Spectral Clustering

Machine Learning – CS5011

Content Credits:

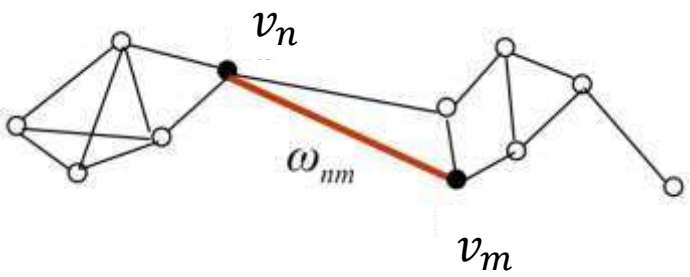
1. Von Luxburg, U., “A tutorial on spectral clustering.”; Statistics and Computing, 17(4), 395-416. Springer (2007).
2. Davide Eynard, “Notes on Spectral Clustering.”; (2012).

Similarity graph

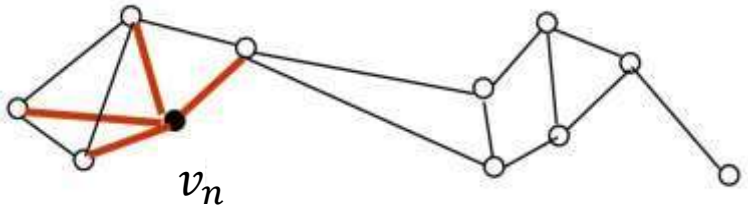
- The objective of a clustering algorithm is partitioning data into groups such that:
  - Points in the **same** group are **similar**
  - Points in **different** groups are **dissimilar**
- **Similarity graph** $G = (V, E)$ [undirected graph]:
  - Vertices v_i and v_j are connected by a **weighted** edge if their similarity is above a given threshold
  - GOAL: find a partition of the graph such that:
    - edges **within** a group have **high weights**
    - edges **across** different groups have **low weights**

Weighted adjacency matrix

- Let $G(V, E)$ be an undirected graph with vertex set
$$V = \{v_1, \dots, v_n\}$$
- **Weighted adjacency matrix** $\boldsymbol{W} = (w_{ij})_{i,j=1,\dots,n}$
  - $w_{ij} \geq 0$ is the weight of the edge between v_i and v_j.
  - $w_{ij} = 0$ means that v_i and v_j are not connected by an edge
  - $w_{ij} = w_{ji}$



  - **Degree of a vertex** $v_i \in V$: $\boldsymbol{d_i} = \sum_{j=1,\dots,n} w_{ij}$
  - **Degree matrix** $\boldsymbol{D} = diag(d_1, \dots, d_n)$



Similarity graphs - variants

- **ε-neighborhood:**
  - Connect all points whose pairwise distance is less than ε
- K-nearest neighbor graph
  - Connect vertex v_i with vertex v_j if v_j is among the k-nearest neighbours of v_i.
- Fully connected
  - *all* points with similarity $w_{ij} > 0$ are connected.
  - use a similarity function like the **Gaussian**:

$$w_{ij} = w(v_i, v_j) = \exp(-\|v_i - v_j\|^2/(2\sigma^2))$$

Graph Laplacians

- Graph Laplacian:
  - $\boldsymbol{L} = D - W$ *(symmetric and positive semi-definite)*
- Properties:
  - Smallest eigenvalue $\lambda_1 = 0$ with eigenvector $\mathbb{1}$
  - n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - the multiplicity k of the eigenvalue 0 of $\boldsymbol{L}$ equals the number of connected components $A_1, \ldots, A_k$ in the graph.

Spectral Clustering algorithm (1)

- Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.
  1. Construct a similarity graph as previously described. Let W be its weighted adjacency matrix.
  2. Compute the unnormalized Laplacian $\boldsymbol{L}$
  3. Compute the first k eigenvectors $u_1, \ldots, u_k$ of $\boldsymbol{L}$
  4. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns
  5. For $i = 1, \ldots, n$ let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U
  6. Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.
- Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

Normalized Graph Laplacians

- *Symmetric*: $\boldsymbol{L_{sym}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}} = I - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}$
  - λ is an eigenvalue of L_{sym} with eigenvector u iff λ and u solve the **generalized eigenproblem** $\boldsymbol{L}u = \lambda \boldsymbol{D}u$
  - 0 is an eigenvalue of $\boldsymbol{L_{sym}}$ with eigenvector $\boldsymbol{D}^{\frac{1}{2}}$
  - $\boldsymbol{L_{sym}}$ is positive semi-definite and have n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - the multiplicity k of the eigenvalue 0 of $\boldsymbol{L_{sym}}$ equals the number of connected components $A_1, \ldots, A_k$ in the graph.

Spectral Clustering algorithm (2)

- Input: Similarity matrix S $\in \mathbb{R}^{n \times n}$, number k of clusters to construct.
  1. Construct a similarity graph as previously described. Let W be its weighted adjacency matrix.
  2. Compute the normalized Laplacian L_{sym}
  3. Compute the first k eigenvectors $u_1, \ldots, u_k$ of $\boldsymbol{L}_{sym}$.
  4. normalize the eigenvectors
- Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i \}$.