Community Detection Based On Markov Edge Similarity

Samya Shah
Ahmedabad University

Diya Patel
Ahmedabad University

Abstract

Community detection can discover the cluster structure hidden in complex networks, which helps people predict network behavior and understand network functions. It is one of the current research hotspots. In this brief, we propose a Markov similarity enhancement method, which obtains the steady-state Markov similarity enhancement matrix through the Markov iterative state transition of the initial network. According to the Markov similarity index, the network is divided into initial community structure. Then, we merge the small communities into its closely connected communities to obtain the final community. The results from our base paper [8] are used to compare the results of our proposed algorithm, numerical simulation experiments show that the proposed algorithm has a good community detection effect.

Keywords

Community detection, Markov similarity, local similarity.

1. Introduction

In this brief, we introduce the Markov chain into community detection. The method discovers communities through steady-state Markov similarity enhancement matrix. Experiments show that the method can obtain stable and high-performance communities in different real networks. The main contributions of this brief are as follows.

1. Propose a new algorithm based on Markov similarity enhancement method, which performs Markov similarity iterative transfer to the initial network, obtains a steady-state Markov similarity enhancement matrix, and conducts community detection on this basis.
2. Propose a Markov similarity index to obtain initial division of the network, and then merge the small communities with the closely connected communities to obtain the final community.

The rest parts of this brief are as follows. Section II is the problem formulation, Section III describes the new algorithm in detail. Section IV gives numerical simulation Section V result analysis. The conclusions are given in Section VI.

2. Problem Formulation

We use Markov to process the network. Distinguishing from previous relevant studies, we use Markov chain to transfer adjacency matrix, and define the final obtained steady-state matrix as the Markov similarity enhancement matrix. Our goal is to extract the Markov similarity enhancement matrix from the network, and then propose the Markov similarity index between nodes to obtain the initial community structure of the network. Finally, we merge the small community with its closely connected community to obtain final community.

For an undirected and unweighted network $G = (V, E)$, where $V = v_1, v_2, \ldots, v_N$ represents the set of nodes, E represents the set of edges between nodes, N represents the number of nodes in the network, $\Gamma(v_i)$ indicates the set of neighbors of node. Therefore, $|E|$ means the number of edges between nodes in the network, $|\Gamma(v_i)|$ means the number of neighbors of node v_i. Let $a_{i,j}$ as the connection state of node v_i and node v_j. If there is an edge between the node v_i and the node v_j, then $a_{i,j} = 1$, otherwise $a_{i,j} = 0$. Then, the adjacency matrix A can be represented as:

$$A = [\begin{matrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{22} & \cdots & a_{2,N} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{matrix}], c$$

$$a_{ij} = \begin{cases} 1, a_{ij} \in E \\ 0, a_{ij} \notin E \end{cases}$$

(1)

Algorithm 1 Community Detection in Networks

```
Input: Network $G(V, E)$, Similarity Matrix $S$, Thresholds $\lambda$, $\tau$
Output: The set of local communities $C$
function GetAdjacencyMatrix($G$)
    Compute adjacency matrix for $G$
    return $adjacency\_matrix$
end function
function GetJaccardCoefficients($G$)
    Compute Jaccard coefficients for all node pairs
    return $adjacency\_matrix$
end function
function NormalizedCoefficients($jaccard\_matrix$)
    Normalize $jaccard\_matrix$ by row sums
    return $normalized\_matrix$
end function
function MatrixOperation($A$, $S$, $G$)
    Calculate $S$ to the power of average node degree $n$
    Perform matrix multiplication $A \times S^n$
    return $result$
end function
function FormInitialCommunities($G$, $result\_matrix$, $\lambda$, $\tau$)
    Initialize each node as its own community
    Update $result\_matrix$ based on $\tau$
    while true do
        Find the pair of communities with the highest average similarity
        Merge the communities with the highest average similarity score
        Break if no more communities to merge or if all communities meet $\lambda$
    end while
    return $initial\_communities$
end function
function MergeSmallCommunities($G$, $initial\_communities$, $\lambda$)
    Classify communities into $small\_communities$ and $large\_communities$ based on $\lambda$
    Initialize $unassigned\_nodes$ to empty
    for each $small\_community$ in $small\_communities$ do
        Find best matching $large\_community$ to merge with
    end for
    Reevaluate $unassigned\_nodes$ and merge with best matching $large\_community$
    Combine all communities after merging
    return $final\_communities$
end function
```

where a_{ij} represents the node pair composed by node i and node j.

3. Community Detection Based on Markov Edge Similarity (MES)

We propose a Markov similarity enhancement method, which obtains the steady-state Markov similarity enhancement matrix through the Markov iterative state transition of the initial network. For the convenience of description, we abbreviate the method proposed in this brief as MES.

3.1. MES Algorithm Framework

The algorithm in this brief is mainly divided into 4 steps.

Perform Markov similarity iterative transfer to the initial network to obtain a steady-state Markov similarity enhancement matrix. Obtain the Markov similarity index between nodes from the Markov similarity enhancement matrix, obtain the most similar node of each node, and then obtain the initial community. In the initial community, merge the small communities with the closely connected communities to form the final community.

3.2. Markov Similarity Enhancement Matrix

From the initial state of the network, we do iterative transfer of Markov similarity to obtain a steady-state Markov similarity enhancement matrix. We use adjacency matrix A as the initial state matrix of the network, and use Jaccard local similarity index

$$\mathrm{sim}(v_i, v_j) = \frac{\Gamma(v_i) \cap \Gamma(v_j)}{\Gamma(v_i) \cup \Gamma(v_j)}, \quad v_j \in \Gamma(v_i), v_j \neq v_i \tag{1}$$

to measure the similarity between nodes.

3.3. Markov Similarity Index and Initial Community Division

We define $S_{i,j}$ as the similarity index of node v_i and v_j. For any node v_i, if

$$j = (S_{i,x}, x \in 1, 2, \ldots, N \text{ and } x \neq i) \tag{2}$$

then the node v_j is its most similar node.

We define the state transition matrix of the network

$$S = \left[\begin{array}{cccc} s_{1,1} & s_{1,2} & \cdots & s_{1,N} \\ s_{2,1} & s_{22} & \cdots & s_{2,N} \\ \cdots & \cdots & \cdots & \cdots \\ s_{N,1} & s_{N,2} & \cdots & s_{N,N} \end{array}\right]$$

$$s_{ij} = \begin{cases} 1, s_{ij} \in E \\ 0, s_{ij} \notin E \end{cases}$$

where $s_{i,j}$ represents the similarity of nodes v_i and v_j, which can be understood as the probability that these two nodes are in the same community, or the probability of transition from the node i state to the node j state. Because the sum of the probabilities of each state transitioning to other states is 1, so we need to do a normalization operation on the matrix S. That is, $s_{i,j} = s_{i,j}/\sum(s_{i,k}), k = 1, 2, \ldots, N$. We define $r_{i,j}(n)$ as the Markov similarity of node v_i to node v_j after n rounds of iterations, then

$$\begin{aligned} r_{i,j}(n) &= P(X_n = v_j | X_0 = v_i) \\ &= \sum_{k=1}^{N} P(X_{n-1} = v_k | X_0 = v_i) * \\ &P(X_n = v_j | X_{n-1} = v_k) \\ &= \sum_{k=1}^{N} P(X_{n-1} = v_k | X_0 = v_i) \\ &P(X_n = v_j | X_{n-1} = v_k) \\ &= \sum_{k=1}^{N} r_{i,k}(n-1) s_{k,j} \end{aligned}$$

where the current state of Markov is only related to the previous state. Let $R(n) = r_{i,j}(n); i \in N, j \in N$, then the matrix form of equation is (3) $R(n) = R(n-1)S$. Therefore, $R(n) = R(0)S^n$. We use normalized adjacency matrix A as the initial state $R(0)$, and let $\bar{S}$ denote $R(n)$, then the final Markov similarity enhancement matrix is

$$\overline{S} = A\mathbb{S}^n$$

State transition matrix S is a local similarity index, which ignores the high-order similarity information. Knowing that $R(1) = AS$, then $R(1)i, j = \sum v_k \in \Gamma(v_i) R(0)i, ksk, j$ captures the local similarity within 2-order of the node, and $R(2)i, j = \sum v_k \in \Gamma(v_i) \sum_{v_l \in \Gamma(v_k)} R(0)i, lsl, j$ captures the local similarity within 3-order of the node. Therefore, when n is large, $\bar{S} = R(n)$ can capture the information of global node similarity through interaction. This shows that, compared with S, the Markov similarity enhancement matrix $\bar{S}$ is able to capture the information of higher-order relation and converge to steady state through iteration. So, it is a global similarity metric with more information, and therefore will have better performance in community detection. Considering the computational cost, we prefer to obtain the Markov similarity enhancement matrix at a small number of iteration rounds. Therefore, we empirically set the maximum number of iteration rounds to $\lfloor |E|/N \rfloor$.

3.4. Formation of Initial Communities

The process of forming initial communities is based on the concept of similarity thresholds. Each node in the network is initially considered as a separate community. The algorithm then iteratively examines the similarity between each pair of nodes or communities, merging them if their similarity exceeds a predefined threshold. This process can be mathematically represented as follows:

Let $G = (N, E)$ be a graph where N is the set of nodes and E is the set of edges. The similarity between two nodes i and j is denoted by $S(i, j)$. The initial communities are formed by the following procedure:

1. Initialize each node as a separate community, i.e., $C_i = \{i\}$ for all $i \in V$.

2. For each pair of nodes (i, j), if $S(i, j) > \theta$, merge their communities. Here, θ is the similarity threshold.

3. Repeat step 2 until no more communities can be merged based on the threshold θ.

The similarity threshold θ is a critical parameter that determines the granularity of the community structure. A higher value of θ results in a larger number of smaller communities, while a lower value leads to fewer, larger communities.

3.5. Merging Small Communities

After the initial communities are formed, small communities are merged into larger ones to refine the community structure. This process aims to enhance the cohesion within communities by merging those that are closely connected. The merging process can be described as follows:

Let $C = \{C_1, C_2, \ldots, C_k\}$ be the set of initial communities. The process of merging small communities involves:

1. Calculate the average similarity between communities, defined as $S(C_i, C_j) = \frac{1}{|C_i||C_j|}\sum_{u\in C_i, v\in C_j} S(u,v)$, where $|C_i|$ and $|C_j|$ are the sizes of communities C_i and C_j, respectively.
2. Find the pair of communities (C_i, C_j) with the highest average similarity $S(C_i, C_j)$.
3. If $S(C_i, C_j) > \lambda$, merge C_i and C_j, where λ is the merging threshold.
4. Repeat steps 1-3 until no more communities can be merged based on the threshold λ.

The merging threshold λ is another important parameter that influences the final community structure. Adjusting λ allows for control over the degree of cohesion within the resulting communities but in our algorithm we have kept it to 2.

4. Numerical Evaluation

We carried out a number of numerical simulation experiments on real networks to test the efficiency of our algorithm. The FastGreedy algorithm, the leading eigenvector algorithm, the label propagation algorithm (LPA), the CNM algorithm, degree clustering information (BLI), Node2vec-SC, and NGLPA were the seven well-known algorithms with which the proposed MES algorithm was compared.

4.1. Evaluation index

1. Modularity (Q): The quality of the divided communities is typically determined by the modularity index [2]. The impact of community detection is better the more modular it is. Conversely, community detection has a worsening effect. Normalized mutual information (NMI): NMI is an important index to measure the similarity between the detected community and the original community.

$$Q = \frac{1}{2|E|}\sum_{i,j}\left(a_{ij} - \frac{d_i d_j}{2|E|}\right)\delta(c_i, c_j)$$

where $\delta(c_i, c_j)$ represents whether the nodes v_i and v_j are in the same community, which is 1; Otherwise, it is 0.

2. $NMI \in [0,1]$; the divided community is closer to the original community the higher the NMI value.

$$NMI(X;Y) = \frac{2I(X;Y)}{H(X)+H(Y)}$$

where $I(X;Y)$ is the mutual information between X and Y, and $H(X)$ is the entropy of X.

5. Results

To test our method, we employ seven well-known real networks: the football network [6], the pollbooks network, the karate network [7], the dolphins[5] network, lesmis network [3], polblogs network [1], and facebook network [4].

The fundamental scale details for the networks are listed in Table I. "Network" displays the network's name, "Node" the number of nodes within it, "Edge" the number of nodes connected to its edge, and "Ground-Truth" the actual number of communities within the network. Table II displays the modularity of the seven other algorithms and MES algorithm following the division of communities. While MES algortihm has underperformed in terms of modularity results when compared to other algorithms. MES achieves the best overall performance, which in three of the four networks yields the best NMI result. Table II— displays the NMI values of the four networks with natural communities for the MES algorithm and the other seven algorithms. In conclusion, the MES algorithm obtains good modularity in 1/7 networks and the best NMI value in 3/4 networks. As a result, MES offers a promising overall performance.

Table 1: Network properties

Network	Node	Edge	Ground-Truth
karate	34	78	2
dolphins	62	159	2
polbooks	105	441	3
football	115	613	12
lesmis	77	254	-
polblogs	1490	19090	-
facebook	4039	88234	-

Table 2: NMI/Number of communities

Networks	LPA	CNM	Fast Greedy	Leading eigen-vector	BLI	NGLPA	Node2vec-SC	MSC	MES
karate	0.563/4	0.690/3	0.692/3	0.677/4	0.699/3	0.441/2	0.574/2	**0.837/2**	**0.837/2**
dolphins	0.552/4	0.572/4	0.572/4	0.449/5	0.510/5	0.711/4	0.889/2	0.777/2	**1.0/2**
polbooks	0.539/4	0.530/4	0.530/4	0.520/4	0.533/6	0.535/3	0.516/3	0.539/5	**0.639/4**
football	0.909/11	0.697/6	0.697/6	0.698/8	0.889/12	0.889/12	**0.921/12**	**0.921/11**	0.860/9

Table 3: Modularity/Number of communities

Networks	LPA	CNM	Fast Greedy	Leading eigen-vector	BLI	NGLPA	Node2vec-SC	MSC	MES
karate	0.375/4	0.380/3	0.380/3	0.393/4	0.370/3	0.331/2	0.256/2	0.418/4	0.386/2
dolphins	0.506/4	0.495/4	0.495/4	0.491/5	0.510/5	0.519/4	0.379/2	0.495/7	0.505/2
polbooks	0.496/4	0.501/4	0.502/4	0.467/4	0.511/6	0.460/3	0.499/3	0.519/5	0.456/4
football	0.582/11	0.549/6	0.549/6	0.492/8	0.550/11	0.585/12	0.552/12	0.600/11	0.591/9
lesmis	0.526/5	0.500/5	0.502/4	0.532/8	0.525/6	0.448/4	0.450/11	0.544/6	0.510/4
polblogs	0.521/6	0.516/20	0.527/20	0.521/6	0.390/97	0.520/9	0.499/3	0.457/9	0.538/6
Facebook	0.796/53	0.777/13	0.774/13	0.799/18	0.801/98	0.713/13	0.756/13	0.812/59	0.703/40

6. Parameter Sensitivity Analysis

We have kept lambda threshold at constant 2 because we don't want any lone communities. We found experimentally that the Similarity threshold (τ) gives optimal results when it has values between $0.03 \leq \tau \leq 0.18$.

The plots of similarity value index versus the node pairs resulted in extremely low p-values (< 0.00001) when comparing power-law to other models is a strong indicator of power-law behavior in almost all datasets.

7. Conclusion

In this brief, inspired by the Markov similarity enhancement method, we proposed an edge-based Markov similarity algorithm to obtain communities from a network. By leveraging iterative state transitions of the initial network, we obtain a steady-state Markov similarity enhancement matrix that utilizes the highest average similarity edges to form initial communities. Our algorithm utilizes a Markov similarity index to obtain to merges smaller initial communities with closely connected counterparts to arrive at the final community structure. We have tested the efficacy of our proposed method across various real networks and believe that it has vast potential for detecting communities.

In future work, we would like to come up with a good heuristic for defining the similarity threshold. Further, we want to test our algorithm on artificial networks and modify the algorithm to work on dynamic networks.

References

[1] Adamic, Lada A & Natalie Glance. 2005. The political blogosphere and the 2004 us election: divided they blog. En *Proceedings of the 3rd international workshop on Link discovery*, 36–43

[2] Blondel, Vincent D, Jean-Loup Guillaume, Renaud Lambiotte & Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008(10). P10008

[3] Knuth, Donald Ervin. 1993. *The stanford graphbase: a platform for combinatorial computing*, vol. 1. AcM Press New York

[4] Leskovec, Jure & Julian Mcauley. 2012. Learning to discover social circles in ego networks. *Advances in neural information processing systems* 25

[5] Lusseau, David, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten & Steve M Dawson. 2003. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology* 54. 396–405

[6] Rozemberczki, Benedek, Carl Allen & Rik Sarkar. 2021. Multi-Scale Attributed Node Embedding. *Journal of Complex Networks* 9(2)

[7] Rozemberczki, Benedek, Oliver Kiss & Rik Sarkar. 2020. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. En *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, 3125–3132. ACM

[8] Yang, Xu-Hua, Gang-Feng Ma, Xiang-Yu Zeng, Yuchao Pang, Yanbo Zhou, Yu-Di Zhang & Lei Ye. 2023. Community detection based on markov similarity enhancement. *IEEE Transactions on Circuits and Systems II: Express Briefs*