Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 3): pp. 4728-4739
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.03.484

Investigation Of Spectral Graph Theory For Graph Clustering And Community Detection

Shilpy Tayal Asst. Professor, Department of Mathematics, Graphic Era Hill University, Dehradun Uttarakhand India.

Abstract

This study investigates how graph grouping and community detection can be accomplished using spectral graph theory. The eigenvalues and eigenvectors of a graph's adjacency or Laplacian matrix are used by spectral graph theory to uncover structural characteristics and underlying patterns. Data analysis and network science core objectives include clustering and community discovery, which seek to locate communities of connected nodes in a graph. Spectral graph theory offers important insights into the structure and connection patterns of the network by examining its spectrum. This study explores numerous spectral clustering algorithms and evaluates how well they identify communities inside various kinds of graphs, including normalised cuts, spectral embedding, and modularity optimisation. It also investigates how spectral graph clustering algorithms perform in relation to graph characteristics like sparsity and size. The findings of this study advance knowledge of spectral graph theory and its practical application to graph clustering and community detection problems. For more precise community discovery, the suggested method makes use of a probability matrix and an enhanced spectral clustering algorithm. The approach first builds a probability matrix by using the Markov chain to determine the transition probabilities between nodes. The mean probability matrix is then used to create a similarity graph. The NCut goal function is then optimised to accomplish community detection. On both synthetic and actual networks, comparisons are made between the proposed algorithm and existing techniques like SC, WT, FG, FluidC, and SCRW to assess its efficacy. The suggested technique produces more precise community detection and demonstrates higher overall clustering performance, according to experimental data.

Keywords: Graph Theory, Clustering, Probability Matrix, Spectral Clustering Algorithm.

I. Introduction

Basic network analysis tasks like graph clustering and community detection have numerous applications in areas including social network analysis, biology, and recommendation systems. Finding node clusters with strong connectedness and similarity in a graph is the aim. These communities offer important insights on the composition and arrangement of complicated networks. A potent paradigm for comprehending the structural characteristics of graphs and deriving useful information from them has emerged: spectral graph theory. It makes use of the eigenvalues and eigenvectors of the adjacency or Laplacian matrix of a network in order to expose hidden patterns and connectedness. By converting the network into a spectral domain, spectral graph theory provides a logical method for grouping graphs and identifying communities. Community discovery has traditionally made use of conventional spectral clustering algorithms like normalised cuts and spectral embedding. However, they frequently have the issue of producing similarity graphs that contain inaccurate community information. This may result in subpar clustering outcomes and insufficient precision in community discovery tasks. This inquiry intends to increase the comprehension and usefulness of spectral methods in network analysis by investigating the use of spectral graph theory in graph clustering and community detection and providing a novel algorithm. The findings of this study could enhance community detection methods and advance the area of network science as a whole.

By dividing a network into various clusters based on node interactions, community identification can disclose the hierarchical structure of the network and make it easier to store, analyse, and analyse network data. The spectral bisection algorithm, the graph segmentation algorithm, the heuristic algorithm, and the objective optimisation algorithm are only a few of the techniques that have been created for community detection.

The spectral bisection algorithm divides the network recursively using eigenvalues and eigenvectors, drawing on the spectral graph theory. The goal of the graph segmentation algorithm is to separate the graph into sections with high internal connectivity and low external connectivity. To find communities, heuristic algorithms use iterative processes based on local optimisation criteria. Algorithms for objective optimisation aim to improve a particular objective function associated with community structure.

II. Review of Literature

A crucial area of research in the study of complex networks is community detection. Spectral clustering stands out among the conventional methods as a well-liked approach for community detection based on network topology [2]. By identifying the primary eigenvectors from the network's similarity matrix, this method uses eigen-decomposition to discover communities. In addition to working with a variety of data types, spectral clustering also makes use of dimensionality reduction to improve computation speed. As a result, scientists have been actively researching and developing spectral grouping.

For instance, a multisimilarity spectral approach for clustering dynamic networks was presented by Qin et al. [6]. To find communities, this strategy bootstraps using a variety of similarity metrics. A strategy for agglomerative spectral clustering that takes conductance and edge weights into account was put out by Ulzii and Sanggil [7]. Based on edge weights and eigenvector space, it combines the nodes that are the most comparable. A semi-supervised spectral clustering algorithm was created as a result of research by Ding et al. [8] into the connection between nonnegative matrix factorization and spectral clustering.

These instances serve as a reminder of the continuous work being done to improve spectral clustering techniques for community detection. To increase the accuracy, scalability, and application of spectral clustering approaches in diverse situations, researchers are enhancing and expanding upon them. These techniques aid in the creation of better community detection algorithms by utilising the benefits of spectral clustering and incorporating cutting-edge improvements.

Accurate community detection in spectral clustering depends on the creation of a trustworthy similarity matrix. It is possible to ignore hidden associations when using the conventional method of employing Euclidean distance between nodes in the similarity matrix, which results in incomplete community information and subpar clustering performance. Thus, strengthening the similarity matrix's generation becomes essential to raising the performance of the spectral clustering technique.

Nataliani and Yang [9] developed a novel approach based on the propagation of neighbour relations to construct an affinity matrix in order to overcome this difficulty. By using this technique, it becomes more likely that two points that ought to be in the same cluster will be identical. The distance requirement, however, makes it vulnerable to the impact of outlier or noisy points. In a different method, Beauchemin [10] constructs affinity matrices using a density estimator based on K-means with subbagging. When there is multiple proximity, it could struggle to function properly. In order to analyse the similarity matrix, Zhang and You [11] created a method that uses random walks. In this method, pairwise similarity is affected not only by the two points themselves but also by their surroundings. However, this approach necessitates manual adjustment of clustering instability.

Nataliani and Yang [9] developed a novel approach based on the propagation of neighbour relations to construct an affinity matrix in order to overcome this difficulty. By using this technique, it becomes more likely that two points that ought to be in the same cluster will be identical. The distance requirement, however, makes it vulnerable to the impact of outlier or noisy points. A different strategy used by Beauchemin [10]The problem of creating a similarity graph that effectively reflects the underlying community structure remains unsolved despite the multiple community detection techniques based on optimising similarity graphs that have been presented. Because of this, this study introduces the idea of a probability matrix, concentrates on computing similarity based on the transition probability between nodes, and then presents an enhanced spectral clustering community discovery algorithm utilising the probability matrix.

This approach seeks to create a more trustworthy similarity matrix that accurately depicts the links between nodes by utilising the transition probabilities. The constraints of conventional similarity graph generation techniques are addressed in the suggested approach, which provides a viable option to improve the performance of spectral clustering for community discovery.uses a K-means-based density estimator with subbagging to create affinity matrices. When there is multiple proximity, it could struggle to function properly. In order to analyse the similarity matrix, Zhang and You [11] created a method that uses random walks. In this method, pairwise similarity is affected not only by the two points themselves but also by their surroundings. However, this technique suffers from clustering instability and needs a human threshold setting for nearby nodes.

III. Spectral Clustering Algorithm: An Improvement

The similarity between nodes is computed during the building of the similarity graph in spectral clustering. In this part, we take a different tack by computing similarity based on the likelihood that nodes will transition. The possibility of switching from one node in the graph to another is represented by the transition probability.

a. Probability Transition

A mathematical model known as a Markov chain illustrates a stochastic process with a series of states. The current state is the only factor that influences each subsequent state, and future states are unrelated to earlier ones [12]. Transition probabilities control the change between states. The 1st transition probability describes the possibility of moving from node i to node j after a single step in the context of a network N with n nodes, represented by an adjacency matrix W. It captures the likelihood of a node in the network shifting from one node to another.

The first transition probability is defined formally as follows: The relationship between nodes i and j is represented by the matrix element w_ij of an adjacency matrix W. The ratio of the connection strength between node i and node j (w_ij) to the total of the connection strengths from node i to all other nodes in the network is used to calculate the first transition probability from node i to node j, denoted as P_ij:

$$\mathrm{P}_{\{\mathrm{ij}\}} = \left\{ \sum \; \frac{\mathrm{w}_{\mathrm{ij}}}{\sum_{\{\mathrm{k}=1\}}{}_{\{\mathrm{ik}\}}^{\{\mathrm{n}\}} \mathrm{W\ ij}} \right\} \qquad (1)$$

The possibility of moving from node i to node j after one step in the Markov chain is expressed in terms of this probability. It is essential to the process of creating the probability matrix, which is employed in the proposed spectral clustering technique for community discovery and records the probabilities of transitions between nodes.

In this equation, W stands for the network's adjacency matrix, and dW0, dW1,..., and dWn-1 are its row sums. The reciprocal of each diagonal element is used to calculate the inverse of the diagonal matrix DW(-1).

The probability of moving from node i to node j in a single Markov chain step is shown by entry prij in the first transition matrix Pr. Based on their connectedness, it measures the likelihood of travelling between network nodes.

The idea of transition probabilities can be expanded to include l-th transition probabilities by building on this idea. The likelihood that a node i will reach node j after l steps in the Markov chain is represented by the l-th transition probability.

b. Probability Matrix

The likelihood of a transition between node in a network are represented by the probability matrix. The first transition probability includes the direct connection between a node and its surrounding nodes, but it may not take into consideration any hidden connections with nodes that are not adjacent.

We provide a technique for building the probability matrices based on the gathering of balanced multiorder transit matrices in order to overcome these restrictions. The probability matrix, P, is characterised as follows:

$$P = \sum_{i=1}^{n} \mathrm{Wi\ Prj} \qquad (2)$$

In this equation, Pr stands for the first transition matrix, as stated before, and Prl for the lth transition matrix, which was produced by iteratively multiplying Pr. Indicating the number of steps taken into consideration for examining the connectedness of the network, L stands for the maximum order of the multistep transition.

c. **Probability Mean Matrix**

The time scale is very important for figuring out how similar the nodes are. Different networks may have different ideal time scales for similarity calculations, though. A fixed time scale may cause the analysis to be inaccurate and mistaken.

We provide the idea of a mean probability matrix to lessen the effects of the parameters, t and ϵ, We generate the mean probability matrix by taking into account several time scales and averaging the probability matrices derived from various time ranges.

With regard to the similarity computation, we specifically obtain probability matrices, indicated as P, utilising different time scales. Every probability matrix has a corresponding time scale. We create the mean probability matrix by averaging these probability matrices.

$\text{Probability}_{\text{mean}}(\text{Pm}) = (1/N) *$ ∑_{i = 1}^{N} P_i (3)

The time scale parameter, represented as t, indicates the number of probability matrices that are added together to get the mean in addition to the size of the time scale for each probability matrix. We efficiently average out the mistakes resulting from variations in t and by adding multiple probability matrices with various time scales.

The mean probability matrix aids in minimising errors and the effects of selecting a particular value for t. As a result, t's value can be picked at random from a set of values. To control computational complexity in our method, we set t to fall between [5, 13].

IV. Spectral Clustering Algorithm with Mean Probability Matrix Improvement

1. Similarity Graph construction

The mean probability matrix PM is used as the foundation for the similarity matrix, abbreviated as WP. The similarity between nodes i and j in a given network N, where PM stands for the mean probability matrix of N, can be defined as:

$$\text{WP}(\text{i},\text{j}) = \begin{cases} \text{Pmij} & \text{where i } \neq \text{ j} \\ 0, & \text{where i } = \text{ j} \end{cases} \qquad (4)$$

The item in the mean probability matrix PM that corresponds to the similarity between nodes i and j is represented in this equation by PM(i, j). The diagonal entries of PM, PM(i, i) and PM(j, j), respectively, capture the self-similarity of nodes i and j.

The similarity metric WP(i, j) is created by normalising the entry PM(i, j) with the square root of the product of PM(i, i) and PM(j, j). This normalisation assures that the similarity value is between [0, 1] and takes into account the self-similarity of each individual node.

For any vector f, the matrix multiplication LW can be expressed as:

$$\begin{aligned} f^T L_W f = f^T D f - f^T W_P f &= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{Pij} f_i f_j, \\ &= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} w_{Pij} f_i f_j + \sum_{j=1}^{n} d_j f_j^2\right), \\ &= \frac{1}{2}\sum_{i,j=1}^{n} w_{Pij}\left(f_i - f_j\right)^2. \end{aligned}$$

The resulting matrix, LW, which was created by multiplying the Laplacian matrix L with the similarity matrix W, is an example of a Laplacian matrix. The connectivity and structural characteristics of the graph are captured by this matrix.

A spectral clustering network can be created using the similarity matrix WP, which was created based on the mean probability matrix PM. The matrix WP depicts a graph, with nodes denoting the network's constituent entities and edges denoting their pairwise similarity.

2. Step wise execution of Algorithm

The technique provides the main steps of the enhanced spectral clustering technique.

Algo:

The following steps describe how to obtain K communities using spectral clustering given a network N, an adjacency matrix W, the desired number of communities K, a time scale t, and a set of weights ws:

(1) Determine the transition probabilities between nodes based on the adjacency matrix W and time scale t. This is done by computing the first transition matrix Pr using equation.

(2) Using equation, which averages the probability matrices acquired from various time scales, compute the mean probability matrix P.

(3) Use equation to construct the similarity matrix WP, where the mean probability matrix P is utilised to specify the pairwise similarity between nodes.

(4) Create the unnormalized Laplacian matrix LW using WP's characteristics.

(5) Create the normalised Laplacian matrix Ln, which is represented by the equation Ln = D(-1/2) * LW * D(-1/2), where D is the diagonal matrix storing the degrees of nodes.

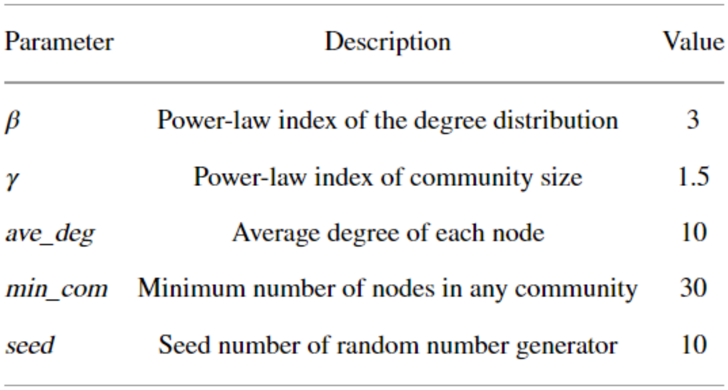
(6) Determine the first K U, or eigenvectors, of Ln. These eigenvectors are employed for community discovery because they capture the spectral data of the graph.

(7) Treat the rows of U as nodes and divide them into K communities using the K-means clustering technique. The K-means algorithm iteratively improves the cluster assignments by assigning each row of U to the closest cluster centroid.

V. Results and Analysis

The synthetic networks used in the LFR benchmark networks were produced computationally. With the help of these networks, numerous factors can be changed to create networks with diverse properties. The mixing parameter and the network size N are the two main parameters that are used in the experiments to evaluate performance.

The average rate at which nodes from various communities are connected is gauged by the mixing parameter. The degree of connectedness between communities in the network is quantified. The value of μ is a number between 0 and 1, where μ=0 denotes completely disconnected communities with no connections between them and μ= 1 denotes a fully interconnected network where each node has an equal chance of connecting to any other node, regardless of community.



Parameter	Description	Value
β	Power-law index of the degree distribution	3
γ	Power-law index of community size	1.5
ave_deg	Average degree of each node	10
min_com	Minimum number of nodes in any community	30
seed	Seed number of random number generator	10

Figure 1: Hyper parameter of LFR benchmark

The performance comparison of the six methods on the mixing parameter is shown in Figure 1. After analysing the data, we see that the ISCP algorithm's NMI trend looks to be smoother than that of the other techniques. Furthermore, ISCP significantly outperforms the other five algorithms in terms of NMI. It is stated in [16] that a greater NMI value equates to a higher grade of community detection. In conclusion, ISCP achieves a greater community clustering impact than the other five techniques.

Overall, ISCP shows improved stability and quicker convergence. It continuously surpasses the competition in terms of community detection precision and displays more gradual performance changes as the mixing parameter is altered. These results demonstrate the improved clustering.

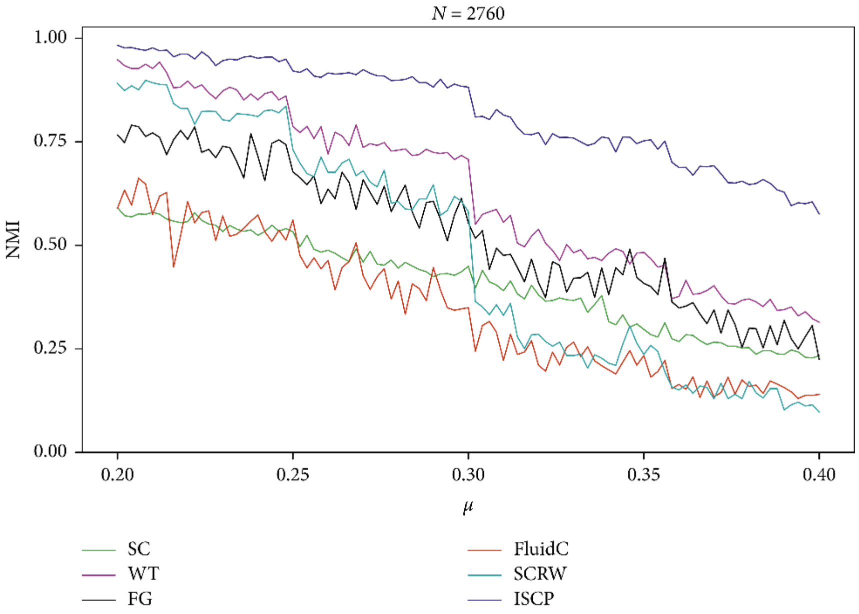


Figure 2: Performance comparison of NMI six algorithms

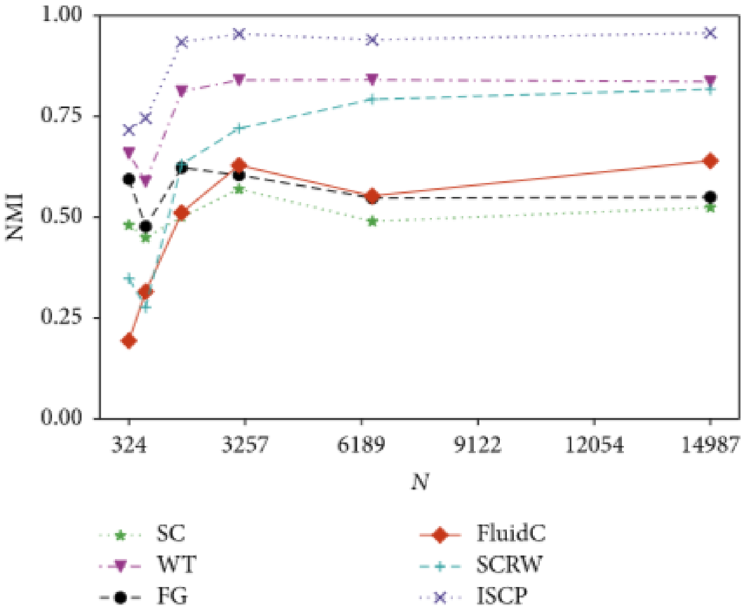


Figure 3: NMI Different Size of NMI Algorithm

The performance comparison of the six methods on various network sizes, abbreviated as N, is shown in Figure 3. It is clear from the findings that the ISCP algorithm regularly achieves a higher NMI than the other five techniques. Additionally, the NMI of ISCP exhibits an expanding trend as the network size grows. Notably, the NMI stabilises and stays around 0.9 when the network size reaches 5000 or greater. According to these results, ISCP's clustering performance is still higher regardless of the network size order of magnitude, whether it is between 1000 and 10,000 nodes.

VI. Conclusion

In the area of community detection, spectral clustering is usually regarded as a key algorithm. However, conventional similarity graphs employed in spectral clustering frequently contain a sizable amount of inaccurate community information, which results in subpar community detection performance. This work discusses the idea of a probability matrix and suggests an enhanced spectral clustering algorithm called ISCP to address this problem. Numerous tests on benchmark networks and actual networks show that ISCP surpasses the majority of conventional community discovery methods and produces more precise grouping outcomes in complicated networks. It's vital to remember that, despite its effectiveness, ISCP can be computationally expensive in terms of time and resources needed. The transition probability matrix must be multiplied by t times in order to create the similarity matrix in ISCP for a network N with n nodes and a time scale of t. The time complexity of the algorithm can still be O(n3ltb) even with optimisation methods like the Fast Power algorithm. The computation of the similarity matrix gets more time-consuming and demands large memory resources as the network size grows. In addition, it should be emphasised that ISCP is tailored for nonoverlapping complex networks and that clustering of overlapping networks is still a topic that needs more investigation.

Future study should therefore concentrate on optimising the computational complexity of ISCP and investigating methods to cut down on time and space needs while maintaining the accuracy of its clustering. Additionally, it would be beneficial to expand the application of community discovery techniques by creating algorithms that can handle overlapping networks.

References:

1. M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” Proceedings of the National Academy of Sciences, vol. 99, no. 12, pp. 7821–7826, 2002.
2. A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: analysis and an algorithm,” Advances in Neural Information Processing Systems, vol. 14, pp. 849–856, 2002.
3. B. W. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,” Bell System Technical Journal, vol. 49, no. 2, pp. 291–307, 1970.
4. M. Rosvall and C. T. Bergstrom, “Maps of random walks on complex networks reveal community structure,” Proceedings of the National Academy of Sciences, vol. 105, no. 4, pp. 1118–1123, 2008.

5. D. He, J. Liu, D. Liu, D. Jin, and Z. Jia, “Ant colony optimization for community detection in large-scale complex networks,” in Proceedings of the Seventh International Conference on Natural Computation, pp. 1151–1155, Shanghai, China, July 2011.
6. X. Qin, W. Dai, P. Jiao, W. Wang, and N. Yuan, “A multi-similarity spectral clustering method for community detection in dynamic networks,” Scientific Reports, vol. 6, no. 1, pp. 31454–31465, 2016.
7. N. Ulzii and K. Sanggil, “Social network community detection using agglomerative spectral clustering,” Complexity, vol. 2017, Article ID 3719428, 10 pages, 2017.
8. S. Ding, H. Jia, M. Du, and Y. Xue, “A semi-supervised approximate spectral clustering algorithm based on HMRF model,” Information Sciences, vol. 429, pp. 215–228, 2018.
9. Y. Nataliani and M. S. Yang, “Powered Gaussian kernel spectral clustering,” Neural Computing and Applications, vol. 31, no. 1, pp. 557–572, 2019.
10. M. Beauchemin, “A density-based similarity matrix construction for spectral clustering,” Neurocomputing, vol. 151, pp. 835–844, 2015.
11. X. Zhang and Q. You, “An improved spectral clustering algorithm based on random walk,” Frontiers of Computer Science in China, vol. 5, no. 3, pp. 268–278, 2011.
12. J. Rhodes and A. Schilling, “Unified theory for finite Markov chains,” Advances in Mathematics, vol. 347, pp. 739–779, 2019.
13. L. Hagen and A. B. Kahng, “New spectral methods for ratio cut partitioning and clustering,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 11, no. 9, pp. 1074–1085, 1992.
14. J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 2000.
15. A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” Physical Review E, vol. 78, no. 4, pp. 46110–46115.
16. P. Zhang, “Evaluating accuracy of community detection using the relative normalized mutual information,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2015, no. 11, pp. 11006–11013, 2015.
17. M. E. J. Newman, “Modularity and community structure in networks,” Proceedings of the National Academy of Sciences, vol. 103, no. 23, pp. 8577–8582, 2006.
18. P. Pons and M. Latapy, “Computing communities in large networks using random walks,” Journal of Graph Algorithms and Applications, vol. 10, no. 2, pp. 191–218, 2006.
19. M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” Physical Review E, vol. 69, no. 6, pp. 133–138, 2004.
20. F. Parés, D. Garcia Gasulla, A. Vilalta et al., “Fluid communities: a competitive, scalable and diverse community detection algorithm,” in Proceedings of the Complex Networks & Their Applications, pp. 229–240, Cham, Switzerland, November 2017.