Community detection for directed graphs using random walk

Phan Thi Ha Duong, Do Duy Hieu and Dang Tien Dat

Institute of Mathematics, Vienam Academy of Science and Technology

ICDMCS, Dalat 2^{nd} December, 2022

Table of contents

Table of contents

Community Detection problem

- Community is a set of nodes having close relationship while the opposite is true with nodes being in different communities.
- The detection of communities provides latent information about the relationship between vertices inside a graph.

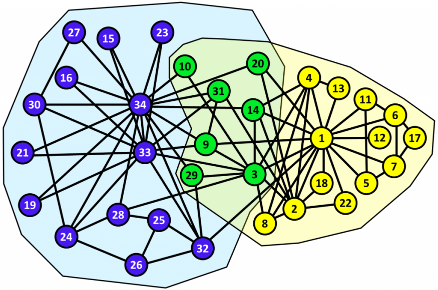

Figure 1.1: Illustration of communities in the Karate Club graph[1].

[1] https://bigdata.oden.utexas.edu/project/graph-clustering

Traditional method

- Graph Partitioning
- Hierarchical clustering
- Partitional clustering
- Spectral clustering
- Divisive algorithms

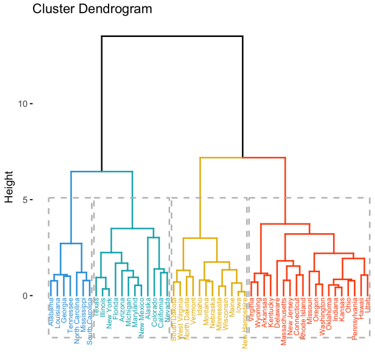

Figure 1.2: Illustration of hierarchical clustering[2].

[2] https://sbme-tutorials.github.io/2019/cv/notes/

1.3 Modularity function

- It is the most widely used and well-known clustering quality evaluation function.
- Range of value is $(-1, 1)$. Modularity determines the quality of clustering.

Modularity of unweighted undirected graph proposed by M. Newman [**1**]:

$$Q_u = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j) \tag{1.1}$$

Notation

C_i is community of node i;
$\delta(C_i, C_j) = 1$ if $C_i = C_j$ while $C_i \neq C_j$ then $\delta(C_i,C_j)= 0$;
A_{ij} is the number of edges between two nodes i, j ;
$\frac{k_i k_j}{2m}$ is the mean of edges between 2 nodes i, j based on the configuration model.

1.3 Illustration of modularity

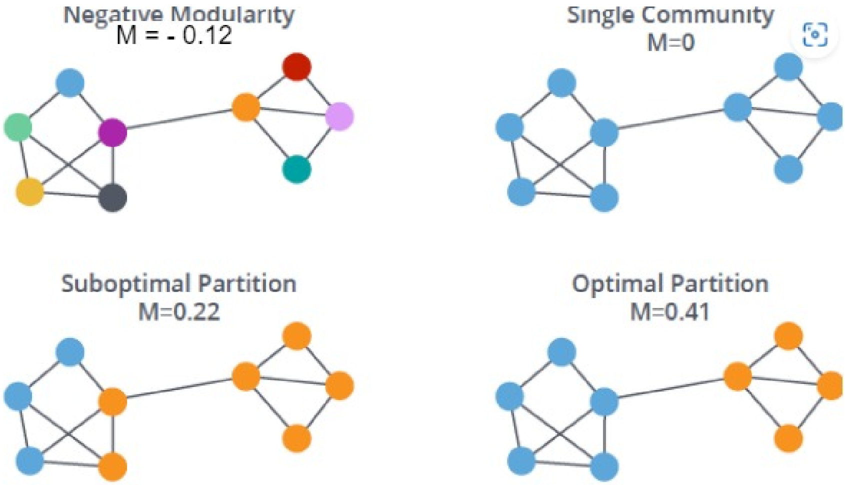

Figure 1.3: Illustration of modularity function.

1.4 Random walk on the graph

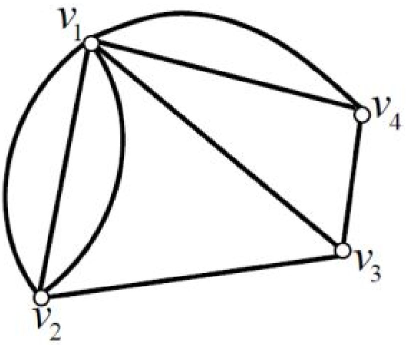

Figure 1.4: Illustration of random walk on graph.

Transition matrix:

$$P = \begin{bmatrix} 0 & 1/2 & 1/6 & 1/3 \\ 3/4 & 0 & 1/4 & 0 \\ 1/3 & 1/3 & 0 & 1/3 \\ 2/3 & 0 & 1/3 & 0 \end{bmatrix}$$

- P_{ij}: probability from node $i$to node j
- $P_{ij}^{(t)}$: probability from node $i$to node j after t steps (transitions) - denoted as P_{ij}^{t} .
- $P^{(t)} = \left[P_{ij}^{(t)}\right]_{i,j=\overline{1,n}} = P^t = P \times P \times \ldots \times P$ is transition matrix after t steps.

Table of contents

Some features

- The community detection algorithm used on undirected graphs.
- The process is similar to the hierarchical clustering algorithm, in addition, the WalkTrap algorithm [**4**] is particularly interested in three criteria:
  - Defining distance between vertices based on random walk on undirected graph.
  - Controlling the association between communities must be based on the objective function which is the average distance from the vertices to the clusters.
  - Finding the optimal slice by using the modularity function.

2.1 Initial idea of the WalkTrap algorithm

- Assuming two nodes being same community should have approximately the same probability to arbitrary node after $t$steps:

$$P_{ik}^{t} \approx P_{jk}^{t} \tag{2.1}$$

- Importance of each node is different $\Rightarrow$ the weighted sum.
- The distance formula from node i to node j:

$$r_{ij} = \sqrt{\sum_{u=1}^{N} \frac{\left(P_{iu}^{t} - P_{ju}^{t}\right)^2}{k_u}} = ||D^{-1/2} P_{i*}^{t} - D^{-1/2} P_{j*}^{t}|| \tag{2.2}$$

- Furthermore, this formula can be represented based on eigenvectors and eigenvalues of transition matrix P.

2.2 Distance formulas

$$r_{ij} = \sqrt{\sum_{u=1}^{N} \frac{\left(P^t_{iu} - P^t_{ju}\right)^2}{k_u}} = ||D^{-1/2}P^t_{i*} - D^{-1/2}P^t_{j*}|| \quad (2.3)$$

$$P^t_{Cj} = \frac{1}{|C|}\sum_{i \in C} P^t_{ij} \quad (2.4)$$

$$r_{C_1C_2} = \sqrt{\sum_{u=1}^{N} \frac{\left(P^t_{C_1u} - P^t_{C_2u}\right)^2}{k_u}} = ||D^{-1/2}P^t_{C_1*} - D^{-1/2}P^t_{C_2*}|| \quad (2.5)$$

- The distance between nodes r_{ij}.
- The probability from a community to a node: P^t_{Cj}.
- The distance between communities: $r_{C_1C_2}$.
- P is transition matrix $(P_{ij} = A_{ij}/k_i)$
- P^t is transition matrix after t steps.
- D is the diag degree matrix.

2.2 Object function and criteria to merge communities

- Object function:

$$\sigma_k = \frac{1}{N} \sum_{C \in \mathcal{P}_k} \sum_{i \in C} r_{iC}^2 \tag{2.6}$$

- Criteria to merge two communities: (C_1, C_2) so tha $\Delta\sigma(C_1, C_2)$ is minimal, let $C_3 = C_1 \cup C_2$:

$$\Delta\sigma(C_1, C_2) = \frac{1}{N} \left(\sum_{i \in C_3} r_{iC_3}^2 - \sum_{i \in C_1} r_{iC_1}^2 - \sum_{i \in C_2} r_{iC_2}^2 \right) \tag{2.7}$$

Determine the optimal number of communities

- From the beginning is N communities, after $N-1$ loops we will get only one community.
- Using the modularity function evaluates the quality of each partition..

Example of the process of WalkTrap

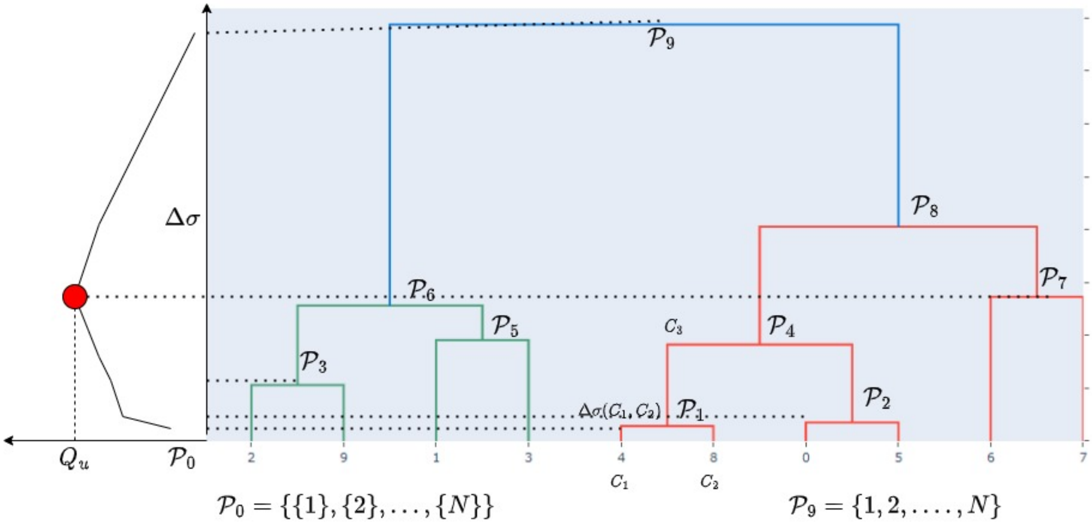

Table of contents

3.1 Our proposed method - The Di-WalkTrap algorithm

Some features

- The community detection algorithm can be applied on **both undirected graph and directed graph.**
- The process is similar to the WalkTrap algorithm, in addition, our proposed method defined the new distances between nodes:
  - Defining the **new distance** formulas based on **hitting times and stationary distribution** on graph (both undirected and directed cases).
  - Proposing the new **relationship with spectral** approaches on undirected case and **singular value decomposition** on directed case.
- **Overcome** the problem with **eigenvalue less than 1** of WalkTrap algorithm.

Hiting times and stationary distribution

$\{X_k\}_{k=0,1,2,...}$ is finite Markov chain and the state space $\mathbb{S} = \{1, 2, ...\}$

- Hitting time H_{ij}: the expected number of steps for the first transition from i to j.

$$T_j = \inf\{l \geq 1 : X_l = j\}$$

$$H_{ij} = E[T_j | X_1 = i]$$

- Stationary distribution $\phi = (\phi_1, \phi_2, ..., \phi_n)$: ϕ_i has meaning is the limitation of probability the chain reach to state $i \in \mathbb{S}$:

$$\phi_i = \lim_{k \to \infty} P(X_k = i)$$

Initial idea

- Replacing transition matrix after $t$steps P^t by expected hitting time matrix H which is steady instead of depending on t.
- Assuming two nodes being same community should have approximately the same number of steps to arbitrary node:

$$H_{ik} \approx H_{jk} \tag{3.1}$$

- Importance of each node is different $\Rightarrow$ the weighted sum.
- Stationary distribution $\phi = (\phi_1, \phi_2, ..., \phi_n)$ not only distinguish nodes but also has the property that sum of each quantile is 1.

$$\sum_{i=1}^{n} \phi_i = 1 \tag{3.2}$$

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \phi_k (H_{ik} - H_{jk})^2} \tag{3.3}$$

- Furthermore, this distance formula is related to eigenvalue, eigenvector and SVD.

The new distance formulas

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \phi_k (H_{ik} - H_{jk})^2} = \|\Phi^{1/2} H_{i\bullet} - \Phi^{1/2} H_{j\bullet}\| \quad (3.4)$$

$$H_{Cj} = \frac{1}{|C|} \sum_{i \in C} H_{ij} \quad (3.5)$$

$$r_{C_1 C_2} = \sqrt{\sum_{k=1}^{n} \phi_k (H_{C_1 k} - H_{C_2 k})^2} = \|\Phi^{1/2} H_{C_1\bullet} - \Phi^{1/2} H_{C_2\bullet}\| \quad (3.6)$$

- The distance between nodes r_{ij}.
- The expected hitting times from a community to a node: P^t_{Cj}.
- The distance between communities: $r_{C_1 C_2}$.
- H is expected hitting time matrix.
- Φ is stationary transition matrix.
- $H_{i\bullet}$ is i^{th}row of expected hitting time matrix.

3.2 The relationship with spectral approachs on undirected graph

Theorem

The distance r is related to the spectral properties of the matrix P by:

$$r_{ij}^2 = \sum_{\alpha=2}^{n} \frac{1}{(1-\lambda_\alpha)^2} (v_\alpha(i) - v_\alpha(j))^2, \tag{3.7}$$

where $(\lambda_\alpha)_{1\leq\alpha\leq n}$ *and* $(v_\alpha)_{1\leq\alpha\leq n}$ *are respectively the eigenvalues and right eigenvectors of the matrix* P.

- *Note:* $1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n \geq -1$.

Algorithm	Relation formula	Weighted
WalkTrap	$r_{ij}^2 = \sum_{\alpha=2}^{n} \lambda_\alpha^{2t} (v_\alpha(i) - v_\alpha(j))^2.$	λ_α^{2t}
Di-WalkTrap	$r_{ij}^2 = \sum_{\alpha=2}^{n} \frac{1}{(1-\lambda_\alpha)^2} (v_\alpha(i) - v_\alpha(j))^2$	$\frac{1}{(1-\lambda_\alpha)^2}$

Comparision with WalkTrap algorithm

- Walktrap algorithm: There isn't much of a difference between distances r_{ij} because $|\lambda_\alpha| \le 1 \ \forall \alpha = \overline{1, n}$
- Our algorithm: makes this difference be clearly when the coffecient is $\dfrac{1}{(1-\lambda_\alpha)^2}$

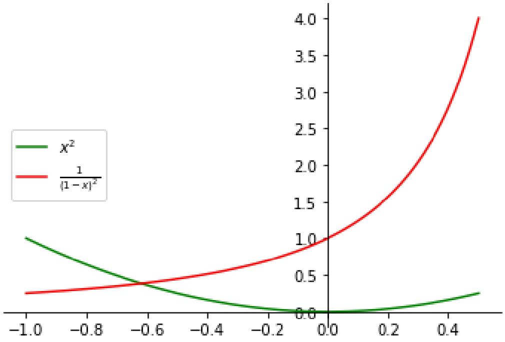

Figure 3.1: Illustration of coffecient function of two algorithms.

3.2 The relationship with singular value decomposition on directed graph

Theorem

The distance r is related to the spectral properties of the matrix P by:

$$r_{ij}^2 = \sum_{\alpha=2}^{n} \frac{1}{{\sigma_\alpha}^2}(w_\alpha(i) - w_\alpha(j))^2, \tag{3.8}$$

where σ_i, v_i be the i^{th} singular value, the corresponding right singular vectors of $\Gamma = \Phi^{1/2}(I - P)\Phi^{-1/2}$ where $\Phi^{1/2} = diag[\sqrt{\phi_i}]$ and $w_\alpha = \Phi^{-1/2}v_\alpha$.

Table of contents

4.1 Some types of random partition graph

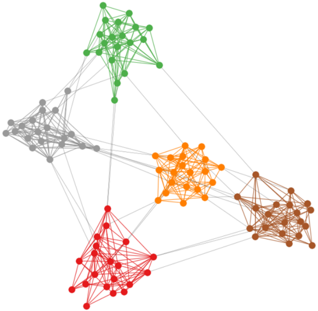

Figure 4.1: Illustration of planted l-partition.

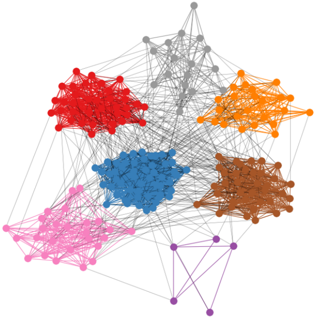

Figure 4.2: Illustration of Gaussian random partition

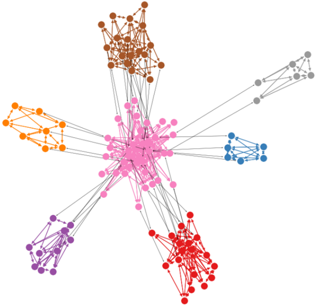

Figure 4.3: Illustration of LFR.

4.2 Results on Undirected graph - LFR benchmark graph

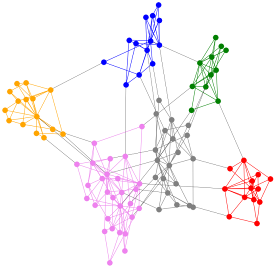

Figure 4.4: Result of Di-WalkTrap ($Q_u = 0.707$).

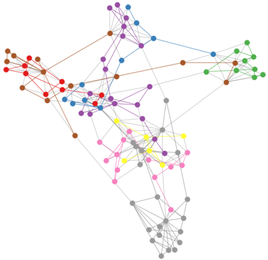

Figure 4.5: Result of WalkTrap ($Q_u = 0.327$).

Heatmap Jaccard Index Matrix

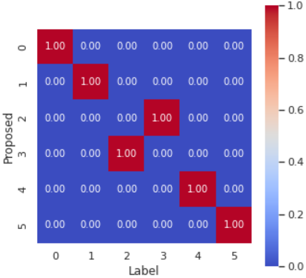

Figure 4.6: Di-WalkTrap - Heatmap Jaccard index Jaccard.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

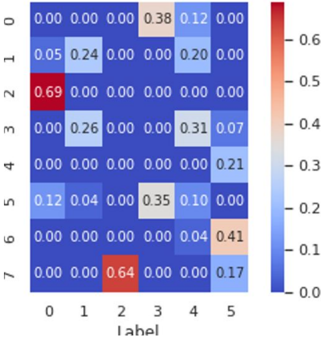

Figure 4.7: WalkTrap - Heatmap Jaccard index matrix.

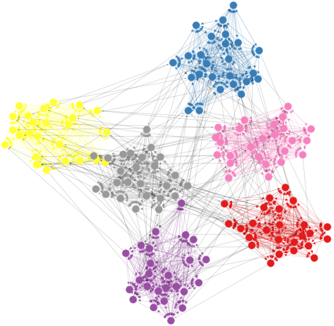

Figure 4.8: Results of Di-Walktrap on directed graph.

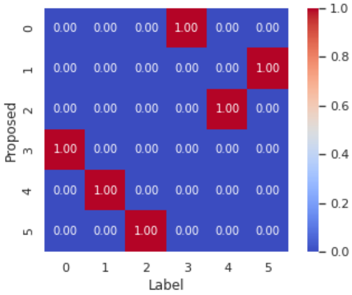

Figure 4.9: Heatmap Jaccard index matrix.

Conclusion and Future work

- Proposed new distance formula based on the hitting times and the stationary distribution.
- After proposing a new definition of distance, we use a mechanism similar to the Walktrap algorithm to perform clustering.
- The relationship with eigenvalues, eigenvectors and SVD demonstrate our effective algorithm.
- In the future, deep intervention into the processes occurring in the graph will yield a lot of hidden information about the relationship between the vertices.

References

[1] Newman, M. (2013). Spectral methods for community detection and graph partitioning. Physical review. E, Statistical, nonlinear, and soft matter physics. 88. 042822. 10.1103/PhysRevE.88.042822.

[2] Leicht, EA & Newman, M. (2008). Community Structure in Directed Networks. Physical review letters. 100. 118703. 10.1103/PhysRevLett.100.118703.

[3] Phan Thi Ha Duong, Do Duy Hieu and Dang Tien Dat, Community detection methods for directed graphs (preprint), 2022.

[4] P. Pons and M. Latapy. Computing communities in large networks using random walks, Journal of Graph Algorithms and Applications, volume 10. no. 2, 2006, Pages 191–218, 2006.

Thanks for your attention.