., Yang, B., Jin, D., Yang, Y., 2013. Decentralized mining social network communities with agents. Math. Comput. Model. 57 (11), 2998–3008.
Hughes, B.D., 1996. Random Walks and Random Environments. Clarendon Press, Oxford.
Hughes, A.L., Palen, L., 2009. Twitter adoption and use in mass convergence and emergency events. Int. J. Emerg. Manag. 6 (3–4), 248–260.
Hui, P., Crowcroft, J., 2007. How small labels create big improvements. In: Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference. IEEE, pp. 65–70.
Hui, P., Yoneki, E., Chan, S.Y., Crowcroft, J., 2007. Distributed community detection in delay tolerant networks. In: Proceedings of 2nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture. ACM, p. 7.
Hui, P., Crowcroft, J., Yoneki, E., 2011. Bubble rap: social-based forwarding in delay-tolerant networks. IEEE Trans. Mobile Comput. 10 (11), 1576–1589.
Jalili, M., Orouskhani, Y., Asgari, M., Alipourfard, N., Perc, M., 2017. Link prediction in multiplex online social networks. Open Sci. 4 (2), 160863.
Java, A., Song, X., Finin, T., Tseng, B., 2007. Why we twitter: understanding microblogging usage and communities. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis. ACM, pp. 56–65.
Ji, P., Jin, J., et al., 2016. Coauthorship and citation networks for statisticians. Ann. Appl. Stat. 10 (4), 1779–1812.
Jia, H., Ding, S., Du, M., 2017. A nyström spectral clustering algorithm based on probability incremental sampling. Soft Comput. 21 (19), 5815–5827.
Ju, C., Xu, C., 2013. A new collaborative recommendation approach based on users clustering using artificial bee colony algorithm. Sci. World J. 2013.
Kanemitsu, T., Higo, Y., Kusumoto, S., 2011. A visualization method of program dependency graph for identifying extract method opportunity. In: Proceedings of the 4th Workshop on Refactoring Tools. ACM, pp. 8–14.
Karrer, B., Levina, E., Newman, M.E., 2008. Robustness of community structure in networks. Phys. Rev. E 77 (4), 046119.
Kelley, S., 2009. The Existence and Discovery of Overlapping Communities in Large-scale Networks Ph.D. thesis. Rensselaer Polytechnic Institute.
Kelley, S., Goldberg, M., Magdon-Ismail, M., Mertsalov, K., Wallace, A., 2011. Handbook of Optimization in Complex Networks.
Kelley, S., Goldberg, M., Magdon-Ismail, M., Mertsalov, K., Wallace, A., 2012. Defining and discovering communities in social networks. In: Handbook of Optimization in Complex Networks. Springer, pp. 139–168.
Kernighan, B.W., Lin, S., 1970. An efficient heuristic procedure for partitioning graphs. Bell Syst. Tech. J. 49 (2), 291–307.
Kim, Y., Jeong, H., 2011. Map equation for link communities. Phys. Rev. E 84 (2), 026110.
Kim, J., Lee, J.-G., 2015. Community detection in multi-layer graphs: a survey. ACM SIGMOD Rec. 44 (3), 37–48.
Krause, A.E., Frank, K.A., Mason, D.M., Ulanowicz, R.E., Taylor, W.W., 2003. Compartments revealed in food-web structure. Nature 426 (6964), 282–285.
Krishnamurthy, B., Wang, J., 2000. On network-aware clustering of web clients. Comput. Commun. Rev. 30 (4), 97–110.
Kumpula, J.M., Kivelä, M., Kaski, K., Saramäki, J., 2008. Sequential algorithm for fast clique percolation. Phys. Rev. E 78 (2), 026109.
Lancichinetti, A., Fortunato, S., 2009. Community detection algorithms: a comparative analysis. Phys. Rev. E 80 (5), 056117.
Lancichinetti, A., Fortunato, S., Radicchi, F., 2008. Benchmark graphs for testing community detection algorithms. Phys. Rev. E 78 (4), 046110.
Lancichinetti, A., Fortunato, S., Kertész, J., 2009. Detecting the overlapping and hierarchical community structure in complex networks. N. J. Phys. 11 (3), 033015.
Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S., 2011. Finding statistically significant communities in networks. PLoS One 6 (4), e18961.
Le, C.M., Levina, E., 2015. Estimating the Number of Communities in Networks by Spectral Methods. arXiv preprint arXiv:1507.00827.
Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755), 788–791.
Lehmann, S., Schwartz, M., Hansen, L.K., 2008. Biclique communities. Phys. Rev. E 78 (1), 016108.
Lei, C., Ruan, J., 2013. A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity. Bioinformatics 29 (3), 355–364.
Lewis, T.G., 2011. Network Science: Theory and Applications. John Wiley & Sons.
Li, Z., Liu, J., 2016. A multi-agent genetic algorithm for community detection in complex networks. Phys. Stat. Mech. Appl. 449, 336–347.
Li, L., Du, M., Liu, G., Hu, X., Wu, G., 2014. Extremal optimization-based semi-supervised algorithm with conflict pairwise constraints for community detection. In: Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference. IEEE, pp. 180–187.
Li, L., Fan, K., Zhangy, Z., Xia, Z., 2016. Community detection algorithm based on local expansion k-means. Neural Netw. World 26 (6), 589.
Liang, Z., Chen, P., 2016. Delta-density based clustering with a divide-and-conquer strategy: 3dc clustering. Pattern Recogn. Lett. 73, 52–59.
Liben-Nowell, D., Kleinberg, J., 2007. The link-prediction problem for social networks. J. Assoc. Inf. Sci. Technol. 58 (7), 1019–1031.
Lin, L., Zheng-Min, X., Li, S.-H., Zhi-Huang, S.-N.L., 2014. Detecting overlapping community structure via an improved spread algorithm based on pca. In: International Conference on Computer Science and Software Engineering (CSSE). DEStech Publications, pp. 115–121.
Lipai, A., 2008. World wide web metasearch clustering algorithm. Inf. Econ. J. 12 (2) issn 1453–1305.
Liu, W., Chen, L., 2013. Community detection in disease-gene network based on principal component analysis. Tsinghua Sci. Technol. 18 (5), 454–461.
Liu, J., Liu, T., 2010. Detecting community structure in complex networks using simulated annealing with k-means algorithms. Phys. Stat. Mech. Appl. 389 (11), 2300–2309.
Liu, T., Rosenberg, C., Rowley, H.A., 2007. Clustering billions of images with large scale nearest neighbor search. In: Applications of Computer Vision, 2007. WACV'07. IEEE Workshop. IEEE. pp. 28–28.
Liu, C., Liu, J., Jiang, Z., 2014. A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. IEEE Trans. Cybernet. 44 (12), 2274–2287.
Liu, X., Wei, Y.-M., Wang, J., Wang, W.-J., He, D.-X., Song, Z.-J., 2016. Community detection enhancement using non-negative matrix factorization with graph regularization. Int. J. Mod. Phys. B 1650130.
Liu, J., Bier, E., Wilson, A., Guerra-Gomez, J.A., Honda, T., Sricharan, K., Gilpin, L., Davies, D., 2016. Graph analysis for detecting fraud, waste, and abuse in health-care data. AI Mag. 37 (2), 33–47.
Lloyd, S., 1982. Least squares quantization in pcm. IEEE Trans. Inf. Theor. 28 (2), 129–137.
Lu, Z., Sun, X., Wen, Y., Cao, G., La Porta, T., 2015. Algorithms and applications for community detection in weighted networks. IEEE Trans. Parallel Distr. Syst. 26 (11), 2916–2926.
Maity, S., Rath, S.K., 2014. Extended clique percolation method to detect overlapping community structure. In: Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference. IEEE, pp. 31–37.
Malliaros, F.D., Vazirgiannis, M., 2013. Clustering and community detection in directed networks: a survey. Phys. Rep. 533 (4), 95–142.
Mankad, S., Michailidis, G., 2013. Structural and functional discovery in dynamic networks with non-negative matrix factorization. Phys. Rev. E 88 (4), 042812.
Maqbool, O., Babri, H.A., 2004. The weighted combined algorithm: a linkage algorithm for software clustering. In: Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings. Eighth European Conference. IEEE, pp. 15–24.
Maruyama, K., Shima, K.-i., 1999. Automatic method refactoring using weighted dependence graphs. In: Software Engineering, 1999. Proceedings of the 1999 International Conference. IEEE, pp. 236–245.
Maslov, S., Sneppen, K., 2002. Specificity and stability in topology of protein networks. Science 296 (5569), 910–913.
McQueen, J., 1967. Some methods for classification and analysis of multivariate observations. Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1967, pp. 281–297.
Meila, M., Shi, J., 2001. A Random Walks View of Spectral Segmentation. Citeseer.
Meilă, M., 2007. Comparing clusterings—an information based distance. J. Multivariate Anal. 98 (5), 873–895.
Moëllic, P.-A., Haugeard, J.-E., Pitel, G., 2008. Image clustering based on a shared nearest neighbors approach for tagged collections. In: Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval. ACM, pp. 269–278.
Moghaddam, A., 2011. Detection of Malicious User Communities in Data Networks Ph.D. thesis. University of Victoria.
Morvan, A., Choromanski, K., Gouy-Pailler, C., Atif, J., 2017. Graph Sketching-based Massive Data Clustering. arXiv preprint arXiv:1703.02375.
Mukkamala, S., Janoski, G., Sung, A., 2002. Intrusion detection using neural networks and support vector machines. In: Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference, vol. 2. IEEE, pp. 1702–1707.
Nepusz, T., Petróczi, A., Négyessy, L., Bazsó, F., 2008. Fuzzy communities and the concept of bridgeness in complex networks. Phys. Rev. E 77 (1), 016107.
Newman, M.E., 2003. The structure and function of complex networks. SIAM Rev. 45 (2), 167–256.
Newman, M.E., 2004. Fast algorithm for detecting community structure in networks. Phys. Rev. E 69 (6), 066133.
Newman, M.E., 2004. Detecting community structure in networks. Eur. Phys. J. B Condens. Matter Complex Syst. 38 (2), 321–330.

Newman, M.E., 2006. Modularity and community structure in networks. Proc. Natl. Acad. Sci. U.S.A. 103 (23), 8577–8582.
Newman, M.E., Girvan, M., 2004. Finding and evaluating community structure in networks. Phys. Rev. E 69 (2), 026113.
Newman, M.E., Reinert, G., 2016. Estimating the number of communities in a network. Phys. Rev. Lett. 117 (7), 078301.
Ng, A.Y., Jordan, M.I., Weiss, Y., et al., 2002. On spectral clustering: analysis and an algorithm. Adv. Neural Inf. Process. Syst. 2, 849–856.
Nguyen, N.P., Dinh, T.N., Shen, Y., Thai, M.T., 2014. Dynamic social community detection and its applications. PLoS One 9 (4), e91431.
Nguyen, H.H., Imine, A., Rusinowitch, M., 2016. Detecting communities under differential privacy. In: Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society. ACM, pp. 83–93.
Nicosia, V., Mangioni, G., Carchiolo, V., Malgeri, M., 2009. Extending the definition of modularity to directed graphs with overlapping communities. J. Stat. Mech. Theor. Exp. 2009 (03), P03024.
Nieves, J.F., Jiao, Y.C., 2009. Data clustering for anomaly detection in network intrusion detection. Res. Alliance Math Sci. 1–12.
Noble, C.C., Cook, D.J., 2003. Graph-based anomaly detection. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 631–636.
Olesen, J.M., Bascompte, J., Dupont, Y.L., Jordano, P., 2007. The modularity of pollination networks. Proc. Natl. Acad. Sci. U.S.A. 104 (50), 19891–19896.
Orman, G.K., Labatut, V., Cherifi, H., 2012. Comparative evaluation of community detection algorithms: a topological approach. J. Stat. Mech. Theor. Exp. 2012 (08), P08001.
Oyelade, O., Oladipupo, O., Obagbuwa, I., 2010. Application of k Means Clustering Algorithm for Prediction of Students Academic Performance. arXiv preprint arXiv:1002.2425.
Ozer, M., Kim, N., Davulcu, H., 2016. Community detection in political twitter networks using nonnegative matrix factorization methods. In: Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference. IEEE, pp. 81–88.
Padmanabhan, K., Chen, Z., Lakshminarasimhan, S., Ramaswamy, S.S., Richardson, B.T., 2013. Graph-based anomaly detection. In: Practical Graph Mining with R, p. 311.
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.
Pan, W.-F., Jiang, B., Li, B., 2013. Refactoring software packages via community detection in complex software networks. Int. J. Autom. Comput. 10 (2), 157–166.
Papadopoulos, S., Zigkolis, C., Tolias, G., Kalantidis, Y., Mylonas, P., Kompatsiaris, Y., Vakali, A., 2010. Image clustering through community detection on hybrid image similarity graphs. In: Image Processing (ICIP), 2010 17th IEEE International Conference. IEEE, pp. 2353–2356.
Papadopoulos, S., Kompatsiaris, Y., Vakali, A., Spyridonos, P., 2012. Community detection in social media. Data Min. Knowl. Discov. 24 (3), 515–554.
Pikovsky, A., Rosenblum, M., Kurths, J., 2003. Synchronization: a Universal Concept in Nonlinear Sciences, vol. 12. Cambridge University Press.
Pinheiro, C.A.R., 2012. Community detection to identify fraud events in telecommunications networks. In: SAS SUGI Proceedings: Customer Intelligence.
Pizzuti, C., 2008. Ga-net: a genetic algorithm for community detection in social networks. In: PPSN. Springer, pp. 1081–1090.
Plantié, M., Crampes, M., 2013. Survey on social community detection. In: Social Media Retrieval. Springer, pp. 65–85.
Pluchino, A., Latora, V., Rapisarda, A., 2005. Changing opinions in a changing world: a new perspective in sociophysics. Int. J. Mod. Phys. C 16 (04), 515–531.
Pompili, F., Gillis, N., Absil, P.-A., Glineur, F., 2014. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. Neurocomputing 141, 15–25.
Pons, P., 2007. Détection de communautés dans les grands graphes de terrain Ph.D. thesis, Paris. p. 7.
Pons, P., Latapy, M., 2005. Computing communities in large networks using random walks. In: International Symposium on Computer and Information Sciences. Springer, pp. 284–293.
Porter, M.A., Onnela, J.-P., Mucha, P.J., 2009. Communities in networks. Not. AMS 56 (9), 1082–1097.
Portnoy, L., Eskin, E., Stolfo, S., 2001. Intrusion detection with unlabeled data using clustering. In: In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001. Citeseer.
Pothen, A., 1997. Graph partitioning algorithms with applications to scientific computing. In: Parallel Numerical Algorithms. Springer, pp. 323–368.
Pothen, A., Simon, H.D., Liou, K.-P., 1990. Partitioning sparse matrices with eigenvectors of graphs. SIAM J. Matrix Anal. Appl. 11 (3), 430–452.
Psorakis, I., Roberts, S., Ebden, M., Sheldon, B., 2011. Overlapping community detection using bayesian non-negative matrix factorization. Phys. Rev. E 83 (6), 066114.
Qadir, J., Ali, A., ur Rasool, R., Zwitter, A., Sathiaseelan, A., Crowcroft, J., 2016. Crisis analytics: big data-driven crisis response. J. Int. Humanit. Action 1 (1), 12.
Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D., 2004. Defining and identifying communities in networks. Proc. Natl. Acad. Sci. U.S.A. 101 (9), 2658–2663.
Raghavan, U.N., Albert, R., Kumara, S., 2007. Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E 76 (3), 036106.
Ralambondrainy, H., 1995. A conceptual version of the k-means algorithm. Pattern Recogn. Lett. 16 (11), 1147–1157.
Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. 66 (336), 846–850.
Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., Samatova, N.F., 2015. Anomaly detection in dynamic networks: a survey. Wiley Interdiscip. Rev.: Comput. Stat. 7 (3), 223–247.
Rattigan, M.J., Maier, M., Jensen, D., 2006. Using structure indices for efficient approximation of network properties. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 357–366.
Rattigan, M.J., Maier, M., Jensen, D., 2007. Graph clustering with network structure indices. In: Proceedings of the 24th International Conference on Machine Learning. ACM, pp. 783–790.
Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabási, A.-L., 2002. Hierarchical organization of modularity in metabolic networks. Science 297 (5586), 1551–1555.
Reddy, P.K., Kitsuregawa, M., Sreekanth, P., Rao, S.S., 2002. A graph based approach to extract a neighborhood customer community for collaborative filtering. In: International Workshop on Databases in Networked Information Systems. Springer, pp. 188–200.
Reichardt, J., Bornholdt, S., 2004. Detecting fuzzy community structures in complex networks with a potts model. Phys. Rev. Lett. 93 (21), 218701.
Reichardt, J., Bornholdt, S., 2006. Statistical mechanics of community detection. Phys. Rev. E 74 (1), 016110.
Reichardt, J., Bornholdt, S., 2007. Clustering of sparse data via network communities—a prototype study of a large online market. J. Stat. Mech. Theor. Exp. 2007 (06), P06016.
Reid, F., McDaid, A., Hurley, N., 2013. Partitioning breaks communities. In: Mining Social Networks and Security Informatics. Springer, pp. 79–105.
Ricci, F., Rokach, L., Shapira, B., 2011. Introduction to Recommender Systems Handbook. Springer.
Ríos, S.A., Videla-Cavieres, I.F., 2014. Generating groups of products using graph mining techniques. Proc. Comput. Sci. 35, 730–738.
Rives, A.W., Galitski, T., 2003. Modular organization of cellular networks. Proc. Natl. Acad. Sci. U.S.A. 100 (3), 1128–1133.
Rodriguez, A., Laio, A., 2014. Clustering by fast search and find of density peaks. Science 344 (6191), 1492–1496.
Rossi, R.A., Gallagher, B., Neville, J., Henderson, K., 2013. Modeling dynamic behavior in large evolving graphs. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining. ACM, pp. 667–676.
Rosvall, M., Bergstrom, C.T., 2007. An information-theoretic framework for resolving community structure in complex networks. Proc. Natl. Acad. Sci. U.S.A. 104 (18), 7327–7331.
Rosvall, M., Bergstrom, C.T., 2008. Maps of random walks on complex networks reveal community structure. Proc. Natl. Acad. Sci. U.S.A. 105 (4), 1118–1123.
Roux, M., 2015. A Comparative Study of Divisive Hierarchical Clustering Algorithms. arXiv preprint arXiv:1506.08977.
Sayyadi, H., Hurst, M., Maykov, A., 2009. Event detection and tracking in social streams. In: ICWSM.
Scharnhorst, A., 2003. Complex networks and the web: insights from nonlinear physics. J. Computer-Mediated Commun. 8 (4) 0–0.
Schwarz, A.J., Gozzi, A., Bifone, A., 2008. Community structure and modularity in networks of correlated brain activity. Magn. Reson. Imag. 26 (7), 914–920.
Semertzidis, T., Rafailidis, D., Strintzis, M.G., Daras, P., 2015. Large-scale spectral clustering based on pairwise constraints. Inf. Process. Manag. 51 (5), 616–624.
Sen, J., Goswami, K., 2010. An Algorithm for Detection of Selfish Nodes in Wireless Mesh Networks. arXiv preprint arXiv:1011.1793.
ŞERBAN, E.C., Bogeanu, A., Tudor, E., 2013. Clustering Techniques in Financial Data Analysis Applications on the US Financial Market.
Shen, H., Cheng, X., Cai, K., Hu, M.-B., 2009. Detect overlapping and hierarchical community structure in networks. Phys. Stat. Mech. Appl. 388 (8), 1706–1712.
Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U., 2002. Network motifs in the transcriptional regulation network of escherichia coli. Nat. Genet. 31 (1), 64–68.
Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22 (8), 888–905.
Singh, R.P., 2014. Application of graph theory in computer science and engineering. Int. J. Comput. Appl. 104 (1).
Soliman, S.S., El-Sayed, M.F., Hassan, Y.F., 2015. Semantic clustering of search engine results. Sci. World J. 2015.
Spielmat, D.A., Teng, S.-H., 1996. Spectral partitioning works: planar graphs and finite element meshes. In: Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium. IEEE, pp. 96–105.
Spirin, V., Mirny, L.A., 2003. Protein complexes and functional modules in molecular networks. Proc. Natl. Acad. Sci. U.S.A. 100 (21), 12123–12128.
Sporns, O., 2013. Structure and function of complex brain networks. Dialogues Clin. Neurosci. 15 (3), 247–262.
Stam, C.J., 2014. Modern network science of neurological disorders. Nat. Rev. Neurosci. 15 (10), 683–695.
Steinwart, I., Hush, D., Scovel, C., 2005. A classification framework for anomaly detection. J. Mach. Learn. Res. 6 (Feb), 211–232.
Tan, F., Xia, Y., Zhu, B., 2014. Link prediction in complex networks: a mutual information perspective. PLoS One 9 (9), e107056.
Tang, X., Xu, T., Feng, X., Yang, G., 2014. Uncovering community structures with initialized bayesian nonnegative matrix factorization. PLoS One 9 (9), e107884.
Tasgin, M., Herdagdelen, A., Bingol, H., 2007. Community Detection in Complex Networks Using Genetic Algorithms. arXiv preprint arXiv:0711.0491.
Tsantalis, N., Chatzigeorgiou, A., 2011. Identification of extract method refactoring opportunities for the decomposition of methods. J. Syst. Software 84 (10), 1757–1782.

Tyler, J.R., Wilkinson, D.M., Huberman, B.A., 2005. E-mail as spectroscopy: automated discovery of community structure within organizations. Inf. Soc. 21 (2), 143–153.
Valverde-Rebaza, J.C., de Andrade Lopes, A., 2012. Link prediction in complex networks based on cluster information. In: Advances in Artificial Intelligence-SBIA 2012. Springer, pp. 92–101.
Viana, M.P., Tanck, E., Beletti, M.E., da Fontoura Costa, L., 2009. Modularity and robustness of bone networks. Mol. Biosyst. 5 (3), 255–261.
Von Luxburg, U., 2007. A tutorial on spectral clustering. Stat. Comput. 17 (4), 395–416.
Wang, X.F., Chen, G., 2002. Synchronization in small-world dynamical networks. Int. J. Bifurc. Chaos 12 (01), 187–192.
Wang, X., Garibaldi, J.M., 2005. A comparison of fuzzy and non-fuzzy clustering techniques in cancer diagnosis. Proceedings of the 2nd International Conference in Computational Intelligence in Medicine and Healthcare, BIOPATTERN Conference, Costa da Caparica, Lisbon, Portugal, vol. 28.
Wang, X., Jiao, L., Wu, J., 2009. Adjusting from disjoint to overlapping community detection of complex networks. Phys. Stat. Mech. Appl. 388 (24), 5045–5056.
Wang, F., Li, T., Wang, X., Zhu, S., Ding, C., 2011. Community discovery using nonnegative matrix factorization. Data Min. Knowl. Discov. 22 (3), 493–521.
Wang, C., Tang, W., Sun, B., Fang, J., Wang, Y., 2015. Review on community detection algorithms in social networks. In: Progress in Informatics and Computing (PIC), 2015 IEEE International Conference. IEEE, pp. 551–555.
Wang, M., Wang, C., Yu, J.X., Zhang, J., 2015. Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework. Proc. VLDB Endow. 8 (10), 998–1009.
Wang, Q., Li, W., Zhang, X., Lu, S., 2016. Academic paper recommendation based on community detection in citation-collaboration networks. In: Asia-Pacific Web Conference. Springer, pp. 124–136.
Wang, Y., Fang, J., Wu, F., 2017. Application of community detection algorithm with link clustering in inhibition of social network worms. IJ Netw. Secur. 19 (3), 458–468.
Waskiewicz, T., 2012. Friend of a friend influence in terrorist social networks. In: Proceedings on the International Conference on Artificial Intelligence (ICAI), the Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p. 1.
Wasserman, S., Faust, K., 1994. Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press.
Wasserman, S., Pattison, P., 1996. Logit models and logistic regressions for social networks: I. an introduction to markov graphs andp. Psychometrika 61 (3), 401–425.
Wilkinson, D.M., Huberman, B.A., 2004. A method for finding communities of related genes. Proc. Natl. Acad. Sci. U.S.A. 101 (Suppl. 1), 5241–5248.
Williams, C.K., Seeger, M., 2001. Using the nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems, pp. 682–688.
Wu, F.-Y., 1982. The potts model. Rev. Mod. Phys. 54 (1), 235.
Wu, S., Tuo, M., Xiong, D., 2015. Community structure detection of shanghai stock market based on complex networks. In: LISS 2014. Springer, pp. 1661–1666.
Xie, J., Szymanski, B.K., Liu, X., 2011. Slpa: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In: Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference. IEEE, pp. 344–349.
Xie, J., Kelley, S., Szymanski, B.K., 2013. Overlapping community detection in networks: the state-of-the-art and comparative study. ACM Comput. Surv. 45 (4), 43.
Xin, L., Haihong, E., Song, J., Song, M., Tong, J., 2013. Book recommendation based on community detection. In: Joint International Conference on Pervasive Computing and the Networked World. Springer, pp. 364–373.
Xu, K.S., Hero, A.O., 2014. Dynamic stochastic blockmodels for time-evolving social networks. IEEE J. Sel. Top. Signal Process. 8 (4), 552–562.
Yang, J., Leskovec, J., 2013. Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining. ACM, pp. 587–596.
Yang, J., Leskovec, J., 2015. Defining and evaluating network communities based on ground-truth. Knowl. Inf. Syst. 42 (1), 181–213.
Yang, H., Xie, F., Lu, Y., 2006. Clustering and classification based anomaly detection. In: International Conference on Fuzzy Systems and Knowledge Discovery. Springer, pp. 1082–1091.
Yang, B., Liu, D., Liu, J., 2010. Discovering communities from social networks: methodologies and applications. In: Handbook of Social Network Technologies and Applications. Springer, pp. 331–346.
Yang, T., Chi, Y., Zhu, S., Gong, Y., Jin, R., 2011. Detecting communities and their evolutions in dynamic social networks—a bayesian approach. Mach. Learn. 82 (2), 157–189.
Yang, J., McAuley, J., Leskovec, J., 2013. Community detection in networks with node attributes. In: Data Mining (ICDM), 2013 IEEE 13th International Conference. IEEE, pp. 1151–1156.
Yang, Z., Algesheimer, R., Tessone, C.J., 2016. A comparative analysis of community detection algorithms on artificial networks. Sci. Rep. 6, 30750.
Yanrui, D., Zhen, Z., Wenchao, W., Yujie, C., 2015. Identifying the communities in the metabolic network using'component'definition and Girvan-Newman algorithm. In: Distributed Computing and Applications for Business Engineering and Science (DCABES), 2015 14th International Symposium. IEEE, pp. 42–45.
Yeung, K.Y., Ruzzo, W.L., 2001. An empirical study on principal component analysis for clustering gene expression data. Bioinformatics 17 (9), 763–774.
Yuan, P., Wang, W., Song, M., 2016. Detecting overlapping community structures with pca technology and member index. In: Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 121–125.
Zanin, M., Cano, P., Buldú, J.M., Celma, O., 2008. Complex networks in recommendation systems. In: Proc. 2nd WSEAS Int. Conf. on Computer Engineering and Applications, World Scientific Advanced Series in Electrical and Computer Engineering. World Scientific Advanced Series In Electrical And Computer Engineering, Citeseer, Acapulco, Mexico.
Zarei, M., Izadi, D., Samani, K.A., 2009. Detecting overlapping community structure of networks based on vertex–vertex correlations. J. Stat. Mech. Theor. Exp. 2009 (11), P11013.
Zeng, Y., Liu, J., 2015. Community detection from signed social networks using a multi-objective evolutionary algorithm. Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, vol. 1. Springer, pp. 259–270.
Zhang, Y., 2017. Cluster Analysis and Network Community Detection with Application to Neuroscience Ph.D. thesis. University of Pittsburgh.
Zhang, X., Cao, G., 2017. Transient community detection and its application to data forwarding in delay tolerant networks. IEEE ACM Trans. Netw. 25 (5), 2829–2843.
Zhang, Y., Yeung, D.-Y., 2012. Overlapping community detection via bounded nonnegative matrix tri-factorization. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 606–614.
Zhang, S., Wang, R.-S., Zhang, X.-S., 2007. Uncovering fuzzy community structure in complex networks. Phys. Rev. E 76 (4), 046103.
Zhang, L., Ye, Q., Shao, Y., Li, C., Gao, H., 2014. An efficient hierarchy algorithm for community detection in complex networks. Math. Probl Eng. 2014.
Zhao, H., Gao, Z., 2007. Modular effects on epidemic dynamics in small-world networks. EPL Europhys. Lett. 79 (3), 38002.
Zhao, K., Zhang, S.-W., Pan, Q., 2010. Fuzzy analysis for overlapping community structure of complex network. In: 2010 Chinese Control and Decision Conference. IEEE, pp. 3976–3981.
Zhou, H., 2003. Distance, dissimilarity index, and network community structure. Phys. Rev. E 67 (6), 061901.
Zhou, H., Lipowsky, R., 2004. Network brownian motion: a new method to measure vertex-vertex proximity and to identify communities and subcommunities. In: International Conference on Computational Science. Springer, pp. 1062–1069.

**Muhammad Aqib Javed** received his bachelor degree in Electrical Engineering from the University of Engineering and Technology Lahore, Pakistan in 2011. Currently, he is doing M.S. in Electrical Engineering from the National University of Sciences and Technology, Islamabad, Pakistan. His research interests include convex optimization and adaptive filtering.

**Siddique Latif** received his bachelor degree in Electronic Engineering from the International Islamic University, Islamabad, Pakistan in 2014, sponsored by National ICT Scholarship Program. Currently, he is doing M.S. in Electrical Engineering from the National University of Sciences and Technology, Islamabad, Pakistan. His research interests include mobile health, deep learning.

**Junaid Qadir** is an Associate Professor at the Information Technology University (ITU)-Punjab, Lahore, Pakistan. He is the Director of the IHSAN Lab at ITU that focuses on deploying ICT for development, and is engaged in systems and networking research. Prior to joining ITU, he was an Assistant Professor at the School of Electrical Engineering and Computer Sciences (SEECS), National University of Sciences and Technology (NUST), Pakistan. At SEECS, he directed the Cognet Lab at SEECS that focused on cognitive networking and the application of computational intelligence techniques in networking. He has been awarded the highest national teaching award in Pakistan—the higher education commission's (HEC) best university teacher award—for the year 2012–2013. His research interests include the application of algorithmic, machine learning, and optimization techniques in networks. In particular, he is interested in the broad areas of wireless networks, cognitive networking, software-defined networks, and cloud computing. He serves as an Associate Editor for IEEE Access, IEEE Communication Magazine, and Springer Nature Big Data Analytics. He is a member of ACM, and a senior member of IEEE.

**Shahzad Younis** received the bachelor's degree from National University of Sciences and Technology, Islamabad, Pakistan, in 2002, the master's degree from the University of Engineering and Technology, Taxila, Pakistan, in 2005, and the PhD degree from University Technology PETRONAS, Perak, Malaysia in 2009, respectively. Before joining National University of Sciences and Technical (NUST), he was Assistant Manager at a research and development organization named AERO where he worked on different signal processing and embedded system design applications. He is currently working as an Assistant professor in the Department of Electrical Engineering in School of Electrical Engineering and Computer Science (SEECS)-NUST. He has published more than 25 papers in domestic and international journals and conferences. His research interests include Statistical Signal Processing, Adaptive Filters, Convex Optimization Biomedical signal processing, wireless communication modeling and digital signal processing.

**Adeel Baig** (S'07–SM'14) received the B.E. from the NED University of Engineering and Technology, Karachi, Pakistan, and the master's and Ph.D degree in Computer Science and Engineering from the University of New South Wales (UNSW), Australia. He is an Assistant Professor at the College of Computer and Information Systems, Al Yamamah University, Saudi Arabia. He is also affiliated with the School of Electrical Engineering and Computer Science at the National University of Sciences & Technology (NUST) Pakistan. Before joining NUST, he held several positions in research and teaching, including with National ICT Australia (NICTA) and with UNSW. His research interests include cognitive and mobile networks, cross layer optimization for wireless networks, software defined networks and IPv6 deployments and transition issues. He has co-authored numerous research articles in well-known journals and conferences. Dr. Baig has been a reviewer, member of technical program committees of several journals and conferences. He is also a senior member of ACM, PEC and ISOC.