Motif-Based Community Detection: A Probabilistic Model Based on Repeating Patterns

H. Hajibabaei 1, V. Seydi 1,2,*, A. Koochari 2

1Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
2Centre for Applied Marine Sciences, School of Ocean Sciences, Bangor University, Menai Bridge, UK.

Introduction

For effective network component monitoring and recognition, network analysis is a key tool. A complex network [1] can consist of cells in biology [2], social networks with friendly communication [3], or a network of scientists doing joint scientific studies [4]. To put it another way, it can be any grid with nodes and edges that can be represented as a graph. One of the most effective methods and methodologies to analyze complex networks is community detection.

Community detection identifies subgraphs of a network whose relationship among their nodes is more robust and dense than those between other nodes of the network [5]. A community can represent an idea, a group, an interest, a focus on a particular topic, and so on. Communities can be used separately or together; the latter are referred to as overlap communities. The machine learning clustering topic’s component,
Community detection, has the potential to be applied in a variety of engineering fields, such as text classification, traffic network optimization, and social network analysis. The goal of community detection is to group the nodes of a network into different communities so that they are strongly connected or have similar node features [5]. A key problem in dynamical network research is the discovery of communities with the aim of revealing hidden features of a complex network, which are frequently densely coupled nodes [6].

Community detection in networks is an NP-hard issue that categorized from different perspectives. These categories include weighted [7], [8] and unweighted [9], [10], directed [11] and undirected [12], global [13] and local [12], overlapping [14], [15], and non-overlapping [16] community discovery techniques. Different community detection techniques were developed based on these criteria. Examples include model-based approaches [7], [14], [17], clique percolation methods [18], modularity-based methods [4], [19], [20], label propagation methods [11], [21]-[23], model-based methods [7], [14], methods for network embedding [24] and community detection methods with deep learning [25]-[27]. It can be seen from examining the different approaches used for community detection and the research on this topic that a straightforward analysis of node properties won’t produce the accuracy needed for community detection in networks; rather, taking a deeper look at the networks’ particulars and using the graphs’ original characteristics, like motif structure, will produce better results.

The PMCD method employs a probabilistic relation to find communities in complex networks. We extend probabilistic model-based methods from edge creation to motif generation. Complex networks commonly contain “motifs”, which are a type of small, linked sub-networks. Based on empirical studies, communities with similar nodes have related motifs. As a result, using motifs with lots of connections can be a useful strategy for finding communities and performing more accurate network analysis [28]. We show that the chance of a triangle motif existing between three nodes in shared communities grows with the observation of more nodes in such communities. In other words, we locate the hidden parameter of the probabilistic model and find the community by using the triangle motif. We define the triangular motif estimator function as a Bernoulli loss function over one node and two of that node’s neighbors for the probabilistic motif generator’s function. We also research how community overlap affects how motifs are generated.

Related Works

The problem of community detection in complex networks gets a lot of attention. Several research projects on different aspects of community detection have been performed over the past few years. The first methods of community detection employed traditional techniques and clustering-based algorithms. These methods presented key ideas for community detection and laid the groundwork for future developments. Traditional approaches include graph partitioning, hierarchical, spectral and partitional clustering [29].

The algorithms used to detect communities based on modularity have been extensively studied and used due to their simple tactics and clear outcomes. However, they also encounter difficulties, such as communities that are unstable and sensitive to seed node selection [30]. One of these, the Louvain technique [5], is frequently applied to weighted graphs. This approach provides a straightforward and quick methodology to detect distinct communities and maximizes modularity by clustering network nodes using the greedy approaches [31]. The Leiden method, however, corrects numerous flaws in the Louvain algorithm [32]. The objective is to change the community developed throughout the iteration cycle while simultaneously speeding up local mobility and transferring nodes to arbitrary neighbors.

The label propagation algorithm (LPA), a practical community detection approach, was initially introduced in [22]. Although its simple design and low complexity are widely respected, there are several downsides, such as the randomness of node selection and label updating. In the LPA technique, a node is selected at random, and through an iterative process, its label is updated with the most prevalent label nearby [33]. To handle the weaknesses in the LPA methodology, the Speaker-Listener Label Propagation Algorithm (SLPA) [34] and the COPRA [35] were created.

Cliques are one of the fundamental ideas in graph analysis and are utilized to detect communities in networks. The clique percolation algorithm (CPM) [18] and CFinder [36] were proposed as overlapping community detection algorithms based on the clique percolation method’s search for local patterns.

The motif is another idea related to the clique. Small, linked sub-networks known as motifs frequently appear in complex networks and are one of the basic elements of the network [37]. In network analysis, motifs are used to detect communities and comprehend network structure [38]. Motifs demonstrate that a community with a high edge density will have relationships between nodes that go beyond their immediate neighbors. Although a few motif-based community detection methods have been proposed [38]-[40], when used on large-scale networks, they frequently encounter high computational complexity. It is still challenging to properly and economically combine lower-order and higher-order structural data into a unified framework for community detection.

The group of methods estimates the probabilistic model to detect communities, in difference to the techniques cited at the top that employ traditional methods to do so. This method creates a generative
sample of the graph and estimates the model parameters [14], [17]. The degree of node dependency on communities is a parameter in the generative model that is estimated using methods [14], [17], [41] using a matrix factorization-based model. The algorithm [42] presents a matrix factorization-based paradigm that makes it easy to add or delete edges. The non-negative matrix factorization model of the community detection issue is also described [43] and a transfer matrix is then used to control the dynamics of the network structure.

Proposed Algorithm Frameworks

In this research, a probabilistic motif-based community detection model (PMCD) is presented that uses the triangle motif and the affiliation graph model to detect community structures. The core idea of the proposed community detection method is that a robust community requires taking the node’s structural model and relationship types into consideration. Two nodes observed in more shared communities are more likely to be connected, according to Yang and Leskovec’s study [14] on the connection between edge (2-clique) likelihood and community overlap. In this paper, we examine the impact of community overlapping on the evolution of the triangle and 3-clique motifs. We show that by increasing the number of nodes observed in shared communities, the probability of the existence of a triangular motif between them increases. This result is in accordance with the fundamental principle that vertices situated in communities’ overlaps are more densely connected than vertices within a single community. By using the optimized Bernoulli loss function for probabilistic estimation, we can therefore enhance and increase the AGM’s ability [14], [17] to generate triangle motifs.

The PMCD model is different from other community detection approaches in that it considers additional properties that were not sufficiently considered in earlier methods, for instance:

- Using edge density to detect communities.
- Triangle motif estimation using a probabilistic approach.
- The conceptual link between community detection and the likelihood of the triangular motif being present or absent.
- Use of evolutionary approaches and maximum likelihood estimation in computations.

Fig. 1 presents an example of a simple graph to illustrate the concept of three-node motifs. Two different three-node motif types discovered in Fig. 1 are shown in Fig. 2. Depending on the characteristics of the networks, the triangle motifs observed in various types of networks can be interpreted in multiple ways. For instance, the 3-node motif (3, e) and the 4-node motif (4, e) are the most widely studied motifs in complex networks [44]. The proposed method uses the triangular motif to build the hidden parameter of the probabilistic model and detect the community.
The PMCD model is based on a network $G(N, E)$, where nodes and edges are referred to as $N$ and $E$, respectively. We create $M_{uv}$, a nonnegative integer, to represent the strength of the node's affiliation with the community. ($M_{uv} = 0$ denotes $u$'s non-membership in $c$.) The degree of reliance between each node and each community is thus shown in the $M$ matrix.

The value of $M$ in PMCD establishes whether or not a triangular motif between three nodes $(u, v, w)$ will appear in a community. Specifically, we presumed that three nodes, $u$, $v$, and $w$, are triangular motifs by taking into account the following likelihood. For the probabilistic motif generator's function, we define the triangular motif estimator function as a loss function over one node and two neighbors of that node, that is,

$$P_s(u, v_1, v_2) = P(u, v_1). P(u, v_2) = \begin{bmatrix} 1 - \exp(-M_{uc} \cdot M_{v_1c}^T) \cdot \left[1 - \exp(-M_{uc} \cdot M_{v_2c}^T) \right] \end{bmatrix}$$

Due to the generative probabilistic approach between two couples of nodes in a triangle motif, each couple of nodes is independently propagated by the Bernoulli model. Thus, per element of the adjacency matrix is formed on the following probabilistic method:

$$P(u, v_1, v_2) = P(u, v_1). P(u, v_2) = \begin{bmatrix} 1 - \exp(-M_{uc} \cdot M_{v_1c}^T) \cdot \left[1 - \exp(-M_{uc} \cdot M_{v_2c}^T) \right] \end{bmatrix}$$

$A_{v_1} \sim \text{Bernoulli}(P_{v_1}) \odot A_{v_2} \sim \text{Bernoulli}(P_{v_2})$

The framework of computation for (1) and (2), which describe a probabilistic generative model, is predicated on the following premises:

- In a community, a triangle motif can exist between two pairs of nodes (one node and two neighbors of that node).
- The probability of the existence triangle motif increases when two pairs of nodes are observed in multiple communities.
- Communities can overlap; communities that overlap have a higher density of triangle motifs.

**Community detection by PMCD model**

We defined the components of the PMCD model before illustrating how to utilize it for community detection in networks. The model parameter discussed in the preceding section is the degree of a node's community membership ($M_{uv}$). By maximizing the likelihood, we can get the optimum $M$ as follows:

$$\text{likelihood } M = l(M) = \log P(G | M)$$

$$M = \arg \max l(M) = \arg \max M \prod_{u,v_1,v_2} p(u, v_1, v_2) \prod_{(u,v_1,v_2)} (1 - p(u, v_1, v_2)) = \frac{\arg \max M \prod_{(u,v_1,v_2)} p(u, v_1, v_2) \cdot p(u, v_2)}{\prod_{(u,v_1,v_2)} (1 - p(u, v_1) \cdot p(u, v_2))}$$

After combining (2) and (3), a natural logarithm is needed to be computed on both sides to correct the multiplication to the aggregate and reduce the next computations. The logarithm is completely ascending; therefore, it won't interfere with the maximum likelihood estimation.

$$L(M) = \sum_{(u, v, w) \in E} \log(1 - \exp(-M_{uv} M_{vw}^T)) - \sum_{(u, v, w) \in E} M_{uv} M_{vw}^T$$

**Updating the Parameter**

The non-linear likelihood function of (4), which contains the latent variable $M$, cannot be maximized by conventional optimization methods. We calculate the objective function in (4) using the Block Coordinate Ascent approach [45], which helps solve optimization problems with latent variables in machine learning. By maintaining fixed neighbors ($M_{uv}$), we update $M_{uv}$ for each node $u$.

$$L(M_u) = \sum_{(u,v) \in E} \log(1 - \exp(-M_{uv} M_{vu}^T)) - \sum_{(u,v) \in E} M_{uv} M_{vu}^T$$

In (5), $N(u)$ is a set of neighbours of $u$. In order to calculate the maximum probability (the diagram's maximum point), we must find a location on the Figurative chart where the gradient equals 0. Thus, it is necessary to derive the partial derivation of the likelihood logarithmic in (5) than $M_u$.

$$\frac{\partial l(M_u)}{\partial M_u} = \left[ \sum_{i \in N(u)} M_{iu} \frac{\exp(-M_{iu} M_{iu}^T)}{1 - \exp(-M_{iu} M_{iu}^T)} - \sum_{i \in N(u)} M_{iu} \right] + \left[ \sum_{i \in N(u)} M_{iu} \frac{\exp(-M_{iu} M_{iu}^T)}{1 - \exp(-M_{iu} M_{iu}^T)} - \sum_{i \in N(u)} M_{iu} \right]$$

The gradient ascent algorithm will eventually update $M_u$ values [46, 47]. A node's belonging strength to a community will be replaced with 0 if it detects it, as it is impossible for it to be negative.

$$M_u(t + 1) = \max \left[ 0, M_u(t) - \eta \left( \frac{\partial l(M_u)}{\partial M_u} \right) \right]$$

In (5), $\eta$ is a learning parameter. As long as the difference between the value from the last step and the current value is lower than the acceptable threshold, the process of updating each $M_u$ at each stage of the algorithm iteration is repeated.

**PMCD Algorithm**

Algorithm 1 displays the proposed PMCD model (probabilistic motif-based community detection). A graph
(G) and the number of communities (k) are the method’s inputs. The model also creates a matrix (Muc) that displays the degree to which each node belongs to each community. When they are observed in different communities, the likelihood that there is an existing motif structure between two groups of nodes increases.

Since the hidden variable (M) is initialized (details of computing M addressed later), the method then begins an iterative process. After the difference among Muc(t+1) and Muc(t) was smaller than a predefined point (in this case, the stop threshold is 0.005), the iterations stop. In order to estimate the model’s unknown parameter in the graph, this iterative method calculates the likelihood of the probabilistic model (L(Muc)). To derive the likelihood function’s logarithm as close as possible to its maximum value (when the line’s slope is 0), the likelihood function’s logarithm is collected from each node u using the formula D(L(Muc)).

### Algorithm 1: Probabilistic motif-based community detection (PMCD)

1: **Inputs:** Graph G = (N; E);
   Number of communities (k);
2: **Output:** Muc belonging of each node u Community c
3: t ← 0
4: M = local_maximun_neighborhood()
5: while |Muc(t + 1) - Muc(t)| <= 0.005 do
6:   t ← t + 1
7:   for i = 1 to |V| do
8:     L(M) = log p(G | M)
9:     D(L(Muc)) = Derivation_finder_L(Muc)
10:   Update: Muc(t + 1) = Gradient_ascent (D(L(Muc)); Muc(t))
11: end for
12: end while
13: for i = 1 to |V| do
14:   for j = 1 to k do
15:     if Muc > threshold then
16:       Add: cu ← u,
17:     end if
18:   end for
19: end for

In line 10 of Algorithm 1, we chose the ascending gradient approach [46], [47] to optimize the probability because the computations were complex. This method is used to update the latent variable of the model (Mu) at each iteration of the algorithm. After the M value has been fixed, each node’s ability to contribute per community is assessed. Since comparing this value to a testing point (such as the median of M), it may be defined as either belonging to or not belonging to the communities, and the output of the model will then be realized.

### Computational Complexity

The count of communities and dense motifs affect the computational complexity of the PMCD method. The core concept of Algorithm 1, as shown in its iteration phases, is the rate of depending on the community, which is updated using (6) and (7). In this case, whether or not two nodes have neighbors who are members of one or more communities determines whether or not those nodes share a theme. Because of this, the computational complexity will depend on the number of communities present and the order of each node’s neighbors (N(u)); in the worst case, this complexity will be O(2k.|E|).

**Initialize**

The matrix of depending strengths for the nodes communities can start in a variety of ways. The first option, which also appears to be the simplest, involves filling in the values at random. The algorithm’s major drawback, however, is that it repeats the steps more frequently, increasing computing complexity as it advances to the model stability phases.

The other choice is the local minim neighborhood approach [48], which has been shown through studies to be an excellent starting point for community discovery algorithms. Using this method has the added benefit of being able to estimate the initial number of communities to start the proposed model’s community detection phase, in addition to minimizing iteration steps and starting the process in a stable state.

**Experiments**

The proposed PMCD method has been implemented in the Spyder environment using the Python programming language. We used five real-world data sets (Table 2) and sixteen synthetic networks (Table 5), respectively, to evaluate the results. Additionally, the statistics include the node’s “ground-truth” community memberships. In these datasets, the proposed method is compared with fundamental algorithms like Louvain [49], Leiden [32], Bigclam [14], [17], CPM [18], Label propagation [35], and SLPA [34]. Table 1 lists these algorithms in brief.

**Evaluation Metrics**

We evaluate the community detection algorithms’ effectiveness and accuracy using three standard evaluation metrics. Modularity [50] is an internal metric for assessing community quality, whereas the F1_Score and NMI are external metrics for assessing community accuracy by comparing them to ground-truth communities [6]. The modularity measure in internal metrics, a popular benchmark for estimating the density in the community is derived from Girvan-Newman [50].
Table 1: The employed methods for PMCD evaluation

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Louvain</td>
<td>Louvain amplifies the modularity value of communities</td>
</tr>
<tr>
<td>Leiden</td>
<td>The Leiden method is an advancement of the Louvain</td>
</tr>
<tr>
<td>Bigclam</td>
<td>The probabilistic community detection method that scales to large networks</td>
</tr>
<tr>
<td>CPM</td>
<td>Find k-clique communities in a graph using the percolation method</td>
</tr>
<tr>
<td>LPA</td>
<td>The label propagation algorithm detects communities by network structure</td>
</tr>
<tr>
<td>SLPA</td>
<td>SLPA is an overlapping community discovery that extends the LPA</td>
</tr>
</tbody>
</table>

By dividing the projected community edges by the expected community edges, the modularity value is calculated. The identified community will perform better if there are more nodes inside the community and if the modularity score of the community is around 1. When comparing the frequency of properly recognizing the nodes in each community using the supplied ground truth data, the F1_Score is a well-known evaluation statistic used in community detection methods. The other outsider statistic is NMI, or mutual information, about the connection found among the recognized groups and the real world.

Real-World Datasets

Five real-world datasets are used in the experiments. Zachary’s karate club network [51] is the first dataset, containing 34 nodes, 78 connecting edges between them, and 2 ground-truth communities. This dataset contains social ties among university karate club members collected by Wayne Zachary in 1977. Dolphins’ online social network [52] is the second dataset, which contains 62 nodes, 159 connecting edges, and two ground-truth communities containing a list of all the links, where a link represents frequent associations between dolphins. The third dataset [53], with 105 nodes, 441 connecting edges, and 3 ground-truth communities, is based on data from the network of books about US politics published around the time of the 2004 presidential election. Edges between books represent frequent co-purchasing of books by the same buyers. The fourth dataset is the American football [4], with 116 nodes, 613 connecting edges, and 12 ground-truth communities. This network contains American football games between Division IA colleges during the fall of 2000. The fifth dataset is a large network generated using email data from a large European research institution [54], [55]. This network contains 1005 members of the institution as nodes, and 25571 edges contain emails sent between members of the institution and people outside of the institution. The dataset also assumes departments at the research institute as the nodes’ ground-truth community memberships. Each individual belongs to exactly one of the 42 departments at the research institute.

The real-world datasets analyzed during the current study are shown in Table 2, where N is the number of nodes, E is the number of edges, and K is the number of ground truths. These datasets are available in the network repository[^1], the KONECT project[^2], and the Stanford Network Analysis Project[^3] (SNAP).

Table 2: The specifics of the real-world dataset used

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>N (#Nodes)</th>
<th>E (#Edges)</th>
<th>K (#Ground_truth)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>34</td>
<td>78</td>
<td>2</td>
</tr>
<tr>
<td>Dolphin</td>
<td>62</td>
<td>159</td>
<td>2</td>
</tr>
<tr>
<td>Pol-Book</td>
<td>105</td>
<td>441</td>
<td>3</td>
</tr>
<tr>
<td>Football</td>
<td>115</td>
<td>613</td>
<td>12</td>
</tr>
<tr>
<td>Email-EU</td>
<td>1005</td>
<td>25571</td>
<td>42</td>
</tr>
</tbody>
</table>

[^1]: https://networkrepository.com/
[^2]: http://konect.cc/
[^3]: https://snap.stanford.edu/
The findings in Table 3 demonstrate that our method has more accuracy than other methods in terms of the internal metrics (modularity maximum and accuracy in the number of communities).

Additionally, Fig. 3 and 4 demonstrate that our suggested method has absolute superiority over probabilistic estimation and clique percolation methods and relative superiority over modularity optimization and label propagation methods in the external assessment criteria (NMI and F1_Score).

Table 3: Experimental results on real-world networks by the modularity metric (Q) and community number (CN)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Louvain</th>
<th>Leiden</th>
<th>Bigclam</th>
<th>CPM</th>
<th>LPA</th>
<th>SLPA</th>
<th>PMCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>0.415</td>
<td>0.116</td>
<td>0.204</td>
<td>0.215</td>
<td>0.354</td>
<td>0.371</td>
<td>0.397</td>
</tr>
<tr>
<td>Dolphin</td>
<td>0.518</td>
<td>0.134</td>
<td>0.185</td>
<td>0.321</td>
<td>0.456</td>
<td>0.470</td>
<td>0.522</td>
</tr>
<tr>
<td>Pol-Book</td>
<td>0.526</td>
<td>0.279</td>
<td>0.347</td>
<td>0.271</td>
<td>0.481</td>
<td>0.493</td>
<td>0.543</td>
</tr>
<tr>
<td>Football</td>
<td>0.604</td>
<td>0.257</td>
<td>0.381</td>
<td>0.283</td>
<td>0.552</td>
<td>0.596</td>
<td>0.632</td>
</tr>
<tr>
<td>Email-EU</td>
<td>0.432</td>
<td>0.226</td>
<td>0.207</td>
<td>0.162</td>
<td>0.274</td>
<td>0.303</td>
<td>0.507</td>
</tr>
</tbody>
</table>

Fig. 3: NMI assessment chart, compare PMCD by community detection models on five real-world data sets.

Fig. 4: F1_Score assessment chart, compare PMCD by community detection models on five real-world data sets.
Artificial Datasets

Utilizing an artificial network for evaluating community detection methods makes sense. Different methods can be used to generate artificial networks.

One of the most famous and often used strategies is the LFR benchmark [57]. The LFR benchmark builds types of artificial graphs with ground truth communities using density and dimension of communities. The network and community parameters can be set up before using LFR to simulate networks. The mixing parameter ($\mu$) is one of the essential LFR parameters. This variable controls how various communities interact. A high mixing parameter value ($\mu$), as indicated in Table 5, will reduce the network’s level of modularity ($Q_{\text{GT}}$). As a result, the LFR-generated datasets are divided into two categories based on the modularity measure and mixing parameter: sparse communities and dense communities. The average degree is another crucial element that might be raised to encourage more intercommunal interaction.

Table 4 displays the key characteristics of the LFR artificial networks. Table 5 details the dataset that was created using our LFR approach.

Table 5: The LFR artificial network properties

<table>
<thead>
<tr>
<th>Graph Name</th>
<th>N</th>
<th>k</th>
<th>$\gamma$</th>
<th>$\beta$</th>
<th>$\mu$</th>
<th>$Q_{\text{GT}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFR-1</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.05</td>
<td>0.895</td>
</tr>
<tr>
<td>LFR-2</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.10</td>
<td>0.844</td>
</tr>
<tr>
<td>LFR-3</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.15</td>
<td>0.800</td>
</tr>
<tr>
<td>LFR-4</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.20</td>
<td>0.739</td>
</tr>
<tr>
<td>LFR-5</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.25</td>
<td>0.699</td>
</tr>
<tr>
<td>LFR-6</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.30</td>
<td>0.647</td>
</tr>
<tr>
<td>LFR-7</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.35</td>
<td>0.603</td>
</tr>
<tr>
<td>LFR-8</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.40</td>
<td>0.560</td>
</tr>
<tr>
<td>LFR-9</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.45</td>
<td>0.504</td>
</tr>
<tr>
<td>LFR-10</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.50</td>
<td>0.460</td>
</tr>
<tr>
<td>LFR-11</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.55</td>
<td>0.407</td>
</tr>
<tr>
<td>LFR-12</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.60</td>
<td>0.364</td>
</tr>
<tr>
<td>LFR-13</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.65</td>
<td>0.321</td>
</tr>
<tr>
<td>LFR-14</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.70</td>
<td>0.275</td>
</tr>
<tr>
<td>LFR-15</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.75</td>
<td>0.229</td>
</tr>
<tr>
<td>LFR-16</td>
<td>1000</td>
<td>20</td>
<td>3</td>
<td>1.5</td>
<td>0.80</td>
<td>0.182</td>
</tr>
</tbody>
</table>

Table 6: Experimental results on sixteen LFR artificial networks by the modularity metric

<table>
<thead>
<tr>
<th>Mixing Parameter ( $\mu$ )</th>
<th>Louvain</th>
<th>Leiden</th>
<th>Bigclam</th>
<th>CPM</th>
<th>LPA</th>
<th>SLPA</th>
<th>PMCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>1.00</td>
<td>0.64</td>
<td>0.89</td>
<td>0.84</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.10</td>
<td>0.99</td>
<td>0.51</td>
<td>0.86</td>
<td>0.81</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
</tr>
<tr>
<td>0.15</td>
<td>0.96</td>
<td>0.42</td>
<td>0.77</td>
<td>0.72</td>
<td>0.93</td>
<td>0.95</td>
<td>0.98</td>
</tr>
<tr>
<td>0.20</td>
<td>0.93</td>
<td>0.41</td>
<td>0.73</td>
<td>0.66</td>
<td>0.88</td>
<td>0.87</td>
<td>0.91</td>
</tr>
<tr>
<td>0.25</td>
<td>0.89</td>
<td>0.37</td>
<td>0.69</td>
<td>0.57</td>
<td>0.83</td>
<td>0.85</td>
<td>0.91</td>
</tr>
<tr>
<td>0.30</td>
<td>0.86</td>
<td>0.23</td>
<td>0.59</td>
<td>0.53</td>
<td>0.76</td>
<td>0.79</td>
<td>0.84</td>
</tr>
<tr>
<td>0.35</td>
<td>0.82</td>
<td>0.22</td>
<td>0.57</td>
<td>0.43</td>
<td>0.69</td>
<td>0.72</td>
<td>0.81</td>
</tr>
<tr>
<td>0.40</td>
<td>0.79</td>
<td>0.19</td>
<td>0.47</td>
<td>0.31</td>
<td>0.52</td>
<td>0.64</td>
<td>0.72</td>
</tr>
<tr>
<td>0.45</td>
<td>0.70</td>
<td>0.14</td>
<td>0.31</td>
<td>0.29</td>
<td>0.43</td>
<td>0.56</td>
<td>0.73</td>
</tr>
<tr>
<td>0.50</td>
<td>0.53</td>
<td>0.12</td>
<td>0.25</td>
<td>0.24</td>
<td>0.41</td>
<td>0.48</td>
<td>0.66</td>
</tr>
<tr>
<td>0.55</td>
<td>0.50</td>
<td>0.09</td>
<td>0.19</td>
<td>0.22</td>
<td>0.36</td>
<td>0.33</td>
<td>0.52</td>
</tr>
<tr>
<td>0.60</td>
<td>0.44</td>
<td>0.05</td>
<td>0.12</td>
<td>0.14</td>
<td>0.28</td>
<td>0.29</td>
<td>0.39</td>
</tr>
<tr>
<td>0.65</td>
<td>0.37</td>
<td>0.04</td>
<td>0.08</td>
<td>0.07</td>
<td>0.24</td>
<td>0.18</td>
<td>0.38</td>
</tr>
<tr>
<td>0.70</td>
<td>0.29</td>
<td>0.01</td>
<td>0.05</td>
<td>0.03</td>
<td>0.15</td>
<td>0.14</td>
<td>0.30</td>
</tr>
<tr>
<td>0.75</td>
<td>0.21</td>
<td>0.00</td>
<td>0.03</td>
<td>0.01</td>
<td>0.11</td>
<td>0.09</td>
<td>0.24</td>
</tr>
<tr>
<td>0.80</td>
<td>0.15</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
<td>0.08</td>
<td>0.05</td>
<td>0.18</td>
</tr>
</tbody>
</table>
**Experimental on Artificial Networks**

In addition to actual graphs, we have also examined LFR artificial networks. We contrast the PMCD by the famous community detection model in Table 1 to demonstrate the result of the optimized loss function for a probabilistic estimate on the community detection utilizing modularity, F1_Score, and NMI measure. For this, sixteen LFR artificial networks are developed with various configurations of mixing parameters ($\mu$) ranging from 0.05 to 0.8, as indicated in Table 5. These networks are created in accordance with the attributes of synthetic networks listed in Table 4. Table 6's experimental findings demonstrate that the communities are dense for low mixing parameter range (e.g., $0.05 \leq \mu \leq 0.4$) and that the compared methods are almost correct in this situation.

However, the major contrast between the methods becomes more apparent when the mixing parameter's ($\mu$) value rises (e.g., $0.4 < \mu \leq 0.8$) and the communities are sparse, making it difficult to identify communities since the edges between communities' rise.

As can be seen in Figs 5 and 6, when the mixing parameter value increases, certain algorithms have NMI and F1_Score values that are equivalent to zero. The majority of frequently used approaches in the range of 0.5 to 0.8 are outperformed by the suggested method.

**Conclusion**

We proposed a probabilistic motif-based method for detecting communities in complex networks. Due to the complexity of combining probabilistic approaches in motif structure, recent community detection methods have given the latent variable of the probabilistic model less consideration. However, the proposed approach leverages the intensity of the node's participation in the community and the relationship of at least two linked edges between three nodes (triangular motif structure) to estimate the hidden variable of the probabilistic model. The research maximized the likelihood function and extracted the model's latent parameters using the well-known block coordinate ascent technique. The association between node membership in communities and edge density is another aspect that helps in the examination of newly detected communities; three nodes are more likely to create a motif structure when seen in various communities. Overlapping in the identification of communities is another benefit of PMCD; according to the findings, communities that overlap have a greater density of triangular motifs. We employed 16 artificial graphs and 5 real graphs to evaluate the performance of the suggested method. In comparison to the other six methods, PMCD was able to achieve a sufficient quorum on real-world networks and surpass them in terms of internal and external assessment criteria. Synthetic network assessments further show that the suggested strategy performs better in sparse datasets than other approaches. Furthermore, a review of the complexity of the execution time reveals that the suggested method outperforms previous approaches. Future research can develop PMCD. A probabilistic generative model can be used to estimate edge weight while taking a latent parameter into account. Also, the suggested method can be enhanced by utilizing network node properties in order to give a more precise description of the found communities.

**Author Contributions**

H. Hajibabaei, V. Seydi, and A. Koochari contributed to the research design and implementation, the analysis of the results, and the writing of the manuscript.

**Acknowledgment**

We thank the editor and all the anonymous reviewers.
Conflict of Interest
The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancy, have been completely observed by the authors.

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMCD</td>
<td>probabilistic motif-based community detection</td>
</tr>
<tr>
<td>NMI</td>
<td>Normalized Mutual Information</td>
</tr>
<tr>
<td>LFR</td>
<td>Lancichinetti–Fortunato–Radicchi Benchmark</td>
</tr>
<tr>
<td>CPM</td>
<td>Clique Percolation Method</td>
</tr>
<tr>
<td>LPA</td>
<td>Label Propagation Algorithm</td>
</tr>
<tr>
<td>SLPA</td>
<td>Speaker-Listener Label Propagation Algorithm</td>
</tr>
<tr>
<td>QTG</td>
<td>Ground Truth Modularity</td>
</tr>
<tr>
<td>SNAP</td>
<td>Stanford Network Analysis Project</td>
</tr>
</tbody>
</table>

References


Biographies

Hossein Hajibabaee received his M.Sc. in Software Engineering in 2014. Currently, he is a Ph.D. candidate in Computer Engineering majoring in artificial intelligence, and is teaching as a teaching assistant at the Islamic Azad University of Science and Research Branch. His areas of interest are social network analysis and deep learning.

Vahid Seydii is a Senior Research Fellow in the School of Ocean Science at Bangor University in Data Science (DS) and Machine Learning (ML). Before Bangor, He was an Assistant Professor at the Department of AI at Azad University South Tehran Branch (Feb 2014 - Sep 2020) and an award-winning lecturer (Oct 2010 Feb 2014). Vahid received a B.Sc. (2005) in software engineering, and M.Sc. (2007), and Ph.D. (2014) in AI, from the Department of Computer Science at Azad University, Science and Research Branch, Tehran Iran. He has been awarded a Global Talen endorsement from the UK Royal Society (2023). His current research fellowship (2020), and a merit-based scholarship for attending the school of AI, Rome, Italy (2019), Also, He has achieved a full scholarship Award from Azad University (2010-2014), KNTU ISLAB Research Fellowship (2007-2010) He secured the first rank among the graduates from 2004-2005. His current research focuses on dedicating machine-learning methods to analyze data associated with digital oceanography, especially in the offshore renewable energy section.

- Email: h.hajibabaee@srbiau.ac.ir
- ORCID: 0000-0005-8063-981X
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA
Abbas Koochari received his Ph.D. in Computer Engineering majoring in artificial intelligence. He is currently an assistant professor and a member of the scientific staff of Islamic Azad University, Science and Research Branch. His areas of interest are image processing, machine vision, speech and natural language processing, and deep learning.

- Email: koochari@srbiau.ac.ir
- ORCID: 0000-0003-0584-6470
- Web of Science Researcher ID: NA
- Scopus Author ID: 36005396600
- Homepage: https://srb.iau.ir/faculty/a-koochari/en

How to cite this paper:

DOI: 10.22061/jecei.2023.9931.663
URL: https://jecei.sru.ac.ir/article_2013.html