2016 International Conference on Internet of Things and Applications (IOTA)
Maharashtra Institute of Technology, Pune, India 22 Jan - 24 Jan, 2016

An Efficient Modularity based Algorithm for Community Detection in Social Network

Ranjan Kumar Behera
Department of Computer Science and Engineering
NIT, Rourkela
jranjanb.19@gmail.com

Santanu Ku. Rath
Department of Computer Science and Engineering
NIT, Rourkela
skrath@nitrkl.ac.in

Abstract— **Community detection process intends to detect clusters in a social network (SN), where nodes within the cluster are densely connected as compared to nodes outside the cluster.**

This process is one of the challenging issues in era of big data analytics particularly in the area of social networking. Graph data structure is often used to represent SN, where nodes can be used to represent actors and edges can be used to represent relationships among the actors. There are several algorithms for community detection purpose in a SN but each one has certain drawbacks in detecting community over a large scale network. In this paper an efficient modularity based community detection algorithm has been proposed. The proposed algorithm has been compared with other existing community detection algorithms using some of the most popular social network datasets. Performance of the algorithm has been assessed using various parameters like modularity, clustering coefficient, execution time etc.

Keywords- Community Detection; Clustering Coefficient; Modularity; Betweenness; Label Propagation; Fast Greedy

I. INTRODUCTION

Community detection process in large scale network is one of the most popular research topics in the study of social network analysis. Social network (SN) is often represented as a graph, consisting of set of nodes to represent actors and set of edges to represent relationship among the actors. It is more likely to have strong tie among the members in a group as compared to members in other groups. A good number of algorithms have been proposed by various authors in literature for community detection process in large scale social network analysis. Various community detection algorithms may be categorized into the following groups [1]:

- Node-Centric Community Detection Algorithm
- Group-Centric Community Detection Algorithm
- Network-Centric Community Detection Algorithm
- Hierarchy-Centric Community Detection Algorithm

In node-centric community detection algorithm every node in a community should satisfy some of the properties like mutuality, reachability, degree etc by finding maximal clique such as k-clique, k-club, k-clans [2] [3]. In group centric community detection algorithm, community is detected based on satisfiability of some of the criteria for the connection inside the group, where single node inside a community may have less tendency for connection, until and unless the group satisfy some predefined criteria [4]. λ-dense quasi clique is the best example of group based community where each community must have density greater than. The value of λ is considered as 1 if the community is a clique [5].

Network-Centric group detection algorithm aims to find the partition of community based on connection of whole network [6]. Partitioning the whole network should satisfy certain criteria in order to have communities. In this type of algorithm, similarity metric among all the nodes is measured and then any conventional clustering algorithm like centroid based clustering, density based clustering or hierarchical based clustering can be applied to find communities. Another type of community detection algorithm is based on hierarchal centric algorithm, where whole network is transformed into a hierarchal structure of community based on the topology of the network [4]. The important criteria used in hierarchical clustering for community detection are based on betweenness centrality, modularity, clustering coefficient etc.

The subsequent sections of the paper are organized as follows: In section 2, the related work in the field of community detection in large scale social network is discussed. In section 3, the methodology is being presented about modularity, clustering coefficient, which are the principle metrics in community detection. In section 4 the proposed approach is being highlighted along with the proposed algorithm. Section 5 indicates its implementation part. In section 6, a comparative study is presented by using the graphical representation. Section 7 concludes the paper and presents the scope for future work.

II. RELATED WORK

Community detection process is one of major challenging issues in area of social network analysis. The main problem in most of the classical community detection algorithms is that they fail to detect community in the bounded time complexity, when large scale social network comes into picture. Newman and Girvan have proposed a metric known as modularity to measure the strength of clustering in the network [7]. A number of algorithms have been proposed to reduce the time complexity by optimizing the modularity value. The structure of community needs to be quantified, based on the metric like normalized cut, conductance etc. Quality of network partition basically depends on modularity maximization. Some of the efficient optimization techniques like max-heap based agglomeration and iterative heuristic scheme were proposed by Newman [8]. Extreme optimization technique, simulated annealing and spectral optimization techniques are often considered to optimize the local subgraph modularity.

A greedy approach to community detection process is proposed by Clauset et al.[9]. In this algorithm each node is assigned to different communities and then merge the nodes according to their maximum modularity elevation value.

It considers two parameters ie, e_{ij} and a_i, for calculating maximum modularity value, where e_{ij} denotes the fraction of edges that connect to vertex i in one community to vertex j in other community and a_i denotes the fraction of end points of edges associated to the vertices in community i. The values of two parameters can be found as:

$$e_{ij} = \frac{1}{2m} \sum_{vw} A_{vw} \tag{1}$$

and

$$a_i = \frac{1}{2m} \sum_{vw} k_{vw} \tag{2}$$

The modularity defined by author Clauset et al.[8] is:

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) = \sum_i e_{ii} - a_i^2 \tag{3}$$

This algorithm has time complexity of $O(n^2 \log n)$ time, However complexity gets affected sometimes as it only aims at detecting larger community by sacrificing the smaller size community. Sometimes smaller community may contribute towards larger modularity value than expected.

A modification to this algorithm is carried out by Dannon et al. [10] where they considered equal priority to every communities by normalizing the change in modularity value by a factor k_i, where k_i is the degree of vertex i as written below:

$$\Delta Q_{ij}(new) = \frac{\Delta Q_{ij}}{k_i} \tag{4}$$

Merging of two very different sized communities may lead to higher computational complexity. To reduce the complexity further modification has been done by Wakitia et al. [11]. In this algorithm they have improved the efficiency by using a consolidation ratio $rcons(c_i, c_j)$ as defined below:

$$rcons(c_i, c_j) = \min \left\{ \frac{|c_i|}{|c_j|}, \frac{|c_j|}{|c_i|} \right\} \tag{5}$$

In this modification, the community pair (i,j) chosen for merging are based on maximum value of $\Delta Q * rcons(c_i, c_j)$, instead of maximum value of ΔQ.

There are also a number of community detection algorithms used to detect the overlapping communities. Clique Percolation method by Derenyi et al. is the most popular algorithm used to detect overlapping community [12]. Gergely et al.[7] have proposed an efficient algorithm for detecting overlapping community. An efficient community detection algorithm has also been proposed by Gergely et al. for weighted complex graph [13]. This method is efficient for detecting large scale network having very less computational complexity. Minimum cut method uses the concept of load balancing for community detection, which is some time inefficient for large scale network.

Finding a maximum clique in a large scale network is a NP complete problem. There are some other algorithms used to find the communities by using maximum clique problem. Graph mining techniques can be used for community detection problem [14]. In graph mining process, prior information about the graph is needed for knowledge extraction. An approximation algorithm is used to detect the community in social network by author Nguyen et al. [15]. In this paper, author used the concept of betweenness properties of edges to optimize the modularity value.

Community detection is one of the applications of graph mining. Different other applications in graph mining is well presented in the paper [16]. Centrality analysis, network modeling, link prediction, network classification, outlier's detection and node influence topics are some of the application areas of graph mining, each of these applications has a relation with community detection.

Clustering in social network can be possible using distance preserving subgraph. Nussbaum et al. have presented an efficient technique for clustering in social network using distance preserving subgraph [17]. A distance preserving subgraph is a subgraph that maintains the distances as in the original graph. Communities can be detected by minimizing the cut between the clusters and maximizing the modularity value which is well illustrated in paper [17].

Latent space model, spectral clustering, modularity maximization are some of the effective techniques for community detection [1]. In this paper authors have discussed on a detailed empirical comparison of community detection algorithm along with their time complexity values. Datasets on youtube and co-authorship network were used for algorithm comparison. Community detection in multi-dimensional network has been discussed in the paper [1].

Different graph sampling algorithms have been presented in paper [18]. Random walk and breadth first sampling are the well-known sampling algorithms which can be useful in detecting communities in social network. Degree distribution and clustering coefficient are discussed by Wang et al. in order to quantify the sample subgraph from large scale social network [18].

III. METHODOLOGY

The proposed methodology is based on few important topics on community detection process.

A. Modularity

Modularity is a metric used to measure the community structure in large scale network. A set of nodes in a network is said to be a community, if it has more number of links among the nodes in the set, as compared to random assignment of edges between them e.g. random graph. Modularity can be used to quantify the quality of community. The value of Modularity can be either positive or negative. Positive value of modularity implies the presence of community structure.

In a random graph having n nodes and m edges, the expected number of edges between any two nodes i and j having degree d_i and d_j respectively is $d_i d_j / 2m$. The actual number of edges between i and j, can be obtained by adjacency matrix A_{ij} . The modularity of a graph is given by the following equation [19]:

$$Q = \frac{1}{2m} \sum_{c} \sum_{i \in C, j \in C} A_{ij} - d_i d_j / 2m \tag{6}$$

B. Clustering Coefficient

Clustering Coefficient is another useful metric that defines the tendency of a node to make a cluster. It is associated with every node of the network. If clustering coefficient of a node is high, it means the node has high tendency to make a cluster. Strength of the community structure is affected by average clustering coefficient of the network. The local clustering coefficient of a node in a graph is given by the following equation [20]:

$$C_v = \begin{cases} \dfrac{2\left|e_{v,w}\right|}{k_u (k_u - 1)} & if \quad k_u > 0 \\ 0 & otherwise \end{cases} \tag{7}$$

The average clustering coefficient of the network is average of clustering coefficient of all the nodes in the network. It can be defined by the following equation:

$$C_{avg} = \frac{1}{n} \sum_{i=1}^{n} C_i \tag{8}$$

In this paper, extensive study has been carried out for several popular community detection algorithms like edge betweenness, fast greedy, walk trap, spin glass and label propagation to measure the effectiveness of the proposed algorithm. Girvan Newman algorithm is used to detect community based on betweenness centrality [21] of the network. The edge betweenness algorithm is used to detect the community by removing the cut edges in iterative manner. Cut edge is that edge in the graph which has highest edge-betweenness value.

Clauset et al. have used the concept of fast greedy community detection approach on [22]. It is quite fast to detect community in a large scale undirected network. In this algorithm merge process is carried out iteratively based on the modularity value. This algorithm tries to optimize the modularity value after each step of merging. As it uses greedy approach it may not always lead to optimal solution in large scale social network [23].

Walk-trap algorithm is based on performing random walk in the community to measure the similarities among nodes [24]. It is used to detect the densely connected subgraph in large scale complex network. The worst case time complexity of this algorithm is $O(en^2)$ and average case time complexity is $O(n^2 \log n)$, where e and n represent the number of edges and nodes respectively.

Spin glass community detection algorithm is another popular approach for finding community detection. This can be used to detect the community in both directed and undirected complex network. This algorithm uses the concept of spin glass model [25] and simulated annealing [26].

Label propagation algorithm is another approach used for community detection and analyzing the complex network. The advantage of this algorithm is that it is quite fast as it doesn't collect prior information about the network [27]. The limitation of this algorithm is that, it does not produce unique solution, but it produces aggregate of multiple solution. In this algorithm each node in the network is leveled and aggregation of nodes is carried out based on the label of neighboring nodes [28] [29].

C. Static and Dynamic Community Detection

There are quite a good number of algorithms used to find the community in the network. Most of the social networks are dynamic in nature, where number of nodes and edges are added repeatedly. Community detection in static network is easy as compared to dynamic network. The algorithm used for community detection in static network is divided into three categories as follows:

- Graph Partitioning method which aims at partitioning the graph with predefined group [19].
- Spectral Bisection method where Laplacian matrix is calculated and then spectral properties of the matrix is evaluated [9].
- Max Flow Min Cut method which is based on the idea that maximum flow is carried out through the minimum cut between two vertices.

The algorithm used for detecting dynamic social network are categorized as Evolutionary algorithm, Graph Score algorithm, Parallel algorithm, FaceNet algorithm, Event-based Framework [30].

In this study, Graph Partitioning approach has been considered as the proposed algorithm.

IV. PROPOSED ALGORITHM

In community detection process, a good amount of time is consumed by optimizing the modularity value. Higher is the modularity value, better is the community structure of the network. An algorithm has been proposed that will have relatively less time complexity as compared to other well-known community detection algorithms.

The proposed algorithm intends to reduce the time complexity for community detection to $O(n^2 \log_k n)$, where 'k' is in the range from 2 to λ, depending on the size of the social network. λ is a threshold value needed to be set before the execution of algorithm. For small network, k should have value 2 and for complex large scale network k should have higher value than 2.

Initial assumption in the algorithm is that the whole network is to be considered as a single community. The network is then divided into k communities by assigning each node of the network into one of the k communities. Each node is assigned with an integer value from the eigen vector corresponding to highest eigen value of modularity matrix (B), which is defined by the following equation:

$$B[i,j] = A[i,j] - \frac{k_i k_j}{2m} \tag{9}$$

Since it is not guaranteed that the value of eigen vector lies in between 0 to k-1, the value assigned to each node is to be normalized by k. After division, strength of the network partition based on modularity value is calculated. The modularity value is then maximized by calculating the eigen vector for highest eigen value of modularity matrix (B) as defined in equation 9. The objective is to find out the value of k so that it will maximize the modularity value. This process is repeated recursively, until there is no more increase in modularity value.

K-way Partitioning Algorithm for Community Detection (KWPA)

Input: The large scale social network G = (V, E)
Output: Network is to be partitioned with different communities,
Where V represent the set of nodes or vertices and E represent set of edges of the network
Step 1: The network is converted to adjacency matrix (A) where:

$$A_{ij} = \begin{cases} 1 & if\ (i,j) \in E \\ 0 & if\ (i.j) \notin E \end{cases} \tag{10}$$

Step 2: Modularity matrix (B) for the given network is calculated by the following equation:

$$B = A - \frac{k_i k_j}{2m} \tag{11}$$

Here 'k_i' and 'k_j' represent the degrees of 'i' and 'j' respectively and 'm' represents the total number of edges possible.
Step 3: Calculate the eigen vector(s) corresponding to maximal eigen value of modularity matrix B. The eigen vector(s) is then normalized by k using the following equation:

$$s[i] = s[i] \mod k \tag{12}$$

Step 4: The values in vector(s) is assigned randomly to each node of the network. Each node is then put into one of the k communities 'c_i', if the value of the node assigned is equal to i.
Step 5: The value of modularity of the network using the following equation is calculated:

$$Q = \frac{1}{2m} \sum_{c} \sum_{i \in C, j \in C} A_{ij} - d_i d_j / 2m \tag{13}$$

Step 6: For each community, step 1 to 6 is repeated until $\Delta Q > 0$
Step 7: Return the community structure of the network as found in step 6

V. IMPLEMENTATION

For testing our proposed algorithm, following four popular social network datasets are considered:

1. Zachary Karate Club: This datasets consist of members of a Karate club, collected from University Karate club by Wayne Zachary [31]. This datasets consist of friendship of 34 members, where some of them have higher influence factors than others, where average clustering coefficient is found to be 0.256.

2. Facebook: Facebook is the most popular social networking website that consists of millions of users. A part of this network is collected that consist of 4039 nodes [32]. This is the network that change dynamically quite rapidly in every millisecond, where average clustering coefficient is found to be 0.6055.

3. Twitter: Twitter is also one of the popular social networking websites consisting of millions of users. A part of Twitter dataset is collected from Stanford Large Network Dataset Collection [32]. This dataset consist of a large community having 81306 number of nodes where average clustering coefficient is found to be 0.5653.

4. Google Plus: Google Plus is a component of Google search engine. This is also a dynamically changing social network. The dataset is collected from Stanford Large Network Dataset Collection [32], this is a part of Google plus consist of 107614 nodes. Average clustering coefficient for this datasets is found to be 0.4901.

Details of the datasets is listed in TABLE I:

TABLE I. DATASETS USED FOR MEASURING PERFORMANCE

Datasets	No. of Nodes	No. of Edges	Clustering Coefficient
Zachary Karate Club[31]	34	78	0.256
Facebook[32]	4039	88234	0.6055
Twitter[32]	81306	1768149	0.5653
Google Plus[32]	107614	13673453	0.4901

The performance of the proposed algorithm i.e. KWPA is compared with the following existing community detection algorithms:

a. Edge-Betweeness Community Detection (EDBN)[33]
b. Fast-Greedy Community Detection (FG) [22]
c. Walk-trap Community Detection (WT) [34]
d. Spin-Glass Community Detection (SG) [35]
e. Label Propagation Community Detection (LP) [15]

The package 'igraph' in R language is used for community detection analysis. Different values of 'k' are considered for different datasets. The value of 'k' is chosen as

2,3,4,8 in the proposed algorithm for Zachary Karate Club, Facebook, Twitter and Google plus dataset respectively. Different values of 'k' are chosen based on size of datasets. Since Zachary Karate club is a small network, value of 'k' is chosen as 2 for this dataset and the value of 'k' is chosen as 8 for Google Plus dataset due to its complexity. The proposed algorithm has been executed in R and compared the performance with other conventional community detection algorithms. A detailed comparison study with graph indicating performance is presented in next section. The modularity value for each of the community partitions is also calculated for each of the datasets. Modularity is the metric used to find out the strength of the community partition.

VI. COMPARATIVE STUDY

The performance of proposed algorithm is compared with other existing algorithms in term of execution time by using the four different datasets of social network. The execution time (in mili second) for different datasets using different algorithms are listed in Table II.

Graphical representation of comparison is presented for different datasets. From the following graphs, it can be easily identified that the proposed algorithm runs faster than others, with its less amount of execution time.

Fig.1 shows the execution times obtained using different algorithms for Zachary Karate dataset. Fig.2 shows the execution times obtained using different algorithm for facebook dataset. Fig.3 shows the execution times obtained using different algorithm for Twitter dataset. Fig.4 shows the execution times obtained using different algorithm for Google-Plus dataset. X-axis represents different community detection algorithms and Y-axis represents the execution time in mili second.

TABLE II. EXECUTION TIME FOR DIFFERENT ALGORITHMS IN MSEC

	EDBT	FG	ET	SG	LP	KWPA
Zachary	0.22	0.23	0.23	0.98	0.25	0.16
Facebook	15.36	16.02	18.22	25.69	18.0	12.2
Twitter	62.53	54.22	46.54	69.85	50.07	39.85
Google Plus	98.56	87.25	69.90	120.3	79.8	63.23

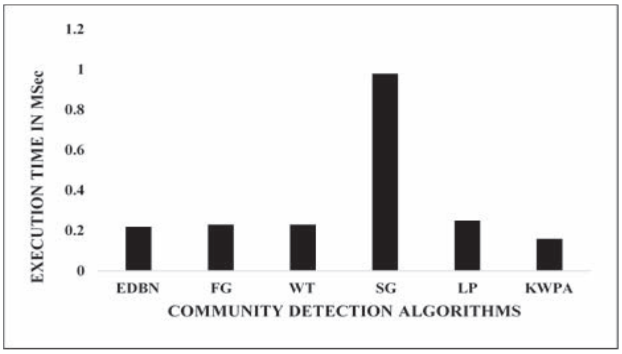

Fig. 1. Performance of algorithms using Zachary Karate dataset

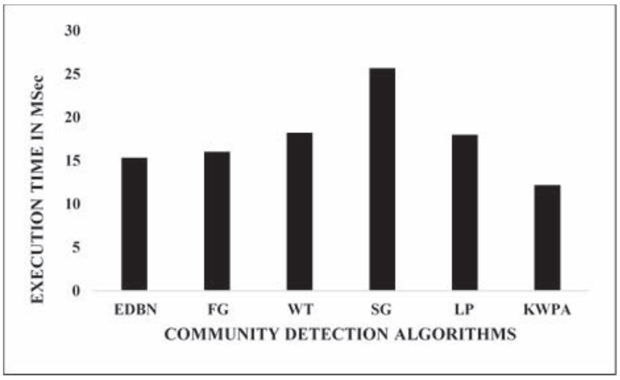

Fig. 2. Performance of algorithms using Fac ebook dataset

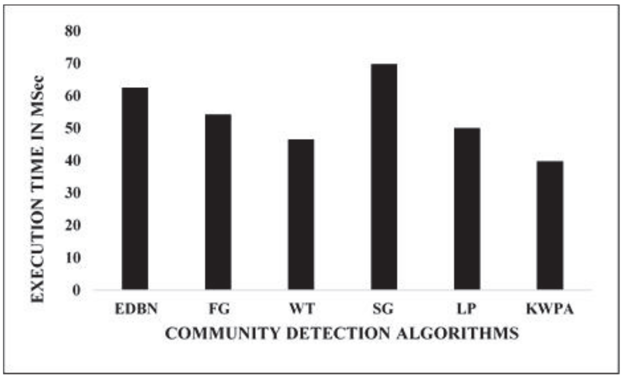

Fig. 3. Performance of algorithms using Twitter dataset

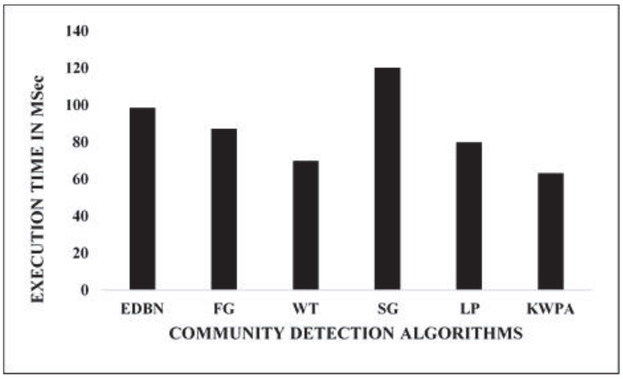

Fig. 4. Performance of algorithms using Google Plus dataset

The proposed algorithm performs better in term of execution time in all the four datasets. Performance of Spin glass community detection algorithm is worst among all the algorithms. All the comparisons clearly show that the proposed algorithm performs better than other existing community detection algorithms.

VII. CONCLUSION AND FUTURE WORK

An effort has been made in this paper to reduce the time complexity of community detection algorithm by using k-way partitioning method recursively. In this paper, the proposed algorithm has been compared with other existing algorithms. It is evident from comparative study that our algorithm performs better in most of the datasets, where number of nodes is quite large.

In this paper, four most popular datasets of social networks are taken for measuring the performance of proposed algorithm. Optimizing the modularity value is the key ingredient for community detection in large scale network. For this reason the proposed algorithm mainly focuses on optimizing the modularity value in recursive manner.

The proposed algorithm has less computational time complexity as compared to other algorithms. It has time complexity of $O(n^2 \log_k n)$ where other algorithms have time complexity of $O(n^2 \log_2 n)$. Some algorithms may have less time complexity of $O(n^2)$, but it is true only for small scale network.

As the social networking size is increasing day by day rapidly, millions of nodes along with their relationships are added in dynamic manner every day, detecting community in such a large scale network in a reasonable time is a challenging task. In future a distributed processing system like Spark or Hadoop system can be used to distribute the computation task in multiple nodes so that it would be faster to detect the community in an efficient manner. Application of big data analytics concept in social networking can be very helpful to detect the communities in large scale network.

REFERENCES

[1] Lei Tang, Xufei Wang, and Huan Liu. Community detection via heterogeneous interaction analysis. *Data Mining and Knowledge Discovery*, 25(1):1–33, 2012.

[2] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, 2012.

[3] Lu´ıs Cavique, Armando B Mendes, and Jorge MA Santos. An algorithm to discover the k-clique cover in networks. In *Progress in Artificial Intelligence*, pages 363–373. Springer, 2009.

[4] Lei Tang and Huan Liu. Graph mining applications to social network analysis. In *Managing and Mining Graph Data*, pages 487–513. Springer, 2010.

[5] Mauro Brunato, Holger H Hoos, and Roberto Battiti. On effectively finding maximal quasi-cliques in graphs. In *Learning and Intelligent Optimization*, pages 41–55. Springer, 2008.

[6] Lei Tang and Huan Liu. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–137, 2010.

[7] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[8] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[9] Mark EJ Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

[10] Vin´ıcius da Fonseca Vieira and Alexandre Gonc¸alves Evsukoff. A comparison of methods for community detection in large scale networks. In *Complex Networks*, pages 75–86. Springer, 2013.

[11] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks:[extended abstract]. In *Proceedings of the 16th international conference on World Wide Web*, pages 1275–1276. ACM, 2007.

[12] Imre Der´enyi, Gergely Palla, and Tam´as Vicsek. Clique percolation in random networks. *Physical review letters*, 94(16):160202, 2005.

[13] Gergely Palla, Imre Der´enyi, Ill´es Farkas, and Tam´as Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[14] Bapuji Rao and Anirban Mitra. A new approach for detection of common communities in a social network using graph mining techniques. In *High Performance Computing and Applications (ICHPCA), 2014 International Conference on*, pages 1–6. IEEE, 2014.

[15] Nam P Nguyen, Thang N Dinh, Ying Xuan, and My T Thai. Adaptive algorithms for detecting community structure in dynamic social networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 2282–2290. IEEE, 2011.

[16] Diane J Cook and Lawrence B Holder. *Mining graph data*. John Wiley & Sons, 2006.

[17] Ronald Nussbaum, Abdol-Hossein Esfahanian, and Pang-Ning Tan. *Clustering social networks using distance-preserving subgraphs*. Springer, 2013.

[18] Tianyi Wang, Yang Chen, Zengbin Zhang, Tianyin Xu, Long Jin, Pan Hui, Beixing Deng, and Xing Li. Understanding graph sampling algorithms for social network analysis. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pages 123–128. IEEE, 2011.

[19] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[20] Peng Zhang, Jinliang Wang, Xiaojia Li, Menghui Li, Zengru Di, and Ying Fan. Clustering coefficient and community structure of bipartite networks. *Physica A: Statistical Mechanics and its Applications*, 387(27):6869–6875, 2008.

[21] Mark EJ Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004.

[22] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[23] Guo-Jun Qi, Charu C Aggarwal, and Thomas Huang. Community detection with edge content in social media networks. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 534–545. IEEE, 2012.

[24] Thomas Aynaud and Jean-Loup Guillaume. Static community detection algorithms for evolving networks. In *Modeling and optimization in mobile, ad hoc and wireless networks (WiOpt), 2010 proceedings of the 8th international symposium on*, pages 513–519. IEEE, 2010.

[25] Eric Eaton and Rachael Mansbach. A spin-glass model for semi supervised community detection. In *AAAI*. Citeseer, 2012.

[26] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical review E*, 72(2):027104, 2005.

[27] NGUYEN Hoang Gia, LUONG Ngoc Lan, TRAN Thi Diem, Anne Dicky, Maylis Delest, and Franc¸ois Queyroi. Label propagation algorithm. 2013.

[28] Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010.

[29] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. *Knowledge and Data Engineering, IEEE Transactions on*, 20(1):55–67, 2008.

[30] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[31] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[32] Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[33] Usha Nandini Raghavan, R´eka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[34] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.

[35] Mason A Porter, Jukka-Pekka Onnela, and Peter J Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.