Heuristic Spectral Clustering for Three-Community Detection

Chenkai Wang

Email: wangck2022@mail.sustech.edu.cn

Abstract

This project explores community detection in complex networks through the lens of spectral clustering. Beginning with the theoretical foundations and motivations for detecting community structures, the study introduces a heuristic method tailored for networks with three communities. By applying spectral analysis recursively and leveraging modularity-based refinement, the approach effectively identifies core node groups while iteratively resolving ambiguous cases.

Keywords: Three-community detection, Spectral clustering, Modularity, Graph partition

Contents

1 Description of Homework 5 **2**

2 A Brief Introduction to Community and its Detection **3**

2.1 Community: Illustration, Definition, and Examples 3

2.2 Community Detection Algorithms 5

3 Spectral analysis for networks with two communities **7**

3.1 Notations and Definitions 7

3.2 A General Instruction Case 8

3.3 Spectral Analysis 9

4 Spectral Clustering for Networks with Three Communities **10**

4.1 Core Idea 10

4.2 Interpretation 11

4.3 Graphical Illustration 12

5 Heuristic Spectral Clustering for Detecting Three Communities **12**

References **16**

1 Description of Homework 5

Generalize the method of spectral analysis to the network with 3 communities.

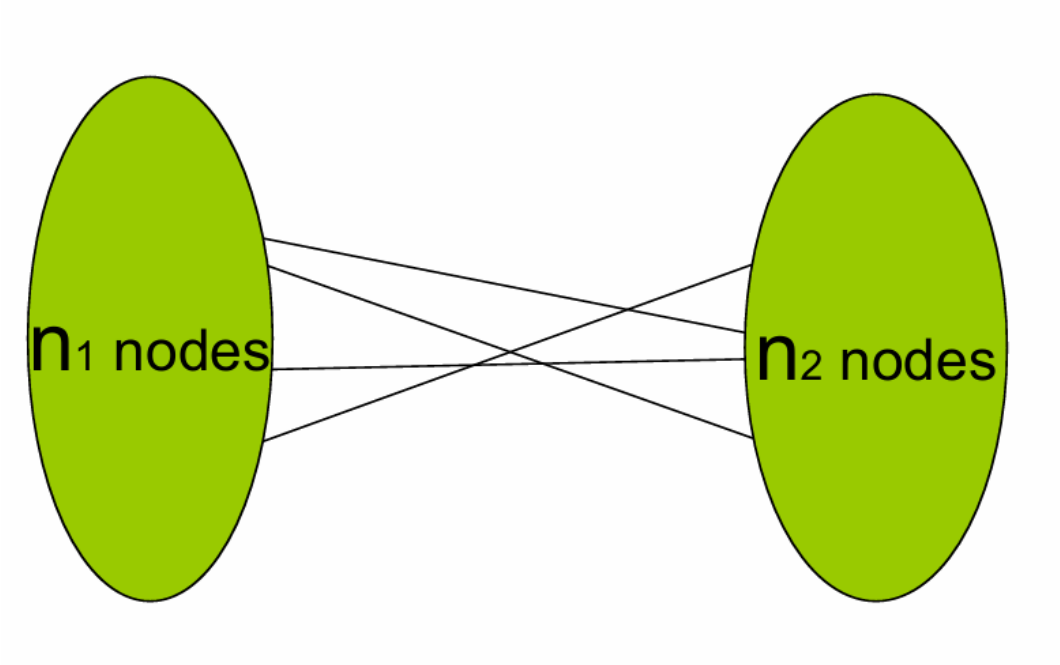


Figure 1: An example of 2 communities

2 A Brief Introduction to Community and its Detection

2.1 Community: Illustration, Definition, and Examples

Social networks are full of easy-to-spot communities, something that scholars have noticed decades ago[7]. The employees of a company are more likely to interact with their coworkers than with employees of other companies. Meanwhile, students at SUSTech tend to communicate with schoolmates rather than those at Shenzhen University. Consequently, workplaces and schools appear as densely interconnected communities within the social network. Communities could also represent circles of friends, a group of individuals who pursue the same hobby together, or individuals living in the same neighborhood.

More precisely, for an unweighted and undirected network $G = (V, E)$ where V and E are the node set and the edge set with $|V| = n$ and $|E| = m$, respectively. A community structure of network G is a partition of the network, denote as $C = \{C_1, C_2, \cdots, C_{\mathcal{K}}\}$, where $C_i \subset V$, $\cup_{i=1}^{\mathcal{K}} C_i = V$ and $C_i \cap C_j = \emptyset$ $(i, j = 1, 2, \cdots, \mathcal{K}$ and $i \neq j)$. With the concept of community, an additional constraint

$$\sum_{i,j=1}^{\mathcal{K}} |\{(u,v) \mid (u,v) \in E, u \in C_i, v \in C_i\}| >>$$
$$\sum_{i,j=1}^{\mathcal{K}} \left|\left\{(u,v) \mid (u,v) \in E, u \in C_i, v \in C_j, i \neq j\right\}\right|$$

is always attached to the partition. Intuitively, a community is a group of nodes with much more edges connecting to nodes within the community than to nodes from other communities.

Figure 2 gives an example of community structure. Several real-world examples help illustrate the importance of community structures.

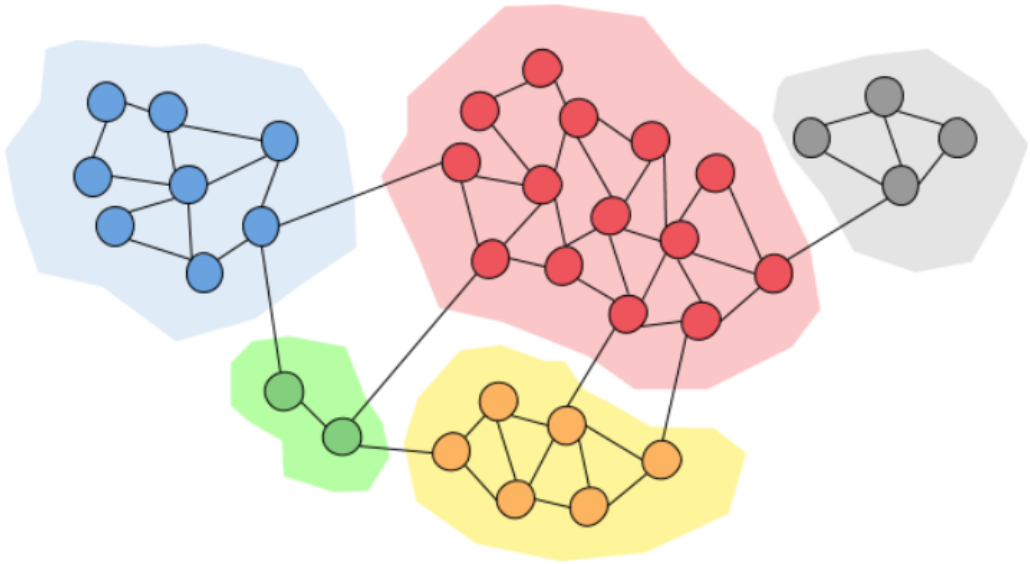


Figure 2: An network with five communities painted in 5 different colors

Communities play a vital role in understanding human diseases. Indeed, proteins involved in the same disease tend to interact with each other[4]. This finding inspired the disease module hypothesis[3], stating that each disease can be linked to a well-defined neighborhood of the cellular network. With these discoveries, we can develop new medicines faster to fight diseases like COVID-19 and keep us better.

Communities are also crucial in the collaboration network of scientists. For beginners in scientific research, through the community, they can quickly find influential scholars to master the general situation of this field better. Scholars in this field can follow the frontier work faster. Moreover, scientists who want to try cross-field can rapidly understand the proximity and development of the target field to make more meaningful work.

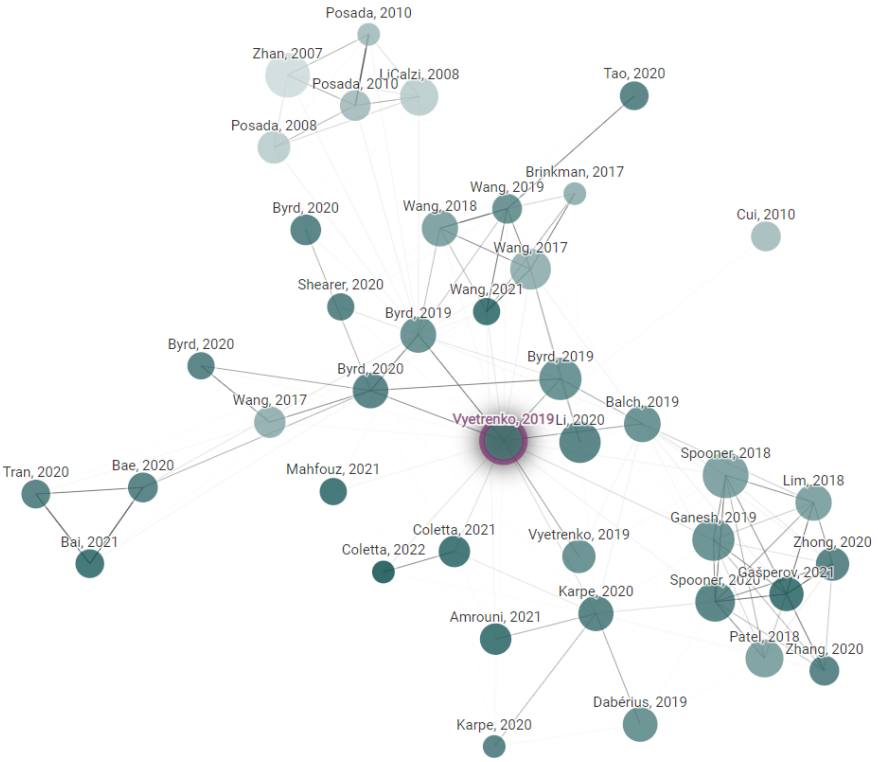


Figure 3: The collaboration network for a specific paper[6] from [1]

A community structure in a network consists of users and their preferences. Internet companies can benefit from such a network by identifying its community structure. For example, the users who have bought basketball and football are divided into two different communities, denoted as A and B, respectively. Then when the company wants to launch an advertisement for a new basketball shoe, the effect of advertising to the users of A will be better than that of club B, which provides a new idea for improving the efficiency and benefit of advertisement and commodity recommendation.

2.2 Community Detection Algorithms

In this section, we will briefly introduce the famous two: the agglomerative algorithm and the divisive algorithm.

The basic idea of the agglomerative algorithm is to calculate the similarity between each pair of nodes in a certain way and then add the edge in each step to the primitive empty network where the number of nodes is n. The process

starts from an empty network with zero edges and adds edges in descending order of similarity. This process can stop at any step, and the community structure is obtained simultaneously. A tree can also represent the whole process from empty to origin, as shown in figure 4. The green circles at the bottom represent the nodes in the network. As the blue horizontal dashed line moves up from the bottom of the tree, the nodes merge into a larger community. When the line moves to the top, the network becomes one community. The line at any point in the tree corresponds to a community structure. In the figure, nodes belonging to the same community are circled by a black ellipse according to the location of the line.

Oppositely, the divisive algorithm tries to find the edge with minimum similarity and remove it in each step. Repeat this process, and the whole network is gradually divided into smaller communities. Similarly, the algorithm can be terminated at any point to yield a corresponding community structure. Similar to the agglomerative algorithm, we can use a tree to represent the division process, which can better describe the continuous process that the whole network is split into several smaller communities. The relationship between the agglomerative algorithm and the divisive algorithm is shown below.

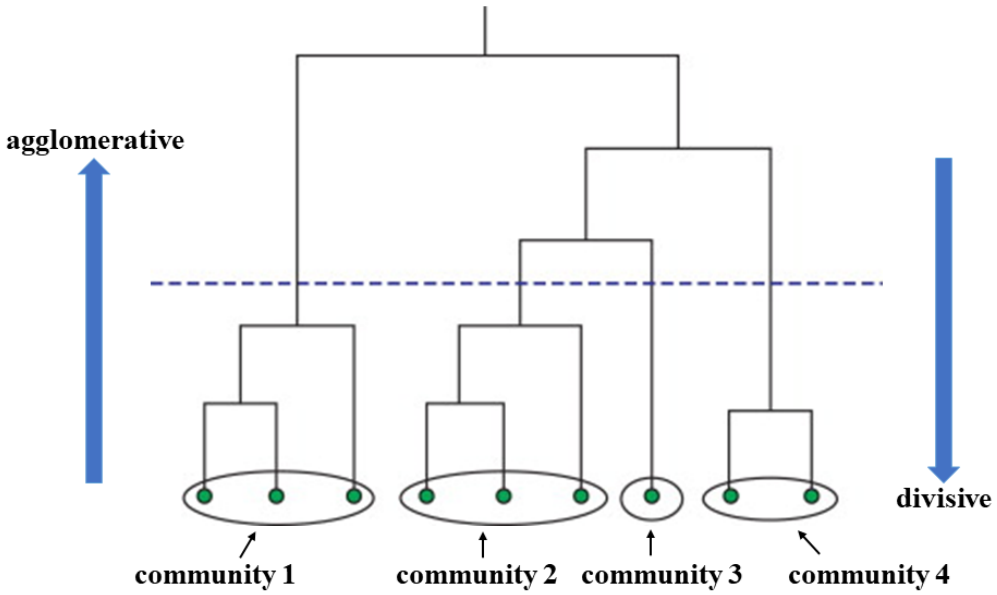


Figure 4: An illustration of agglomerative algorithm and divisive algorithm

In the next section, we will introduce the spectral analysis algorithm, specifically for meeting the requirement and its sound theoretical principles and

appliances.

3 Spectral analysis for networks with two communities

3.1 Notations and Definitions

For community detection, the spectral methods utilize the eigenspectra of various types of the network-associated matrix to identify the community structure. Before stating the algorithm formally, we first introduce some notations in the below chart.

notation	meaning
A	adjacency matrix
D	diagonal matrix that D_{ii} = degree of node i
L	Laplacian matrix which equals $D - A$
S	column vector records nodes information
k_{ij}	number of edges of node i in community j

Table 1: some notations and the corresponding meanings

The notation above will be illustrated by an example from figure 5.

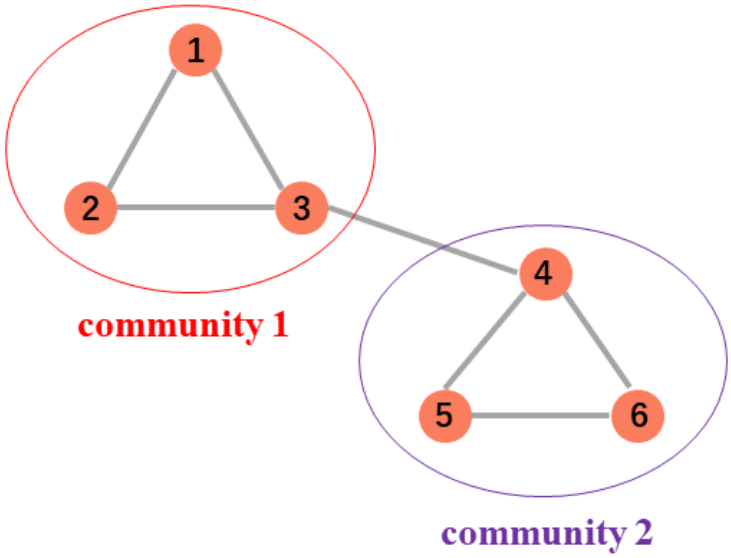


Figure 5: An illustration network with two communities

For the network shown above, we have

$$S = [1, 1, 1, -1, -1, -1]^T$$

where T represents transpose.

Meanwhile,

$$k_{11} = k_{21} = k_{31} = 2, \sum_{i=1}^{3} k_{i1} = 6 \text{ and } k_{42} = k_{52} = k_{62} = 2, \sum_{i=4}^{6} k_{i2} = 6$$

Similarity we have

$$k_{12} = k_{22} = 0, k_{32} = 1, \sum_{i=1}^{3} k_{i2} = 1 \text{ and } k_{41} = 1, k_{51} = k_{61} = 0, \sum_{i=4}^{6} k_{i2} = 1$$

Moreover,

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

and $L = D - A$. In the above formulas, red and purple indicates that they record the information within community 1 and community 2 in figure 5, respectively. In addition, brown records the edges between communities.

3.2 A General Instruction Case

Now consider a general case with $|V| = n$, $\mathcal{K} = 2$. Also, there are m and $n - m$ nodes in community 1 and community 2, respectively. Suppose we are already familiar with the community structure. Namely, S is undisputed where

$$S = [1, \cdots, 1, -1, \cdots, -1]$$

We are interested in $S^T LS$, which is

$$S^T LS = S^T (D - A)S = S^T DS - S^T AS$$

After simplification,

$$S^T DS = (\sum_{i=1}^{m} k_{i1} + \sum_{i=m+1}^{n} k_{i2}) + (\sum_{i=1}^{m} k_{i2} + \sum_{i=m+1}^{n} k_{i1})$$

Similarly,

$$-S^T AS = -(\sum_{i=1}^{m} k_{i1} + \sum_{i=m+1}^{n} k_{i2}) + (\sum_{i=1}^{m} k_{i2} + \sum_{i=m+1}^{n} k_{i1})$$

Therefore, $S^T LS = 2 * \left(\sum_{i=1}^{m} k_{i2} + \sum_{i=m+1}^{n} k_{i1}\right)$, which is four times the number of edges between two communities. Then According to our definition of community, finding the best partition is equivalent to minimizing $S^T LS$.

3.3 Spectral Analysis

This process is only applicable when there are two communities in the network. If satisfies, we can minimize $S^T LS$ to distinguish them. However, we can only write S precisely once we work on the community structure. There is no need to find S. We start from the properties of L, the Laplacian matrix.

Consider a network $G = (V, E)$ with $V = n$ and $\mathcal{K} = 2$, where $\mathcal{K} = 2$ comes from the prior knowledge or our observation. Here we do not add any constraint to $|E|$. Then L corresponding to G has the following properties we care about:

- $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ where λ_i represents eigenvalue of L,
- $< v_i^T, v_i >= 1, < v_i^T, v_j >= 0$ for $i \neq j, i, j = 1, \cdots, n$ where $< \bullet, \bullet >$ and v_i represents inner product of vectors and eigenvector of λ_i, respectively.

Then

- $\text{span}\{v_1, v_2, \cdots, v_n\} = \mathbb{R}^n$
- $\forall S \in \mathbb{R}^n, S = a_1 v_1 + a_2 v_2 + \cdots + a_n v_n$

Hence we have

$$\begin{aligned} S^T L S &= (a_1 v_1 + a_2 v_2 + \cdots + a_n v_n)^T L (a_1 v_1 + a_2 v_2 + \cdots + a_n v_n) \\ &= (a_1 v_1 + a_2 v_2 + \cdots + a_n v_n)^T (\lambda_1 a_1 v_1 + a_2 v_2 + \cdots + a_n v_n) \\ &= (a_1 v_1^T + a_2 v_2^T + \cdots + a_n v_n^T)(\lambda_1 a_1 v_1 + a_2 v_2 + \cdots + a_n v_n) \\ &= a_1^2 \lambda_1 + a_2^2 \lambda_2 + \cdots + a_n^2 \lambda_n \end{aligned}$$

where $a_i = v_i^T S$. Then minimizing $S^T L S \approx$ maximizing $a_2^2 \lambda_2$ subject to $\sum_{i=1}^n a_i^2 = n$.

4 Spectral Clustering for Networks with Three Communities

Unlike the analysis in Section 3.3, which uses the unnormalized Laplacian $L = D - A$, this section adopts the normalized Laplacian $L_{\text{sym}} = I - D^{-1/2} A D^{-1/2}$, which is often preferred in spectral clustering tasks involving more than two communities, particularly when node degrees vary significantly.

In the previous section, we have introduced the spectral analysis method for partitioning networks into two communities. When it comes to networks with more than two communities, the classical spectral bisection approach becomes insufficient. In this section, we introduce a mainstream generalization of spectral methods for detecting $\mathcal{K}$ communities, with a focus on the case $\mathcal{K} = 3$.

4.1 Core Idea

Here we adopt the normalized Laplacian L_{sym} as it is more suitable for spectral clustering with more than two clusters, especially in graphs with heterogeneous degrees.

Spectral clustering extends the idea of graph partitioning by using the eigenvectors of the normalized Laplacian L_{sym} to embed nodes into a low-dimensional Euclidean space, where standard clustering techniques such as K-means can be applied.

Given an undirected graph $G = (V, E)$ with n nodes, the process can be summarized as follows:

- Construct the normalized Laplacian matrix:

$$L_{\text{sym}} = I - D^{-1/2} A D^{-1/2}$$

  where A is the adjacency matrix and D is the degree matrix.

- Compute the eigenvectors corresponding to the smallest k eigenvalues of L_{sym}. For $k = 3$, we denote these eigenvectors as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$.

- Form a matrix $U \in \mathbb{R}^{n \times 3}$ with each row corresponding to a node and each column to an eigenvector:

$$U = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$$

  Since $\mathbf{v}_1$ is usually a constant vector and uninformative for clustering, we may alternatively use only $\mathbf{v}_2$ and $\mathbf{v}_3$ to construct $U \in \mathbb{R}^{n \times 2}$.

- Normalize the rows of U to unit length, which is commonly used in practice.

- Apply the K-means algorithm to the rows of U, treating each row as a point in $\mathbb{R}^2$.

- Assign the cluster label returned by K-means to each node. The result gives the final partition into three communities.

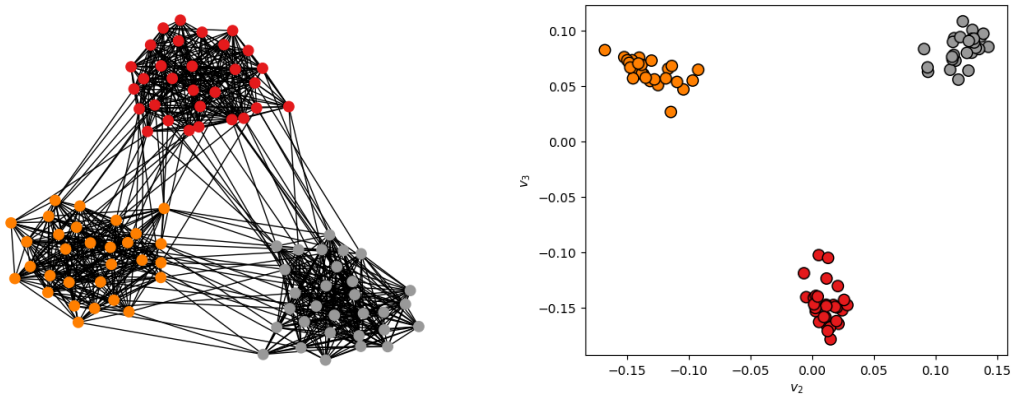
4.2 Interpretation

Each node is mapped to a point in a low-dimensional space based on its spectral embedding using the normalized Laplacian L_{sym}. Intuitively, nodes within the same community will be close to each other in this space, allowing clustering algorithms to identify groupings effectively.

The success of this method is theoretically supported by perturbation results in spectral graph theory and has been widely used in real-world applications such as image segmentation, recommendation systems, and social network analysis.

4.3 Graphical Illustration

Figure 6 presents an illustrative example of applying spectral clustering with $\mathcal{K} = 3$. The left shows a network with ground-truth communities, and the right shows the corresponding spectral embedding colored by the detected clusters.



(a) A network with three ground-truth communities.

(b) A network with three detected communities.

Figure 6: An illustrative example of spectral clustering with $\mathcal{K} = 3$.

5 Heuristic Spectral Clustering for Detecting Three Communities

Consider a network that is expected to contain three communities. An illustration example is shown below.

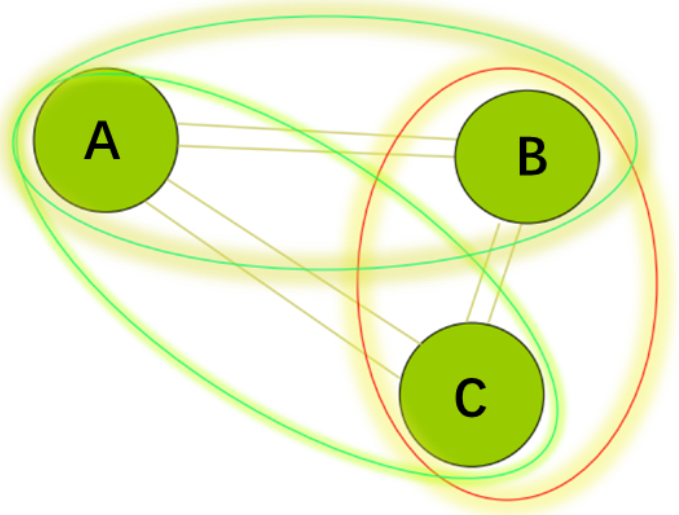


Figure 7: An illustration network with three communities

To detect the community structure similar to figure 7, we can apply the spectral algorithm for three rounds and use model-selecting thinking to derive the final result.

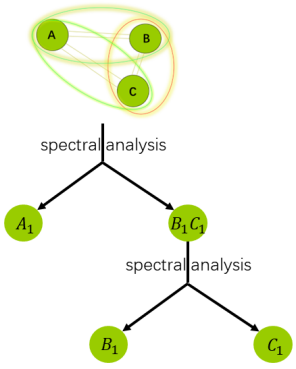


Figure 8: round 1

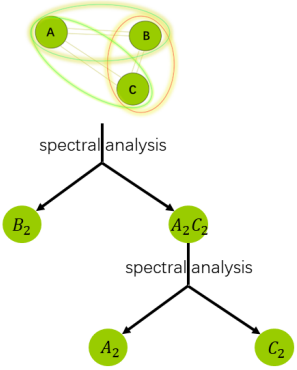


Figure 9: round 2

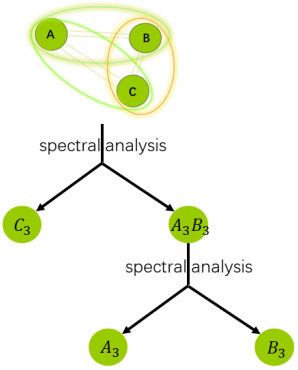


Figure 10: round 3

Here is the description. For graph $G = (V, E)$, suppose $A_{true}B_{true}, C_{true}$ are the true communities and A_i, B_i, C_i are communities from round i, $i = 1, 2, 3$. In round 1, we use spectral analysis twice, the first time for the entire network to get A_1 and B_1C_1 as a whole, the second time for B_1C_1 to get B_1 and C_1. In rounds 2 and 3, we do the same thing as in round 1, which obtain A_2, B_2, C_2 and A_3, B_3, C_3, respectively. We call the whole process as ***bisection_process***.

The next step is finding the difference. Here we apply the following algorithm.

Algorithm 1 Find common elements across three clusterings

Require: Community clusterings A, B, C, each a list of node sets.

Create sets S_{same} and $S_{different}$.
Let V be the union of all nodes in A, B, and C.
for each node u in V **do**
    Find A_u: the community in A containing u
    Find B_u: the community in B containing u
    Find C_u: the community in C containing u
    if $A_u = B_u = C_u$ **then**
        Add u to S_{same}
    else
        Add u to $S_{different}$
    end if
end for

Ensure: $S_{same}, S_{different}$

We run the algorithm 1, with parameters $A \leftarrow A_1, B \leftarrow A_2, C \leftarrow A_3$, which gives S_{same_A}. For B_is and C_is we repeat it similarly and denote the output as S_{same_B} and S_{same_C} respectively.

Define $S_{total\_same} = S_{same\_A} \cup S_{same\_B} \cup S_{same\_C}$, $S_{total\_different} = V \backslash S_{total\_same}$. Next, we consider arranging the elements in $S_{total\_different}$ to attain the best community structure. To achieve this, we use modularity [5] to evaluate the arrangement. We will not present its formula here for brevity. Instead, we will give an intuitive figure below to explain what it means.

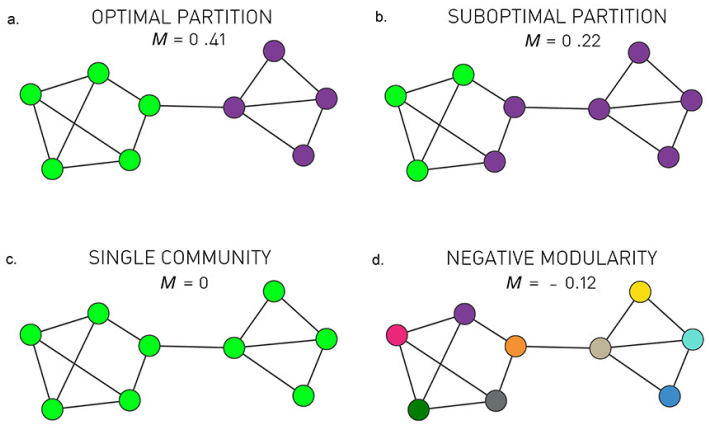


Figure 11: An illustration for modularity from [2]

In short, the better the community is divided, the higher the modularity should be. We use this principle and the idea of greed to deal with S, which is summarized as algorithm 2 presented in Appendix 5. The input parameters are

$$\begin{aligned} A &\leftarrow S_{same\_A} \\ B &\leftarrow S_{same\_B} \\ C &\leftarrow S_{same\_C} \\ S &\leftarrow S_{total\_different}. \end{aligned}$$

In conclusion, for a network $G = (V, E)$ with $\mathcal{K} = 3$, we can get the community structure via the following steps:

- Get A_i, B_i, C_i by ***bisection_process***, $i = 1, 2, 3$.
- Get S_{total_same} and $S_{total\_different}$ via Algorithm 1.
- Refine the partition using Algorithm 2.

References

[1] https://www.connectedpapers.com/.

[2] http://networksciencebook.com/chapter/9.

[3] Albert-Laszlo Barabasi, Natali Gulbahce, and Joseph Loscalzo. "Network medicine: a network-based approach to human disease". In: *Nature reviews genetics* 12.1 (2011), pp. 56–68.

[4] Kwang-Il Goh et al. "The human disease network". In: *Proceedings of the National Academy of Sciences* 104.21 (2007), pp. 8685–8690.

[5] Mark EJ Newman. "Spectral methods for community detection and graph partitioning". In: *Physical Review E* 88.4 (2013), p. 042822.

[6] Svitlana Vyetrenko et al. "Get real: Realism metrics for robust limit order book market simulations". In: *Proceedings of the First ACM International Conference on AI in Finance*. 2020, pp. 1–8.

[7] Wayne W Zachary. "An information flow model for conflict and fission in small groups". In: *Journal of anthropological research* 33.4 (1977), pp. 452–473.

Appendix

Algorithm 2 Greedy assignment of remaining nodes based on modularity

Require: Communities A, B, C, unassigned node set S.

Initialize temporary set $S_{\text{temp}} \leftarrow \emptyset$.
for each node u in S **do**
  Compute modularity gain M_A by temporarily assigning u to A.
  Compute modularity gain M_B by temporarily assigning u to B.
  Compute modularity gain M_C by temporarily assigning u to C.
  $M_{\max} \leftarrow \max\{M_A, M_B, M_C\}$
  if $M_{\max} = M_A$ **then**
    Assign u to A.
  else if $M_{\max} = M_B$ **then**
    Assign u to B.
  else if $M_{\max} = M_C$ **then**
    Assign u to C.
  else
    Add u to S_{temp}.
  end if
end for
for each node u in S_{temp} **do**
  Repeat modularity comparison as above.
  if no improvement found **then**
    Generate random number $e \in [0, 1]$.
    if $e < \frac{1}{3}$ **then**
      Assign u to A.
    else if $e < \frac{2}{3}$ **then**
      Assign u to B.
    else
      Assign u to C.
    end if
  end if
end for

Ensure: Final communities A, B, C.