Dombi and Dhama *Appl Netw Sci* (2021) 6:85
https://doi.org/10.1007/s41109-021-00424-0

Applied Network Science

RESEARCH **Open Access**

Theory of preference modelling for communities in scale-free networks

József Dombi[1,2†] and Sakshi Dhama[1*]

*Correspondence: sakshi@inf.u-szeged.hu
†József Dombi is contirbuted equally to this work.
[1] Department of Algorithms and Artificial Intelligence, Institute of Informatics, University of Szeged, Árpád Tér 2, Szeged 6720, Hungary
Full list of author information is available at the end of the article

Abstract

Detecting a community structure on networks is a problem of interest in science and many other domains. Communities are special structures which may consist nodes with some common features. The identification of overlapping communities can clarify not so apparent features about relationships among the nodes of a network. A node in a community can have a membership in a community with a different degree. Here, we introduce a fuzzy based approach for overlapping community detection. A special type of fuzzy operator is used to define the membership strength for the nodes of community. Fuzzy systems and logic is a branch of mathematics which introduces many-valued logic to compute the truth value. The computed truth can have a value between 0 and 1. The preference modelling approach introduces some parameters for designing communities of particular strength. The strength of a community tells us to what degree each member of community is part of a community. As for relevance and applicability of the community detection method on different types of data and in various situations, this approach generates a possibility for the user to be able to control the overlap regions created while detecting the communities. We extend the existing methods which use local function optimization for community detection. The LFM method uses a local fitness function for a community to identify the community structures. We present a community fitness function in pliant logic form and provide mathematical proofs of its properties, then we apply the preference implication of continuous-valued logic. The preference implication is based on two important parameters ν and α. The parameter ν of the preference-implication allows us to control the design of the communities according to our requirement of the strength of the community. The parameter α defines the sharpness of preference implication. A smaller value of the threshold for community membership creates bigger communities and more overlapping regions. A higher value of community membership threshold creates stronger communities with nodes having more participation in the community. The threshold is controlled by δ which defines the degree of relationship of a node to a community. To balance the creation of overlap regions, stronger communities and reducing outliers we choose a third parameter δ in such a way that it controls the community strength by varying the membership threshold as community evolves over time. We test the theoretical model by conducting experiments on artificial and real scale-free networks. We test the behaviour of all the parameters on different data-sets and report the outliers found. In our experiments, we found a good relationship between ν and overlapping nodes in communities.

Keywords: Overlapping-communities, Continuous-value-logic, Preference-implication, Power-law, Networks

Introduction

In network modelling graphs are used to model abstract relationships of inter-related data. Graphs with nontrivial features are also known as complex networks. The non-trivial features may consist of a clustering coefficient, heavy-tailed degree distribution, reciprocity, community structure and others (Kim and Wilhelm 2008). Real-world examples of such graphs implemented to solve complex problems include social networks, social graphs API such as Facebook's graph, a recommendation engine such as yelp GraphQL API, and path optimization algorithms such as Google maps platforms(maps, routes, car navigation). For many years researchers have been interested in finding the communities in networks. The definition of a community states that a group of nodes has a higher likelihood of connecting than other nodes in the network (Barabási et al. 2016; Gulbahce and Lehmann 2008). There are two hypotheses that can be used to define the communities (Barabási et al. 2016). In the first fundamental hypothesis, the community structure can be discovered by looking at the connections in the network. There is a ground truth about the community structure and wiring of the network. The second hypothesis, connected to the density hypothesis, states that a community is a connected subgraph. In other words, two isolated subgraphs in a network cannot belong to the same community. An example of a community structure is shown in Fig. 1. Nodes in the community have more connections with the members of the same community than with the members outside the community. Maximal cliques automatically form the basis for the second hypothesis. A clique is a fully connected subgraph and it has a maximal link density. However, the clique method for community identification is limited as larger cliques are rare (Carter and Park 1993). The communities are therefore defined as stronger or weaker communities and this relaxes the rigid definition of communities based on cliques (Barabási et al. 2016). We have used these definitions to define a community in detail in "Strength of communities based on links" section. A real network may consist

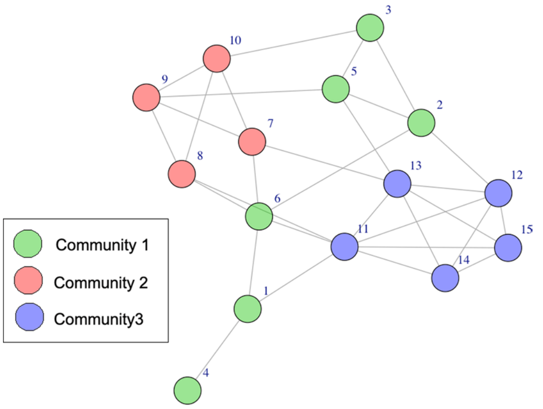


Fig. 1 Strong and Weak Community. An example of a strong and weak community network. The community in green is a weak community as it strictly violates the definition of strong community for each node. However the community in red can be viewed as a strong community if node 6 is ignored. In the case of the purple community, all the nodes strictly follow the definition of a strong community from 1

of nodes that can belong to more than one community depending on the real world it models (Palla et al. 2005). The overlapping regions are common in networks where some nodes can exhibit properties of more than one community (Barabási et al. 2016). Community detection in networks with more than one membership is of great interest as it resembles more closely the real-world networks (Palla et al. 2005). The identification of these community structures can provide a solution for many risky situations. Controlling community infection earlier and in the time of pandemic like the current one is of great interest (Vanhems et al. 2013). The theory of multi-criterion decision making and fuzzy sets involves capturing the indecisive human thinking. Fuzzy sets involve representing objects with unclear boundaries. Continuous-valued logic is a natural generalization of discrete logic. However, the general structure of continuous-valued is different from discrete logic as operations like negation cannot be defined in terms of addition in a similar way (Levin 2007). Preference modelling is an approach used in multi-criterion decision making where multiple criteria are taken in consideration at the same time (Csiszár et al. 2020; Dombi and Jónás 2020). Community has a vague definition as there is no control on inner and outer links even while preserving the definition of a community. In other words, there is no strict threshold for the number of inner and outer links except for the inequality operator which looks for a higher density of links between the community members. The existing approaches which deal with the strength of a community define it after the communities have been detected. In Fig. 1 the communities detected by a community detection algorithm are unable to control the creation of these communities with respect to certain controls on number of inner and outer links. To provide a solution to such scenarios we provide a mathematically proven approach on community detection which is based on continuous-valued logic and it can describe the vagueness of a community definition.

Related work

Lots of methods have been developed over the past three decades for community detection and we here we present summary of some of them. The use of social networks and online communities provides a good incentive for industry and science to gain a better insight into community detection.

One of the best known methods is the clique percolation method (CPM), which assumes that there is an overlapping set of fully connected graphs in the community (Derényi et al. 2005). It searches for the adjacent cliques for example to identify cliques of size k in a network. Afterwards, a new graph is constructed such that each node in the graph denotes one of these k cliques. Connected components identify the cliques which form communities. The presence of a node in many k cliques introduces an overlap between any two communities. However, it is beneficial for the network that has a dense connection type of structure. One disadvantage of this algorithm is that it looks for a pattern or localized structure in the network, and it runs with polynomial time complexity. The overlapping regions in the network can be studied by establishing the relationship between the structure of the network and the function that identifies the communities (Newman 2003). Many papers have presented the studies and evaluation of community structures (Schaub et al. 2017; da Fonseca Vieira et al. 2020; Cherifi et al. 2019). Some algorithms use a local function to characterize the densely connected group of nodes. One method was provided by Baumes, (Baumes et al. 2005) who introduced a two-step procedure. In the first step, the network nodes are ranked based on the PageRank algorithm (Page et al. 1999). Then the higher value nodes are removed step-by-step until small disjoint clusters are formed. Then, an Iterative scan (IS) adds or removes the nodes until the local density function improves. The disadvantage of IS is that it also produced disconnected components in many cases. This drawback was corrected by Kelly in a new method called connected iterative scan (CIS) (Kelley 2009). His new method checked for connectedness in each iteration. The local function optimization approach is also used by Lancichinetti in the (LFM) to detect overlapping community (Lancichinetti et al. 2009). LFM introduces a fitness function for the definition of a community, as shown in Eq. 3. The random seed nodes from the network form the community until the fitness function in Eq. 3 is locally maximal. In the OSLOM method, the local optimization of the fitness function is also used, which determines the statistical significance of clusters with respect to random fluctuations (Lancichinetti et al. 20011). First, it identifies the relevant cluster until the local fitness function converges. Then, an internal analysis of these clusters is performed on their union. Lastly, it identifies the hierarchical structure of these clusters. This method offers a comparable performance with those of other existing algorithms on synthetic networks. The main advantage of this method is that it can also be used to improve the clusters generated by other algorithms (Dombi and Dhama 2020).

Among other popular methods available, one of the best approaches is given by fuzzy community detection. In this approach, the membership for each node is calculated and the dimension of this membership vector can be calculated from the data or it can be chosen by the user (Gregory 2011; Xie et al. 2013; Nepusz et al. 2008).

There have been more overlapping algorithms presented in literature surveys which we will describe in detail (Xie et al. 2013). CESNA is a scalable community detection algorithm which develops communities from node attributes and the edge structure in the network (Yang and Leskovec 2013). It also models the interaction of nodes attributes and the network structure for detecting the community. It also handles robustness even when there is noise in the network. In the BIGCLAM method, the authors introduce a new approach to model k communities in different ways categorized as non-overlapping, overlapping, and nested using non-negative matrix factorization which as we call non-convex. This method calculates the community with a reliable accuracy and it has an F_1 score of community membership greater than 0.85. For a large network, this approach reduces the memory requirements by introducing the sparsity to matrix by ℓ_1 regularization. The main findings of their experiment revealed that overlap regions of a community are more densely connected than non-overlapping regions of a community (Yang and Leskovec 2013). An improved faster version has also been implemented recently which parallelizes the computations and hence improves the time complexity, making it suitable for use on larger networks (Liu and Chamberlain 2017). In another approach presented recently the author deals with the problem of the identification of central nodes, which had not been handled properly by any previous algorithm (Li et al. 2016). A kernel function is used to calculate the leadership of every node in the network. A higher value of leadership is the deciding factor in the selection of central nodes. After the central nodes have been identified, the discrete-time dynamical system framework is used to assign the dynamic community membership to nodes. This method also has several conditions for the convergence of the discrete-time dynamical system. These conditions guarantee the convergence of the node dynamic trajectory and it should reveal any hidden hierarchical community structure. The advantage of this approach is that it can be applied to any community detection algorithm that uses a local community membership optimization function. In some recent work considering the structural point of view, the authors suggest that a deeper comprehension is needed for designing more efficient community detection methods (da Fonseca Vieira et al. 2020). The author also points out importance of strength of weak ties theory (Friedkin 1980). In another work on community structures, the authors have used modularity maximization as basis for designing communities (Cherifi et al. 2019). They propose some efficient deterministic strategies to control the epidemic outbreaks, if the structural information about the network is available. Random intersection graph with communities is a different method to model networks with community structure (Vadon et al. 2019). In this work the authors derive an asymptotic description of local structure of graph which yields the important structural properties on overlapping structures of communities. Some work has also been done on dynamic communities in temporal networks. In one of the paper the authors have used Markov-chain model with community structure (Peixoto and Rosvall 2017). Using the Bayesian inference framework they are also able to explain the temporal interaction data.

Benchmarks are useful for generating artificial networks to test the community detection algorithms. Many generative models for real-world networks exist. These models are also used to test community detection algorithms. The first benchmark for networks introduced by Girvan and Newman in 2001 is known as the GN benchmark (Girvan and Newman 2002). In the GN benchmark, the networks consist of 128 nodes with a similar degree of 16. The network is divided into four groups each of size 32 nodes. The probability of the existence of links between the node and member nodes from its community is given by Z_m. The probability of links between nodes and nodes outside its community is given by $1 - Z_m$ (where $Z_m \in (0, 1)$). The drawbacks of this method are that it has the same degree distribution and that it has a similar community size in the network. Lancichinetti and Fortunato introduced a widely used LFR benchmark.

Strength of communities based on links

An important and well-known hypothesis about the definition of the community asserts that the community is a locally dense connected subgraph in a network (Barabási et al. 2016). Extending this definition, various methods for community detection have been proposed. Communities can be categorized as a strong or weak community.

Strong community

For each node (i) in the community, the number of internal links is more than the external links (Table 1). That is,

$$k_i^{int}(c) > k_i^{ext}(c) \tag{1}$$

Weak community

The total number of internal links of the community is greater than the total number of external links. That is,

Table 1 The notations and their meaning used in the preference relation approach

Meaning	Abbreviation
Input graph on which communities are to be detected	$G = (V, E)$
Out-degree: number of links outside graph *G*	$K_{out}^{\mathcal{G}}$
In-degree: number of links inside graph *G*	$K_{in}^{\mathcal{G}}$
Internal-links: for node *i*, links shared with neighboring nodes belonging to community *c*	$k_i^{int}(c)$
External-links: for node *i*, links shared with neighboring nodes not belonging to community *c*	$k_i^{ext}(c)$
Preference implication for comparing *x* and *y*	$P_\nu^{(\alpha)}(x, y)$

$$\sum_{i\in c} k_i^{int}(c) > \sum_{i\in c} k_i^{ext}(c) \tag{2}$$

The LFM method employs the definition of community as defined by the connectivity and density hypotheses (Barabási et al. 2016; Lancichinetti et al. 2008, 2009). The LFM method also introduced new features of heterogeneity in-degree distribution and community size distribution.

In the following section, we explain how we were inspired by the LFM method in our approach of community detection. In the case of the LFM method, the weak community definition is used to define the fitness function of the community and the fitness function for a community is :

$$f_{\mathcal{G}} = \frac{K_{in}^{\mathcal{G}}}{(K_{in}^{\mathcal{G}} + K_{out}^{\mathcal{G}})^{\alpha_1}}, \tag{3}$$

where $\mathcal{G}$ is the subgraph or community, $K_{in}^{\mathcal{G}}$ is the total number of internal links, $K_{out}^{\mathcal{G}}$ is the total number of links of each member relative to the told graph and α_1 is a positive real-valued parameter which controls the size of communities. In the method used by LFM, a new node 'a' is added in the community if

$$f_{\mathcal{G}'} > f_{\mathcal{G}}, \tag{4}$$

where $f_{\mathcal{G}}$ is the fitness function of community prior to addition of node a and $f_{\mathcal{G}'}$ is the fitness of community after the addition of node a and α_1 is a real value arbitrary constant that controls the size of the community.

$$f_{\mathcal{G}'} = \frac{K_{in}^{\mathcal{G}'}}{(K_{in}^{\mathcal{G}'} + K_{out}^{\mathcal{G}'})^{\alpha_1}} \tag{5}$$

The LFM approach works well and it has been used by lot of network scientists, biologists and statisticians. However, the approach used in LFM method and many other community detection approach focus on local optima. There is very little scope to control the strength of the community. We enlist some more improvements which gives the user an edge over the existing methods to somehow control the creation of the partitions of existing networks, while adhering to the original detection method.

Some limitations of community detection approaches

- In a real-world type of network, every community has a threshold for membership criteria. In the LFM method, the inequality operator is limited as there are no criteria to control the strength of community relative to the strength of a community(threshold).
- LFM has been used to detect communities in networks with overlapping structures. However this method, limits that the overlap regions that may belong to more than one community with a different degree, which is difficult to decide based on Eq. 4.
- There is no parameter that measures the membership contribution of an overlapping node or a non overlapping node towards its own community. For instance, a node may belong to two communities with same number of links but it can still make a different contribution to each community.

These limitations are crucial even to other community detection algorithms which use the principle of a number of links while detecting the communities based on any local function optimization. Some type of tuning on these detection algorithms can project different decision boundaries for community detection in different scenarios. We define this type of control on the community detection algorithm as 'controlling the strength of the community'. This approach overcomes some of these limitations of Eq. 4. We propose the so-called preference implication-based method, where the threshold that defines the strength of the community can be controlled. The user can control the strength of the community by changing the threshold value and hence controlling the decision boundary for each community to be detected and for each member that is part of the community to be detected. For example the social networks differ from other networks (Radicchi et al. 2004). In a social network, when a community starts with one member the joining criteria to become the a member of this community is very low as there are no members in the community, but after a certain period of time the joining criteria become more strict as the community has now matured.

Problem statement

To detect nc communities on an undirected graph $G = (V, E)$. The preference implication-based method finds nc communities of different strengths defined by the threshold parameter δ.

Preference relations

The relation R between A and B in set theory is defined as the subset of the Cartesian product (i.e. $R \subset A \times B$), where $A \times B$ is the set of all ordered pairs (a, b) and $a \in A$,$b \in B$. To define the preference operator we will use the following example.

$$aRb \Leftrightarrow b \text{ is more likable than } a.$$

In the classical sense, preference is a binary relation related to implication. The preference implication gives the degree of truth of a statement. Hence,

$$P(a, b) = \text{truth of } (a < b)$$

This new method based on preference-implication can be used to define overlap structures in networks. The preference relation has the monotone property and here we define it so that it can be used to make multi-criteria decisions (Dombi et al. 2006).

The preference relation $P_\nu^{(\alpha)}(x, y)$ tells us how true is $(x < y)$ sometimes, which in our case also indicates how strong the community is. Here, $x = f_{\mathcal{G}}$ and $y = f_{\mathcal{G}'}$

$$P_\nu^{(\alpha)}(x, y) \qquad\qquad \text{where } x < y \text{ and } x, y, \nu \in (0, 1) \tag{6}$$

In Table 2 the domain of all the parameters we have used in the preference-based method is explained.

$$P_\nu^{(\alpha)}(x, y) = degree(x < y) \tag{7}$$

The key parameters ν, α and δ

The parameter ν controls the intensity of the truth of the inequality operator. In simpler words, $100 > 10$, $100 > 50$, $100 > 90$, $100 \geq 100$. In all of the cases the degree of greater is not the same. For instance $100 > 10$ represents a much higher difference than $100 > 50$, $100 > 90$, $100 \geq 100$. Selection of nodes in a community which has a greater contribution towards the community can be controlled by the user by varying ν parameter. The sharpness of preference is controlled by the parameter α. The sharpness tells us the slope of the curve, which is also a measure of how fast is the change in the curve shown in Fig. 2. The ν parameter is important for selecting the difference of threshold in the community fitness value when there is the addition of a new member. The inequality operator in LFM is only capable of selecting the members that increase the fitness value of the community. However, the ν in preference can control this threshold by taking any user-defined membership strength. This threshold can be varied in different situations according to the application of the algorithm on different types of data. These different values can be handled by the δ parameter, which is directly related to ν. Although the stopping criteria of the algorithm is when there is no increase in the fitness value of subgraph, the ν parameter also allows for tweaking this definition in some rare and exceptional situations. For instance, giving a node chance to become a member of community, by giving them some discount on the membership requirement. This is very similar to situations of trade-off in supply and demand. When there is great competition to become member of community then a higher membership threshold criteria can be introduced, and when the competition is less in those situations, a lower membership threshold is similar to offering a discount

Table 2 The range of the parameters values in the preference relation

Parameters	Domain
Preference relation	$P_\nu^{(\alpha)}(x, y) \in (0, 1)$
Threshold	$\nu \in (0, 1)$
Sharpness parameter	$\alpha \in (0, 1)$
$x = f_{\mathcal{G}}$	$f_{\mathcal{G}} \in (0, 1)$
$y = f_{\mathcal{G}'}$	$f_{\mathcal{G}'} \in (0, 1)$
$\delta \in deltaset$	$(deltaset) \in (0, 1)$

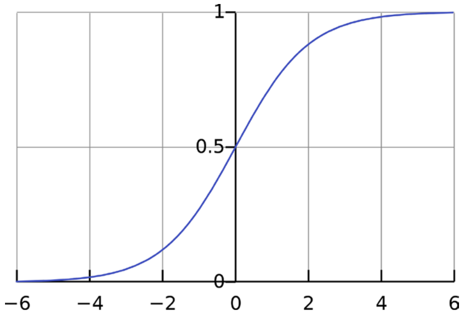


Fig. 2 The preference relation sigmoid function

and relaxing the strictness of membership criteria for nodes to become members of a community. The preference implication can not only be used to enhance the applicability of existing LFM method for community detection, but it can also be used in LFR benchmark for generating networks with such type of communities. The $alpha1$ parameter in LFM is used in Eqs. 41 and 1 for community sizes. The preference method is quite different from $alpha1$ as it measures the threshold of every node which not only controls the size of the community but the quality of nodes in terms of strength. The preference method is an approach which handles both the quantitative and qualitative aspects of a community.

Preference implication in Boolean algebra

In Boolean algebra, the preference implication has a special form in the range (0, 1) range. In terms of an equality relation, we can define the preference operator like so :

Case 1

$P(x, y) \in (0, 1)$ and in Boolean algebra we have following possibility : $P(0, 1) = 1$ as we know 0 is less than 1, so the the truth of statement has the highest value of 1.

Case 2

$P(1, 0) = 0$ The statement 1 is less than 0 is false, so the preference value of statement is 0.

The current inequality operator of any community detection algorithm uses this boolean form of preference to determine the membership of nodes. However, to be able to add more parameters to control the strength, we will define the continuous-valued logic form of preference implication.

Preference implication in continuous-valued logic

As we are dealing with the strength of communities, so the preference implication has continuous values in the (0, 1) interval. For example, the preference value based relation of truth of $(x < y)$ have these three calculated possible values,

1. $P(x < y) = 0.9$
2. $P(x < x) = 0.5$
3. $P(y > x) = 0.1$

$P_\nu^{(\alpha)}(x,y) > \nu$ *if and only if* $x < y$. Now, let us assume that the threshold ν is 0.5 and that the sharpness parameter α is 1. Consider the following un-normalized values of $x = 3$ and $y = 9$ to explain in detail the above-mentioned three scenarios.

Case 1$P_\nu^{(\alpha)}(3,8)$

The truth value of statement $(3 < 8)$ is 0.9, which greater than ν as $0.9 > 0.5$. Hence, we establish the truth statement $3 < 9$ (using preference relation) with a strength greater than ν(threshold).

Case 2 $P_\nu^{(\alpha)}(3,3)$

For the truth value of statement $(3 < 3)$ is 0.5 which is just the threshold value. So in this case $3 < 3$ is a weak statement as it is at the threshold, but still it establishes the truth of the statement.

Case 3 $P_\nu^{(\alpha)}(3,8)$

The truth value of statement $(3 > 8)$ is 0.1, which is less than ν as $0.5 > 0.1$. So, it is a very weak statement or in other words it is a false statement. And hence, $3 > 8$ is not a true statement.

Here the sharpness threshold defined by α and ν is the threshold used for comparing the truth values. The intensity of the preference is controlled by the parameter of this function. The parameter is $\nu \in (0,1)$ and f is a generator of a strict t norm. The preference implication in pliant logic form is (Dombi et al. 2006) :

$$P_\nu^{(\alpha)}(x,y) = \begin{cases} 1, & \text{if } (x,y) \in (0,0),(1,1) \\ f^{-1}\left(f(\nu)\left(\frac{f(y)}{f(x)}\right)^{\alpha}\right), & \text{otherwise} \end{cases} \tag{8}$$

We can also define our own function in Eq. 8. We define a special function for our purpose which has the monotonic property (Dombi 1982; Dombi and Jónás 2018). Using the Dombi operator for the preference relation in f_G, we get

$$P_\nu^{(\alpha)}(x,y) = \frac{1}{1+\frac{1-\nu}{\nu}\left(\frac{1-y}{y}\frac{x}{1-x}\right)^{\alpha}} \tag{9}$$

$$P_\nu^{(\alpha)}(x,y) > \nu \textit{ if and only if } x < y, \tag{10}$$

$$P_\nu^{(\alpha)}(x,y) = \begin{cases} > \frac{1}{2}, & \text{if } y > x \\ = \frac{1}{2}, & \text{if } x = y \\ < \frac{1}{2}, & \text{if } x > y \end{cases} \tag{11}$$

This method allows us to control the size and number of these overlapping regions. The threshold parameter ν of preference allows us to design the communities according to our requirement of the strength of a community. The value of ν is desirable when $\nu > 0.5$, as the Dombi operator system is a sigmoid function. The graphical form of the sigmoid function is shown in Fig. 2.

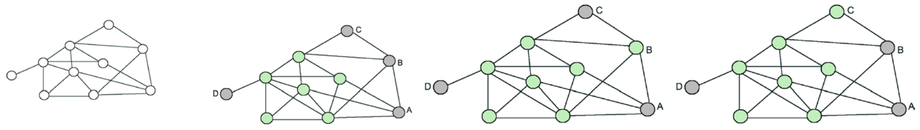
An example of the use of the preference relation in community detection is shown in the example in Fig. 3. As we see in the Fig. 3, the following possible members to be added to the community. Here, the classical method selects the one best node with a higher threshold to be added to the community. The preference relation allows one to select more than one node and also to control the threshold in different situations. For a higher value of δ, we can create a community with a strong membership if node B is chosen as shown in 2 of Fig. 3. As node A has three in-going edges and node B has two in-going edges they count as strong members of a community, where the in-going edges are greater than the out-going. Hence a high value of δ can be used in these circumstances to establish the membership for the community. Node C has only one in-going edge so it can viewed as a weak member as shown in 3 of Fig. 3. In the case of node D, the membership will be very weak with a very low value of δ (Fig. 3 and Table 3).

Table 3 Preference rule table: rule for the new node addition in the subgraph based on the threshold ν value and preference value

Preference value $\mathbf{P}_{\nu}^{(\alpha)}(\mathbf{x}, \mathbf{y})$	**Fitness function rule**	**Decision of a new node addition**
$0 < P_{\nu}^{(\alpha)}(x,y) < 0.5 \quad P_{\nu}^{(\alpha)}(x,y) > \delta$	$f_{\mathcal{G}} > f_{\mathcal{G}'}$	Not desirable for a community membership but it can have a very weak overlapping membership
$P_{\nu}^{(\alpha)}(x,y) = 0.5 \quad P_{\nu}^{(\alpha)}(x,y) > \delta$	$f_{\mathcal{G}} = f_{\mathcal{G}'}$	Desirable for Community membership and it can have a weak overlapping membership
$1 > P_{\nu}^{(\alpha)}(x,y) > 0.5 \quad P_{\nu}^{(\alpha)}(x,y) > \delta$	$f_{\mathcal{G}} < f_{\mathcal{G}'}$	Strong community membership and it can have a strong overlapping membership

Different types of networks for community detection

We previously introduced the preference implication for undirected and unweighted graphs networks and collected some results on them (Dombi and Dhama 2020). However, we did not introduce the other different cases of graph networks. In this section, we will provide an extension to handle the weighted undirected graph networks. For the case of unweighted directed and weighted directed graph networks, there are some state-of-the-art methods that provide different approaches of how to detect communities on the directed graph networks (Malliaros and Vazirgiannis 2013; Kim et al. 2010; Leicht and Newman 2008). One of the approaches available in the literature is to transform the directed graph network to an undirected graph network by preserving the direction of edges by introducing the weights. Then, we apply the community detection algorithm of the undirected graph network on the transformed graph network. The louvian method based on modularity optimization has an improved version for directed networks where the modularity definition is based on the community connection matrix (De Meo et al. 2011). Then, the community detection method is applied on a directed graph. In another approach, the edge direction is preserved in form of weights and directed network is transformed into an undirected bipartite network (Dugué and Perez 2015). Then, a new modularity measure is defined by modifying the modularity of undirected networks. A partition with the highest modularity value is treated as a community of networks. However, this method is limited due to it not being too efficient on large scale networks (Good et al. 2010). Some improved and faster community detection methods based on



(a) Figure A (b) Figure B (c) Figure C (d) Figure D

Fig. 3 This figure shows three different scenarios of node addition in the community. The node selection based on the rules in Table 3 in a community

modularity maximization have been proposed recently (Li et al. 2018; Gach and Hao 2013; Zhuang et al. 2019; Bhowmick and Srinivasan 2013; Que et al. 2015).

Undirected and unweighted graphs

For unweighted graphs, we will use the fitness function defined in Eq. 3.

$$f_{\mathcal{G}} = \frac{K_{in}^{\mathcal{G}}}{(K_{in}^{\mathcal{G}} + K_{out}^{\mathcal{G}})^{\alpha_1}},$$

Undirected and weighted graphs

In weighted and undirected graphs we normalize the weights for each node links. For a vertex $v_i \in V$ and all the links $e_1, e_2 ... e_m \in E$ of graph G, we introduce the normalized weighted fitness function for subgraph $\mathcal{G}$ as follows:

$$\begin{aligned} W_{\mathcal{G}} &= \sum_{j=1}^{d} w_j, where\ w_j\ is\ weight\ of\ edge\ j \\ K_{in} &= \sum_{j=1}^{m_1} k_j \left(\frac{w_j}{W_i} \right)\ where\ m_1\ is\ number\ of\ inner\ links, \\ K_{out} &= \sum_{j=1}^{m_2} k_j \left(\frac{w_j}{W_i} \right)\ where\ m_2\ is\ number\ of\ outer\ links, \\ d &= m_1 + m_2, \\ f_{\mathcal{G}} &= \frac{K_{in}^{\mathcal{G}}}{(K_{in}^{\mathcal{G}} + K_{out}^{\mathcal{G}})^{\alpha_1}}, \end{aligned}$$

In our method, we have only normalized weights of edges of community. However, the weights of all the edges of a graph can also be normalized before applying the normalization on community.

Preference relation properties

Theorem 1 *The necessary and sufficient conditions for satisfying all the four distributivity equations are* (Dombi and Jónás 2018)*:*

1. *The conjunction and disjunction are weighted operators.*

2. *Negation is a strong negation.*
3. *The De Morgan laws are valid for the above triple.*
4. *The implication is a fuzzy implication which is continuous except for the points* $(0, 0)$ *and* $(1, 1)$.
5. *The law of contrapositive is valid. And these conditions can only be satisfied if the operators are elements of a pliant system and the implication is a preference implication. That is,*
   $c(x, y) = f^{-1}(uf(x) + vf(y))$,
   $d(x, y) = f^{-1}\left(\frac{f(x)f(y}{vf(x)+uf(y)}\right)$,
   $\eta(x) = f^{-1}\left(\frac{f^2(\nu)}{f(x)}\right)$,
   $\begin{cases} 1, & \text{if } (x, y) \in (0,0), (1,1), \\ f^{-1}\left(f(\nu)\frac{f(y)}{f(x)}\right), & \text{otherwise,} \end{cases}$
   for all $x, y \in [0, 1]$ *where* $u, v \in (0, \infty)$ and $\nu \in (0, 1)$

Definition 1 For $x, y \in [0, 1]$, $P(x, y)$ has the reciprocity property when

$$P(x, y) + P(y, x) = 1 \tag{12}$$

Definition 2 A preference relation p is multiplicative transitive if

$$\frac{p(x, y)p(y, z)}{p(y, z)p(z, y)} = \frac{p(x, z)}{p(z, x)} \tag{13}$$

for all x, y, z in [0, 1] and the above formula is well defined.

Note 1 - We define this special function for our purposes and it has the monotonic property, using the Dombi operator for the preference relation in Eq. 8.

Definition 3 A preference implication p is reciprocal if

$$p(x, y) + p(y, z) = 1 \qquad x, y \in [0, 1] \tag{14}$$

Here, we will prove that for preference implication,

$$P_\nu^{(\alpha)}(x, y) > \nu \qquad \text{if and only if } x < y \tag{15}$$

Proof

We know the form of preference implication. It is :

$$P_\nu^{(\alpha)}(x, y) \qquad \text{where } x < y \text{ and } x, y, \nu \in (0, 1). \tag{16}$$

Using the Dombi operator for the preference relation from Eq. 9, and recalling note 1, we have

$$P_{\nu}^{(\alpha)}(x,y) = \frac{1}{1+\frac{1-\nu}{\nu}\left(\frac{1-y}{y}\frac{x}{1-x}\right)^{\alpha}}. \tag{17}$$

Now, from Eq. 9 we get the expression given below,

$$\frac{1}{1+\frac{1-\nu}{\nu}\left(\frac{1-y}{y}\frac{x}{1-x}\right)^{\alpha}} > \nu. \tag{18}$$

Taking the reciprocal of the LHS and RHS, we get

$$\frac{1-\nu}{\nu}\left(\frac{1-y}{y}\frac{x}{1-x}\right)^{\alpha} < \frac{1-\nu}{\nu}. \tag{19}$$

Subtracting 19 from 1, we get

$$\left(\frac{1-y}{y}\frac{x}{1-x}\right)^{\alpha} < 1. \tag{20}$$

After cross-multiplication of the above term we get the following reduced form :

$$(1-y)x < y(1-x), \tag{21}$$

$$x - xy < y - xy. \tag{22}$$

Cancelling the common term $-xy$ on each side of the equation we get,

$$x < y, \tag{23}$$

and hence it is proved. Therefore,

$$P_{\nu}^{(\alpha)}(x,y) > \nu \qquad \text{if and only if } x < y. \tag{24}$$

□

Commutative property

Here x is the fitness value of the sub-graph before the addition of a new node and y is the fitness value of the sub-graph after the addition of a new node. Here, n and o for simplicity denote k_{in} and k_{out}, respectively.

As defined above, we know that

$$x = \frac{k_{in}}{(k_{in}+k_{out})^{\alpha_1}} \tag{25}$$

For simplicity we choose $\alpha_1 = 1$, and we get

$$x = \frac{k_{in}}{k_{in} + k_{out}}. \tag{26}$$

Now, taking the reciprocal of x and subtracting 1 from it, we get

$$\frac{1-x}{x} = \frac{k_{out}}{k_{in}}. \tag{27}$$

Taking reciprocal of both the sides we get,

$$\frac{x}{1-x} = \frac{k_{in}}{k_{out}}. \tag{28}$$

Using the commutative property, we have

$$\frac{k_{in}}{k_{out}} = \frac{n}{o} \tag{29}$$

Similarly, for y when $\alpha_1 = 1$, we get

$$y = \frac{k'_{in}}{k'_{in} + k'_{out}} \tag{30}$$

Now we repeat the same steps for y as we did for x

Taking the reciprocal and subtracting and taking reciprocal again, we get

$$\frac{1-y}{y} = \frac{k'_{in}}{k'_{out}}. \tag{31}$$

Also, in other notation we have

$$\frac{k'_{in}}{k'_{out}} = \frac{o'}{n'}. \tag{32}$$

From Eq. 9 we get

$$P_{\nu}^{(\alpha)}(x,y) = \frac{1}{1 + \frac{1-\nu}{\nu}\left(\frac{1-y}{y}\frac{x}{1-x}\right)^{\alpha}}. \tag{33}$$

We will introduce the threshold δ and from Eq. 10, we get

$$P_{\nu}^{(\alpha)}(x,y) > \delta. \tag{34}$$

Therefore,

$$\frac{1}{1 + \left(\frac{o'}{n'}\frac{n}{o}\right)^{\alpha}} > \delta.$$

Taking the reciprocal we get

$$1 + \left(\frac{o'}{n'}\frac{n}{o}\right)^{\alpha} < \frac{1}{\delta}, \tag{35}$$

$$\left(\frac{o'}{n'}\right)\left(\frac{n}{o}\right) < \left(\frac{1}{\delta} - 1\right)^{\frac{1}{\alpha}}. \tag{36}$$

Let us assume that the RHS of above inequality is k. Then

$$\left(\frac{1}{\delta} - 1\right)^{\frac{1}{\alpha}} = k. \tag{37}$$

And we get,

$$\left(\frac{o'}{n'}\right)\left(\frac{n}{o}\right) < k. \tag{38}$$

Taking the log on both the sides of above expression, we get

$$\ln(o') - \ln(o) + \ln(n) - \ln(n') < \ln(k) \tag{39}$$

$$\Delta o - \Delta n < k', \text{ where } k' = \frac{1}{\alpha}(\ln(1-\delta) - \ln(\delta)) \tag{40}$$

On a logarithmic scale the difference between the two situations is where a new member is added to the community and there is an increase in the number of inner links of the community, and also a comparatively small increase in the number of outer links. This increase is directly related to k, where k is $\left(\frac{1}{\delta} - 1\right)^{\frac{1}{\alpha}}$. This term denotes the strict threshold for the addition of a new member to the community.

Algorithm and implementation of the preference-based method

A brief version of the algorithm is given in our paper (Dombi and Dhama 2019).

Table 5 Statistics of unweighted real networks are given below (Csardi 2015; Csardi and Nepusz 2006)

I	Network name	Nodes/size	Edges	Average degree	Maximum degree	Density	Diameter	Transitivity
1	Zachary's Karate Club	34	78	4	17	0.1390	5	0.2556
2	Facebook Caltech	769	16656	43	248	0.0564	6	0.2912
3	Tortoise Social	787	1506	3	30	0.0048	21	0.4199
4	EU Road	1174	1417	2	10	0.0020	62	0.0338
5	Facebook Nips	2888	2981	2	769	0.0007	9	0.2912
6	Indo-china Webgraph	11,358	47606	8	199	0.0007	27	0.5669
7	Facebook Simmons	1518	32988	43	300	0.0286	7	0.2123

These networks were taken from a network repository (Csardi and Nepusz 2006; Traud et al. 2011, 2012; http://networkrepository.com)

Algorithm 1: Fitness Function to calculate fitness of subgraph

```
Input: $K_{in}, K_{out}$ /* In-degree, Out-degree of subgraph */
Output: $f$ /* Fitness value of the subgraph */
Function Fitness($K_{in}$,$K_{out}$):
    $alp \longleftarrow 1$
    $f \longleftarrow \frac{K_{in}}{\left(K_{in}+K_{out}\right)^{alp}}$
    return $f$
End Function
```

We presented the unweighted graph community detection in our previous papers (Dombi and Dhama 2020, 2019). In this paper we also implement the weighted graph network community detection by incorporating the weights in the fitness function.

Algorithm 2: Main Function

```
Input: $G, nc$ /* G: Graph on which community detection is to be performed, nc :
    number of communities */
Output: $L$ /* List of communities detected on the Graph, i.e. each index of $L$
    has one community subgraph */
Function Calculate($i$):
    foreach $i \in neighborhoodof\ c$ do
        /* Select next node using Preference */
        $Y \longleftarrow Fitness(K_{in}, K_{out})$ /* $\mathcal{G}'_i = \mathcal{G} + i$ */
        $PreferenceListofc_i \longleftarrow Preference(X, Y)$
    return $PreferenceListofc_i$
End Function
Function Main($G$, $nc$):
    $L \longleftarrow head$ /* Each community is initialised with one node, which is the head
        node */
    $First\_itr \longleftarrow 1$ /* Flag for first iteration */
    foreach $c \in L$ do
        $delta \longleftarrow (0.1, 0.2, 0.3, ...0.9)$
        /* Initialize threshold */
        $\mathcal{G} \longleftarrow L[c]$
        while $\mathcal{G}' > \mathcal{G}$ or $First\_itr$ /* Stopping condition */
         do
            $\mathcal{G} \longleftarrow \mathcal{G}'$
            $X \longleftarrow Fitness(K_{in}, K_{out})$ /* Fitness of $\mathcal{G}$ */
            $PreferenceListofc_i \longleftarrow Calculate(i)$
            foreach $i$ in $PreferenceListofc$ do
                if $PreferenceListofc_i > \delta$ then
                    /* $i^{th}node$ is member of the community $c$ */
                    $\mathcal{G}' \longleftarrow \mathcal{G}_i$
                    $X \longleftarrow Fitness(K_{in}, K_{out})$ /* update the new $X$ which has
                        community with new members */
            $L[c] \longleftarrow \mathcal{G}$
            End while
    return L
```

Algorithm 3: Function for Preference Relation Implication value

```
Input: $F_{\mathcal{G}}, F_{\mathcal{G}'}$ /* Fitness of subgraph $F_{\mathcal{G}}$, Fitness of the subgraph $F_{\mathcal{G}'}$ */
Output: $P$ /* Preference value of the subgraph */
Function Preference($F_{\mathcal{G}}$, $F_{\mathcal{G}'}$):
    $alpha \longleftarrow 1$
    $X \longleftarrow Normalize(F_{\mathcal{G}})$
    $Y \longleftarrow Normalize(F_{\mathcal{G}'})$
    if $(X > 0\ \&\ X < 1\ \&\ Y > 0\ \&\ Y < 1)$ then
        $P \longleftarrow \frac{1}{1+\frac{1-\nu}{\nu}\left(\frac{1-Y}{Y}\frac{X}{1-X}\right)^{\alpha}}$
    end
    else
        $P \longleftarrow null$
    end
    return $P$
```

This strategy is similar to the unweighted graph network community detection. The preference-based method for the weighted graph network works in the following way.

1. We start the process of community creating by randomly selecting seed nodes for each community c. Centrality measures such as PageRank, betweenness, and other centralities can also be used to determine the seed nodes. Each community c is represented by $\mathcal{G}$, which is a subgraph with one node and one virtual edge.
2. We find all the neighbours of the community. To make the selection of best neighbour to add in the community, we create a new subgraph $\mathcal{G}_i' = \mathcal{G} + i$ corresponding to each neighbour i. Now, we have to choose the best subgraph out of this list; i.e. the best next node or nodes to be added in $\mathcal{G}$ to improve the local fitness function of the community. We also consider the strength of community denoted by δ when making this selection.
3. We create a preference list corresponding to each subgraph of neighbouring node i of $\mathcal{G}$. If the preference value of any subgraph is greater than the δ parameter value, then these nodes are included in the community.
   $\mathcal{G}' = \sum \mathcal{G}_i$: a new subgraph which includes nodes with a preference value greater than δ.
4. The process is repeated from step 4 using a while loop until it satisfies the stopping criteria.

Experiments and results

We selected artificial and real networks to test our algorithm. For the artificial network, we generated different sizes of networks from the LFR benchmark (Lancichinetti et al. 2008). The real networks were selected standard repositories (Csardi and Nepusz 2006; Traud et al. 2011, 2012; http://networkrepository.com).

Table 4 Unweighted artificial networks generated from the LFR benchmark. The mixing parameter $\mu = 0.1, t = 1, t_2 = 1$

Size	Nodes	Edges	Average degree	Maximum degree	Density	Diameter	Transitivity
1	30	126	6	10	0.2896	4	0.4263
2	60	491	12	20	0.2774	3	0.4819
3	100	444	5	10	0.0896	4	0.1779
4	200	2901	20	40	0.1457	3	0.4070
5	500	7619	20	50	0.0610	4	0.4444
6	1000	15,340	20	100	0.0307	4	0.3471
7	2000	272,024	272	398	0.1360	3	0.6058
8	5000	234,632	93	517	0.0187	4	0.2777
9	10000	1,268,003	150	300	0.0253	3	0.2690

The degree distribution of those networks which obey a power law is also shown in Fig. 4

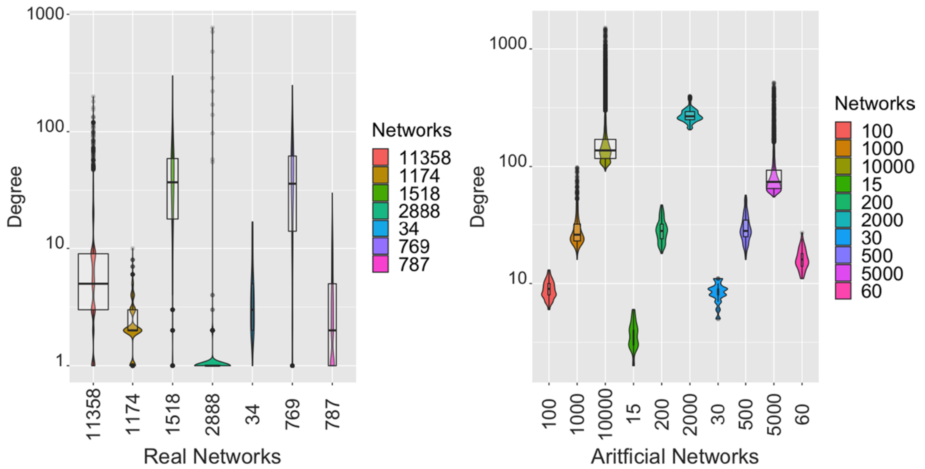


Fig. 4 The degree distribution of unweighted real networks is shown on the left and the degree distribution of unweighted artificial networks used in the simulation is shown on the right

Study on unweighted networks

We generated communities on the unweighted artificial networks of different sizes with different network statistics shown in the Table 4. The structure of the degree distribution of each unweighted real and artificial network is shown in Fig. 4. For comparison purposes, we did not normalize the degree values. The initial seed node of communities to be created can be chosen by the user or it can be randomly selected. Since the selection of a seed is random, the performance of the algorithm is highly dependent on the seed values. There are many centrality measures that are available for choosing the initial seed nodes. In our experiments we have randomly selected the seed nodes.

The effect of network properties degree, eigen centrality, closeness centrality and many more on the detection of community can be further explored in the future. As can be seen from the Table 4, the network density decreases as the size increases, but the diameter does not vary much.

These networks were generated using the LFR benchmark (Lancichinetti et al. 2008). The real and artificial networks we selected for tests have quite similar characteristics in terms of the diameter, density, transitivity, maximum and average degree depending on the size.

Study of the parameter values

For bench-marking purposes, the parameters values were tested on the graphs of the LFR benchmark. To test the parameters, we chose a small-sized networks. The plots for this network and determination of the parameter ν and α are shown in Fig. 5. The key parameters of our algorithm are ν and α. To show the behaviour of these parameters, we choose an artificial network of size 15 and set the initial community number; i.e. the number of communities to be created.

The behaviour of α

We excluded the testing of $\alpha > 0.80$ as the preference implication has an oscillating behavior for values closer to 1. Also, the value of α has a similar oscillatory behavior for

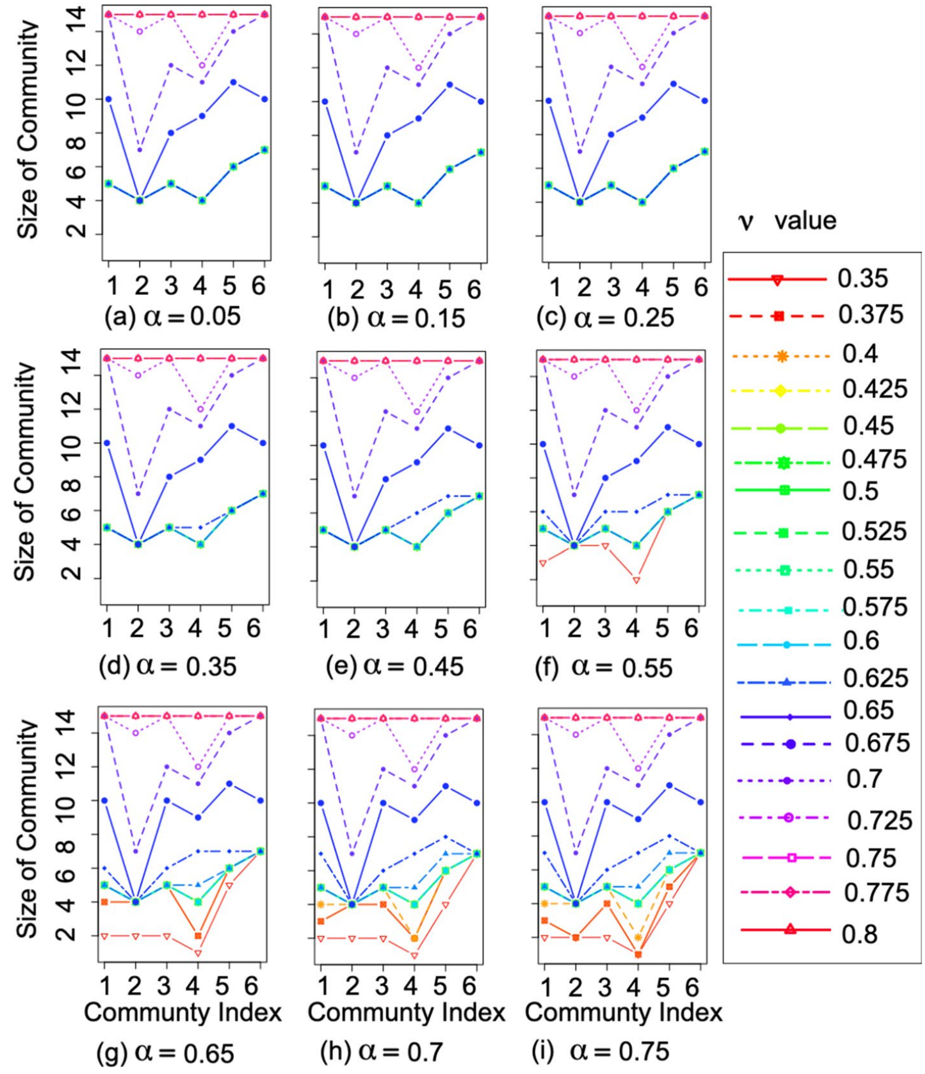


Fig. 5 From the LFR benchmark the statistics of network that we used are $N = 15$, $k = 3$, $maxk = 5$, $\nu = 0.2$, $t1 = 2$, $t2 = 1$, $minc = 3$, $maxc = 5$, $on = 5$, $om = 2$. The value of α is tested for $\alpha \in [0.05, 0.8]$. For each α the ν is tested for $\nu \in [0.35, 0.8]$. The initial seed nodes were the same for all the runs. The value of number of communities to be created is the same in all the cases; i.e. $nc = 6$. The delta set is same in each cases with ν and α with the values [0.24, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95]

values close to 0. However, when α is greater than 0.5 and less than 0.8, the effect of ν is more visible, as each value of ν generates different sizes of communities. This behavior can be seen in the plots in Fig. 5 with α between 0.55 and 0.8. This behavior can be seen in the plots in Fig. 5 with α between 0.55 and 0.8. A range of α between [0.2, 0.8] is desirable as it ensures a non-oscillating behavior of the preference implication (Dombi and Baczyński 2019).

Behaviour of ν

When ν is close to 1, it behaves in a similar way for all the values of α. All the community sizes are close to the network size. This is not a suitable value as all the network nodes belong to one community, which is not the goal of the community detection algorithm. For ν values less than 0.55, the algorithm has the same behavior for $\alpha < 0.55$. The community of size between 4 and 7 was generated for all seed nodes. This behavior is similar to a fixed-size membership of overlapping nodes in the LFR benchmark. However, for $\alpha > 0.55$ and $\nu < 0.5$, the communities of different size are generated for different ν values. The algorithm does not produce too many overlapping structures for ν close to 0.

Results on unweighted artificial and real networks

In Tables 6 and 7 the results for the artificial and real networks of community detection are listed with different percentages of initial seed nodes. When the number of seeds for community is increased it increases the number of overlapping nodes but there was little change in the community size behavior as we used the same threshold for all the seeds.

Analysis of ν and α

Based on our analysis of the ν and α we generated communities for $\nu \in [0.3, 0.95]$ and $\alpha \in (0.05, 0.80)$. For a larger network, the community size distribution is shown in Fig. 6. The community size distribution is different for different values of ν but it has a similar value for community sizes with the same threshold. For a smaller value of $\nu < 0.5$ smaller size communities are created and they all have similar sizes. However, for values greater than 0.5, larger community sizes are observed.

Different seed node percent on artificial data

We plotted the behavior of community sizes using the violin plot in Fig. 8. For smaller networks, the shape of the violin and mean both suggest a uniform distribution of community sizes for all the communities. In the case of larger networks, the power-law behavior is much more obvious, as can be seen in Fig. 6 from the shape of violins and in Fig. 9. We also observed a similar behavior for different seed sizes in three different networks of size 200, 500, 1000 in Fig. 7. Community detection was performed using different seed sizes of 5%, 10%, 20% and 30% of the network. We observed a similar behavior for violins in all the cases that have the characteristics of a power law. The size of the communities formed in this way follows a power-law, as can be seen in Fig. 9.

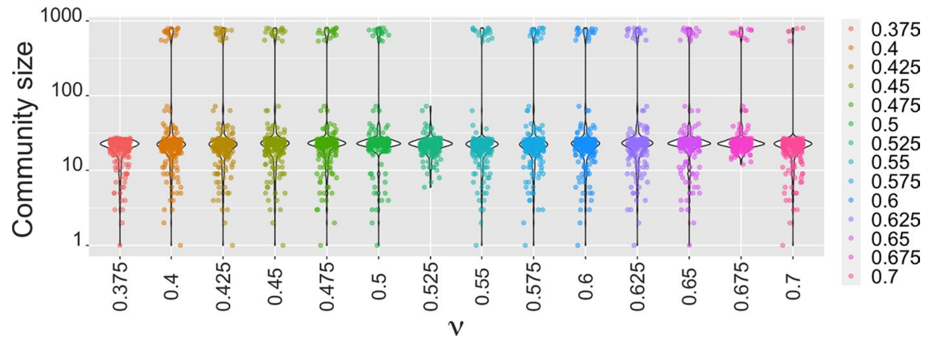


Fig. 6 The community size distribution is shown for a fixed value of ($\alpha = 0.70$) and varying value of $\nu \in [0.375, 0.70]$. The size of network$= 1000$, Average degree $k = 150$, Maximum degree $K_{max} = 200$. The mixing parameter $\mu = 0.1, t = 1, t_2 = 1$. The initial seed node and threshold δ are the same for each value of ν

Table 6 Some statistics of Preference Implication-based method results of communities created on the artificial networks from Table 4 and different sizes of seed nodes

I	Network size	Seed node percent (%)	Number of community	Maximum membership	Maximum community size	Minimum community size
1	15	30	6	5	8	4
2	30	20	6	4	10	7
3	60	10	6	7	16	12
4	100	20	20	15	25	13
5	200	5	10	8	44	22.
6	200	10	20	8	44	22 .
7	200	20	40	8	46	21.
8	200	30	60	13	44	19
9	500	5	25	14	55	19
10	500	10	50	16	55	18
11	500	20	100	14	55	19
12	500	30	150	17	49	17
13	1000	10	100	13	98	22
14	1000	20	200	21	98	22
15	1000	30	300	27	53	20
16	2000	30	600	51	391	212
17	5000	10	500	42	508	56
18	10000	10	1000	0.6	1515	86

Here, $\alpha = 0.75$ and $\nu = 0.6$ for all the entries in the table

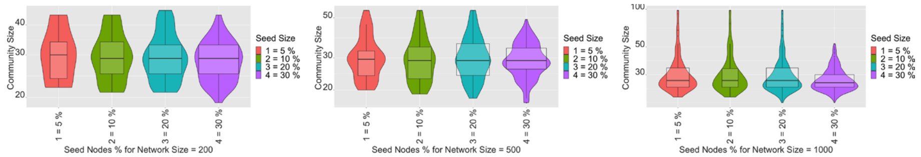


Fig. 7 A violin plot of community size distribution for different sizes of seed nodes on networks with sizes ranging from 200 to 1000. The value of parameters for $deltaset = 0.3, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, \alpha = 1$and $\nu = 0.6$ are the same in all cases. The green color corresponds to the tests with seed size = 10% of the network size, the pink color corresponds to tests with seed size = 5% of the network size, the blue color corresponds to tests with seed size = 20% of the network size, the purple color corresponds to tests with a seed size = 10% of the network the size in all the cases

Study on weighted networks

We collected synthetic weighted networks from LFR benchmark. The statistics of these networks are shown in Table 8.

We created community on the weighted artificial networks with ten percent of seed nodes (Fig. 10).

Evaluation

Many methods are available for evaluating the partitions detected (McDaid et al. 2011; Lancichinetti et al. 2009; Liu et al. 2019; Dao et al. 2018). In the recent work on the evaluation of ground-truth data, the authors have discussed the importance

Table 7 Some statistics of the Preference Implication-based method results of communities created on the unweighted real networks from Table 5 of real networks with different sizes of seed nodes

I	Networ size	Seed node percent (%)	Number of community	Maximum membership	Maximum community size	Minimum community size
1	Zachary's Karate Club	5	2	2	19	17
2	Facebook Caltech	10	76	11	176	17
3	Facebook Caltech	20	152	14	208	12
4	Facebook Caltech	30	228	16	246	11
5	Tortoise Social Network	10	8	4	21	8
6	EU Road Network	10	6	5	82	6
7	Facebook Nips	10	200	2	288	2
8	Indo-china Webgraph	5	500	17	120	8
9	Facebook Simmons	10	150	21	182	7

Also, here $\alpha = 0.75$ and $\nu = 0.6$ for all the entries in the table

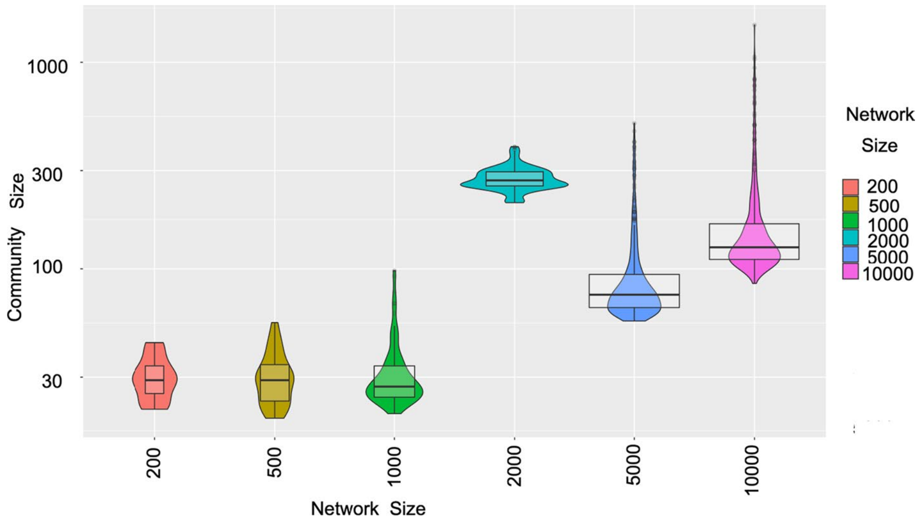


Fig. 8 This figure shows the violin plot of a community size distribution for 10% of the seed nodes on networks from Table 4. The value of the parameters are $\delta = 0.3, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95$, $\alpha = 1$and $\nu = 0.6$

of the topological features of the community structure (Jebabli et al. 2018). In their work, they have investigated the relationship between the topological properties of the community structure and the alternative evaluation measures such as quality metrics and clustering metrics. The authors concluded that the different evaluation measures present different views and therefore must be combined for the evaluation of the community detection algorithms. In another work on the evaluation of community detection methods, the authors have used a topological approach (Orman et al. 2012). They studied and tested, the importance of community-oriented topological measures. Topological measures are used to qualify the communities and evaluate their deviation from the reference structure (Orman et al. 2012). The authors also researched about use of artificially generated realistic networks for evaluation

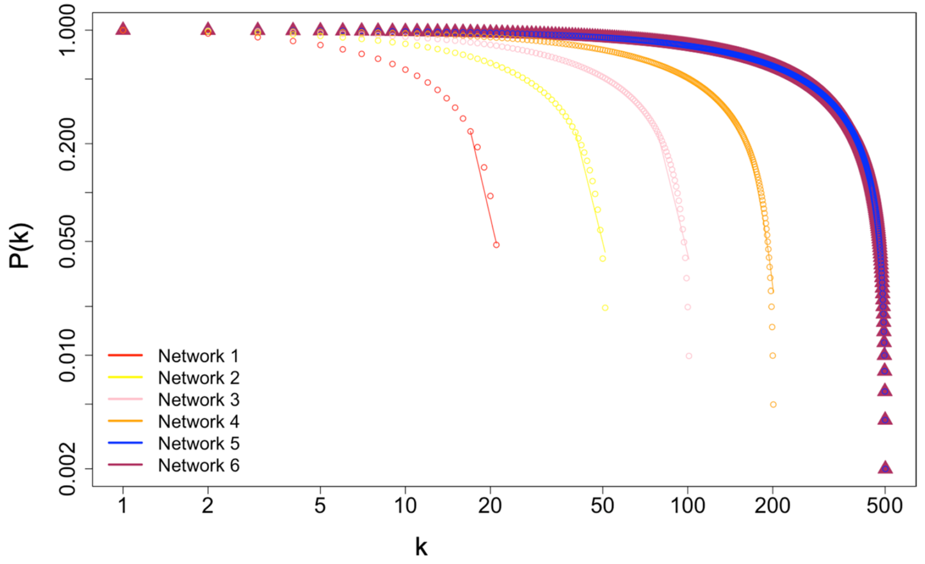


Fig. 9 A power law distribution for the community size of artificial networks with sizes ranging from 200 to 10000. The parameter values are the same as those in the violin plot in Fig. 8

Table 8 Weighted undirected artificial networks generated from the LFR benchmark

Size	Nodes	Edges	Average degree	Maximum degree	Density	Diameter	Transitivity
1	1000	3366	6.7	67	0.00673	6	0.08524
2	2000	6574	6.5	79	0.00328	7	0.08549
3	3000	7907	5.2	177	0.00175	8	0.05006
4	6000	15,397	5.1	495	0.00085	9	0.03210

The mixing parameter $\mu = 0.3, t = 1, t_2 = 1, \mu_w = 0.3$

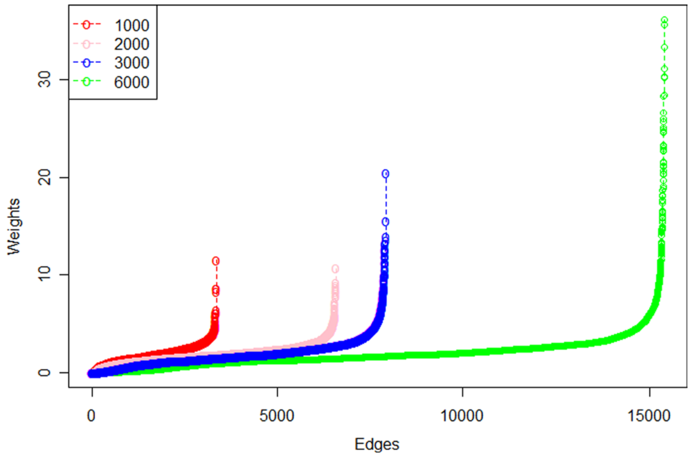


Fig. 10 This figure shows the distribution of weights on edges on artificial weighted networks of size from 1000 to 6000. The value of $\mu_W = 0.3$

purposes. And, they found no equivalence between these approaches. This concluded that high performance does not necessarily guarantee correct topological properties, and vice-versa. They have emphasized using both approaches to perform a complete and accurate assessment. We have used different evaluation measures to compare preference-based methods from the existing literature (Liu et al. 2019).

Normalized information

Normalized mutual information (NMI) is based on information theory is based on entropy. For evaluation of overlapping communities, NMI has a special form.

Normalized information for overlapping communities

Lancichinetti et al. (2009) introduced a version of normalized mutual information for overlapping communities. Let's consider there are two overlapping community detection algorithms that generate partitions X and Y. Each node in the network can belong to more than one cluster. So, for each node, a binary array is stored. The length of the binary array is denoted by the number of the communities detected by the algorithm. The partitions X and Y define community assignments x_i and y_i for each node i in the graph. The entropy for X is

$$H(X) = \sum_{x} P(x) \log P(x), \tag{41}$$

where $P(x)$ is the probability of a node chosen randomly and assigned to the community x. Let's assume that partition X generates k communities. A node i of partition X denoted by x_i^kwill have value 1 if it belongs to k^{th} community i.e. $x_i^k = 1$ and $x_i^k = 0$ otherwise. This relates to the kth entry of x_i to a random variable X_k of probability distribution $P(X_k = 1) = \frac{n_k}{N}$, $P(X_k = 0 = 1 - \frac{n_k}{N})$, where n k is the number of nodes of community k and N is total number of the nodes in the network. Similarly, lets assume that y_i in the l_{th} cluster of Y (Emmons et al. 2016). Further, the joint probability distributions of $P(X_k = 1, Y_l = 1)$, $P(X_k = 0, Y_l = 1)$, $P(X_k = 1, Y_l = 0)$, and $P(X_k = 0, Y_l = 0)$. The additional information of a given X_k is to a given Y_l

$$H(X_k|Y_l) = H(X_k, Y_l) - H(Y_l) \tag{42}$$

To determine X_k from Y, they considered the minimum additional information to determine X_k from all choices of Y_l form L total communities, giving the follows

$$H(X_k|Y) = \min_{l \in 1,2...L} H(X_k|Y_l). \tag{43}$$

Then, normalizing the expression by dividing by $H(X_k)$ and averaging the value of each assignment k, from total K communities, results in the normalized entropy of X conditional to Y. It is denoted as follows

$$H(X|Y)_{norm} = \frac{1}{K} \sum_{k} \frac{H(X_k|Y)}{H(X_k)} \tag{44}$$

Then, they also defined the symmetric conditional entropy as

$$H(Y|X)_{norm} = \frac{1}{K}\sum_{k}\frac{H(Y_l|X)}{H(Y_l)} \tag{45}$$

These conditional entropy in Eqs. (44, 45) are then used to construct the overlapping normalized mutual information between two clustering and is defined as

$$I_{norm}(X,Y) = 1 - \frac{1}{2}[H(X|Y)_{norm} + H(Y|X)_{norm}] \tag{46}$$

Different versions of NMI such as rNMI, cNMI are also available to deal with the problems of finite size effect and reverse finite size effect (McDaid et al. 2011; Lai and Nardini 2016; Zhang 2015). We have used some popular algorithms to compare the preference-based method and computed NMI similarity matrix.

Unweighted and undirected networks

In a network with with an overlapping community it is highly unlikely that all overlapping nodes have the same membership ie that each overlapping node belongs to the same number of communities. In the LFR benchmark approach, the overlapping nodes have a similar membership value. So, we need some other alternatives for the evaluation. We have used different existing overlapping community detection

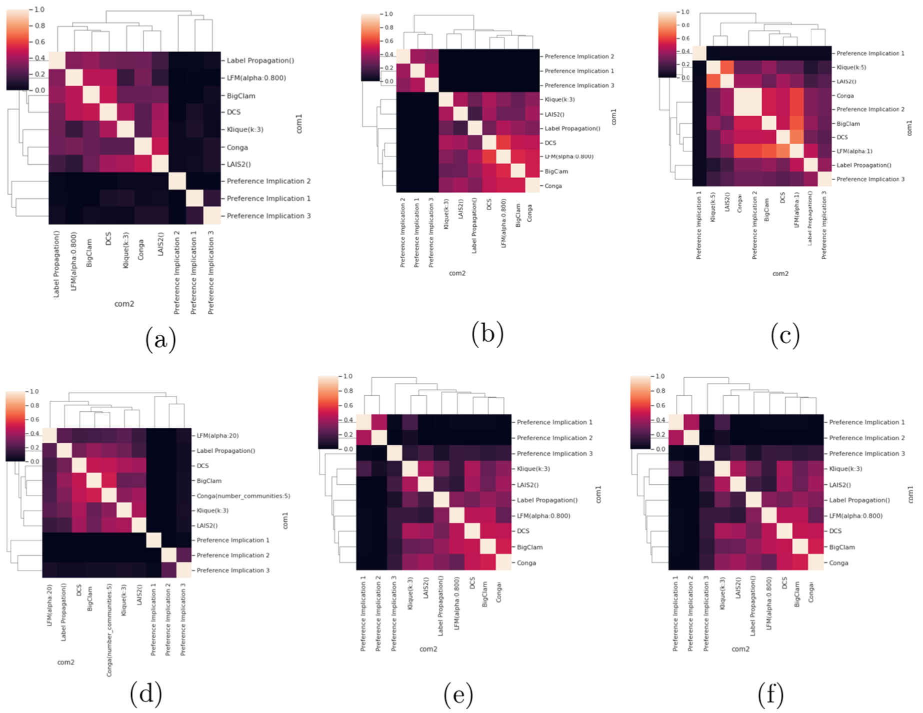


Fig. 11 This figure above is the NMI similarity matrix of the preference-value based method for six real networks. Here, **a** Zachary's Karate Club, **b** Tortoise Social, **c** Facebook Caltech, **d** EU Road, **e** Facebook Nips and **f** Indo-china Webgraph. The three values of $\nu = [065, 0.75, 0.85]$ have been compared to other existing algorithms based on NMI and results are compared through this similarity matrix of clustering scoring

methods to overcome the shortcomings of evaluation strategies. In Fig. 11, the similarity matrix computed on six real networks is shown. We found that for different values of ν, the similarity matrix on some graphs has partitions similar to other detection methods but not always. The values in NMI similarity matrix do not follow a particular behaviour. The results could also be different in other simulations, as we have chosen the seed values randomly. The selection of number of clusters and type of seed nodes is another direction of research in itself, which can be further explored in future. In Fig. 12, we have compared the different ν values on the six real networks. We found the behaviour of ν quite promising to create different clustering. In Fig. 13 the measures are plotted for Zachary karate club for different measures on evaluation. There are three values of ν shown in preference 1-3 with other algorithms. We found similar behaviour on other real networks. We have used F1 score, link modularity and average internal degree as other quality metric. The F1 score is as follows :

$$\text{F1 Score} = 2.\left(\frac{(precision.recall)}{(precision + recall)}\right) \tag{47}$$

The preference-based method has a higher F1 score compared with Big-clam and K-clique even for a smaller value of ν, as shown in Fig. 13. However, we also observed a higher value of F1 score for other combinations of α and ν.

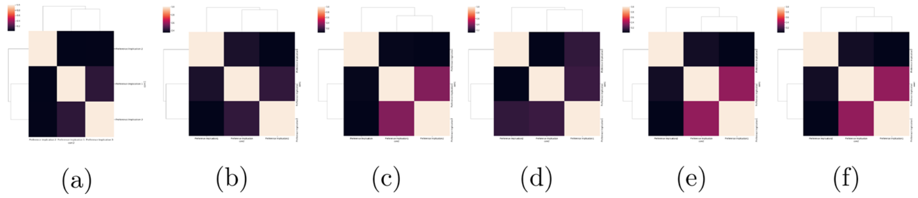
The average internal degree is similar in all the clustering algorithms.

Evaluation on weighted undirected networks

The communities generated by the preference-based networks are analysed by comparing with the ground truth data available from synthetic networks. Some of the existing overlapping community detection methods are used for comparison. The similarity matrix of NMI scores is plotted in Fig. 14.

Comparison using ground truth data

The ground truth data for communities from synthetic networks are used for comparison with our approach. The NMI scores of the preference-based method with the ground truth data were in the range of [0.2, 0.5]. These scores seem to show different behaviour by changing the number of communities. However, we could not establish any correlation between the NMI scores and the number of communities (Fig. 15).



(a) (b) (c) (d) (e) (f)

Fig. 12 This figure shows the NMI similarity matrix for $\nu = [065, 0.75, 0.85]$ for preference-based method for different clustering scores. Also, here **a** Zachary's Karate Club, **b** Tortoise Social, **c** Facebook Caltech, **d** EU Road, **e** Facebook Nips and **f** Indo-china Webgraph

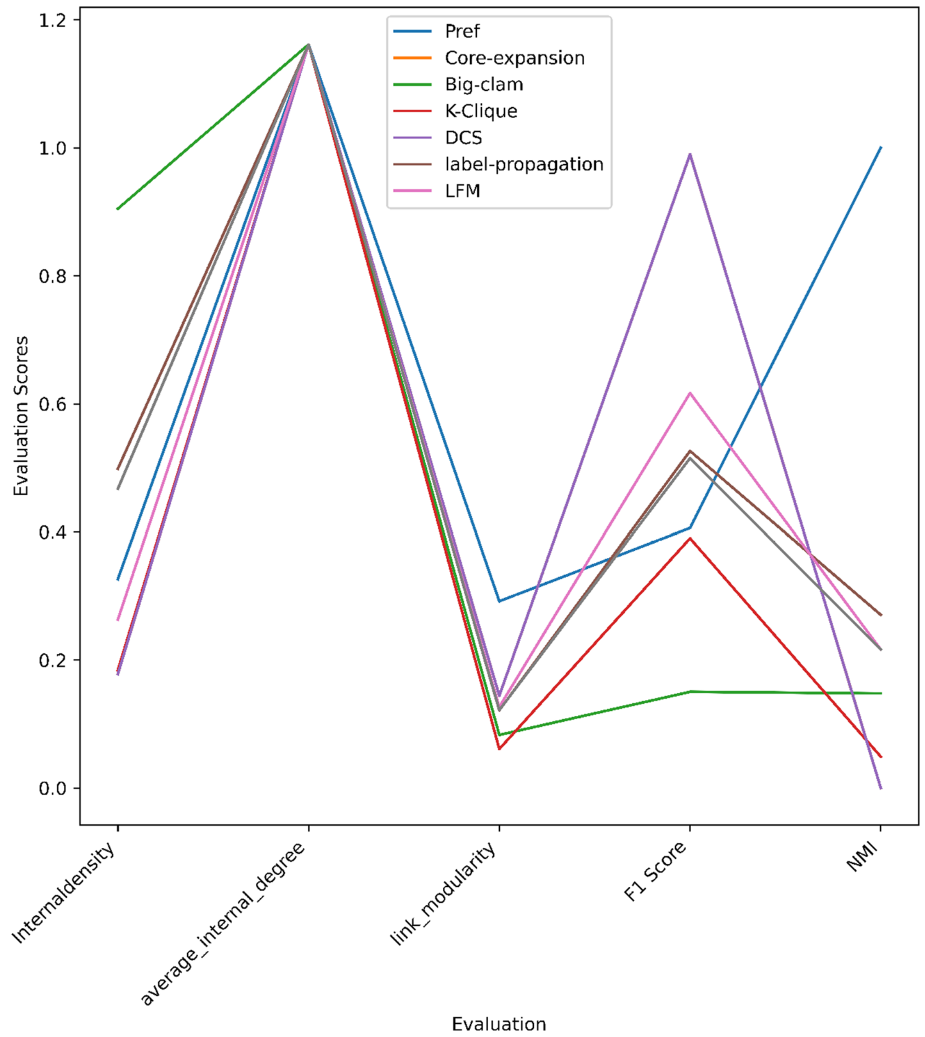


Fig. 13 Different measures such as internal degree, link modularity, F1 score, NMI are plotted for Zachary karate club using community detection algorithms. We have used Core-expansion, Big-clam, K-clique, DCS, label-propagation and LFM to show further comparison

Conclusion and discussion

The theory of preference modeling of a community introduces continuous-valued logic into graph theory. The existing method of community detection and other potential problems on graphs can be handled by the continuous-valued logic of other methods and operators. In particular, for $np - hard$ problems like partitions and community detection, the unsupervised learning is keeping up with the enormous data and resources available at the user's end. Network clustering in general is different characterization from the other data clustering methods due to the presence of relationships among the nodes. Overlaps in partitions are the soft decision boundaries that show the fuzziness in the classification of nodes. The choice of community strength for the community is an important aspect of the algorithm. This aspect is incorporated into preference-based method, as the user-controlled parameter. The community strength is not a self-learned feature of the proposed method. This feature of preference-based approach is crucial for

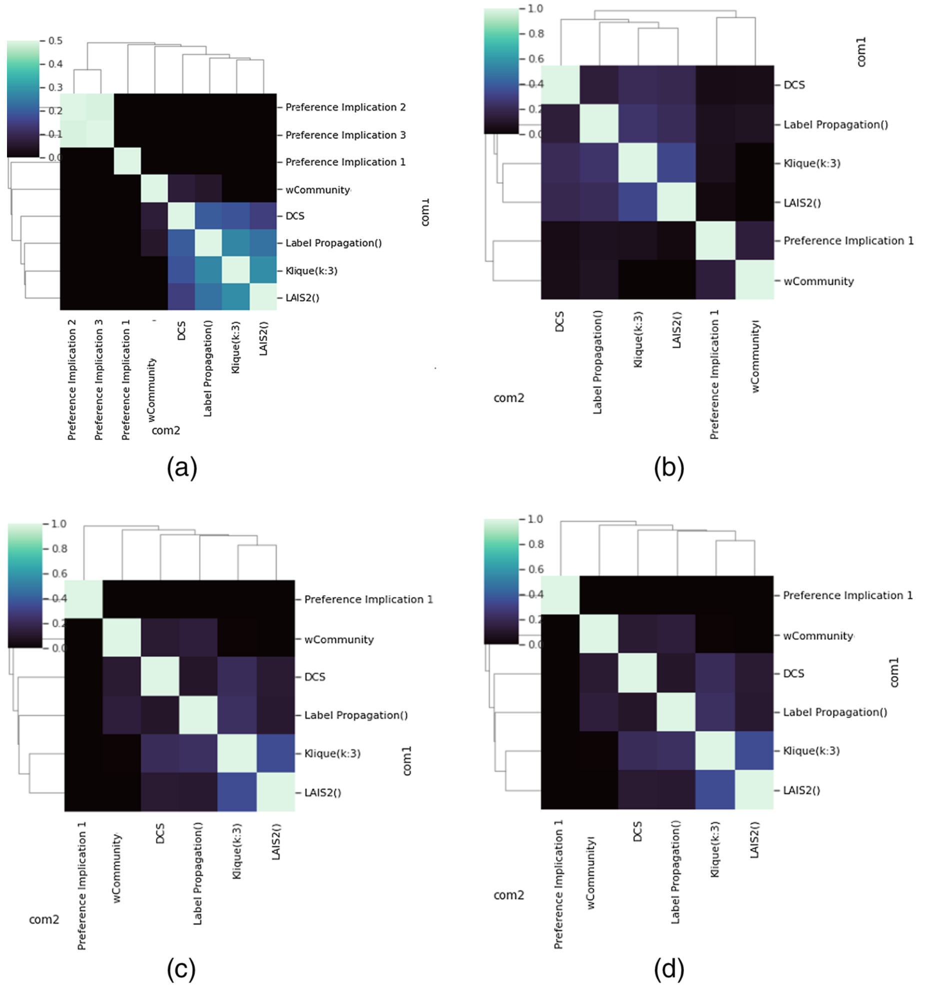


Fig. 14 The figure above is the NMI similarity matrix of the preference-value based method for four artificial weighted networks. The preference-based method with a value of $\nu = 0.65$ and $\alpha = 0.75$ is compared with other existing algorithms using an NMI similarity matrix. We plot the results for networks of sizes **a** 1000, **b** 2000, **c** 3000, **d** 6000 respectively

manually tuning the community strength in different situations for community identification. A user has the flexibility of detecting communities of different strengths depending on the application area. In data-mining this feature can be useful for generating a threshold-based representation of any group. For a network, many partitions can exist for different strengths of communities. The behavior of the preference implication can be comprehended from the sigmoid function graph. We conducted our tests on artificial and real networks collected from standard repositories and benchmarks. The statistical characteristics of networks on which the tests were performed were analyzed to discover the property of these networks. The best results of the proposed approach are not seen on smaller networks. The approach has promising results on artificial and real graph networks with a power-law distribution of community sizes. However, the choice of initial seed nodes is a crucial criterion for the creation of communities. We chose the random-based approach to select the initial seed nodes. The randomly selected nodes

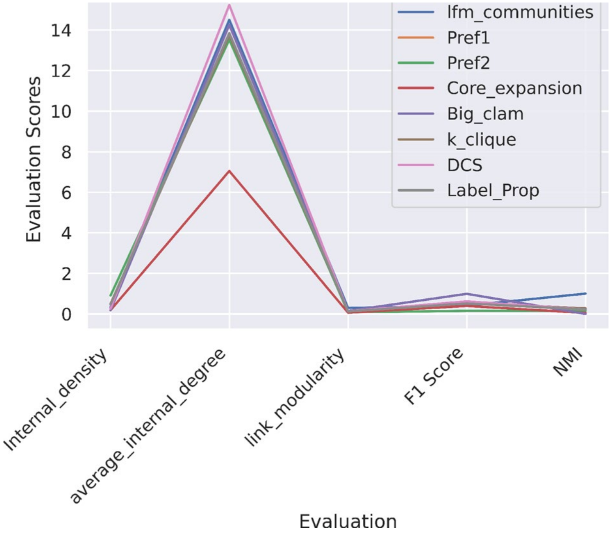


Fig. 15 Different measures such as internal degree, link modularity, F1 score, NMI are plotted for undirected weighted network of UK faculty using community detection algorithms (Csardi and Nepusz 2006). We have used Core-expansion, Big-clam, K-clique, DCS, label-propagation and LFM to show further comparison

were not in the neighborhood of each other. The proposed preference-based method is a novel approach for controlling the overlapping structures and strength of the communities. The parameters δ and α can be used to control the strength of the communities and overlapping regions. The outliers observed were less than 15% and they were handled with a feedback loop by replacing the faulty seed nodes with new seed nodes. In large networks, the search complexity of community detection increases with stopping criteria that depend on the strength of the community. This situation can be dealt in two ways. In the first solution to the problem, the delta parameter can be a high value for new nodes at the later stage of community creation. This implies the initial motivation of creating more real-like communities. In the second solution keeping a membership for all the nodes (i.e. a fixed delta) and defining the maximum size of a community in the network can be used as one of the criteria to terminate the search.

Abbreviations

LFM: Lancichinetti Fortunato method; GN: Girvan and Newman; CESNA: Communities from edge structure and node attributes; BigCLAM: Cluster affiliation model for big networks; DCS: Divide and conquer strategy; CONGA: Cluster-overlap Newman Girvan algorithm; CONGO: CONGA optimized; NMI: Normalized mutual information; wCommunity: Overlapping communities of weighted networks via a local algorithm; rNMI: Relative normalized mutual information; cNMI: Corrected normalized mutual information.

Acknowledgements

We would like to thank the anonymous reviewers who gave us insightful and meaningful suggestions for further analysis and evaluation.

Authors' contributions

JD and SD furnished the idea, method and general direction of the study. JD also conceptualised the idea of the theory used in the paper. SD has contributed to the implementation and result generation for analysis purposes. SD contributed to the data cleaning and manipulation and the analysis, under the supervision of JD. SD drafted the manuscript and JD has read, edited and assessed the manuscript. Both authors read and approved the final manuscript.

Funding

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program (MILAB).

Availability of data and materials

The data-sets and analysis code used in the current research is available on the github repository. Further details about the results generation is also available at https://github.com/sakshidhama/PrefCommunity.git.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Algorithms and Artificial Intelligence, Institute of Informatics, University of Szeged, Árpád Tér 2, Szeged 6720, Hungary. [2]Hungarian Academy of Sciences, Budapest, Széchenyi István tér 9, Budapest 1051, Hungary.

Received: 11 March 2021 Accepted: 28 August 2021
Published online: 02 November 2021

References

Barabási A-L et al (2016) Network science. Cambridge University Press, Cambridge

Baumes J, Goldberg MK, Krishnamoorthy MS, Magdon-Ismail M, Preston N (2005) Finding communities by clustering a graph into overlapping subgraphs. IADIS AC 5:97–104

Bhowmick S, Srinivasan S (2013) A template for parallelizing the louvain method for modularity maximization. In: Dynamics on and of complex networks, vol 2. Springer, pp 111–124

Carter R, Park K (1993) How good are genetic algorithms at finding large cliques: an experimental study. Technical report, Citeseer

Cherifi H, Palla G, Szymanski BK, Lu X (2019) On community structure in complex networks: challenges and opportunities. Appl Netw Sci 4(1):1–35

Csardi G (2015) Igraphdata: a collection of network data sets for the igraph package. R package version 1.0.1

Csardi G, Nepusz T et al (2006) The igraph software package for complex network research. Int J Complex Syst 1695(5):1–9

Csiszár O, Csiszár G, Dombi J (2020) How to implement mcdm tools and continuous logic into neural computation? Towards better interpretability of neural networks. Knowl Based Syst 210:106530

da Fonseca Vieira V, Xavier CR, Evsukoff AG (2020) A comparative study of overlapping community detection methods from the perspective of the structural properties. Appl Netw Sci 5(1):1–42

Dao V-L, Bothorel C, Lenca P (2018) Estimating the similarity of community detection methods based on cluster size distribution. In: International conference on complex networks and their applications. Springer, pp 183–194

De Meo P, Ferrara E, Fiumara G, Provetti A (2011) Generalized louvain method for community detection in large networks. In: 2011 11th international conference on intelligent systems design and applications. IEEE, pp 88–93

Derényi I, Palla G, Vicsek T (2005) Clique percolation in random networks. Phys Rev Lett 94(16):160202

Dombi J (1982) Basic concepts for a theory of evaluation: the aggregative operator. Eur J Oper Res 10(3):282–293

Dombi J, Baczyński M (2019) General characterization of implication's distributivity properties: the preference implication. IEEE Trans Fuzzy Syst PP, 1–1

Dombi J, Dhama S (2019) Preference relation and community detection. In: 2019 IEEE 19th international symposium on computational intelligence and informatics and 7th IEEE international conference on recent achievements in mechatronics, automation, computer sciences and robotics (CINTI-MACRo). pp 33–36

Dombi J, Dhama S (2020) Using preference intensity for detecting network communities. In: International conference on complex networks and their applications. Springer, pp 137–151

Dombi J, Jónás T (2018) Approximations to the normal probability distribution function using operators of continuous-valued logic. Acta Cybern 23(3):829–852

Dombi J, Jónás T (2020) Advances in the theory of probabilistic and fuzzy data scientific methods with applications. Springer, Berlin

Dombi J, Gera Z, Vincze N (2006) On preferences related to aggregative operators and their transitivity. LINZ 56

Dugué N, Perez A (2015) Directed louvain: maximizing modularity in directed networks. PhD thesis, Université d'Orléans

Emmons S, Kobourov S, Gallant M, Börner K (2016) Analysis of network clustering algorithms and cluster quality metrics at scale. PLoS ONE 11(7):0159161

Friedkin N (1980) A test of structural features of granovetters strength of weak ties theory. Soc Netw 2(4):411–422

Gach O, Hao J-K (2013) Improving the louvain algorithm for community detection with modularity maximization. In: International conference on artificial evolution (evolution artificielle). Springer, pp 145–156

Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826

Good BH, De Montjoye Y-A, Clauset A (2010) Performance of modularity maximization in practical contexts. Phys Rev E 81(4):046106

Gregory S (2011) Fuzzy overlapping communities in networks. J Stat Mech Theory Exp 2011(02):02017

Gulbahce N, Lehmann S (2008) The art of community detection. BioEssays 30(10):934–938

Jebabli M, Cherifi H, Cherifi C, Hamouda A (2018) Community detection algorithm evaluation with ground-truth data. Physica A 492:651–706

Kelley S (2009) The existence and discovery of overlapping communities in large-scale networks. PhD thesis, Rensselaer Polytechnic Institute

Kim J, Wilhelm T (2008) What is a complex graph? Physica A 387(11):2637–2652

Kim Y, Son S-W, Jeong H (2010) Finding communities in directed networks. Phys Rev E 81(1):016103
Lai D, Nardini C (2016) A corrected normalized mutual information for performance evaluation of community detection. J Stat Mech Theory Exp 2016(9):093403
Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E 78(4):046110
Lancichinetti A, Fortunato S, Kertész J (2009) Detecting the overlapping and hierarchical community structure in complex networks. New J Phys 11(3):033015
Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S (2011) Finding statistically significant communities in networks. PLoS ONE 6(4):e18961
Leicht EA, Newman ME (2008) Community structure in directed networks. Phys Rev Lett 100(11):118703
Levin VI (2007) Basic concepts of continuous logic. Stud Logic Grammar Rhetor 11(24):67–84
Li H-J, Bu Z, Li A, Liu Z, Shi Y (2016) Fast and accurate mining the community structure: integrating center locating and membership optimization. IEEE Trans Knowl Data Eng 28(9):2349–2362
Li L, He X, Yan G (2018) Improved louvain method for directed networks. In: International conference on intelligent information processing. Springer, pp 192–203
Liu C, Chamberlain BP (2017) Speeding up bigclam implementation on snap. arXiv preprint arXiv:1712.01209
Liu X, Cheng H-M, Zhang Z-Y (2019) Evaluation of community detection methods. IEEE Trans Knowl Data Eng 32(9):1736–1746
Malliaros FD, Vazirgiannis M (2013) Clustering and community detection in directed networks: a survey. Phys Rep 533(4):95–142
McDaid AF, Greene D, Hurley N (2011) Normalized mutual information to evaluate overlapping community finding algorithms. arXiv preprint arXiv:1110.2515
Nepusz T, Petróczi A, Négyessy L, Bazsó F (2008) Fuzzy communities and the concept of bridgeness in complex networks. Phys Rev E 77(1):016107
Newman ME (2003) The structure and function of complex networks. SIAM Rev 45(2):167–256
Orman GK, Labatut V, Cherifi H (2012) Comparative evaluation of community detection algorithms: a topological approach. J Stat Mech Theory Exp 2012(08):08001
Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab
Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043):814
Peixoto TP, Rosvall M (2017) Modelling sequences and temporal networks with dynamic community structures. Nat Commun 8(1):1–12
Que X, Checconi F, Petrini F, Gunnels JA (2015) Scalable community detection with the louvain algorithm. In: 2015 IEEE international parallel and distributed processing symposium. IEEE, pp 28–37
Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D (2004) Defining and identifying communities in networks, vol 101. National Academy of Sciences, Berlin, pp 2658–2663
Rossi RA, Ahmed NK The network data repository with interactive graph analytics and visualization. In: AAAI. http://networkrepository.com
Schaub MT, Delvenne J-C, Rosvall M, Lambiotte R (2017) The many facets of community detection in complex networks. Appl Netw Sci 2(1):4
Traud AL, Kelsic ED, Mucha PJ, Porter MA (2011) Comparing community structure to characteristics in online collegiate social networks. SIAM Rev 53(3):526–543
Traud AL, Mucha PJ, Porter MA (2012) Social structure of Facebook networks. Physica A 391(16):4165–4180
Vadon V, Komjáthy J, van der Hofstad R (2019) A new model for overlapping communities with arbitrary internal structure. Appl Netw Sci 4(1):1–19
Vanhems P, Barrat A, Cattuto C, Pinton J-F, Khanafer N, Régis C, Kim B-A, Comte B, Voirin N (2013) Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. PLoS ONE 8(9):e73970
Xie J, Kelley S, Szymanski BK (2013) Overlapping community detection in networks: the state-of-the-art and comparative study. ACM Comput Surv (CSUR) 45(4):1–35
Yang J, Leskovec J (2013) Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the sixth ACM international conference on web search and data mining. pp 587–596
Zhang P (2015) Evaluating accuracy of community detection using the relative normalized mutual information. J Stat Mech Theory Exp 2015(11):11006
Zhuang D, Chang MJ, Li M (2019) Dynamo: dynamic community detection by incrementally maximizing modularity. IEEE Trans Knowl Data Eng

Publisher's Note