Towards effective discovery of natural communities in complex networks and implications in e-commerce

Chattopadhyay, Swarup; Basu, Tanmay; Das, Asit K.; Ghosh, Kuntal; Murthy, Late C. A.

DOI:
10.1007/s10660-019-09395-y

Electronic Commerce Research
https://doi.org/10.1007/s10660-019-09395-y

Towards effective discovery of natural communities in complex networks and implications in e-commerce

Swarup Chattopadhyay[1,3] · Tanmay Basu[2] · Asit K. Das[3] · Kuntal Ghosh[1] · Late C. A. Murthy[1]

Abstract
Automated community detection is an important problem in the study of complex networks. The idea of community detection is closely related to the concept of data clustering in pattern recognition. Data clustering refers to the task of grouping similar objects and segregating dissimilar objects. The community detection problem can be thought of as finding groups of densely interconnected nodes with few connections to nodes outside the group. A node similarity measure is proposed here that finds the similarity between two nodes by considering both neighbors and non-neighbors of these two nodes. Subsequently, a method is introduced for identifying communities in complex networks using this node similarity measure and the notion of data clustering. The significant characteristic of the proposed method is that it does not need any prior knowledge about the actual communities of a network. Extensive experiments on several real world and artificial networks with known ground-truth communities are reported. The proposed method is compared with various state of the art community detection algorithms by using several criteria, viz. normalized mutual information, f-measure etc. Moreover, it has been successfully applied in improving the effectiveness of a recommender system which is rapidly becoming a crucial tool in e-commerce applications. The empirical results suggest that the proposed technique has the potential to improve the performance of a recommender system and hence it may be useful for other e-commerce applications.

Keywords Complex networks · Clustering · Community discovery · Social network analysis · Recommender systems · Machine learning

✉ Swarup Chattopadhyay
swarupchatt@gmail.com

✉ Tanmay Basu
welcometanmay@gmail.com

1 Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India

2 Institute of Cancer and Genomic Sciences, University of Birmingham, Birmingham, UK

3 Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

Published online: 06 January 2020

1 Introduction

Understanding and various modeling aspects of large scale real world complex networks have been widely explored during the last decade [3, 12, 33, 34, 57]. The term complex network refers to any large, dynamic, random graph that corresponds to a complex system, where the nodes of the network represent the individuals and the edges symbolize the relations between them [35]. Examples of real world complex networks include World Wide Web (WWW), biological networks, communication networks, citation networks, social networks etc. Recently, social networks e.g., Twitter, Facebook have gained popularity through the involvement of large number of people and the exchange of information between them. In spite of the differences in the interpretation of vertices and edges, complex networks display appreciable topological similarities and therefore it is important to study those topological properties that ensure the similarities. Community structure is an important topological property of complex networks and in recent years, detecting communities is of great importance in sociology, biology and computer science, where systems are often represented as graphs [56]. A community is defined as a subset of vertices that are densely connected in a relatively sparse neighborhood. Modules, motifs, and communities are other terminologies that refer to such dense sub graphs. The issue of community discovery closely corresponds to the idea of data clustering in a system. Clustering algorithms partition a data set into several groups such that the data points in the same group are close to each other and the points across groups are far from each other [7]. The task of community discovery is to segregate a network into groups of vertices having high density of edges within groups, and low density of edges between groups [4]. A metric is required for such real world network clustering to quantify the existence of a node in a particular community, which is known as node similarity. In the earlier studies, researchers have proposed different models for community discovery by using existing distance functions e.g., Jaccard distance, Hub Promoted Index etc. to find similarity between nodes [2, 26, 43, 61]. These studies are mostly focused on finding connection between any two nodes based on their local information, but the local information may not represent the actual community structure in a network. An effective node similarity measure should determine the similarity between two nodes by considering their pairwise connectedness across the entire network. However, such an approach is lacking till date and it holds promise if one such algorithm can be developed.

Therefore a node similarity measure between a pair of nodes is proposed in this spirit and it is named as *nodality*. Intuitively, nodality determines the similarity between two nodes by finding their connections with every other node in the network. It assigns a non negative score to each pair of nodes to measure the degree of similarity. A negative nodality score denotes that a pair of nodes are not connected. Subsequently, a community detection technique is introduced using the idea of hierarchical data clustering and the proposed nodality measure. In principle, the proposed algorithm groups two nodes with high node similarity between them in the same community. Initially, the method treats every node as

a community and then merges two communities that have a minimum distance. Therefore, it finds next minimum distant communities and merges them and so on. The process continues until there exists no two communities with finite distance. The proposed algorithm never merges two communities with infinite distance. Thus, the proposed community detection algorithm determines the number of communities automatically. The distance of two communities is determined by nodality of the nodes between the communities. The distance between two communities is infinite when the nodality between every pair of nodes, taking one from each cluster is negative. The significant characteristic of the proposed community detection algorithm is that it does not require any prior knowledge about the number of desired communities. The main contributions in this article are, a new node similarity measure and an agglomerative hierarchical community detection technique that can explicitly identify two dissimilar communities. The performance of the proposed method is compared with several state of the arts and traditional community detection algorithms on various well known real world networks. The analysis of the results shows that the proposed algorithm successfully identifies the natural communities present in a network.

Furthermore, the proposed community detection algorithm is used for collaborative filtering based recommender system, a typical e-commerce application [8, 52]. A recommender system for an e-commerce site recommends products that are likely to be suitable to user needs. Collaborative filtering (CF) [8, 16] is an useful recommender system technology to date, and is used in many successful recommender systems on the web [52]. Most collaborative filtering based recommender systems build a neighborhood of like minded customers. Once a neighborhood of users is formed, these systems use several algorithms to produce recommendations. The aim is to integrate the proposed community detection method with the neighborhood based recommender systems. To this end we have used the adsorption algorithm [5], for recommending items using implicit user preferences. Through comprehensive experimental analysis on a DBLP co-author dataset, the approach of integrating the proposed community detection technique with the adsorption algorithm is shown to deliver good performance in recommending collaborators for an user.

The remainder of the paper is organized as follows. Section 2 describes some related works. The general idea of node similarity is described in Sect. 3. The proposed node similarity measure and the community detection technique are explained in Sect. 4. Section 5 presents the experimental results with a detailed analysis of the results and its application in the filed of e-commerce. Finally we conclude and discuss about the scopes of further works in Sect. 6.

2 Related works

Community discovery has been well studied in the past few years and many methods have been developed. The discovery of communities in a network provides an understanding about the structural topology of each community and its organization principles, e.g., a community in social networks usually contains users having similar characteristics that make them different from the others [29]. Identifying

communities in a network is nothing but partitioning a graph into set of disjoint sub graphs having similar properties within the graph. Let $G = (V, E)$ be a graph, where V is the set of vertices and E is the set of edges. Detecting non overlapping communities of the graph G is equivalent to partition G into k disjoint sub graphs $G_i = (V_i, E_i)$, in which $V_i \cap V_j = \emptyset \; \forall \; i \neq j$, and $V = \bigcup_{i=1}^{k} V_i$. The number of sub-graphs, k, may be denoted as a priori. The sub graphs V_i, V_j may overlap for overlapping communities, i.e. $V_i \cap V_j \neq \emptyset$. Simple undirected graph is considered throughout the article. Some methods for detecting both overlapping and non overlapping communities in a network are discussed in this section. A comprehensive reviews on both disjoint and overlapping community detection have been presented by Coscia et al. [14]. Xie et al. [55] contrasted the performance of 14 state-of-the-art algorithms for overlapping community detection on both synthetic graphs and on real-world social networks with no known ground-truth communities. Similarly, Leskovec et al. [24] evaluated the structural quality of the communities identified by various algorithms on real-world networks.

The majority of algorithms for community detection find disjoint communities; that is, each node belongs to at most one community. Several graph theoretic and probabilistic techniques are used to detect the communities over real world and artificial networks, such as, finding cliques, partitioning graph, maximizing the modularity, random walk, stochastic block models, etc. [4, 20, 39, 58, 60]. One of the graph partitioning methods, known as the Min-max cut method, makes a partition of a graph into two communities, say A and B, with the principle of minimizing the number of connections between A and B and maximizing the number of connections within each of A and B [17]. The algorithm searches for those two communities (or sub graphs), whose cut is minimized while maximizing the remaining edges. The top-down hierarchical approach has to be followed for finding k communities, by splitting the graph into two communities, and then further splitting these communities, and so on, until k communities have been detected. The major limitation of any graph partitioning method is that the method requires the number of partitions a priori, which may not be known in advance. Several metrics such as modularity have been proposed as a quality measure of network clustering [13, 24, 30]. The Louvain [9] method (LOUVN) is a widely used heuristic greedy algorithm for disjoint community detection by network modularity optimization. Clauset et al. proposed the CNMA (Clauset Newman Moore Algorithm) method based on a fast greedy algorithm proposed by Newman et al. [31], that builds an explicit hierarchical tree from small clusters to large ones. In order to achieve speedy performance, it maintains a data structure that tracks the change of modularity at each iteration, and iteratively generates the optimal level of the hierarchy structure [13]. Recenet research also integrates the node neighbourhood information's with the modularity structure to detect the communities present in a network [11]. The Scalable Community Detection Algorithm (SCDA) creates a set of disjoint partitions of the input graph. By combining different strategies, SCDA partitions the graph by maximizing the Weighted Community Clustering (WCC), a recently proposed community detection metric based on triangle analysis [40]. Another traditional method of spectral clustering by calculating the Leading Eigenvector (LEADE) of the modularity matrix was proposed by Newmann [32] to identify the disjoint communities in a network.

Adamcsek developed a method CFinder, which is an implementation of the clique percolation method [1]. It finds all the maximal cliques in a graph and then forms communities by merging cliques with common nodes. The Core Groups Graph Cluster (CGGC) method is an ensemble learning method, which combines the output of different clustering methods to determine the final partitions of the network [36]. Another heuristic algorithm is Walktrap (WLKTP) [39] that measures the similarity between vertices based on random walks in order to detect the communities in a network. The COmplex Network CLUster DEtection (CONCLUDE) algorithm aims to combine the accuracy of global methods with the efficiency of local methods using random walk [15]. Label Propagation Algorithm (LPA) has been proposed by Raghavan et al. [41] to detect both disjoint and overlapping communities by propagating labels representing community membership between nodes in a graph. Here every node is initialized with a unique label and at every step each node adopts the label that most of its neighbors currently have. Speed and Performance In Clustering (SPICi) [19] is a greedy heuristic algorithm that produces an incomplete clustering and is designed to work on large biological networks. The major limitation of these randomized algorithms is that they might get stuck at densely connected regions of a graph corresponding to a community. Top Graph Clusters (TopGC) [27] is a probabilistic clustering algorithm that finds the top well-connected clusters in a graph. Lancichinetti et al. proposed OSLOM (Order Statistics Local Optimization Method) [22] for detecting overlapping communities, which tests the statistical significance of a community with respect to a random graph model. Table 1 summarize the above mentioned recent community detection methods.

During the last decade, in the filed of e-commerce, many researchers [38, 45, 46, 53, 54, 59] have addressed the important problems such as recommendation of items to a user, opinions of the users on different items, buying behavior patterns of the users etc. through the methods of clustering the users or items into different meaningful groups. Many researchers [38, 46, 59] have applied the community detection methods to cluster the users based on similarity of their rating or co-purchasing a product and have further used the clusters to generate recommendations. Sarwar et al. [47] improve the performance issue of neighborhood based approaches [8, 16, 52] by accumulating the neighborhood formation process through the use of clustering. Here, Collaborative Filtering (CF) is a popular and widely used neighborhood-based approach for recommender systems regardless of the application domain [8, 16, 52] and adsorption [5] is one such neighborhood based algorithm which is used in applications such as recommending Youube videos, movies and sentiment analysis of text data. The Adsorption algorithm is a random walk based approach and works by propagating preference information through graphs. The intuition behind the algorithm is that a user's preference for items is likely to match the items commonly preferred by similar users. Recently, Parimi et al. [38] used the modularity based community detection method [9] to generate neighborhood for users and applied collaborative filtering [8, 52] on the neighborhood for recommending collaborators and books to users. They have integrated the identified communities with the neighborhood based recommender systems [16], specifically, the Adsorption algorithm [5], for recommending items using implicit user preferences. Similar to the approaches as discussed in [38, 47] is also adopted here to integrate the proposed

Table 1 Overview of the state of the art community detection methods

Algorithm	Short description	Param	Complexity[a]	Platform	Year	Source
CFinder	Implementation of the clique percolation method	–	Exponential	Java	2005	[1]
CGGC	Ensemble learning method	–	–	C++	2012	[36]
LPA	Random walk label propagation algorithm	–	$O(l)$	R	2007	[41]
LEADE	Spectral clustering with the leading eigenvector	–	$O(l(l+M))$	R	2006	[32]
LOUVN	Heuristic modularity optimization algorithm	–	$O(M\log(M))$	R	2008	[9]
WLKTP	Random walk based similarity between vertices	–	$O(lM^2)$	R	2005	[39]
CONCLUDE	Mixing local and global information	Yes	$(\hat{k}l + \underline{d}(V)^2M + \hat{\gamma}M)$	Java	2011	[15]
CNMA	Hierarchical modularity optimization algorithm	–	$O(M\log^2(M))$	R	2004	[31]
OSLOM	Order statistics local optimization method	–	$O(M^2)$	Java	2011	[22]
SCDA	Scalable community detection algorithm	Yes	$O(l\log(M))$	C++	2014	[40]
SPICi	Randomized greedy heuristic algorithm	Yes	$O(M\log(M)+l)$	C++	2010	[19]
TopGC	Probabilistic clustering algorithm	–	–	Java	2010	[27]

[a] $M = |V|$ number of vertices, $l = |E|$ number of edges, $\underline{d}(V)$ average degree of the graph, *Param* required input parameters

nodality based community detection technique with the Adsorption algorithm for recommending collaborators to a user in a DBLP co-author data set. The applicability of the proposed method in the filed of e-commerce has thus been explored.

3 Notion of node similarity

The similarity between two nodes in a network is a measure of closeness based on their behaviors across the whole network. Two nodes are considered to be similar, if they have many common features associated with them. Several node similarity measures have been developed based on local information or features to predict the missing links between nodes and to reveal the community structure in complex networks [26, 61]. Let $G = (V, E)$ be a undirected graph, where V is the set of vertices and E is the set of edges. Let us consider G as a simple graph, i.e., it does not contain multiple edges and self loops. Usually two nodes, $a, b \in V$, are more similar, if they have many common neighbors (CN). Therefore the simplest measure of similarity S_{ab} between two nodes a and b can be defined by simply counting the common neighbours as follows:

$$S_{ab}^{CN} = |\Gamma(a) \cap \Gamma(b)|,$$

where $\Gamma(i)$ is the neighbourhood of $i, i = a, b$. Several other similarity measures have been proposed based on the number of common neighbors, yet with different normalization methods, such as Jaccard Index [2], Hub Promoted Index [43], Hub Depressed Index [43] etc. Therefore, we can measure the similarity of each pair of nodes according to the above measures, but it can not guarantee the existence of a direct link between them. Hence these measures may some time affects the discovery of natural communities in a network. Additionally, some of the above metrics are unable to capture the indirect connectivity between pair of nodes, which may result inaccurate detection for community structure in the networks.

4 A similarity assessment technique for community detection of a network

The existing node similarity measure finds the relation between two nodes mostly by using the local information e.g., the common neighbors of two nodes. The local information may not be useful to identify the relation between two nodes. Instead the similarity between two nodes should be determined by checking all of their neighbors and non-neighbors in a network. Therefore, if two nodes are highly similar then they should have similar connectedness with most of the other nodes across the network i.e., in ideal condition, if two nodes x and y are connected and if x have a connection to any other node z then y must have a connection to z. This significant phenomenon is not observed in the existing node similarity measures.

4.1 A node similarity measure

A new similarity measure, *nodality* is proposed in this spirit to find the similarity between two nodes. The similarity measure exhaustively checks all the nodes in a network to determine the relation between two nodes. Nodality between two nodes is determined depending on their connection with every other node in the network. Intuitively, two nodes have highest similarity, if they are connected and they have almost the same connectedness with every other node in the network (i.e., both are either connected or disconnected to all the other nodes). The nodality is thus designed to find the grade of similarity or relation of a pair of nodes that are connected to each other. Let us define a score $D_{i,j}$ between node n_i and n_j as follows:

$$D_{i,j} = |\Gamma(i) \cup \Gamma(j)| - |\Gamma(i) \cap \Gamma(j)|, \tag{1}$$

where $\Gamma(i), \Gamma(j)$ respectively denote the set of neighbors of any two nodes say, n_i and n_j. Here neighbors of a node n_i indicate the nodes with which n_i is connected. Therefore the nodality between nodes n_i and n_j, $\forall i, j$ has been defined as

$$\text{nodality}(n_i, n_j) = \begin{cases} M - D_{i,j}, & \text{if } n_i \,\&\, n_j \text{ are connected} \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

here M denotes the total number of nodes in the entire network. Two nodes n_i, n_j have maximum nodality M, if n_i and n_j connected to each other and they are also connected to every other node in the network. However, this is an ideal case and hardly occur in real life networks. The minimum value of nodality becomes zero when $D_{i,j} = M$. Unlike other node similarity measures, nodality takes into account the connections of the said two nodes n_i, n_j with all the other nodes in the network when measuring the relation between them. $D_{i,j}$ indicates the number of nodes with which, if n_i is connected then n_j is not connected and the vice versa i.e., the togetherness of n_i and n_j is not observed in these cases. As the $D_{i,j}$ value increases, the relation between the nodes n_i and n_j decreases. If $D_{i,j} = 0$ then n_i and n_j are exactly similar. Actually $D_{i,j}$ denotes a grade of dissimilarity and it indicates that n_i and n_j have different connectedness with $D_{i,j}$ number of nodes. The nodality is used to define the distance between two communities in the first stage of the proposed community detection method. Nodality has some interesting properties which are as follows.

- It is symmetric. For every pair of nodes n_i and n_j, we have $\text{nodality}(n_i, n_j) = \text{nodality}(n_j, n_i)$.
- If $n_i = n_j$ i.e., if the nodes are same then $\text{nodality}(n_i, n_j) = 0$. However $\text{nodality}(n_i, n_j) = 0$ indicates that $D_{i,j} = M$ i.e., the nodes are connected to each other and both the nodes are connected to all the other nodes in the network, but still they are different nodes i.e., $n_i \neq n_j$ Hence nodality is not a metric.
- It should be noted that the only negative value of nodality is -1. This negative nodality value denotes the complete dissimilarity between two nodes. For the rest of the cases, nodality in general is positive and zero only when $D_{i,j} = M$.

4.2 Proposed method for community discovery

A distance function is defined to determine the distance between two communities of a network. It finds the distance between two communities say, CM_x and CM_y. Let W_{xy} be a multi-set consisting of the nodality values between each pair of nodes, one from CM_x and the other from CM_y and it is defined as follows:

$$W_{xy} = \{\text{nodality}(n_i, n_j) : \text{nodality}(n_i, n_j) \geq 0, \\ \forall n_i \in CM_x \text{ and } n_j \in CM_y\}$$

Note that W_{xy} consisting of all the occurrences of the same nodality values (if any) for different pairs of nodes. W_{xy} remains an empty set when the nodality value between each pair of nodes, one from CM_x and the other from CM_y is negative. The proposed distance between two communities CM_x and CM_y can be defined as

$$\text{comm\_dist}(CM_x, CM_y) = \begin{cases} \infty, & \text{if } W_{xy} = \emptyset \\ M - \max(W_{xy}), & \text{otherwise} \end{cases} \tag{3}$$

The function comm_dist finds the distance between two communities CM_x and CM_y as the maximum of the multi set of non-negative nodality values. The distance between two communities is infinite, if there exist no two nodes that have a non-negative nodality value i.e., no connected pair of nodes are present in CM_x and CM_y. The significant characteristic of the function comm_dist is that it would never merge two communities with infinite distance between them. The proposed technique segregates two different communities from each other by using the infinite distance property of this distance function. Note that the network size M is used in the definition of Eqs. 2 and 3 just to interpret nodality as the similarity measure and comm_dist as the distance measure.

The steps of the proposed community detection method is presented in Algorithm 1. Initially each node is treated as a community and the algorithm starts with M individual communities. A $M \times M$ distance matrix $Dis[i][j]$ is created in the first stage, whose (i, j)th entry denotes the distance between communities CM_i and CM_j. It is a square matrix and has M rows and M columns for M number of communities, where each row or column represents a community.

In each iteration, the algorithm merges two communities whose distance is minimum and therefore the nodality matrix is updated. This process is continued till there exist no two communities with non negative distance. In other words, the algorithm is terminated when distance between every pair of communities is infinite. Note that some of the communities may remain as singletons when the algorithm terminates. The merging procedure stated in step 15 of Algorithm 1 merges two rows say i and j and the corresponding columns of the distance matrix by following a convention regarding numbering. It merges two rows into one, the resultant row is numbered as minimum of i, j, and the other row is removed. Similar numbering follows for columns too. Then the index structure of the distance matrix is updated accordingly.

Algorithm 1 Community Detection using Nodality

Input: a) A set of M communities, $CM = \{CM_1, CM_2, ..., CM_M\}$ and $noc = |CM|$, number of communities.
b) $CM_i = \{n_i\}\ \forall i \in M$, where n_i is the ith node of the network.
c) A distance matrix $Dis[i][j] = \text{comm\_dist}(CM_i, CM_j),\ \ \forall i, j \in [1, M]$.

Steps of the Algorithm:

```
X ← 0,  Y ← 0
while noc > 1 and X ≥ 0 and Y ≥ 0 do
    min_dist ← M
    X ← −1,  Y ← −1
    for i = 1 to noc − 1 do
        for j = i + 1 to noc do
            if min_dist ≥ comm_dist(CM_i, CM_j) and
            comm_dist(CM_i, CM_j) ≥ 0 then
                min_dist ← comm_dist(CM_i, CM_j)
                X ← i,  Y ← j
            end if
        end for
    end for
    if X ≥ 0 and Y ≥ 0 then
        CM_X ← CM_X ∪ CM_Y
        Dis ← merge(Dis, i, j)
        noc ← noc − 1
    end if
end while
return CM
```

It should be noted that the algorithm never merges two communities, if they have infinite distance. Thus the negative distance property is used as the stopping criterion in the proposed algorithm. Consequently, the method can automatically identify the natural communities in the network and does not require a prior information of desired number of communities. Note that the nodality function not only determines the relation between nodes, but also describes the underlying structure of a network. Ideally, within a community the nodality values between each pair of nodes are very high and the distance between every pair of communities is infinite at the end of the algorithm.

4.3 Discussion

The idea of nodality has resemblance with the co-citation index of bibliometric studies [50]. Co-citation index is a semantic similarity measure for articles that makes use of citation relationship between articles. It is defined as the frequency with which two articles, say a_1 and a_2 are cited together by other articles. If at least one other article, say a_0 cites a_1 and a_2 in common then a_1 and a_2 are said to be co-cited. The main difference between nodality and co-citation index is that co-citation index

does not count the number of other nodes with which the said two nodes (a_1 and a_2) are not connected or cited. The nodality between two nodes counts the number of nodes with which both of these two nodes are either connected or disconnected. Thus nodality takes into account both connected and disconnected nodes for a pair of nodes, whereas co-citation index considers only the connected nodes for the same pair of nodes. Moreover, the nodality explicitly denote the absence of a link between two nodes by assigning a negative value, but the co-citation index has no such scope.

Moreover, the potential of nodality has been used to develop an effective algorithm for community discovery. Some interesting properties of the proposed community detection method are described in the following theorems. The quality of the resultant communities created by the proposed technique can be observed from these theorems.

Theorem 1 *Let CM_x and CM_y be any two resultant communities of the proposed method and M be the number of nodes in the given network then*

(a) $CM_x \cap CM_y = \emptyset$, *i.e., if* $n_i \in CM_x$ *then* $n_i \notin CM_y$
(b) $\exists\, n_i \in CM_x$ and $n_j \in CM_y$ *such that* n_i*and* n_j *are not connected for all* $i, j = 1, 2, \ldots, M$ and $i \neq j$.

Proof 1.a) It can be proved by the method of contradiction. Let us consider that $CM_x \cap CM_y \neq \emptyset$, i.e., $\exists\, n_i$ such that $n_i \in CM_x$ and $n_i \in CM_y$. Note that $nodality(n_i, n_i) = M$. Consequently CM_x, CM_y will be merged after some iterations, which is contradicting the assumption. Hence $CM_x \cap CM_y = \emptyset$.

1.b) This is also proved by the method of contradiction. Let us assume that the statement is not true. It means that there exists no $n_i \in CM_x$ and no $n_j \in CM_y$ such that they are not connected to each other, i.e., n_i and n_j are connected $\forall n_i \in CM_x$ and $\forall n_j \in CM_y$. Therefore nodality $(n_i, n_j) \geq 0, \forall n_i \in CM_x$ and $\forall n_j \in CM_y$. As a result comm_dist $(n_i, n_j) \neq \infty$, and CM_x, CM_y will be merged at some iteration, contradicting the assumption. Hence $\exists\, n_i \in CM_x$ and $n_j \in CM_y$ such that they are not connected. □

5 Experimental evaluation

The performance of the proposed algorithm is compared with the different community discovery algorithms using various well-known real-world[1] and artificial networks [21] having ground truths. The experimental analysis of the proposed algorithm and different competing techniques on these networks are discussed below. The performance metric of the proposed and the other methods were analyzed and

[1] http://snap.stanford.edu/data.

compared in R 3.0.2. Each of the competitive algorithms were run 20 times over each network.

5.1 Description of data sets and preprocessing

5.1.1 Real-world networks with ground truth communities

Six real world complex networks with disjoint ground-truth communities have been used in the experiments [18]. The networks are undirected and unweighted and they are selected from different application domains e.g., biological network, social network. The overview of these networks are presented in Table 2.

Protein-protein interaction network in budding yeast is a biologocal network. The network is viewed as a graph whose nodes correspond to proteins, where a connection between two proteins indicates an interaction between them. Bu et al. collected the data and identified different ground-truth communities in the network [10]. Amazon[2] is an online commercial network for purchasing products. Here nodes represent products and an edge exists between two products, if they are frequently purchased together. Each product (i.e. node) belongs to one or more product categories. Each ground-truth community is defined using hierarchically nested product categories that share a common function [57]. DBLP is a bibliographic network of Computer Science publications. Here nodes represent authors and an edge between two nodes represent co-authorship. Ground-truth communities are defined as sets of authors who published in the same journal or conference [57]. LiveJournal is a free on-line blogging community where users declare friendship to each other. It is a social network, where nodes represent users and edges represent friendship between two users. Ground-truth communities are defined by allowing users to form a group based on their common interest where other members can then join [57]. Orkut is a free online social network where users form friendship to each other. Orkut also allows users to form a group based on their common interest. One can join an existing group in Orkut. These groups are considered as ground-truth communities [57]. Youtube is a website to share videos and considered as a social network. Each user in the Youtube network is considered as a node and the friendship between two users is denoted as edge. Moreover, an user can create a group where other Youtube users can be a member through their friendship. These user created groups are considered as ground-truth communities [57].

The networks described above have several connected components and each connected component consisting of more than 3 nodes are considered as a separate ground-truth community. Leskovec et al. observed that the average goodness metric of the top k communities first remain flat with increasing k, but then after approximately 5000 communities, degrades rapidly [3]. Therefore they have implemented some community detection algorithms using different goodness metrics on the top 5000 communities of some of the networks described above. Eventually,

[2] www.amazon.com.

Table 2 Overview of the real world networks

Data sets	NV[a]	NE	MID	NC	MaxCS	MinCS	MaxDeg	AvgDeg	AvgCS	MCO	MCPN
AMAZON	1644	5022	0.3247	131	25	4	20	6.11	12.549	0	1
DBLP	5831	18,733	0.4889	676	27	6	29	6.43	8.626	0	1
LIVEJOURNAL	17,969	434,075	0.6667	668	407	3	399	48.31	26.898	0	1
ORKUT	1247	15,511	0.1645	43	211	3	204	24.877	29	0	1
YOUTUBE	3088	6695	0.2857	595	29	2	88	4.34	5.819	0	1
YEAST	2361	6646	0.00157	13	586	46	64	5.63	181.62	0	1

[a] *NV* number of vertices, *NE* number of edges, *MID* minimum internal density, *NC* number of communities, *MaxCS* maximum community size, *MinCS* minimum community size, *MaxDeg* maximum degree, *AvgDeg* average degree, *AvgCS* average community size, *MCO* maximum community overlap, *MCPN* maximum communities per node

Table 3 Training and test subsets based on 6 splits of the DBLP co-author dataset

Subset name	Training	Test
Subset1	$D_1 \cup D_2$	D_3
Subset2	$D_2 \cup D_3$	D_4
Subset3	$D_3 \cup D_4$	D_5
Subset4	$D_4 \cup D_5$	D_6

they obtained nice results in terms of finding communities in those networks. Following the same idea we have used only the top 5000 ground-truth communities of each of these networks in the experimental evaluation. Moreover, a community should be compact i.e., it should have high internal density rather than only having high value of goodness metric. Therefore, we have filtered the top 5000 communities of each network by removing the bottom quartile communities having lowest internal densities. Duplicate communities are also eliminated, if any. To get the networks with disjoint ground-truth communities, the maximum independent set of disjoint ground-truth are found.Therefore, the nodes and their incident edges that do not belong to any of these communities are removed. The resulting networks and ground-truth communities are used to evaluate the algorithms for community detection in the experiments.

Furthermore, we have considered a DBLP co-author dataset [44] to study the usefulness of the proposed community detection algorithm to recommend collaborators. The dataset has information about user to user collaborations between years 1940 and 2013 and consists of approximately 1.3M users and 18.9M collaboration records with four columns, specifically, the IDs of two individual users, weight of the connection, and the timestamp [44]. We have used a subset of this data set with collaboration between the years 2000 and 2013. This subset has approximately 1.1 million users and 17.1 million collaboration records. Given that timestamps are available for the DBLP data set, we have used the timestamps to generate training and test data sets. Specifically, we have divided the data into six splits (viz. $D_1, D_2, D_3, D_4, D_5, D_6$) according to years of collaboration. Using these six splits, we have generated four subsets of training and test data as shown in Table 3.

5.1.2 Artificial networks with ground truth communities

We use the Lancichinetti–Fortunato–Radicchi (LFR) networks [21] with ground-truths to study the behavior of a proposed community detection algorithm and to compare the performance across various competitive algorithms. The LFR model involve with a set of parameters which controls the network topology. In this model, degree distribution and community size distribution follow power laws with exponents γ and β, respectively. Moreover, we can also specify the other parameters such as number of vertices n, average degree k_{avg}, maximum degree k_{max}, minimum community size c_{min}, maximum community size c_{max}, and mixing parameter μ. We vary these parameters depending on our experimental needs. The critical parameter is the mixing parameter μ, which indicates the proportion

Table 4 Overview of the artifical networks

Name	NV[a]	NE	NC	k_{avg}	k_{max}	c_{min}	c_{max}	AvgCS
LFR2000	2000	3893	198	4	15	5	20	10.11
LFR4000	4000	7712	380	4	15	5	20	10.526
LFR6000	6000	12,391	360	4	20	10	40	16.666
LFR8000	8000	16,750	504	4	20	10	40	15.873
LFR10000	10,000	17,457	274	4	50	20	80	36.496
LFR12000	12,000	18,983	339	4	50	20	80	35.398

[a]*NV* number of vertices, *NE* number of edges, *NC* number of communities, k_{avg} average degree, k_{max} maximum degree, c_{min} minimum community size, c_{max} maximum community size, *AvgCS* average community size

of relationships a node shares with other communities. Six artificial networks are produced for experimental evaluation using the following parameter setting, $\gamma = -2, \beta = -1, \mu = 0.01$ as mentioned by Lancichinetti et al. [21]. Table 4 provides the details of the other parameters to produce these artificial networks. The results presented on these networks are the average of 50 runs.

5.2 Evaluation measures

In this section, some evaluation functions are described to measure the quality of a community discovery algorithm. The networks under consideration have labeled nodes i.e., whether an actual assignment of nodes into communities are known a priori (also known as ground truth communities). Therefore the evaluation functions based on the labeled networks are used in the experimental analysis. Normalized mutual information and f-measure are such evaluation functions and are used by a number of researchers [6, 51, 56] to measure the quality of different communities produced by an algorithm using the ground-truth communities of the network.

Let $\mathfrak{R}'(\mathfrak{R}'')$ be the partition of nodes represents the actual and predicted set of communities resulting into $|\mathfrak{R}'|$ number of actual communities and $|\mathfrak{R}''|$ number of predicted communities observed in a network.There are a total of M number of nodes in the network i.e., both $\mathfrak{R}'$ and $\mathfrak{R}''$ individually contains M nodes. Let n_k be the number of nodes belonging to actual community $\mathfrak{R}'_k$ of $\mathfrak{R}'$, m_l be the number of nodes belonging to predicted community $\mathfrak{R}''_l$ of $\mathfrak{R}''$ and n_{kl} be the number of nodes belonging to both actual community $\mathfrak{R}'_k$ and predicted community $\mathfrak{R}''_l$, for all $k = 1, 2, \ldots, |\mathfrak{R}'|$ and $l = 1, 2, \ldots, |\mathfrak{R}''|$.

Mutual information is a symmetric measure to quantify the statistical information shared between two distributions, which provides an indication of the shared information between two partitions [51]. Let $I(\mathfrak{R}', \mathfrak{R}'')$ denotes the mutual information between $\mathfrak{R}'$ and $\mathfrak{R}''$ and $E(\mathfrak{R}')$ and $E(\mathfrak{R}'')$ be the entropy of $\mathfrak{R}'$ and $\mathfrak{R}''$ respectively. $I(\mathfrak{R}', \mathfrak{R}'')$ and $E(\mathfrak{R}')$ can be defined as

$$I(\mathfrak{R}', \mathfrak{R}'') = \sum_{k=1}^{|\mathfrak{R}'|} \sum_{l=1}^{|\mathfrak{R}''|} \frac{n_{kl}}{M} \log\left(\frac{Mn_{kl}}{n_k m_l}\right),$$

$$E(\mathfrak{R}') = -\sum_{k=1}^{|\mathfrak{R}'|} \frac{n_k}{M} \log\left(\frac{n_k}{M}\right)$$

$E(\mathfrak{R}'')$ can be defined in the same way as $E(\mathfrak{R}')$. There is no upper bound for I(R,S), so for easier interpretation and comparisons a normalized mutual information that ranges from 0 to 1 is desirable [7]. The normalized mutual information (NMI) is defined by Strehl et. al. [51] as follows:

$$NMI(\mathfrak{R}', \mathfrak{R}'') = \frac{I(\mathfrak{R}', \mathfrak{R}'')}{\sqrt{E(\mathfrak{R}')E(\mathfrak{R}'')}}$$

F-measure determines the recall and precision value between each actual community $\mathfrak{R}'_k \in \mathfrak{R}'$ and each predicted community $\mathfrak{R}''_l \in \mathfrak{R}''$. Therefore, recall, precision and f-measure of an actual community $\mathfrak{R}'_k$ and predicted community $\mathfrak{R}''_l$ are given as follow.

$$Recall_{kl} = \frac{n_{kl}}{n_k}, \forall k, l, \quad Precision_{kl} = \frac{n_{kl}}{m_l}, \forall k, l$$

$$F_{kl} = \frac{2 \times Recall_{kl} \times Precision_{kl}}{Recall_{kl} + Precision_{kl}}, \forall k, l$$

If there is no common node between an actual and a predicted community (i.e., $n_{kl} = 0$) then we shall assume $F_{kl} = 0$. The value of F_{kl} will be maximum when $Precision_{kl} = Recall_{kl} = 1$. Thus the value of F_{kl} lies between 0 and 1. The best f-measure among all the predicted communities is selected as the f-measure for a particular ground-truth community i.e.,

$$F_k = \max_{l \in [0, |\mathfrak{R}''|]} F_{kl}, \ \forall k$$

The f-measure of all the ground-truth communities is the weighted average of the sum of their individual f-measures, $F = \sum_{k=1}^{|\mathfrak{R}'|} \frac{n_k}{M} F_k$. We would like to maximize both f-measure and normalized mutual information to achieve good quality communities.

In addition, an evaluation measure is required that finds the similarity between the sets of communities obtained by two different algorithms and it also depicts the closeness of each of these sets of communities to the set of ground-truth communities. Recently Malliaros et al. [28] presented a similar criterion function that finds pairwise similarity between the sets communities generated by two competing algorithms. Moreover, it shows the performance of each competing algorithm in terms of ground-truth communities. This similarity gives an alternative way of measuring the closeness of each predicted communities with the ground-truth communities. Let us consider that $C_P(v), \forall v \in V$ represents the community of a node v assigned by an

algorithm P. Therefore the similarity between the resultant communities of two different algorithms P and Q can be defined as follows.

$$Sim(P, Q) = \frac{1}{M} \sum_{v \in V} \frac{|C_P(v) \cap C_Q(v)|}{|C_P(v) \cup C_Q(v)|} \tag{4}$$

where $C_P(i), C_Q(i), \forall i = 1, 2, \ldots, M$ are the communities assigned by algorithm P and Q respectively and M is the number of nodes in the network. The value of this similarity measure lies between 0 and 1 and P and Q would achieve highest similarity when the value is 1. The pairwise comparison of each algorithm with every other algorithm as well as the performance of each method in terms of ground-truth communities results in a two dimensional matrix of similarity values. The similarity values in this matrix can be replaced by different colors and therefore the matrix can viewed as a two dimensional image. The image is useful for quick visual inspection of the performance of different algorithms, which is discussed in the next section (Figs. 1, 2).

5.3 Analysis of results

The performance of the proposed community discovery method on different real world and artificial networks described in the earlier section is compared with CFinder [1], LPA [41], LEADE [32], CGGC [36], LOUVN [9], WLKTP [39], CONCLUDE [15], CNMA [13], OSLOM [22], SCDA [40], SPICi [19], TopGC [27]. These algorithms have been executed with their default parameters mentioned in the individual references. The performance of the proposed method and the competing algorithms are measured using NMI, f-measure and Sim measure. Tables 5, 6, 7, and 8 show the NMI and f-measure values respectively for all the networks and for all the methods. The values marked in bold font in each row corresponding to the Tables 5–8 signify the best performance of a particular method. It can be observed from Table 5 that the proposed method is performing better than CFinder, CGGC, LEADE, LOUVN, CONCLUDE, CNMA, OSLOM, SCDA, SPICi and TopGC for all the real world networks. Note that CFinder could not be implemented on LiveJournal within the allotted time frame. Moreover, Table 5 shows that the proposed method performs better than LPA and WLKTP for all the data sets except LIVEJOURNAL. The NMI values of LPA and WLKTP have an edge over the NMI value of the proposed method for LIVEJOURNAL network. Therefore, it can be concluded from Table 5 that the proposed method performs better than the other methods in 70 cases and the other methods have an edge over the proposed one in the rest 2 cases. On the other hand in case of artificial networks the proposed method is also performing better than CFinder, CGGC, LEADE, LOUVN, CNMA, OSLOM, SCDA, SPICi and TopGC for almost all the networks as shown in Table 6. It has also been observed from Table 6 that the performances of CFinder, SPICi, SCDA and TopGC consistently decreases as number of vertices increases. In few cases, the performance of OSLOM, WLKTP, CGGC and CONCLUDE have an edge over the performance of the proposed method. It can be concluded from Table 6 that the

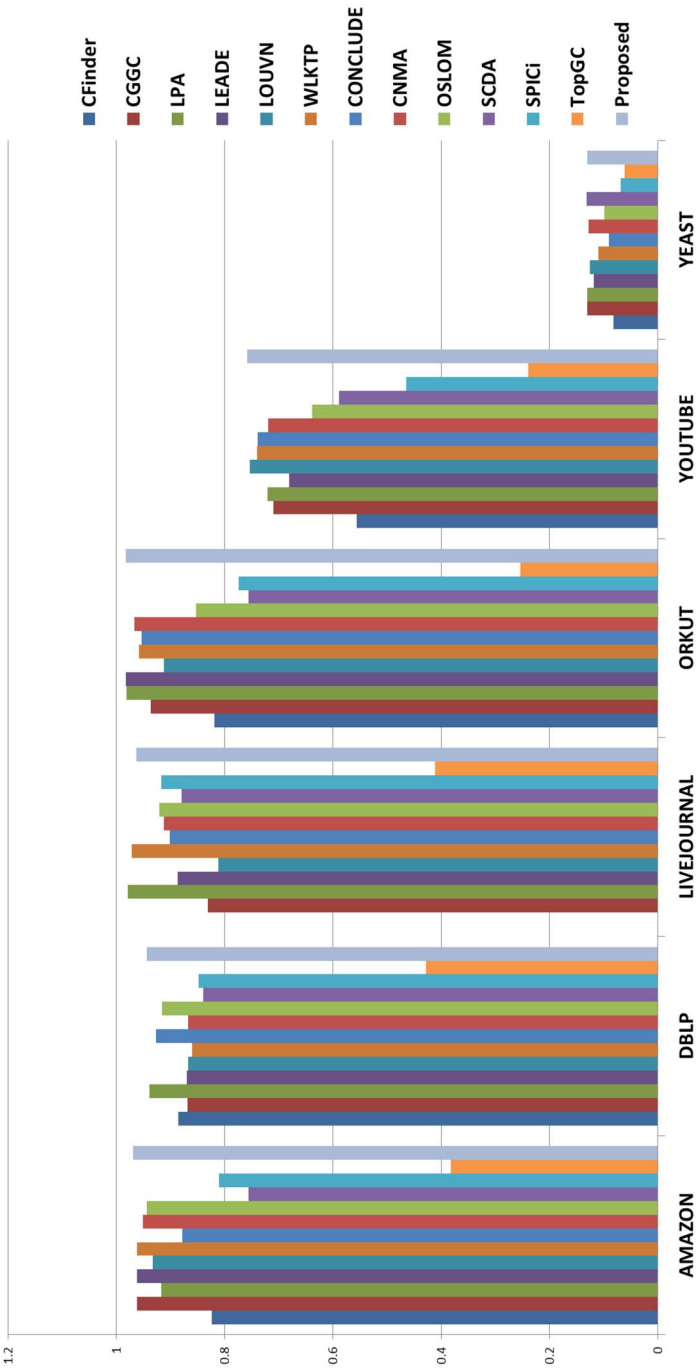

Fig. 1 NMI values of the different competing methods over real world networks have been plotted based on the results provided in Table 5. The plot demonstrates the efficacy of the proposed method. (Color figure online)

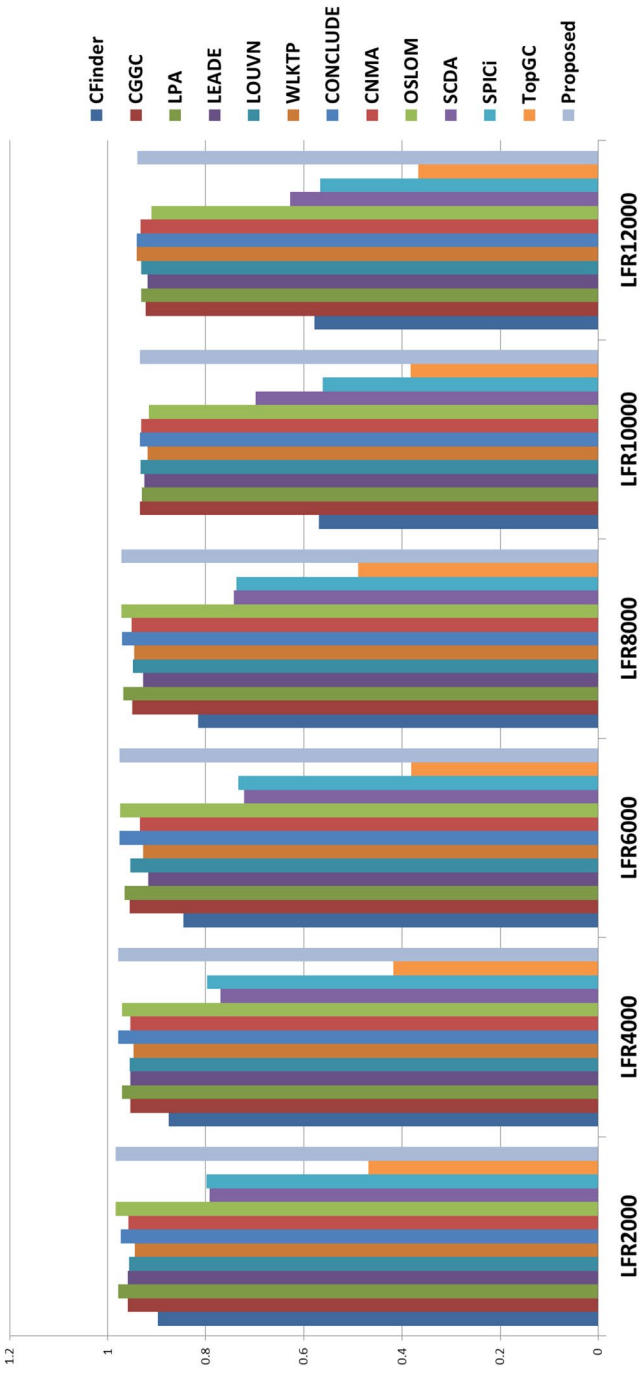

Fig. 2 NMI values of the different competing methods over artificial networks have been plotted based on the results provided in Table 6. The plot demonstrates the efficacy of the proposed method. (Color figure online)

Table 5 Comparison of various community detection methods using NMI on real world networks

Data sets	Normalized mutual information (NMI)												
	CFinder	CGGC[a]	LPA	LEADE	LOUVN	WLKTP	CONCLUDE	CNMA	OSLOM	SCDA	SPICi	TopGC	Proposed
AMAZON	0.824	0.961	0.917	0.961	0.933	0.962	0.878	0.951	0.943	0.756	0.810	0.382	**0.969**
DBLP	0.885	0.868	0.938	0.869	0.867	0.860	0.926	0.867	0.916	0.839	0.848	0.428	**0.943**
LIVE JOURNAL	–[b]	0.831	**0.978**	0.886	0.811	0.971	0.901	0.912	0.920	0.879	0.917	0.411	0.963
ORKUT	0.819	0.936	0.981	**0.982**	0.912	0.958	0.953	0.966	0.853	0.755	0.774	0.254	**0.982**
YOUTUBE	0.556	0.709	0.720	0.681	0.753	0.740	0.738	0.719	0.638	0.588	0.465	0.240	**0.758**
YEAST	0.082	0.130	0.131	0.118	0.126	0.110	0.090	0.128	0.099	0.132	0.069	0.061	**0.139**

[a] *CGGC* core groups graph cluster, *LPA* label propagation algorithm, *LEADE* leading eigenvector, *LOUVN* cluster louvain, *WLKTP* cluster walktrap, *CONCLUDE* complex network cluster detection, *CNMA* clauset newman moore algorithm, *SCDA* scalable community detection algorithm, *SPICi* speed and performance in clustering, *TopGC* top graph clusters

[b] CFinder could not complete its iterations within the allotted time frame for LIVE JOURNAL

Table 6 Comparison of various community detection methods using NMI on artificial (LFR) networks

Data sets	Normalized mutual information (NMI)												
	CFinder	CGGC	LPA	LEADE	LOUVN	WLKTP	CONCLUDE	CNMA	OSLOM	SCDA	SPICi	TopGC	Proposed
LFR2000	0.898	0.959	0.978	0.959	0.956	0.944	0.973	0.958	**0.984**	0.792	0.798	0.469	**0.984**
LFR4000	0.876	0.954	0.971	0.954	0.955	0.947	0.978	0.954	0.971	0.770	0.797	0.418	**0.979**
LFR6000	0.845	0.955	0.965	0.917	0.954	0.928	**0.976**	0.934	0.974	0.722	0.734	0.381	**0.976**
LFR8000	0.816	0.950	0.968	0.928	0.949	0.946	0.971	0.951	**0.972**	0.743	0.737	0.489	**0.972**
LFR10000	0.569	**0.934**	0.930	0.925	0.933	0.919	**0.934**	0.932	0.916	0.698	0.561	0.382	**0.934**
LFR12000	0.578	0.923	0.931	0.918	0.932	**0.941**	**0.941**	0.933	0.911	0.628	0.567	0.367	0.939

Table 7 Comparison of various community detection methods using F-measure on real world networks

Data sets	F-measure												
	CFinder	CGGC	LPA	LEADE	LOUVN	WLKTP	CONCLUDE	CNMA	OSLOM	SCDA	SPICi	TopGC	Proposed
AMAZON	0.877	0.956	0.948	0.954	0.925	0.961	0.898	0.946	0.946	0.871	0.885	0.360	**0.970**
DBLP	0.806	0.816	**0.928**	0.809	0.759	0.748	0.921	0.789	0.923	0.905	0.895	0.286	0.889
LIVE JOURNAL	–	0.830	0.931	0.715	0.820	**0.983**	0.918	0.829	0.918	0.929	0.922	0.412	0.899
ORKUT	0.790	0.948	0.96	0.963	0.964	0.974	0.908	0.964	0.912	0.911	0.869	0.189	**0.976**
YOUTUBE	0.338	0.424	0.507	0.358	0.469	0.486	0.418	0.502	0.492	0.526	0.417	0.135	**0.538**
YEAST	0.142	0.187	0.169	0.147	0.175	0.174	0.132	0.177	0.148	0.142	0.136	0.123	**0.188**

Table 8 Comparison of various community detection methods using F-measure on artificial (LFR) networks

Data sets	F-measure												
	CFinder	CGGC	LPA	LEADE	LOUVN	WLKTP	CONCLUDE	CNMA	OSLOM	SCDA	SPICi	TopGC	Proposed
LFR2000	0.798	0.817	0.917	0.811	0.799	0.764	0.913	0.817	0.916	0.721	0.765	0.301	**0.919**
LFR4000	0.818	0.775	0.929	0.778	0.776	0.755	0.928	0.776	**0.931**	0.711	0.773	0.189	**0.931**
LFR6000	0.792	0.774	**0.919**	0.702	0.761	0.886	0.899	0.763	**0.919**	0.671	0.660	0.138	0.898
LFR8000	0.809	0.741	**0.927**	0.715	0.736	0.764	0.901	0.744	0.924	0.668	0.672	0.367	0.887
LFR10000	0.508	0.786	0.794	0.768	0.787	0.753	0.811	0.786	**0.815**	0.444	0.437	0.121	**0.815**
LFR12000	0.513	0.766	0.792	0.731	0.723	0.786	**0.798**	0.765	0.796	0.424	0.438	0.114	**0.798**

proposed method performs better than the other methods in 70 cases and the other methods beat the proposed one in the rest 2 cases.

Similarly, Table 7 shows that the proposed algorithm beats CFinder, CGGC, LEADE, LOUVN, CNMA, TopGC in terms of f-measure for all the considered real world networks. The proposed method performs better than LPA, CONCLUDE, CNMA, OSLOM, SCDA, SPICi for all real world networks except DBLP and LIVEJOURNAL. The f-measure values of LPA, CONCLUDE, OSLOM, SCDA, SPICi have an edge over the f-measure values of the proposed method for DBLP and LIVEJOURNAL. The proposed method also beat WLKTP for all the data sets except LIVEJOURNAL. Thus, Table 7 shows that the proposed method beat the other methods in 61 cases and the other methods perform better than the proposed one in the rest 11 cases. It may be noted from Table 8 the proposed algorithm performs better than CFinder, CGGC, LEADE, LOUVN, CNMA, SCDA, SPICi, TopGC in terms of f-measure for artificial networks. The f-measure values of LPA, CONCLUDE, OSLOM for few artificial networks are greater than the f-measure values of the proposed method. The f-measure values of CFinder, SPICi, SCDA and TopGC decreases as the number of vertices increases in Table 8. Hence, it can be concluded that the proposed method performs better than the other methods in 66 cases and the other methods beat the proposed one in the rest 6 cases in Table 8.

A statistical significance test has been performed to check whether the differences of f-measure and NMI values between the proposed method and every other method through Tables 5, 6, 7, and 8 are statistically significant. A generalized version of paired *t-test* is suitable for testing the equality of means when the variances are unknown. This problem is the classical Behrens-Fisher problem in hypothesis testing and a suitable test statistic[3] is described and tabled in [23, 42], respectively. The level of significance is fixed as 0.05. It has been found in Table 5 that out of those 70 cases, where the proposed algorithm performed better than the other algorithms, the differences are significant in 67 cases. For the rest 2 cases the differences are significant. Hence the performance of the proposed method is found to be significantly better than the other algorithms in 95.71% (67/70) cases using NMI. The results where the proposed algorithm beat the other methods in Table 6 are statistically significant in 60 out of 70 cases and for the rest 2 cases the results are significant. The proposed technique performs significantly better than the other methods in 85.71% (60/70) cases in Table 6. Similarly, in Table 7 the results are significant in 59 out of 61 cases when proposed method performed better than the other methods and all the rest 11 cases are statistically significant. Thus in 84.28% (59/70) cases the proposed algorithm performs significantly better than the other methods Table 7. The results where the proposed algorithm beat the other methods in Table 8 are statistically significant in 58 out of 66 cases and for the rest 6 cases the results are significant in 5 cases. The proposed technique performs significantly better than the other methods

[3] The test statistic is of the form $t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{v_1^2/n_1 + v_2^2/n_2}}$, where $\bar{x}_1, \bar{x}_2$ are the means, v_1, v_2 are the standard deviations and n_1, n_2 are the number of observations.

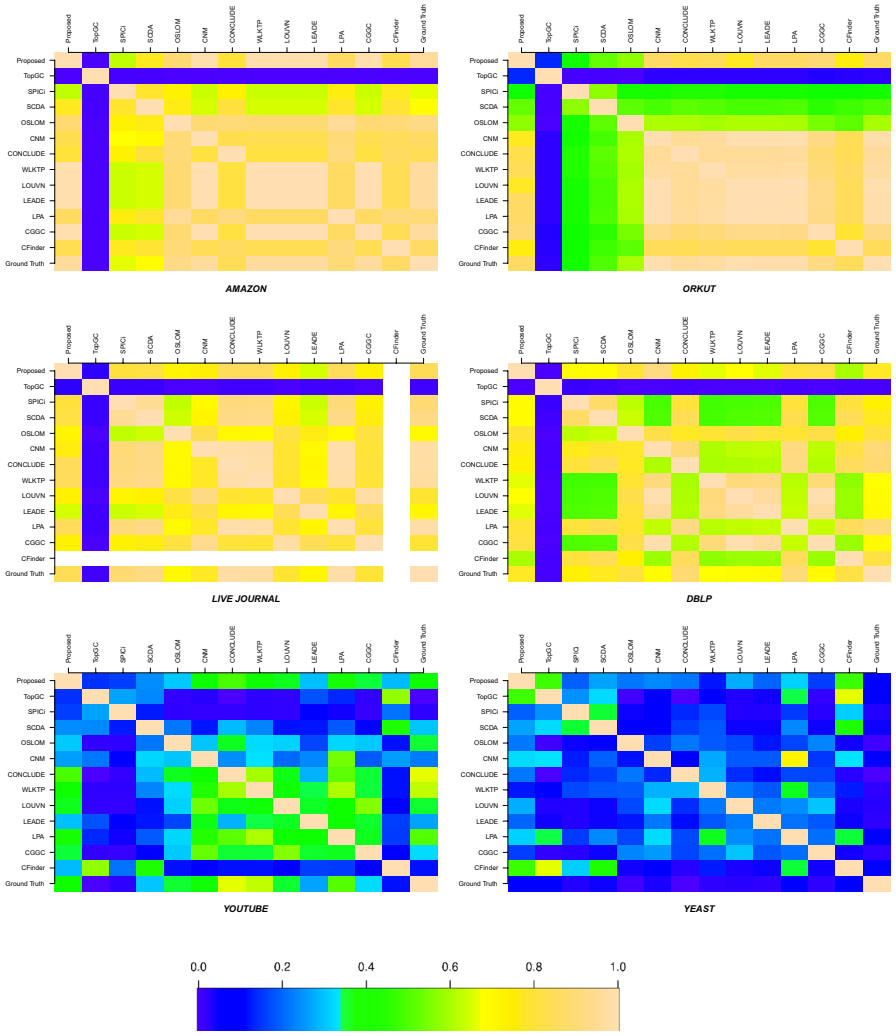

Fig. 3 Matrices of pairwise similarity scores for the community detection algorithms including proposed method and the ground-truths over each real world network. Each colored cell of each matrix provides the similarity values between the algorithms corresponding to that cell's row and column. White cell represents that the similarity could not be computed, because one of the algorithms could not produce result within allotted time frame. (Color figure online)

in 92.06% (58/63) cases in Table 8. These results clearly demonstrate the effectiveness of the proposed method for community discovery.

The performance of different methods are also evaluated by using Eq. 4. Therefore twelve different similarity matrices are obtained from six real world networks and six artificial networks used in the experiments. The similarity matrices corresponding to different networks are shown in Figs. 3 and 4 as a colored two dimensional image. The color scale shown at the bottom of Figs. 3 and 4 represents the

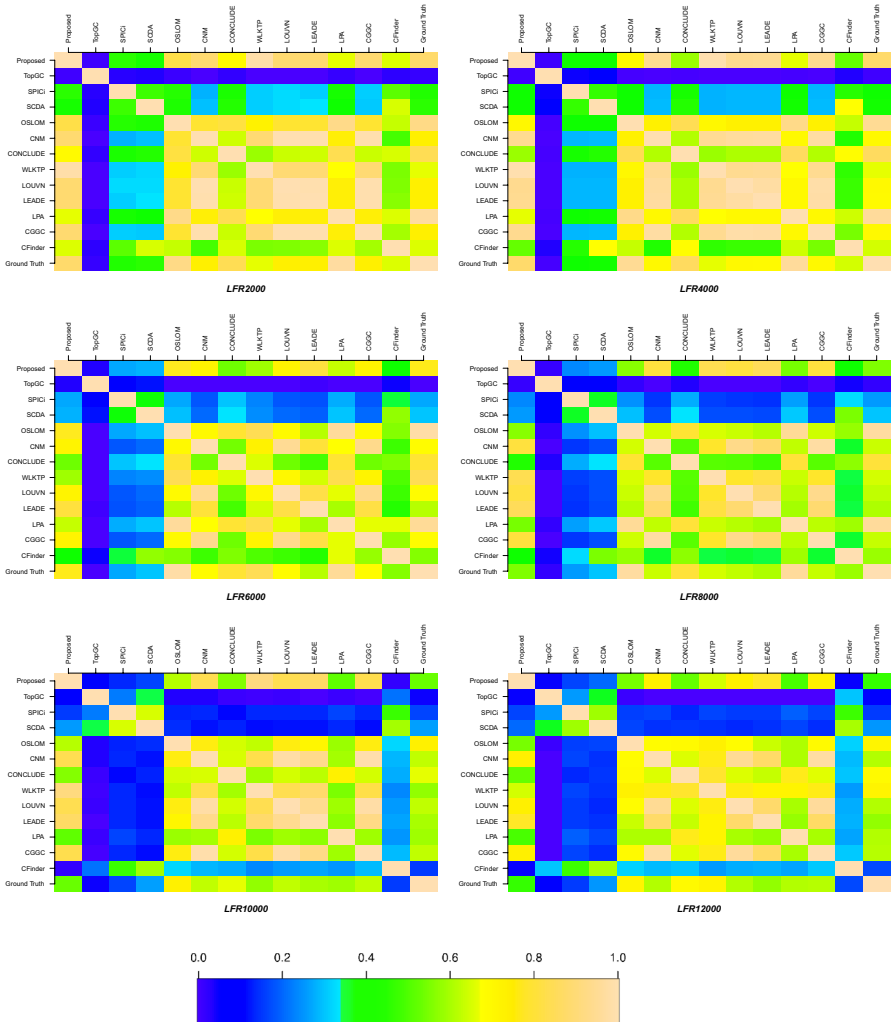

Fig. 4 Matrices of pairwise similarity scores for the community detection algorithms including proposed method and the ground-truths over each artificial network. Each colored cell of each matrix provides the similarity values between the algorithms corresponding to that cell's row and column

correspondence of similarity values with the colors used to generate the two dimensional image for each network. The similarity scale ranges from 0 to 1. Dark blue color corresponds to the similarity value 0 and off white color corresponds to the maximum similarity value 1. The similarity value ranging from 0 to 0.33 are depicted by shades of blue. The range of similarity values from 0.33 to 0.67 are represented by shades of green and the highest range of similarity values from 0.67 to 1 are shown by shades of yellow.

Consider the similarity matrix corresponding to the AMAZON network shown in top left side of Fig. 3. It can be observed that this image consists of mostly shades of

yellow, except for single row and column corresponding to TopGc. The image indicates that most of the algorithms including the proposed one produce high similarity values, i.e., most of the algorithms are agreed with each other and with the ground-truths. Thus the similarity matrix demonstrates the uniformity in the structure of the AMAZON network. It may be noted here that the color of the cell that demonstrate the similarity of the resultant communities of the proposed method with the ground-truth is off white, which indicates maximum similarity. Similar phenomenon can be observed from Fig. 3 for networks, viz., ORKUT, LIVE JOURNAL and DBLP. In case of the LIVE JOURNAL network, the row and column corresponding to the CFinder algorithm are missing, since the algorithm could not complete its iterations within the allotted time frame.

Each column of the similarity matrix represents the similarity values of the respective algorithm in comparison to other competitive algorithms including the proposed one. It can be observed from Fig. 3 that, some of the similarities obtained are relatively consistent across different networks (e.g., Amazon, DBLP, Live journal and Orkut). In those networks, the set of communities found by the proposed method are much closer to the ground-truth communities and it is also consistent with the set of communities found by other competitive algorithms viz. WLKTP, CONCLDE, LOUVN, LEADE and LPA. This phenomenon is also consistent with the NMI and f-measure values of these methods as shown in Tables 5 and 7 respectively. Note that the proposed method outperforms these algorithms in terms of NMI and f-measure. In most of the cases in Fig. 3, the performance of TopGC, SPICi and SCD are considerably different from the results of the other algorithms and also from the ground-truths. However, the images of Fig. 3 corresponding to the YOTUBE and YEAST networks consist mostly of shades of blue, which means most of the algorithms produce low similarity values and there is little agreement between each other and with the ground-truths. This inconsistent behavior of most these algorithms is a result of the irregular structures of these networks as mentioned by Paolillo [37]. It indicates that the similarity values of all the competitive algorithms in terms of ground-truths are very low for these networks, which is also consistent with their NMI and f-measure values as depicted in Tables 5 and 7.

Figure 4 represents the the similarity matrices of six generated artificial networks viz. LFR2000, LFR4000, LFR6000, LFR8000 and LFR10000 and LFR12000. Here also the set of communities found by the proposed method are much closer to the ground-truth communities and they are also consistent with the set of communities found by other competitive algorithms viz. CNMA, WLKTP, CONCLDE, LOUVN, LEADE and CGGC. Moreover, this observation is consistent with the NMI and f-measure values of these methods as shown in Tables 6 and 8 respectively. Note that the proposed method outperforms these algorithms in terms of NMI and f-measure values. The performance of TopGC is consistently worse across all the artificial networks. However the performances of CFinder, SPICi and SCDA degrades as the number of vertices increases, which can be easily seen from Fig. 4. The images of Fig. 4 corresponding to the LFR8000, LFR10000 and LFR12000 networks consist mostly of shades of blue corresponding to the columns of TopGC, CFinder, SPICi and SCDA respectively. This indicates that those columns contain low similarity values and there are a little agreements between each other and with

the ground-truths. This finding is consistent with their NMI and f-measure values as depicted in Tables 6 and 8.

5.4 Performance of the proposed method in e-commerce domain

Recommender systems are known to have been proposed to address the information overload problem resulting out of the rapid growth of the Internet. This is being done by filtering the relevant information and suggesting items of interest to users, for example, interesting movies to watch, books to read, people for collaboration etc. Thus the recommender systems can be considered as one of the most important applications in the field of e-commerce. Here our objective is to evaluate the performance of the proposed community detection technique in neighborhood based recommender systems, specifically, the Adsorption algorithm, a random walk based label propagation approach for recommending items using implicit user preferences [38, 47]. The reason to opt this collaborative filtering (CF) based recommender system is that it has three important characteristics to increase the accuracy of a recommender system, i.e., (a) normalization of data, (b) computation of similarity weights, and (c) selection of neighbors [8, 16, 52]. In general, the nearest neighbors of a user can be selected based on the similarity between individual users. Different similarity measures have been proposed to select neighbours without explicitly considering the underlying community structure existing between the users. Thus the identification of communities in a user to user graph may be important for constructing the n-nearest neighbors and consequently for generating recommendations. The aim is to explore the performance of the proposed community detection technique to construct the n-nearest neighbors of individual users. Subsequently the selected nearest neighbors will be used in Adsorption algorithm for recommending collaborators in a DBLP co-author dataset.

5.4.1 Method of finding nearest neighbors

Network communities correspond to a principled way of organizing vertices in a graph into densely connected clusters. Therefore, we claim that such densely connected clusters of users will result in better neighborhoods compared to the neighborhoods generated in a straightforward way for some applications. To justify the claim, we have compared the quality of the n-nearest neighbors selected by the baseline approach [38] with the n-nearest neighbors identified by the proposed community detection technique in recommending collaborators using mean average precision score.

Baseline approach The weight ($w_{u,v}$) between two users u and v is defined as follows:

$$W(u, v) = \frac{|items(u) \cap items(v)|}{|items(u)| + |items(v)|} \tag{5}$$

here items(u) and items(v) be the set of items preferred by u and v respectively and $W(u, v) \in [0,\ 0.5]$ [38]. We select n-nearest neighbors according to the weights.

Proposed technique The proposed technique generates the nearest neighbors in the following steps:

1. Compute the weight between every pair of users u and v in the data set according to the the above weight Eq. 5 and normalized it to [0, 1].
2. Construct a user-user graph after extracting every pair of users (u, v) for which $W(u, v)$ is above a predefined threshold $t \in [0, 1]$. The number of edges will be maximum when $t = 0$ i.e., there will be an edge between every pair of users for which $W(u, v) > 0$ and the number of edges decreases as t increases.
3. Perform the proposed community detection technique over the generated user-user graph to obtain different communities.
4. The n-nearest neighbors of an user u are selected as the top n neighbors of u say, $v_1, v_2, \ldots, v_n$ following the w_{u,v_i} scores, where u and v_i, $\forall i$ belong to the same community.

5.4.2 Experimental setup

The users belong to the test set of the DBLP co-author data that do not appear in the training set are needed to be removed because, if a user is not present in the training set, there is no way to make recommendations for that user in the test set. Secondly, remove the items corresponding to a user from the test set that have already been by that user in the training set. This is because of the fact that we want to recommend new unseen items to users.

We have designed our experiments to verify how the proposed method identifying communities before generating nearest neighbours on one side, and factors like the threshold (t) for constructing user to user graph and the number of nearest neighbours on the other, finally influence better recommendation of collaborators for an user. The parameters of the Adsorption algorithm are set to (0, 0.8, 0.2). We have used three values for number of nearest neighbours n : (5, 10, 15) and five values for the weight threshold parameter t : (0, 0.2, 0.4, 0.6, 0.8). The number of algorithm recommendations i.e., k is set to 10 as this is a standard value used in most of the works in the literature of recommender systems [16, 38, 52].

The output of the Adsorption algorithm is a list of *(item, preference_score)* tuples which correspond to the recommendations for each user in the data set. Note that preference score refers to the quality of preference of an item say i corresponding to an user say u and it is defined as [5]

$$preference\_score(u, i) = \sum_{v \in users(i)} W(u, v) \tag{6}$$

here $users(i)$ denotes the set of users who prefers item i in the training set. Subsequently, all the tuples that are referring to the items which belong to the training set

Table 9 MAP score of adsorption algorithm with baseline approach and proposed technique for the DBLP co-author dataset

No. of neighbours	Name. data set	Baseline approach	Proposed technique				
			t = 0.00	t = 0.20	t = 0.40	t = 0.60	t = 0.80
n = 5	Subset1	0.0171	0.0183	**0.0187**	0.0186	0.0165	0.0127
	Subset2	0.0169	0.0179	0.0180	**0.0181**	0.0161	0.0123
	Subset3	0.0174	0.0185	**0.0189**	0.0187	0.0166	0.0131
	Subset4	0.0177	0.0189	**0.0199**	0.0197	0.0169	0.0134
	Average	0.0173	0.0184	**0.0189**	0.0188	0.0165	0.0128
n = 10	Subset1	0.0161	0.0184	**0.0191**	0.0189	0.0171	0.0129
	Subset2	0.0160	0.0181	**0.0188**	**0.0188**	0.0172	0.0126
	Subset3	0.0166	0.0189	**0.0197**	0.0195	0.0175	0.0135
	Subset4	0.0169	0.0199	**0.0204**	0.0200	0.0181	0.0137
	Average	0.0164	0.0188	**0.0195**	0.0193	0.0175	0.0132
n = 15	Subset1	0.0149	0.0184	**0.0192**	0.0190	0.0173	0.0131
	Subset2	0.0146	0.0180	**0.0189**	0.0188	0.0176	0.0127
	Subset3	0.0151	0.0188	**0.0199**	0.0198	0.0181	0.0136
	Subset4	0.0153	0.0198	**0.0206**	0.0203	0.0186	0.0139
	Average	0.0149	0.0187	**0.0197**	0.0195	0.0179	0.0133

The threshold (t) on edge weight for the proposed technique varied from 0.0 to 0.8. The neighbourhood size (n) is varied from 5 to 15 and number of recommendations (k) is set to 10

of a user, are removed, as our aim is to recommend only new items to users. Finally, we create an ordered list of k tuples which are sorted from the highest to the lowest *preference_score*. From this list, we compute the average precision at top k ranks for each user, and eventually aggregate these to compute the mean average precision (MAP) at k [38, 49].

5.4.3 Experimental analysis

It can be seen from Table 9 that the performance of the Adsorption algorithm with neighbors constructed through the proposed community detection technique outperforms the performance of the algorithm with neighborhood constructed from the baseline approach. The values marked in bold font in each row of Table 9 signify that the proposed method performs the best as compared to the baseline corresponding to a particular threshold value t. The result is consistent across all the neighborhood sizes considered and reaches to the optimal corresponding to the threshold value of t lying between [0.2, 0.4]. The authors in the co-author network often collaborate with other authors who have similar research interests and are acquaintances. Thus, there exists a natural community structure between the authors in the co-author domain. The proposed method capture this underlying community structure between the authors and use it to select the top n neighbors for authors, thus achieving good MAP score. On the other hand baseline method fail to capture the

global community structure in selecting the top n neighbours and end up with low MAP value as clearly seen in the Table 9.

The threshold t controls the number of edges in a user to user graph G. The performance of the recommendation system improves with increasing values of t and reach a maximum for some value of t and then decreased when t is further increased. The higher values of t produces highly sparse user to user graph G and becomes very difficult to detect meaningful clusters associated with it which leads to small MAP value. Moreover, it can be observed from Table 9 that the performance of the baseline approach in recommending the collaborators becomes less personalized as increasing values of number of neighbours (n). On the other hand, MAP score of the adsorption algorithm using the proposed community detection method increases with the number of neighbours increases. For example, for $t = 0.2$, the MAP score for all the subset increases when n is varied from 5 to 15 as clearly depicted in Table 9. Thus we can conclude that the proposed community detection method is better suited for constructing neighborhood for the co-author domain and leads to the better recommendation for the collaborators of a user in a DBLP co-author data set.

It may be noted that such recommendation of authors in a particular area or topic may help the publisher to identify co-authored publications in the respective fields to promote potential publication of books and articles in future. In a similar way the proposed community detection method can also be applied over a user to item graph based data set e.g., the AMAZON product data set to generate user based and item based nearest neighbours as an important e-commerce application as discussed by Linden et al. [25] and Schafer et al. [48]. In principle the proposed method can be used to generate predictions for users based on ratings from similar users for such applications. Likewise, the method can also be successfully applied to generate predictions based on similarities between items by considering item based nearest neighbours in a user to item graph of an e-commerce data like AMAZON.

5.5 Analysis of computational time

The computation of nodality between two nodes takes $O(M)$ time in worst case for a network with M number of nodes. Let us consider there exist l number of links in the entire network. Therefore the time required to develop the nodality matrix is $l \times M$. The distance matrix of Algorithm 1 is created directly from the nodality matrix and it takes $l \times M$ time. The computational time of the hierarchical steps of the algorithm is $k \times l$, where k merges have been made at the termination of the proposed algorithm. Thus the computational time of Algorithm 1 is $((l \times M) + (k \times l))$. In the worst case, when $l \approx M^2$ and $k \approx M$, the time complexity of Algorithm 1 turns out to be $O(M^3)$. However, this occurs only when the network represents a complete graph, which is hardly possible for a real life complex network. Since most of the real world networks are sparse and hierarchical in nature, then $l << M^2$ and $k << M$. Hence in practice, the computational complexity of the proposed algorithm would be $O(M^2)$, which is quadratic to the number of nodes in a network. The distance matrix of Algorithm 1 requires $\frac{M \times (M-1)}{2}$ memory locations, and to store M

Table 10 Processing times of different community detection techniques on real world networks

Data sets	Community discovery techniques												
	CFinder	CGGC	LPA	LEADE	LOUVN	WLKTP	CONCLUDE	CNMA	OSLOM	SCDA	SPICi	TopGC	Proposed
AMAZON	3.850 s[a]	0.122 s	0.034 s	0.247 s	0.197 s	0.197 s	5.001 s	1.010 s	5.101 s	1.012 s	0.011 s	1.003 s	3.900 s
DBLP	8.321 s	0.180 s	0.567 s	0.596 s	0.925 s	0.371 s	56.01 s	3.012 s	19.00 s	1.420 s	0.033 s	1.001 s	9.120 s
LIVE JOURNAL	–[b]	11.00 s	2.120 s	6.227 s	3.867 s	6.080 s	3.700 h	13.16 s	13.86 m	4.150 s	1.800 s	6.030 s	1.002 m
ORKUT	5.230 s	1.020 s	0.079 s	0.193 s	0.208 s	0.107 s	35.06 s	1.560 s	38.00 s	1.011 s	0.020 s	2.013 s	8.081 s
YOUTUBE	4.210 s	2.160 s	0.137 s	0.578 s	0.489 s	0.206 s	29.30 s	1.421 s	35.00 s	0.732 s	0.110 s	1.010 s	7.180 s
YEAST	4.250 s	1.001 s	0.267 s	0.393 s	0.283 s	0.206 s	10.00 s	1.662 s	41.06 s	0.312 s	0.010 s	0.100 s	5.178 s

[a] h, m, and s denote processing times in hours, minutes and seconds respectively

[b] CFinder could not complete its iterations within the allotted time frame for LIVE JOURNAL

Table 11 Processing times (in seconds) of different community detection technique on artificial (LFR) networks

Data sets	Community discovery techniques												
	CFinder	CGGC	LPA	LEADE	LOUVN	WLKTP	CONCLUDE	CNMA	OSLOM	SCDA	SPICi	TopGC	Proposed
LFR2000	0.10	0.16	0.045	0.297	0.130	0.025	5.08	0.066	5.16	0.140	0.012	0.131	1.360
LFR4000	1.00	0.33	0.067	0.568	0.199	0.083	14.30	0.129	9.92	0.413	0.023	0.220	2.149
LFR6000	1.30	0.73	0.148	1.357	0.287	0.186	34.72	0.890	19.33	0.575	0.031	0.350	3.851
LFR8000	2.10	0.78	0.167	1.95	0.355	0.317	64.01	1.267	24.34	0.877	0.045	0.396	5.736
LFR10000	3.21	0.88	0.286	2.12	0.537	0.469	68.32	1.968	69.76	1.430	0.056	0.457	8.980
LFR12000	4.34	0.99	0.377	3.01	0.777	0.668	92.19	2.299	80.05	1.668	0.081	0.579	14.02

communities, initially, M memory locations are needed. Thus the space required by the proposed algorithm is $\frac{M\times(M-1)}{2}+M$ i.e., the space complexity of the method is $O(M^2)$.

The processing times of all the community detection algorithms on different real world and artificial networks are reported in Tables 10 and 11 respectively. Note that the tool used to implement the competing algorithms likely have an impact on the processing time. The proposed algorithm has been implemented in R. Some of the algorithms, viz. LPA, LEADE, LOUVN and WLKTP are available directly as in-built functions in R tool. The source code of the rest of the algorithms, are also available from their respective references as described in Sect. 5.2. Tables 10 and 11 shows that the computational times of the proposed method, LPA, LEADE, LOUVN, CGGC, SCDA, SPICi and WLKTP are lower than CFinder, CONCLUDE and OSLOM methods for each network. The computational time of SPICi is least among all the methods, but its performance in terms of community discovery is poor for each network. Note that CFinder could not be implemented on LiveJournal within the allotted time frame. The proposed method is computationally efficient than the CFinder, CONCLUDE and OSLOM methods. It is also computationally competitive compared to the other methods (viz. LPA, LOUVN, WLKTP, CNMA, SCDA and SPICi). The proposed method is updating the nodality matrix in each iteration and hence its computational time is little bit higher than the other methods as mentioned before for each network. Although the effectiveness of nodality has been observed through Tables 5, 6, 7, and 8. Note that the proposed algorithm has been implemented by the authors in R using simple data structures for a sparse matrix. Therefore the efficiency of the proposed technique can be improved by using efficient data structures.

6 Conclusions and future scope

A node similarity measure is introduced for effective community discovery in complex networks. The potential of the node similarity is used to design an agglomerative hierarchical technique to detect the natural communities in a network. The significant characteristics of the method is that it does not require the prior knowledge about number of communities. It confirms that a pair of connected nodes with higher similarity is more likely to be grouped into the same community. It also ensures the grouping of similar nodes into a community even though they are not directly connected. The quality of the clusters produced by the proposed method is justified through the experiments. The empirical study suggests the effectiveness of the proposed technique. It should be noted that the proposed method may not be useful for some networks with overlapping clusters, although the performance of the proposed algorithm is better than several other techniques (e.g., TopGc, OSLOM) that are typically developed to identify overlapped communities in a network. In future, the potential of nodality could be used to develop a method for detecting overlapped communities in complex networks. It should be noted that the proposed technique is computationally little expensive compared to some of the other methods. However, the algorithm significantly outperforms the other methods and the time taken by it is

reasonable for some data sets of large size (e.g, AMAZON, YOUTUBE, YEAST). Hence we may allow the computational burden of the proposed algorithm to obtain natural communities in a complex network without having any prior idea about the structural organization of the network. The future scope also includes the development of a computationally efficient version of the proposed technique.

Moreover, we have demonstarted that the proposed community detection method has the potential to be useful for improving the effectiveness of recommender systems that are rapidly becoming a crucial tool in e-commerce applications. It is empirically validated that the proposed algorithm is able to improve the performance of adsorption algorithm by generating suitable neighborhoods for individual user towards recommending collaborators in a DBLP co-author data set. Similarly, this algorithm may be effective in various commercial and specifically, e-commerce applications from recommending popular movies or frequent items corresponding to a user in a movie data set to a co-purchasing product data set e.g., AMAZON. The experimental results suggest that the proposed community detection method is effective in improving the recommendation accuracy and is hence suitable for e-commerce applications. In future, we would like to compare the performance of the method of generating recommendations using the proposed community detection technique with the other approaches such as popularity based recommendations, random walk, matrix factorization, etc. on various state of the art data sets. Likewise in future, a better prediction generation scheme along with the proposed community discovery algorithm can be used to improve the prediction quality of a recommender system.

Acknowledgements The authors would like to thank the editors and the reviewers for their valuable comments to improve the quality of the article. The first author gratefully acknowledge the financial assistance received from Indian Statistical Institute and Visvesvaraya Ph.D. Scheme awarded by the Government of India.

References

1. Adamcsek, B., Palla, G., Farkas, I. J., Derényi, I., & Vicsek, T. (2006). CFinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics*, *22*(8), 1021–1023.
2. Ahn, Y. Y., Bagrow, J. P., & Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature*, *466*, 761.
3. Albert, R., & Barabasi, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*(1), 47–97.

4. Amelio, A., & Pizzuti, C. (2014). Overlapping community discovery methods: A survey. In *Social networks: Analysis and case studies* (pp. 105–125).
5. Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., & Aly, M. (2008). Video suggestion and discovery for youtube: Taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web* (pp. 895–904). ACM.
6. Basu, T., & Murthy, C. A. (2013). Cues: A new hierarchical approach for document clustering. *Journal of Pattern Recognition Research*, *8*(1), 66–84.
7. Basu, T., & Murthy, C. A. (2015). A similarity assessment technique for effective grouping of documents. *Information Sciences*, *311*, 149–162.
8. Bell, R. M., & Koren, Y. (2007). Improved neighborhood-based collaborative filtering. In *KDD cup and workshop at the 13th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 7–14). Citeseer.
9. Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008.
10. Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Hongchao, L., et al. (2003). Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic Acids Research*, *31*(9), 2443–2450.
11. Chattopadhyay, S., Basu, T., Das, A. K., Ghosh, K., & Murthy, C. A. (2019). A similarity based generalized modularity measure towards effective community discovery in complex networks. *Physica A: Statistical Mechanics and its Applications*, *527*, 121338.
12. Chattopadhyay, S., Das, A. K., & Ghosh, K. (2019). Finding patterns in the degree distribution of real-world complex networks: Going beyond power law. *Pattern Analysis and Applications*. https://doi.org/10.1007/s10044-019-00820-4.
13. Clauset, A., Newman, M. E. J., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, *70*(6), 066111.
14. Coscia, M., Giannotti, F., & Pedreschi, D. (2012). A classification for community discovery methods in complex networks. In *CoRR*. arXiv:abs/1206.3552.
15. De Meo, P., Ferrara, E., Fiumara, G., & Provetti, A. (2014). Mixing local and global information for community detection in large networks. *Journal of Computer and System Sciences*, *80*(1), 72–87.
16. Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook* (pp. 107–144). Springer.
17. Ding, C., He, X., Zha, H., Gu, M., & Simon, H. (2001). A min–max cut algorithm for graph partitioning and data clustering. In: Proceeding of ICDM, New York, USA.
18. Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., et al. (2014). Community detection in large-scale networks: A survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, *6*(6), 426–439.
19. Jiang, P., & Singh, M. (2010). SPICi: A fast clustering algorithm for large biological networks. *Bioinformatics*, *26*(8), 1105–1111.
20. Lancichinetti, A., & Fortunato, S. (2009). Community detection algorithms: A comparative analysis. *Physical Review E*, *80*(5), 056117.
21. Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E*, *78*(4), 046110.
22. Lancichinetti, A., Radicchi, F., Ramasco, J. J., & Fortunato, S. (2011). Finding statistically significant communities in networks. *PLoS ONE*, *6*(4), e18961.
23. Lehmann, E. L. (1976). *Testing of statistical hypotheses*. New York: Wiley.
24. Leskovec, J., Lang, K. J., & Mahoney, M. (2010). Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web* (pp. 631–640). ACM.
25. Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *1*, 76–80.
26. Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, *390*(6), 1150–1170.
27. Macropol, K., & Singh, A. (2010). Scalable discovery of best clusters on large graphs. *VLDB*, *3*, 693–702.
28. Malliaros, F. D., & Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. *Physics Reports*, *533*(4), 95–142.

29. Nguyen, D. T., Thai, M. T., Nguyen, N. P., & Dinh, T. N. (2011). Overlapping community structures and their detection on social networks. In *Proceedings of international conference on social computing* (pp. 35–40), Boston: IEEE.
30. Newman, M. E. J. (2006). Modularity and community structure in networks. *PNAS*, *103*(23), 8578–8582.
31. Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, *69*(6), 066133.
32. Newman, M. E. J. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, *74*(3), 036104.
33. Newman, M. E. J. (2001). The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, *98*(2), 404–409.
34. Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256.
35. Newman, M. E. J., Strogatz, S. H., & Watts, D. J. (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, *64*(2), 026118.
36. Ovelgönne, M., & Geyer-Schulz, A. (2012). An ensemble learning strategy for graph clustering. *Graph Partitioning and Graph Clustering*, *588*, 187.
37. Paolillo, J. C. (2008). Structure and network in the YouTube core. In *Proceedings of international conference on system sciences* (p. 156).
38. Parimi, R., & Caragea, D. (2014). Community detection on large graph datasets for recommender systems. In *2014 IEEE international conference on data mining workshop* (pp. 589–596). IEEE.
39. Pons, P., & Latapy, M. (2005). Computing communities in large networks using random walks. In *Computer and information sciences-ISCIS 2005* (pp. 284–293). Springer.
40. Prat-Pérez, A., Dominguez-Sal, D., & Larriba-Pey, J.-L. (2014). High quality, scalable and parallel community detection for large real graphs. In *Proceedings of the 23rd international conference on World Wide Web* (pp. 225–236). ACM.
41. Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, *76*(3), 036–106.
42. Rao, C. R., Mitra, S. K., Matthai, A., & Ramamurthy, K. G. (Eds.). (1966). *Formulae and tables for statistical work*. Calcutta: Statistical Publishing Society.
43. Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., & Barabási, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, *297*, 1551.
44. Rossi, R. A., & Ahmed, N. K. (2015). The network data repository with interactive graph analytics and visualization. In *Proceedings of the twenty-ninth AAAI conference on artificial intelligence*.
45. Rout, J. K., Choo, K.-K. R., Dash, A. K., Bakshi, S., Jena, S. K., & Williams, K. L. (2018). A model for sentiment and emotion analysis of unstructured social media text. *Electronic Commerce Research*, *18*(1), 181–199.
46. Sahebi, S., & Cohen, W. W. (1997). Community-based recommendations: A solution to the cold start problem. In *Proceedings of WOODSTOCK'97*.
47. Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2002). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology* (Vol. 1, pp. 291–324).
48. Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291–324). Springer.
49. Schröder, G., Thiele, M., & Lehner, W. (2011). Setting goals and choosing metrics for recommender system evaluations. In *UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA* (Vol. 23, p. 53).
50. Small, H. (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, *24*(4), 265–269.
51. Strehl, A., & Ghosh, J. (2003). Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, *3*, 583–617.
52. Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*. https://doi.org/10.1155/2009/421425.
53. Roung-Shiunn, W., & Chou, P.-H. (2011). Customer segmentation of multiple category data in e-commerce using a soft-clustering approach. *Electronic Commerce Research and Applications*, *10*(3), 331–341.
54. Wang, D., Li, J., Kaiquan, X., & Yizhen, W. (2017). Sentiment community detection: Exploring sentiments and relationships in social networks. *Electronic Commerce Research*, *17*(1), 103–132.

55. Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, *45*, 43.
56. Yang, J., & Leskovec, J. (2012). Community-affiliation graph model for overlapping network community detection. In *2012 IEEE 12th international conference on data mining (ICDM)* (pp. 1170–1175). IEEE.
57. Yang, J., & Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, *42*(1), 181–213.
58. Yang, Z., Algesheimer, R., & Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, *6*, 30750.
59. Ying, J.-C., Shi, B.-N., Tseng, V. S., Tsai, H.-W., Cheng, K. H., & Lin, S.-C. (2013). Preference-aware community detection for item recommendation. In *2013 conference on technologies and applications of artificial intelligence* (pp. 49–54). IEEE.
60. Zhongying, Z., Shaoqiang, Z., Li, C., Jinqing, S., Liang, C., & Francisco, C. (2018). A comparative study on community detection methods in complex networks. *Journal of Intelligent & Fuzzy Systems*, pages 1–10.
61. Zhou, T., Lü, L., & Zhang, Y. C. (2009). Predicting missing links via local information. *European Physical Journal B*, *71*, 623–630.