



# Community detection in weighted networks using probabilistic generative model

Hossein Hajibabaei<sup>1</sup> · Vahid Seydi<sup>1,2</sup> · Abbas Koochari<sup>1</sup>

Received: 14 March 2022 / Revised: 14 August 2022 / Accepted: 26 August 2022 /  
Published online: 26 September 2022  
© The Author(s) 2022

## Abstract

Community detection in networks is a useful tool for detecting the behavioral and inclinations of users to a specific topic or title. Weighted, unweighted, directed, and undirected networks can all be used for detecting communities depending on the network structure and content. The proposed model framework for community detection is based on weighted networks. We use two important and effective concepts in graph analysis. The structural density between nodes is the first concept, and the second is the weight of edges between nodes. The proposed model advantage is using a probabilistic generative model that estimates the latent parameters of the probabilistic model and detecting the community based on the probability of the presence or absence of weighted edge. The output of the proposed model is the intensity of belonging each weighted node to the communities. A relationship between the observation of a pair of nodes in multiple communities and the probability of an edge with a high weight between them, is one of the important outputs that interpret the detected communities by finding relevancy between membership of nodes to communities and edge weight. Experiments are performed on real-world weighted networks and synthetic weighted networks to evaluate the performance and accuracy of the proposed algorithm. The results will show that the proposed algorithm is more density and accurate than other algorithms in weighted community detection.

**Keywords** Community detection · Weighted graph · Complex networks · Matrix factorization · Probabilistic model

---

✉ Vahid Seydi  
V.seydi@bangor.ac.uk

Hossein Hajibabaei  
h.hajibabaei@srbiau.ac.ir

Abbas Koochari  
koochari@srbiau.ac.ir

<sup>1</sup> Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

<sup>2</sup> Centre for Applied Marine Sciences, School of Ocean Sciences, Bangor University, Menai Bridge, UK

## 1 Introduction

Network analysis is a critical tool for accurate network component monitoring and detection. A complex network (Sia et al., 2019) can be a network of friendly communication in social networks (McAuley & Leskovec, 2012), a network of cells in biology (Ahn et al., 2010), or a network of scientific researchers engaged in collaborative scientific research (Girvan & Newman, 2002). To put it another way, it can be any grid with nodes and edges that can be represented as a graph. For example, if social network users, author of an article, a cell of a protein is considered a node in various networks, then friendship between social network users, shared articles between the two authors, and the relationship between cells can all be considered edges.

Community detection is one of the usable approaches and techniques for analyzing complex networks. Detection of the community means Finding a subgraph of a graph whose connections between its nodes are stronger and denser than the connections between the graph's other nodes (Blondel et al., 2008). Communities can represent a way of thinking, a category, an interest, a topic orientation, and so on. Communities can be separate, and they can also be shared, which are called overlap communities.

The proposed model uses a probabilistic model to detect the community in weighted networks without direction. To get acquainted with the proposed model approach, we will have an overview of the community detection approaches in complex networks.

Modularity-based methods are one of the most well-known community detection approaches. The Louvain method (Blondel et al., 2008) is one of them, and it is often used in weighted graphs. This method presents a simple and fast algorithm for detecting separate communities in complex weighted and unweighted networks. The greedy approach (Sánchez-Oro & Duarte, 2018) is used in this algorithm to cluster graph nodes and maximize modularity. However, the Louvain algorithm has flaws, which the Leiden algorithm addresses (Traag et al., 2019). The goal is to accelerate the local movement and transfer nodes to random neighbors while also modifying the community discovered during the iteration cycle.

Label propagation methods, which were first introduced in Raghavan et al. (2007), are another type of community detection method in weighted networks. These approaches use a local network's architecture and structural features for community detection. To address the flaws in the label propagation process, the COPRA (Gregory, 2010) and Speaker-listener Label Propagation Algorithm (SLPA) (Xie et al., 2011) were proposed, in which each node has only one label that is modified repeatedly using the maximum label in its neighborhood. Separate communities can be identified after the algorithm converges. In Chen et al. (2010) presents an algorithm based on the basic concepts outlined in the label propagation methods in the following section. Its primary strategy is to create an initial partial community based on a node with the highest node strength, then add strong nodes to expand the partial community and increase modularity.

In contrast to the methods mentioned above, which use traditional approaches to detect communities in weighted graphs, newer methods estimate the probabilistic model to detect communities. The model parameters are estimated using this method, which generates a potential generative model from the network graph. Community detection in unweighted networks is addressed using methods (Yang & Leskovec, 2012; 2013). Most community detection methods in complex networks have relied on undirected and unweighted graphs up until now (Sánchez-Oro & Duarte, 2018). Many of these techniques necessitate specific assumptions and constraints. In other words, the accuracy of these approaches is limited to

detailed descriptions of the problem conditions, and they lack comprehensiveness in terms of accuracy, scalability, and data compatibility (Kumar et al., 2021).

Consequently, today due to the immense communication space of networks, the presence of big data, providing accurate analysis, and the importance of accuracy of performance, more features such as edge weight can be used to achieve this aim, that the advantage of the proposed model is covering these items. The proposed model framework for community detection is based on using probabilistic models on weighted networks. For this purpose, the proposed model develops the idea of Yang & Leskovec (2012; 2013) for weighted networks, in which the intensity of each node's dependence on each community is considered the model's latent parameter. The sum of these interdependencies is compiled into a matrix with nodes in the rows and communities in the columns. The graph adjacency matrix will be factorized into this matrix, which will be non-negative. Nodes are the components of the graph adjacency matrix; graph nodes can be found in both the row and column of this matrix. The number 1 will be inserted in the matrix if there is an edge between two nodes; otherwise, 0 will be used. Since the values of the factorized matrix, which is the intensity of each node's dependence on each community (an unknown parameter in the model), must have non-negative values, the proposed model employs the non-negative matrix factorization (NMF) approach (Lee & Seung, 1999; Hsieh & Dhillon, 2011). This is because it is assumed that a node does not belong to a community. In this case, the cell in the factorized matrix will have a value of zero or belong to a low or high-intensity community, which should be a positive value. The proposed model's goal is to estimate the unknown and hidden values of model parameters in such a way that the best cluster of communities is created based on a combination of a node belonging to the community and edge weight. The proposed model will be compared quantitatively and qualitatively with other community detection methods in weighted networks, to determine its accuracy. Modularity, conductance, NMI, and F1Score are well-known evaluation criteria in community density quality and accuracy measurement, were used in this study.

## 2 Related works

Detecting communities in complex networks is a complex problem that has been approached from various perspectives. The form of problem assumptions and the input graph has a significant influence on community detection methods. Community detection has gotten much attention because of these variations. Due to the importance of community detection in network analysis, various categories of approaches have been presented in authoritative sources (Fortunato & Hric, 2016; Javed et al., 2018) by researchers of different disciplines that we review some of these studies in this section based on the proposed model's concepts and evaluation criteria.

### 2.1 Traditional clustering approaches

Over the years various algorithms based on clustering were proposed, which are called traditional approaches, hierarchical clustering, spectral clustering, partitional clustering and graph partitioning are some well-known traditional algorithms which introduced important concepts of community detection and paved path for future advancements (Javed et al., 2018).

Hierarchical clustering techniques are appropriate for the networks having hierarchical structures. These algorithms develop a binary tree that merges similar clusters based on the similarity between vertices. In hierarchical clustering, there is no need to specify the number of clusters beforehand as compared to partitional clustering. Two popular approaches of hierarchical clustering are agglomerative and divisive algorithms (Fortunato, 2010). In the agglomerative algorithms, clusters are combined iteratively (bottom-up) if they have a high similarity index (or similarity score) (Maqbool & Babri, 2004). The maximal clique and hierarchical link-based clustering are examples of agglomerative hierarchical clustering algorithms (Shen et al., 2009). Divisive algorithms are a top-down approach that starts with all nodes in one cluster, and then partitioning using a flat clustering algorithm is executed that removes the edges which connect low similarity vertices and the highest edge betweenness (Morvan et al., 2017). This process continues recursively until communities are detected by cutting edges of the graph. Girvan and Newman proposed algorithms for community detection in the form of divisive algorithms (Girvan & Newman, 2002; Newman & Girvan, 2004; Newman, 2006). The newer work of hierarchical clustering utilizes leading trees. The leading tree is an efficient granule calculation model (GrC) for hierarchical clustering and requires two elements: the distance between the grains and the calculated density in Euclidean space (Fu et al., 2021). For non-Euclidean network data, vertices must be embedded in Euclidean space before calculating density.

Spectral clustering divides a graph into clusters using the eigenvectors of the input data matrix. It converts a given set of nodes into a set of nodes in multidimensional space, whose coordinates are the eigenvector components. This transformation reveals implicit properties of the initial dataset (Javed et al., 2018). In spectral clustering algorithms, the first contribution was submitted by Donath and Hoffman (Donath & Hoffman, 2003), in which they used eigenvectors of adjacency matrix and eigenvalues of similarity matrix for graph partitioning.

Partitional clustering divides the graph nodes into  $k$  clusters with maximizing or minimizing a loss function based on the distance between them. Minimum  $K$ -clustering,  $K$ -clustering sum,  $K$ -center,  $K$ -median, and  $K$ -means are some of the partitional clustering approaches. Among these methods, the  $K$ -means (MacQueen, 1967) is a famous algorithm for partitional clustering which minimizes the squared loss function of intra-cluster distance. In developing the  $K$ -means, Lloyd's algorithm (Celebi, 2014) helps to speed up the minimization of the loss function. Fuzzy  $K$ -means (FKM) clustering (Nie et al., 2020) assigns each data point to multiple clusters with some degree of certainty.  $U$ - $k$ -means (Sinaga & Yang, 2020) proposed a novel unsupervised  $K$ -means clustering algorithm with automatically finding an optimal number of clusters that solved the  $K$ -means problem with initializations with a necessary number of clusters a priori. Recently, novel methods have used combined partitional clustering and network embedding techniques to detect communities in complex networks. The network embedding technique (Kumar et al., 2021) represents the nodes of the input graph into vector space and preserves their inherent and topological features and can contribute significantly to various applications in network analysis. In Kumar et al. (2021) firstly, nodes of the graph are embedded in the feature space of dimensions, and then a low-rank approximation is applied to avoid the results from being affected by noise or outliers. Further,  $K$ -means clustering is employed to find the centroids of the clusters in the network and followed by a gravitational search algorithm to improve the results of centroids of clusters.

Graph partitioning divides the vertices into  $g$  groups whose size is predefined to get the minimum number of links between identified groups. The number of vertices running among the clusters is termed as cut size. If one does not give the number of clusters in

advance and inflict a partition with minimum cut size, a trivial solution will be the output (Javed et al., 2018).

## 2.2 Modularity optimization approaches

Several methods focused on modularity optimization are moving towards optimal community detection in weighted networks. In general, the modularity optimization approaches lead to a classification of more densely connected nodes than others, to construct related classes of nodes with the similar structural characteristics. The Louvain method (Blondel et al., 2008) is one of the most critical approaches in this area. This algorithm uses a greedy approach to form communities and optimize them, focusing on modularity maximization at each stage. This process is divided into two parts:

- Small communities are formed initially due to local optimization
- Core communities are formed as a result of merging small communities with the ability to create larger communities, and the method continues to evolve

These two steps are repeated until the communities remain unchanged, and the modularity of the method is maximized. The random selection of a neighboring node in the first phase is one of this method's drawbacks. Comprehensibility of algorithm phases, simplicity, and speed are also among features of Louvain. To improve and solve Louvain's problems, the Leiden algorithm was proposed (Traag et al., 2019). Because the Louvain algorithm is greedy, this method may combine incompatible communities. In the worst-case scenario, it could even cut off a newly discovered real community's connection and split it into two or more parts or move nodes between communities, while that node may serve as a strong link between the various parts of its previous community. This unpleasant event will be exacerbated by the algorithm's repeated execution to reach the optimal point, to the point where it may appear that the algorithm's qualitative performance based on modularity will improve. However, the desired result will not be achieved in practice. This is due to the randomness with which the neighboring node is chosen in the first phase of this method. To address this issue, Leiden proposes accelerating local movement and transferring nodes to random neighbors, as well as altering the community discovered during the iteration cycle. There are three main steps in the Leiden algorithm:

- Local clustering of nodes
- Modification and improvement of clusters
- Network integration and community detection based on improved clusters

## 2.3 Label propagation approaches

Another class of weighted network community detection methods is based on the label propagation method, which was first introduced in Raghavan et al. (2007). This algorithm extracts communities by detecting strong label connections by propagating each node's label. Label propagation methods are typically used to detect communities based on network structure. In these cases, optimizing a predefined objective function or utilizing supplementary and content data are less effective in detecting communities. Each node is given a distinct label in Raghavan et al. (2007). The key node for developing and completing a community is then chosen as the node with the most neighbors. Due to the random

approach in choosing the neighboring node, this process is correlated with issues such as community instability discovered in multiple runs and detection of heterogeneous communities. To fix the shortcomings of the label propagation process, the COPRA method for directed and weighted networks was implemented in Gregory (2010), followed by the Speaker-listener Label Propagation Algorithm (SLPA) to extract higher quality communities in Xie et al. (2011). Each node in Xie et al. (2011) has only one label, modified repeatedly using the maximum label in its neighborhood. Separate communities are detected after the algorithm has been converging and consolidating. Method (Chen et al., 2010) is another label propagation algorithm that is discussed in the following sections. It calculates the concept of power per node first, then creates an initial community centered on the node with the most power and its node neighbors. The initial community is then expanded by repeatedly adding nodes with maximum modularity to the community and discovering overlap nodes.

## 2.4 Model-based approaches

The probabilistic model estimation approach is used to solve the problem of community detection in another category of methods (Yang & Leskovec, 2012; 2013; Yang et al., 2013). The primary axes of these methods are estimating a probability distribution function and use the intensity of nodes affiliation to communities as a parameter in the probabilistic model. Following the research process (Yang & Leskovec, 2012), the Bigclam algorithm (Yang & Leskovec, 2013) is presented, which estimates the probability of the presence or absence of edges in the graph, using non-negative matrix factorization (NMF) methods (Lee & Seung, 1999; Hsieh & Dhillon, 2011). The NMF method divides the adjacency matrix into two matrices whose rows are nodes and columns are communities, and unknown probabilistic model parameters are estimated using maximum likelihood estimation (MLE) (Myung, 2003) on the factorized matrix. However, the assumptions of the Bigclam method have some limitations. One of the most significant disadvantages is that this method is only presented in unweighted graphs and does not provide a solution for graphs with the weighted edge. Another limitation is that Bigclam only works structurally in community detection because it ignores the content of the node and the weight of the edge between two nodes, focusing instead on the relationship and density of communication with neighboring nodes. This has hampered the method's ability to work with various datasets. The CESNA method (Yang et al., 2013), which combines node content and graph structure, is presented as part of the Bigclam research process. However, the role of edge weight between two nodes in community detection is not mentioned in CESNA, and there is almost no solution for weighted graphs. Another problem with CESNA is that in probabilistic calculations, node properties are assumed to be limited.

## 3 Proposed model frameworks

Simply paying attention to the structure or content of the nodes will not provide the required accuracy for community detection in complex networks; instead, a closer look at the details of the problem and the use of the inherent features of the initial graph is required. One of these features is the weight of the edge between the two nodes, which due to the difficulty of implementation and limitation of results on the weighted dataset, has received less attention in community detection methods. On the other hand,

most of the community detection approaches in weighted networks, focus on traditional methods such as hierarchical clustering (Liu et al., 2014; Li et al., 2019), segmentation (Qi et al., 2012), and modularity maximization (Chen et al., 2010; Beckett, 2016; Nicolini et al., 2017). In this paper, a weighted community detection algorithm based on a probabilistic generative model is presented. The proposed method is referred to as WSCD (Structured Community Detection in Weighted Networks). It will be demonstrated in this approach that using edge weight for community detection performs better than unweighted methods. The edge weight indicates the content relationship between the nodes in a way that the previous methods do not, and by which community detection in the graph can be done with acceptable accuracy and quality. In comparison to previous community detection methods, the model presented in WSCD takes a different approach. This method presents characteristics that have demonstrated correct performance in experiment results. Characteristics that were less taken into account in previous approaches. Among these are:

- Using the probabilistic weighted model and matrix factorization method
- Heeding node connection density in detecting communities and combining it with edge weight
- Conceptual connection of community detection with the probability presence or absence of weighted edge
- Using evolutionary methods and MLE in calculations

The proposed model is a probabilistic generative model that will be presented on a weighted graph and is based on the following assumptions:

- A weighted edge is possible between pairs of nodes in a community
- When a pair of nodes are observed in multiple communities, the possibility of a high weight edge between them is increased
- Communities can be overlap; overlap communities have higher weight density

The components of the proposed model are described below. We assume a weighted network  $G(N,E,W)$  in the WSCD model, where  $N,E$ , and  $W$  respectively represent nodes, edges, and edge weight between two nodes. We assign a nonnegative value  $I_{uc}$  between node  $u \in N$  and community  $c \in C$ . ( $I_{uc} = 0$  means  $u$  does not belong to  $c$ .) Therefore the dependency strength of each node to each community is represented in the  $I$  matrix. In WSCD the probability of the presence or absence of an edge between two nodes  $u$  and  $v$  in community  $c$  is depending on the value of  $I$  and the edge weight between  $u$  and  $v$ , Each community  $c$  generates edges independently. In particular, we assumed that two nodes  $u$  and  $v$  are connected by considering the following probability,

$$P_c(u, v) = 1 - \exp(-I_{uc} \cdot I_{vc}^T \cdot B_{uv}) \quad (1)$$

In this case, the weight of the edge between the two nodes ( $B_{uv}$ ) would have a direct impact on determining the likelihood of the presence or absence of an edge between two nodes in addition to the intensity of each node's belonging to the community ( $I_{uc}$ ) and the likelihood of a weighted edge between two nodes and their membership in common communities grows as a result of this process. In other words, nodes that belong to multiple communities are more likely to share edges with strong weight. According to the generative probabilistic procedure between any pair of nodes, each pair of nodes are independently distributed

by Bernoulli distribution. Therefore, each component of the adjacency matrix is generated based on the following generative approach.

$$P(u, v) = 1 - \exp(-I_u \cdot I_v^T \cdot B_{uv})$$

$$A_{uv} \sim \text{Bernoulli}(P_{uv}) \tag{2}$$

In some real-world datasets, the values of the weighted edges are natural numbers (greater than 1). These values must be normalized between 0 and 1 to calculate the probability in (2). Hence we consider ( $W_{uv}$ ) as an abnormal weight and ( $B_{uv}$ ) as the weight of the normalized edge between two nodes. By (3), we normalize the values of the weighted edges.

$$B(u, v) = \frac{W_{uv}}{\max W_{uv}} \tag{3}$$

Instead of 0 in the  $B_{uv}$  value, a minimal value (0.0001) is used because the absence of an edge between nodes in the weighted matrix would mean 0 and prevent those nodes from participating in subsequent calculations.

### 4 Community detection by WSCD model

We will show how to use the WSCD model for community detection in undirected and weighted networks after describing the components of the model. The belonging degree of a node to the community ( $I_{uc}$ ) and edge weight between two nodes are model parameters described in the previous section. We find the optimal  $I$  by maximizing the likelihood as follows,

$$l(I) = \log P(G | I, B) \tag{4}$$

$$I = \underset{I > 0}{\operatorname{argmax}} L(I) = \underset{I > 0}{\operatorname{argmax}} \prod_{(u,v) \in E} P(u, v) \prod_{(u,v) \notin E} (1 - P(u, v)) \tag{5}$$

After inserting (1) into (5), a natural logarithm can be obtained from both sides to convert the multiplication to the sum and simplify subsequent calculations. The maximum likelihood estimation procedure will not be hampered by the logarithm because it is strictly ascending

$$L(I) = \sum_{(u,v) \in E} \ln(1 - \exp(-I_u \cdot I_v^T \cdot B_{uv})) + \sum_{(u,v) \notin E} (-I_u \cdot I_v^T \cdot B_{uv}) \tag{6}$$

#### 4.1 Updating the parameter

To solve the objective function in (6) and estimate the maximum likelihood function, we must use the non-linear optimization procedures. For this purpose, we used the well-known Block Coordinate Ascent algorithm (Lin, 2007; Hsieh & Dhillon, 2011). We update  $I_u$  for each node  $u$  by keeping fix the neighbors ( $I_v$ ) and weight between them ( $B_{uv}$ ). We solve the following subproblem for each  $u$ :

$$L(I_u) = \sum_{v \in N(u)} \ln(1 - \exp(-I_u \cdot I_v^T \cdot B_{uv})) + \sum_{v \notin N(u)} (-I_u \cdot I_v^T \cdot B_{uv}) \tag{7}$$



where  $N(u)$  represents the set of neighbors of  $u$ . To estimate maximum likelihood (maximum point of the diagram) we must look for a point on the diagram where the slope is 0. Therefore, it is required to derive the partial derivative of the log-likelihood function (7) with respect to  $I_u$ .

$$\frac{\partial l(I_u)}{\partial I_u} = \sum_{v \in N(u)} I_u B_{uv} \frac{\exp(-I_u I_v^T B_{uv})}{1 - \exp(-I_u I_v^T B_{uv})} - \sum_{v \notin N(u)} I_v B_{uv} \quad (8)$$

Finally,  $I_u$  values will be updated using the gradient ascent method (Lin, 2007; Hsieh & Dhillon, 2011) and alternating iterations. Because the belonging intensity of a node to a community cannot be negative, if it detects one, it will be replaced with 0:

$$I_u(t+1) = \max \left( 0, I_u(t) + \eta \left( \frac{\partial l(I_u)}{\partial I_u} \right) \right) \quad (9)$$

where  $\eta$  is a learning rate parameter; The process of update each  $I_u$  at steps of the algorithm iteration is continued as far as the difference between the previous step's value and current value be less than the desired threshold.

## 4.2 WSCD Algorithm

Algorithm 1 presents the proposed WSCD model (structural community detection in weighted networks). In this Algorithm, an undirected and weighted graph ( $G$ ) and the number of communities ( $k$ ) are the inputs of the model. The weights of all edges are normalized to a value between 0 and 1, and the input graph is assumed to be weighted. A matrix containing the intensity of belonging each node to each community ( $I_{uc}$ ) is also generated by the model. The relationship between observing a pair of nodes in multiple communities and the probability of a high-weight edge is another significant output that intuitively follows from the understanding of the proposed model. Following that, after the latent variable of the model ( $I$ ) is initialized (how to initialize  $I$  will be discussed below), the algorithm starts in an iterative loop. When the difference between  $I_u(t+1)$  and  $I_u(t)$  is less than a defined threshold (*stop threshold* = 0.001), the iterations will come to an end. The likelihood function of the probabilistic generative model ( $L(I_u)$ ), which aims to estimate the unknown parameter of the model in the normalized weighted graph, is calculated in this iterative method. The logarithm of the likelihood function is extracted from each node  $u$  to get the likelihood function close to its maximum value (where the slope of the line is 0) ( $D(L(I_u))$ ). We used ascending gradient method (Lin, 2007; Hsieh & Dhillon, 2011) to maximize the likelihood instead of the closed-form method due to the difficulty of the calculations.

Finally, after updating the ( $I_u$ ) at each step of the algorithm iteration and fixing the  $I$  values, the belonging intensity of each node to each community will be calculated. After comparing it to an experimental threshold (for example, the average of  $I$  values), this value is labeled as belonging or not belonging to the communities.

## 4.3 Computational complexity

The number of communities and the density of weighted edges determine the WSCD algorithm's computational complexity. As seen in Algorithm 1's iteration steps, the degree of belonging to the community, which is at the heart of the algorithm, is updated using (8)

---

```

1: Input:  $G = (V; E; W)$ ; Number of communities ( $k$ );
2: Output:  $I_{uc}$  belonging of each node  $u$  to Community  $c$ 
3: Compute:  $B(u, v)$ =Normalized weight from  $w(u, v)$ 
4:  $t \leftarrow 0$ 
5: while  $| I_u(t + 1) - I_u(t) | \leq 0.001$  do
6:    $t \leftarrow t + 1$ 
7:   for  $i = 1$  to  $| V |$  do do
8:      $L(I) = \log p(G | I, B)$ 
9:      $D(L(I_u)) = \text{FindDerivation}L(I_u)$ 
10:     $\text{Update} : I_u(t + 1) = \text{UpdatePropGA}(D(L(I_u)); I_u(t))$ 
11:   end for
12: end while
13: for  $i = 1$  to  $| V |$  do
14:   for  $j = 1$  to  $| k |$  do
15:     if  $I_{uc} > \text{threshold}$  then
16:        $\text{Add} : c_j \leftarrow u_i$ 
17:     end if
18:   end for
19: end for

```

---

**Algorithm 1** Structural community detection in weighted networks (WSCD)

and (9). In this case, whether or not two nodes have a weighted edge depends on whether or not their neighbors belong to one or more communities. As a result, the computational complexity will be determined by the order of each node's neighbors ( $N(u)$ ) and its combination with the number of communities, which in the worst case will be  $O(k \cdot |E|)$ .

#### 4.4 initialization

The matrix of belonging intensity the nodes to each community can be initialized in two ways. The first solution, which appears simplest solution, is to fill the values randomly. However, the main drawback is that to reach the model stability stages, the algorithm repeats the steps longer and this increasing the computational complexity. The second option is to use the local minimum neighborhood method (Gleich & Seshadhri, 2012), which has been proven to be a good starting point for community detection algorithms through experiments. Other benefit of using this approach is that it can help predict the initial number of communities to start the community detection phase in the proposed model, in addition to minimizing iteration steps and beginning the algorithm in a steady state.

## 5 Experiments

The proposed WSCD method has been implemented in the spyder environment and using the Python programming language. We used 3 real-world datasets (Table 2) and 8 synthetic networks (Table 4) to evaluate the results. The real-world datasets analyzed during the

**Table 1** Summarizes of the utilized algorithms

Method Name	Description
Louvain	Louvain maximizes a modularity score for each community
Leiden	The Leiden algorithm is an improvement of the Louvain
Label propagation	The LPA detects communities using network structure alone
Greedy modularity	The CNM algorithm uses the modularity to find the communities structures
ASLPAW	ASLPAW can be used for disjoint and overlapping community detection
wCommunity	Algorithm to identify overlapping communities in weighted graphs

**Table 2** Details of the utilized dataset

Dataset Name	N (Number of Nodes)	E (Number of Edges)
Net Science	379	914
Wiki-Vote	879	2914
Twitter	1003	25779

current study are available in the network repository.<sup>1</sup> The LFR-Benchmark\_UndirWeight-Ovp<sup>2</sup> generates the synthetic networks. Also, the proposed method is compared with basic weighted algorithms such as Louvain (Lancichinetti et al., 2008; Lancichinetti & Fortunato, 2009), Leiden (Traag et al., 2019), Label propagation (Gregory, 2010), Greedy modularity (Clauset et al., 2004), ASLPAW (Xie et al., 2011), and wCommunity (Chen et al., 2010) on the above datasets. Table 1 summarizes these algorithms (Table 2).

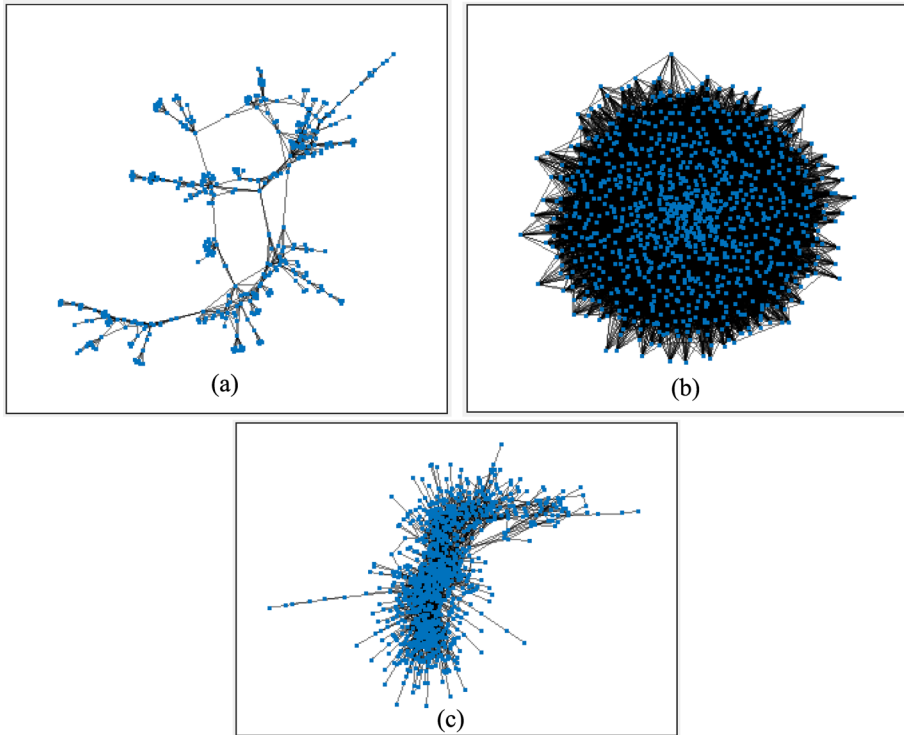
## 5.1 Evaluation metrics

Two well-known evaluation metrics are applied to measure the quality and accuracy of the community detection algorithms. The modularity (Clauset et al., 2004) and the conductance (Shi & Malik, 2000) as internal metrics for assessing the quality of communities and also, the F1Score alongside NMI<sup>3</sup> as external metrics for measuring the accuracy of communities by compared to the ground-truth communities (Fortunato & Hric, 2016). In internal metrics, the modularity criterion is derived from the Girvan-Newman algorithm (Clauset et al., 2004) and is the well-known criteria for calculating the density of edges in communities. Modularity value is equal to the ratio of edges estimated within the community to the expected edges. In this case, the closer a community's modularity value is to 1, nodes with higher density in communities and the higher the quality of the detected community. The conductance criterion is a method of calculating the ratio of edges that have left the community to the total edges of the graph. As a result, the lower the numerical value of conductance, the quality of a community is higher. In external metrics, the F1Score is a famous evaluation measure in community detection algorithms, which

<sup>1</sup> <https://networkrepository.com/>

<sup>2</sup> [https://github.com/eXascaleInfolab/LFR-Benchmark\\_UndirWeightOvp/](https://github.com/eXascaleInfolab/LFR-Benchmark_UndirWeightOvp/)

<sup>3</sup> Normalized mutual information



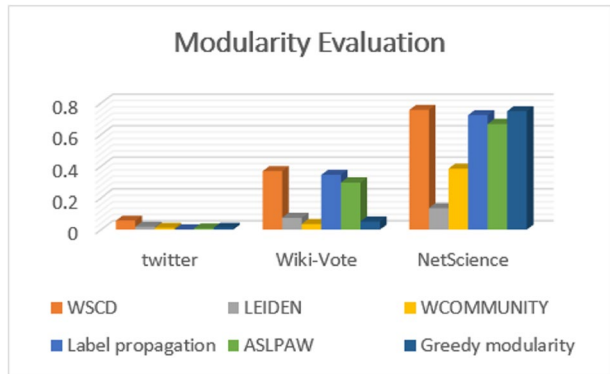
**Fig. 1** Networks density representation (a) Netscience, (b) Twitter , and (c) Wiki-Vote datasets

quantifies the comparative frequency of the number of correct detections of the nodes in each community based on the ground-truth information. The second external metric is NMI which is the mutual information of the association detected between the discovered communities and the ground-truth.

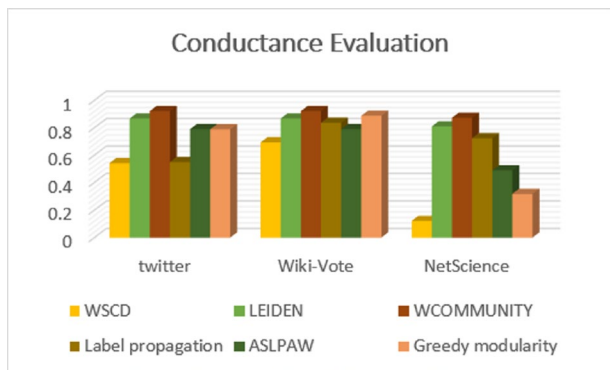
## 5.2 Real-world Networks

Net Science (Rossi & Ahmed, 2015) is the first dataset containing 379 nodes and 914 connecting edges between them. This dataset contains a scientific research communication network wherein researchers are as nodes. The grid contains an undirected weighted edge from  $i$  to  $j$  if node  $i$  and node  $j$  has at least one article in common. The normalized number of joint articles between two nodes equals the weight of the edge. Wiki-Vote (Rossi & Ahmed, 2015) is the second dataset, which contains 879 nodes and 2914 connecting edges. A poll's voter data is contained in this dataset. The network's nodes represent network users and connecting node  $i$  to node  $j$  denotes the edge that user  $i$  voted to user  $j$ . The normalized number of votes is also the weight of connecting edges. The third dataset, with 1003 nodes and 25779 connecting edges, is based on data from the social network Twitter (Kumar et al., 2014). The nodes in this graph represent social network users, and the edge between node  $i$  and node  $j$  represents tweets of node  $i$  retweet by node  $j$ . The normalized number of these retweets is also used to calculate the weight of edges (Fig. 1).

**Fig. 2** WSCD modularity evaluation with five weighted community detection methods on three datasets



**Fig. 3** WSCD conductance evaluation with five weighted community detection methods on three datasets

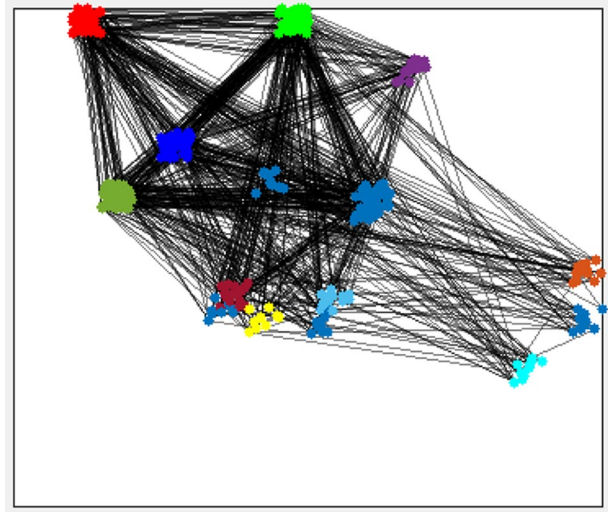


### 5.2.1 Experimental results on real-world networks

We will compare the proposed model with two Categories of conventional community detection methods, namely methods based on label propagation and methods based on modularity optimization, to assess the efficiency and accuracy of WSCD in weighted community detection. Some of these methods were briefly discussed in the preceding sections. Five weighted algorithms are used to evaluate the proposed method with modularity and conductance criteria. In terms of modularity maximum and conductance minimum, the results show that our method is superior. The results in Fig. 2 show that WSCD has higher modularity than other methods. In addition, Fig. 3 show that WSCD has lower conductance than other methods, which indicates minimum edge exit from the community in the proposed method.

Figure 4 shows the results of the communities detected by the WSCD method in the NetScience weighted graph. As seen, the detected communities have acceptable node densities and no scattered nodes outside communities. Also, the dispersion of blue communities in Fig. 4 indicates the division of nodes according to the weight of the edge between them in different communities and the intuitive interpretation of the relationship between the weight of the edge and the belonging of the node to the community in WSCD, which is calculated in (7). In case of weight of the edge is removed from the calculations, the proposed method would identify the communities only based on the belonging of nodes to

**Fig. 4** Community detection by the WSCD method on the NetScience dataset



the communities, in which case all the blue communities are aggregated and represented as a single community. This fact indicates the role of using edge weight in community detection and the accuracy of the proposed method. This interpretation is also true for the other datasets used in the proposed method, but due to the high volume of nodes and edges, the images of the output communities do not have the necessary clarity for intuitive understanding.

### 5.3 Synthetic networks

There are different ways to create synthetic networks for evaluate community detection algorithms, one of the well-known approaches widely used is the LFR<sup>4</sup> benchmark. The LFR benchmark (Lancichinetti et al., 2008) generates undirected and unweighted synthetic networks with ground-truth communities by taking the degree and community size distributions. For creating simulated networks with LFR, should set parameters of network and communities. One of the significant parameters in LFR is the mixing parameter ( $\mu$ ). This parameter controls the interactions between communities. A high value of the mixing parameter will reduce the modularity value of the network. The LFR generated datasets are split into two types of sparse and dense communities according to the modularity criterion. Another important parameter is the average degree. The increasing value of the average degree will cause to increasing the interactions between communities. The extended version of the LFR Benchmark for Undirected and Weighted networks is presented in Lancichinetti and Fortunato (2009) to evaluate community detection algorithms. The principal change in parameters is appending the mixing parameter for the weights ( $\mu_w$ ) alongside the mixing parameter for the topology ( $\mu_t$ ). The main parameters of the LFR synthetic datasets are shown in Table 3. The characterization of the created dataset through the LFR approach that we used, is displayed in Table 4.

<sup>4</sup> Lancichinetti–Fortunato–Radicchi

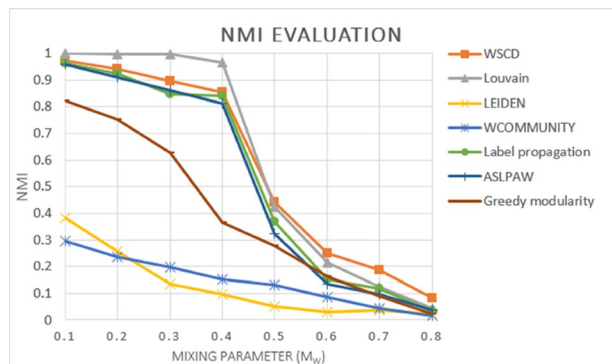
**Table 3** Parameters of LFR synthetic datasets (Lancichinetti & Fortunato, 2009)

Parameter	Description
N	Number of nodes
K	Average degree
Min K	Minimum degree of nodes
Max K	Maximum degree of nodes
$\mu_t$	Mixing parameter for the topology
$\mu_w$	Mixing parameter for the weights
Min C	Minimum for the community sizes
Max C	Maximum for the community sizes

**Table 4** The details of the LFR synthetic network generated

#N	$\mu_t$	$\mu_w$	K	#Edges	#Communities	Min C	Max C
1000	0.1	0.1	20	7642	27	20	60
1000	0.2	0.2	20	7680	28	20	60
1000	0.3	0.3	20	7687	29	20	60
1000	0.4	0.4	20	7762	29	20	60
1000	0.5	0.5	20	7602	30	20	60
1000	0.6	0.6	20	7743	32	20	60
1000	0.7	0.7	20	7817	34	20	60
1000	0.8	0.8	20	7756	35	20	60

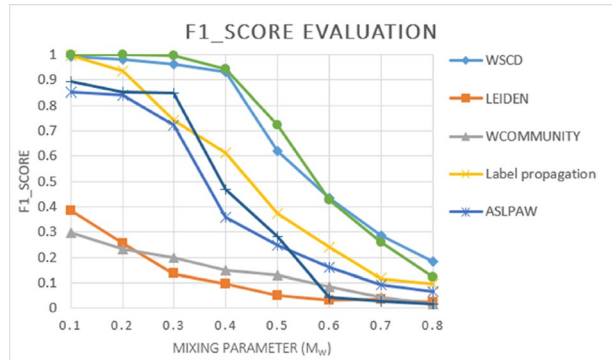
**Fig. 5** NMI evaluation diagram, comparing WSCD with six weighted community detection methods on eight LFR datasets



### 5.3.1 Experimental results on Synthetic networks

In addition to the real-world networks, we have implemented experiments on LFR synthetic networks. We compare the WSCD method with the well-known weighted community detection methods in Table 1 to show the effect of edge weights on the community detection process by using F1Score and NMI criteria. For this purpose, according to the properties of synthetic networks that are given in Table 3, eight LFR synthetic networks are created with different configurations of mixing parameter for the weights ( $\mu_w$ ) varying from

**Fig. 6** F1Score evaluation diagram, comparing WSCD with six weighted community detection methods on eight LFR datasets



0.1 to 0.8, showed in Table 4. By the results, for the small of the mixing parameter values (e.g.  $0.1 \leq \mu_w \leq 0.4$ ), the communities are dense and all compared algorithms have near accuracy in this case. But the central difference between the weighted algorithms occurs when the value of the mixing parameter ( $\mu_w$ ) increasing. (e.g.  $0.4 < \mu_w \leq 0.8$ ), because edges between communities increase, the communities are sparse, and detecting communities are difficult. As shown in Figs. 5 and 6, some algorithms have NMI and F1Score values equal to zero as the mixing parameter value increases and in the range of  $0.4 < \mu_w \leq 0.8$ , the proposed method (WSCD) is better than most of the well-known weighted methods.

## 6 Conclusion

In this paper, we present a probabilistic model-based method for community detection in weighted networks. The proposed approach, WSCD, used the weight of the edge between the two nodes and the intensity belonging of node to the community to estimate latent variable of probabilistic model, which because of the hardness of implementation and restriction of results on the weighted dataset, has received less consideration in recent community detection methods. We employed the well-known Block Coordinate Ascent algorithm to maximize the likelihood function and estimate latent parameters of the model. Moreover, WSCD helps with the interpretation of detected communities by finding relevancy between membership of nodes to communities and edge weight, when a pair of nodes is observed in multiple communities, the possibility of a high-weight edge between them is amplified. For evaluation proposed method 3 real-world and 8 synthetic weighted networks were used. In 3 real-world weighted networks, WSCD was able to obtain a proper quorum when compared to the other five methods on internal evaluation metrics. Also, evaluation of the proposed method on 8 synthetic weighted networks by external evaluation metrics show that WSCD outperforms on the sparse networks than other methods. In addition, the proposed method's execution time complexity analysis reveals that outperforms other methods. There is some future work for researchers to develop a proposed method. Such as considering edge weight as a latent parameter and the estimate it by a probabilistic generative model, extending the method by employing node attributes in the weighted network to present a more accurate interpretation of detected communities.



**Data availability** The real-world datasets generated during and/or analysed during the current study are available on the network repository,<sup>5</sup> and the LFR-Benchmark\_UndirWeightOvp<sup>6</sup> generates the synthetic networks.

## Declarations

**Conflict of interests** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Ahn, Y., Bagrow, J., & Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature*, *466*, 761–764.
- Beckett, S. (2016). Improved community detection in weighted bipartite networks. *Royal Society Open Science*, *3*, 140536.
- Blondel, V., Guillaume, J., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*, P10008.
- Celebi, M. (2014). *Partitional clustering algorithms*. Springer.
- Chen, D., Shang, M., Lv, Z., & Fu, Y. (2010). Detecting overlapping communities of weighted networks via a local algorithm. *Physica A: Statistical Mechanics and its Applications*, *389*, 4177–4187.
- Clauset, A., Newman, M., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, *70*, 066111.
- Donath, W., & Hoffman, A. (2003). Lower bounds for the partitioning of graphs. *Selected Papers Of Alan J Hoffman: With Commentary*, 437–442.
- Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, *659*, 1–44.
- Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, *486*, 75–174.
- Fu, S., Wang, G., Xu, J., & Xia, S. (2021). IbLT: An effective granular computing framework for hierarchical community detection. *Journal Of Intelligent Information Systems*, 1–22.
- Girvan, M., & Newman, M. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, *99*, 7821–7826.
- Gregory, S. (2010). Finding overlapping communities in networks by label propagation. *New Journal of Physics*, *12*, 103018.
- Gleich, D., & Seshadhri, C. (2012). Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings Of The 18th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 597–605).
- Hsieh, C., & Dhillon, I. (2011). Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings Of The 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1064–1072).
- Javed, M., Younis, M., Latif, S., Qadir, J., & Baig, A. (2018). Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, *108*, 87–111.
- Kumar, S., Panda, B., & Aggarwal, D. (2021). Community detection in complex networks using network embedding and gravitational search algorithm. *Journal of Intelligent Information Systems*, *57*, 51–72.
- Kumar, S., Morstatter, F., & Liu, H. (2014). *Twitter data analytics*. Springer.
- Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E*, *78*, 046110.

<sup>5</sup> <https://networkrepository.com/>

<sup>6</sup> [https://github.com/eXascaleInfolab/LFR-Benchmark\\_UndirWeightOvp/](https://github.com/eXascaleInfolab/LFR-Benchmark_UndirWeightOvp/)

- Lancichinetti, A., & Fortunato, S. (2009). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, *80*, 016118.
- Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.
- Li, C., Bai, J., Wenjun, Z., & Xihao, Y. (2019). Community detection using hierarchical clustering based on edge-weighted similarity in cloud environment. *Information Processing & Management*, *56*, 91–109.
- Lin, C. (2007). Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, *19*, 2756–2779.
- Liu, R., Feng, S., Shi, R., & Guo, W. (2014). Weighted graph clustering for community detection of large social networks. *Procedia Computer Science*, *31*, 85–94.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, *1*, 281–297.
- Maqbool, O., & Babri, H. (2004). The weighted combined algorithm: A linkage algorithm for software clustering. In *8th European Conference on Software Maintenance and Reengineering, CSMR 2004. Proceedings.* (pp. 15–24).
- McAuley, J., & Leskovec, J. (2012). Learning to discover social circles in ego networks. *NIPS*, 548–56.
- Morvan, A., Choromanski, K., Gouy-Pailler, C., & Atif, J. (2017). Graph sketching-based massive data clustering. arXiv:1703.02375.
- Myung, I. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, *47*, 90–100.
- Newman, M., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*, 026113.
- Newman, M. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, *103*, 8577–8582.
- Nie, F., Zhao, X., Wang, R., Li, X., & Li, Z. (2020). Fuzzy K-means clustering with discriminative embedding. *IEEE Transactions on Knowledge and Data Engineering*.
- Nicolini, C., Bordier, C., & Bifone, A. (2017). Community detection in weighted brain connectivity networks beyond the resolution limit. *NeuroImage*, *146*, 28–39.
- Qi, G., Aggarwal, C., & Huang, T. (2012). Community detection with edge content in social media networks. In *2012 IEEE 28th international conference on data engineering* (pp. 534–545).
- Raghavan, U., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, *76*, 036106.
- Rossi, R., & Ahmed, N. (2015). The network data repository with interactive graph analytics and visualization. In *Proceedings Of The AAAI Conference on Artificial Intelligence*, 29.
- Sánchez-Oro, J., & Duarte, A. (2018). Iterated Greedy algorithm for performing community detection in social networks. *Future Generation Computer Systems*, *88*, 785–791.
- Sia, J., Jonckheere, E., & Bogdan, P. (2019). Ollivier-ricci curvature-based method to community detection in complex networks. *Scientific Reports*, *9*, 1–12.
- Sinaga, K., & Yang, M. (2020). Unsupervised K-means clustering algorithm. *IEEE Access*, *8*, 80716–80727.
- Shen, H., Cheng, X., Cai, K., & Hu, M. (2009). Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, *388*, 1706–1712.
- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation departmental papers (CIS).
- Traag, V., Waltman, L., & Van Eck, N. (2019). From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, *9*, 1–12.
- Xie, J., Szymanski, B., & Liu, X. (2011). Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *2011 IEEE 11th international conference on data mining workshops* (pp. 344–349).
- Yang, J., & Leskovec, J. (2012). Community-affiliation graph model for overlapping network community detection. In *2012 IEEE 12th international conference on data mining* (pp. 1170–1175).
- 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).
- Yang, J., McAuley, J., & Leskovec, J. (2013). Community detection in networks with node attributes. In *2013 IEEE 13th international conference on data mining* (pp. 1151–1156).