

Spectral Clustering Algorithms

Learner

L^AT_EX by Xin Wang

Notes

Contents

1	Graph theory	4
1.1	Basic terminologies	4
1.2	Graphs	5
1.3	Edges	6
1.4	Matrices	7
2	Classical Spectral Clustering Algorithm	10
2.1	Classical Spectral Clustering Algorithm	10

List of Figures

List of Tables

Chapter 1

Graph theory

1.1 Basic terminologies

Definition 1.1.1: Network or Graph

A network (or graph) consists of a set of nodes (or vertices, actors) and a set of edges (or links, ties) that connect those nodes[1].

Definition 1.1.2: Neighbor

Node j is called a neighbor of node i if (and only if) node i is connected to node j .

Definition 1.1.3: Degree

The number of edges connected to a node. Node i 's degree is often written as $\deg(i)$.

Definition 1.1.4: Walk

A list of edges that are sequentially connected to form a continuous route on a network.

Definition 1.1.5: Trail

A walk that doesn't go through any edge more than once.

Definition 1.1.6: Path

A walk that doesn't go through any node (and therefore any edge, too) more than once.

Definition 1.1.7: Cycle

A walk that starts and ends at the same node without going through any node more than once on its way.

Definition 1.1.8: Subgraph

Part of the graph.

Definition 1.1.9: Connected graph

A graph in which a path exists between any pair of nodes.

Definition 1.1.10: Connected component

A subgraph of a graph that is connected within itself but not connected to the rest of the graph.

1.2 Graphs

Definition 1.2.1: Complete graph

A graph in which any pair of nodes are connected.

Definition 1.2.2: Regular graph

A graph in which all nodes have the same degree. Every complete graph is regular.

Definition 1.2.3: Bipartite (n-partite) graph

A graph whose nodes can be divided into two (or n) groups so that no edge connects nodes within each group.

Definition 1.2.4: Tree graph

A graph in which there is no cycle. A graph made of multiple trees is called a forest graph. Every tree or forest graph is bipartite.

Definition 1.2.5: Planar graph

A graph that can be graphically drawn in a two-dimensional plane with no edge crossings. Every tree or forest graph is planar.

1.3 Edges

Definition 1.3.1: Undirected edge

A symmetric connection between nodes. If node i is connected to node j by an undirected edge, then node j also recognizes node i as its neighbor. A graph made of undirected edges is called an undirected graph. The Adjacency matrix of an undirected graph is always symmetric.

Definition 1.3.2: Directed edge

An asymmetric connection from one node to another. Even if node i is connected to node j by a directed edge, the connection isn't necessarily reciprocated from node j to node i . A graph made of directed edges is called a directed graph. The Adjacency matrix of a directed graph is generally asymmetric.

Definition 1.3.3: Unweighted edge

An edge without any weight value associated to it. There are only two possibilities between a pair of nodes in a network with unweighted edges; whether there is an edge between them or not. The Adjacency matrix of such a network is made of only 0's and 1's.

Definition 1.3.4: Weighted edge

An edge with a weight value associated to it. A weight is usually given by a non-negative real number, which may represent a connection strength or distance between nodes, depending on the nature of the system being modeled. The definition of the Adjacency matrix can be extended to contain those edge weight values for networks with weighted edges. The sum of the weights of edges connected to a node is often called the node strength, which corresponds to a node degree for unweighted graphs.

Definition 1.3.5: Multiple edges

Edges that share the same origin and destination. Such multiple edges connect two nodes more than once.

Definition 1.3.6: Self-loop

An edge that originates and ends at the same node.

Definition 1.3.7: Simple graph

A graph that doesn't contain directed, weighted, or multiple edges, or self-loops. Traditional graph theory mostly focuses on simple graphs.

Definition 1.3.8: Multigraph

A graph that may contain multiple edges. Many mathematicians also allow multigraphs to contain self-loops. Multigraphs can be undirected or directed.

Definition 1.3.9: Hyperedge

A generalized concept of an edge that can connect any number of nodes at once, not just two. A graph made of hyperedges is called a hypergraph (not covered in this textbook).

1.4 Matrices**Definition 1.4.1: Degree matrix**

Given a graph $G = (V, E)$ with $|V| = n$, V is the vertex set, and E is the edge set, the degree matrix $D_{n \times n}$ for G is a $n \times n$ diagonal matrix defined as

$$D_{ij} := \begin{cases} \deg(v_i) & i = j \\ 0 & i \neq j \end{cases}$$

where the degree $\deg(v_i)$ of a vertex counts the number of times an edge terminates at that vertex.

In an **undirected graph**, this means that each loop increases the degree of a vertex by two.

In a **directed graph**, the term degree may refer either to indegree (the number of incoming edges at each vertex) or outdegree (the number of outgoing edges at each vertex)

Definition 1.4.2: Adjacency matrix

Given a graph $G = (V, E)$ with $|V| = n$, V is the vertex set, and E is the edge set, the Adjacency matrix $A_{n \times n}$ for G is a $n \times n$ matrix defined as

$$A_{ij} := \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$$

where (v_i, v_j) is an edge in E .

A matrix with rows and columns labeled by nodes, whose i -th row, j -th column component a_{ij} is 1 if node i is a neighbor of node j , or 0 otherwise.

Corollary 1.4.3

The diagonal elements of the Adjacency matrix are all zero, since edges from a vertex to itself (loops) are not allowed in simple graphs.

$$A_{ii} = 0$$

Corollary 1.4.4

For an undirected graph, the matrix is symmetric, meaning.

$$A_{ij} = A_{ji}$$

For a directed graph, the matrix may not be symmetric.

Definition 1.4.5: Adjacency list

Adjacency list A list of lists of nodes whose i -th component is the list of node i 's neighbors.

Definition 1.4.6: Laplacian matrix

Given a simple G with n vertices v_1, \dots, v_n , its Laplacian matrix $L_{n \times n}$ is defined element-wise as

$$L_{ij} := \begin{cases} \deg(v_i) & i = j \\ -1 & i \neq j \text{ and } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

or equivalently by the matrix

$$L_{n \times n} = D_{n \times n} - A_{n \times n}$$

where $D_{n \times n}$ is the degree matrix and $A_{n \times n}$ is the Adjacency matrix of the graph G .

Since G is a simple graph, A only contains 1s or 0s and its diagonal elements are all 0s.

Corollary 1.4.7

The Laplacian matrix L is a symmetric matrix, and its eigenvalues are real and non-negative.

Corollary 1.4.8

For the undirected graph that both the Adjacency matrix A and the Laplacian matrix L are symmetric, and that row- and column-sums of the Laplacian matrix are all zeros (which directly implies that the Laplacian matrix is singular).

Definition 1.4.9: Eigendecomposition

Let \mathbf{A} be a square $n \times n$ matrix with n linearly independent eigenvectors q_i (where $i = 1, \dots, n$). Then \mathbf{A} can be factored as

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$$

where \mathbf{Q} is the square $n \times n$ matrix whose i -th column is the eigenvector q_i of \mathbf{A} , and $\mathbf{\Lambda}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\Lambda_{ii} = \lambda_i$. Note that only diagonalizable matrices can be factorized in this way.

The n eigenvectors q_i are usually normalized, but they don't have to be. A non-normalized set of n eigenvectors, v_i can also be used as the columns of \mathbf{Q} .

Claim: How to Perform Eigendecomposition?

To perform Eigendecomposition on a matrix, follow these steps:

1. Find the Eigenvalues:

Solve the characteristic equation:

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

Here, \mathbf{A} is the square matrix, λ is the eigenvalue, and \mathbf{I} is the identity matrix of the same dimension as \mathbf{A} .

2. Find the Eigenvectors:

For each eigenvalue λ , substitute it back into the equation:

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = 0$$

This represents a system of linear equations where \mathbf{v} is the eigenvector corresponding to the eigenvalue λ .

3. Construct the Eigenvector Matrix \mathbf{V} :**4. Form the Diagonal Matrix $\mathbf{\Lambda}$:****5. Calculate the Inverse of \mathbf{V} :**

Chapter 2

Classical Spectral Clustering Algorithm

2.1 Classical Spectral Clustering Algorithm

Bibliography

- [1] NICE CXone. Libretexts mathematics library, Jun 2024.