Spectral Clustering

CE-725: Statistical Pattern Recognition
Sharif University of Technology
Spring 2013

Soleymani
Outline

- Partitioning of a similarity graph
  - Graph cut
  - Normalized cuts
  - Laplacian matrix
  - Spectral graph theory as origin of spectral clustering

- Spectral clustering: objective functions and solutions
  - Normalized cut, Ratio cut

- Multi-way algorithms
Spectral Clustering

- Spectral clustering: a class of methods that approximate the problem of partitioning nodes in a weighted graph as eigenvalue problems
  - Related to “spectral graph theory”
    - study of properties of a graph in relationship to eigenvalues, and eigenvectors of matrices associated to the graph (such as Laplacian matrix).

- Compared to the “traditional algorithms” such as k-means or single linkage, spectral clustering has many advantages.
Similarity Graph

- Represent dataset as a **weighted graph** $G(V,E)$

$$V = \{v_1, \ldots, v_N\} \quad \text{Set of } N \text{ vertices representing data points}$$

$$E = \{e(i,j)\} \quad \text{Set of edges connecting vertices if similarity of the corresponding data points is high}$$

$$W_{ij} \quad \text{Weights of edges showing pair-wise similarity between points}$$
Graph Construction

- Nodes $V = \{v_1, ..., v_N\}$ corresponding to data points
- Edges $E = \{e(i,j)\}$ connecting close vertices (data points)

Main Approaches of graph construction:
- $\epsilon$-neighborhood graph
  - we connect all points whose pairwise distances are smaller than $\epsilon$
- $k$-NN graph
  - connect $v_i$ with $v_j$ if $v_j$ is among the $k$-nearest neighbors of $v_i$
  - Two ways of making this graph undirected:
    - $k$-NN graph: ignore the directions of the edges
    - Mutual $k$-NN graph: connect if both of the directions exist
- Fully connected graph
  - only useful if weights model local neighborhoods
Similarity Graph

[Data points]

[epsilon-graph, epsilon=0.3]

[kNN graph, k = 5]

[Mutual kNN graph, k = 5]

[Luxberg, 2007]
Weight of Edges

- $W_{ij}$: Weight of edge $e(i, j)$ connecting $v_i$ and $v_j$
  - measure of the “closeness” of the respective neighbors $x^{(i)}$ and $x^{(j)}$.

Example of weight:

$$W_{ij} = \begin{cases} 
\exp \left(-\frac{||x^{(i)} - x^{(j)}||^2}{\sigma^2}\right), & \text{if } v_i \text{ and } v_j \text{ are connected} \\
0, & \text{otherwise}
\end{cases}$$
Graph Partitioning Algorithms

- Single-link (SL) and complete link (CL) hierarchical algorithms can also be considered as graph-based clustering algorithms
  - SL is related to Maximal Spanning Tree
  - CL is related to Maximal Clique
- CL tends to find compact clusters
- We will introduce spectral clustering as a more proper approach for graph partitioning
Drawbacks of SL and CL Methods: Example

Single-link (10 clusters)  Complete-link (2 clusters)

Single-link (2 clusters)  Complete-link (2 clusters)
Single Link (Noise Problem)
Graph Partitioning Objective: Graph Cuts

Partitioning objectives as a function of the “edge cut”.

\[ \text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \]

Find the partitioning which provides the minimal cut between groups (Minimize weight of between-group connections)

\[ \text{cut}(A, B) = 0.25 \]
Graph Cut Criteria

- **Minimum-cut**
  - Find partitioning that minimizes the weight of connections between groups:

  \[
  \min \text{cut}(A, \bar{A})
  \]

- Degenerate case:
Min-Cut Drawback

- Min-cut favours isolated clusters

Problem:
- Only considers external cluster connections and does not consider the internal cluster density
Volume

- \( \text{cut}(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} S_{ij} \)
- \( d_i = \sum_{j=1}^{N} S_{ij} \)
- \( \text{Vol}(A) = \sum_{i \in A} d_i = \sum_{i \in A} \sum_{j=1}^{N} S_{ij} \)

\( \text{Vol}(A) \): The total weight of the edges originating from group A.
Graph Cut Criteria

- Criterion: Normalised-cut (Shi & Malik,’97)

\[ ncut(A, \bar{A}) = \frac{cut(A, \bar{A})}{vol(A)} + \frac{cut(A, \bar{A})}{vol(\bar{A})} = cut(A, \bar{A}) \left[ \frac{1}{vol(A)} + \frac{1}{vol(\bar{A})} \right] \]

- Normalize the association between groups by their volume.

- Solving this problem is NP-hard
  - It is relaxed and then solved as using an eigen-decomposition
Spectral Graph Theory

- Represent a similarity graph as a matrix
  - The eigenvalues and eigenvectors of a matrix provide global information about its structure.

- Spectral Graph Theory
  - Analyze the “spectrum” of matrix representing a graph.
    - Spectrum: The eigenvectors of a graph, ordered by the magnitude(strength) of their corresponding eigenvalues.
Adjacency Matrix

- Adjacency matrix or weight matrix $\mathbf{W}$
  - $N \times N$ matrix
  - $\mathbf{W} = [w_{ij}]$: edge weight between vertex $x^{(i)}$ and $x^{(j)}$

```
0  0.8  0.7  0  0.1  0  0  
0.8  0  0.8  0  0  0  0  0
0.7  0.8  0  0.15  0  0  0  0
0  0  0.15  0  0.8  0.9  1  0
0.1  0  0  0.8  0  0.8  0  0
0  0  0  0.9  0.8  0  0.8  0
0  0  0  1  0  0.8  0  0
```

Symmetric matrix
Degree Matrix

- **Degree matrix** ($D$)
  - $N \times N$ diagonal matrix
  - $D_{ii} = \sum_{j=1}^{N} W_{ij}$: total weight of edges incident to vertex $v_i$
Laplacian Matrix

- **Laplacian matrix** \( (L) \)
  - \( N \times N \) symmetric matrix

\[
L = D - W
\]

- **Important properties:**
  - Symmetric, positive semi-definite
  - Eigenvalues and eigenvectors provide an insight into the properties of the graph (e.g., connectivity)
Spectral Bi-partitioning

\[ L = D - W \]

\[
\begin{pmatrix}
1.6 & -0.8 & -0.7 & 0 & -0.1 & 0 & 0 \\
-0.8 & 1.6 & -0.8 & 0 & 0 & 0 & 0 \\
-0.7 & -0.8 & 1.65 & -0.15 & 0 & 0 & 0 \\
0 & 0 & -0.15 & 2.85 & -0.8 & -0.9 & -1 \\
-0.1 & 0 & 0 & -0.8 & 1.7 & -0.8 & 0 \\
0 & 0 & 0 & -0.9 & -0.8 & 2.5 & -0.8 \\
0 & 0 & 0 & -1 & 0 & -0.8 & 1.8
\end{pmatrix}
\]

\[ L = V \Lambda V^T \]

\[
V = \begin{pmatrix}
0.38 & -0.43 & 0.07 & -0.73 & 0.37 & 0.03 & 0.03 \\
0.38 & -0.46 & -0.05 & 0.06 & -0.80 & -0.01 & 0.02 \\
0.38 & -0.41 & -0.07 & 0.67 & 0.48 & 0 & -0.08 \\
0.38 & 0.31 & -0.06 & 0.06 & 0.03 & -0.04 & 0.87 \\
0.38 & 0.31 & 0.74 & 0.04 & -0.04 & -0.4 & -0.24 \\
0.38 & 0.34 & 0.02 & 0 & -0.04 & 0.82 & -0.25 \\
0.38 & 0.35 & -0.66 & -0.11 & 0 & -0.41 & -0.34
\end{pmatrix}
\]

\[ \lambda_1 = 0 \]
\[ \lambda_2 = 0.14 \]
\[ \lambda_3 = 1.73 \]
\[ \lambda_4 = 2.32 \]
\[ \lambda_5 = 2.44 \]
\[ \lambda_6 = 3.32 \]
\[ \lambda_7 = 3.74 \]
Relation of Cut Problem to Laplacian Matrix

- Express a bi-partition \((A, B)\) as a vector: 
  \[ p_i = \begin{cases} 
  1, & x^{(i)} \in A \\
  -1, & x^{(i)} \in B 
  \end{cases} \]

- Cost function of cut problem:
  \[
  f(p) = \sum_{i,j \in V} W_{ij} (p_i - p_j)^2 = p^T L p
  \]

\[
  f(p) = \sum_{i=1}^{N} p_i^2 \sum_{j=1}^{N} W_{ij} + \sum_{j=1}^{N} p_j^2 \sum_{i=1}^{N} W_{ij} - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} p_i p_j W_{ij}
\]

\[
  = \sum_{i=1}^{N} p_i^2 D_{ii} + \sum_{j=1}^{N} p_j^2 D_{jj} - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} p_i p_j W_{ij}
\]

\[
  = 2p^T L p
\]
Problem Relaxation

- Relax indicators $p_i$ to continuous values $q_i$

- $\min_{q} f(q)$ can be found by solving an eigen-decomposition problem
  - The optimal solution is the eigen-vector corresponding to the second smallest eigen-value of $L$ (Rayleigh Theorem).
Laplacian Matrix: Properties

- Properties of Laplacian Matrix
  - $L$ is positive semi-definite: $x^T L x \geq 0$ for any $x$.
    - $f(x) = \sum_{i,j \in V} W_{ij} (x_i - x_j)^2$ is always a nonnegative scalar
  - When the similarity graph has one connected component, first eigenvector of $L$ is $(1, \ldots, 1)^T = 1^T$ with eigen value $\lambda_1 = 0$.
    - Second eigenvector can be the desired solution.
      - The smaller $\lambda_2$, the better quality of the partitioning.
  - When the similarity graph has $c$ connected components, the $c$ smallest eigenvalues of $L$ are zero $\lambda_1 = \lambda_2 = \cdots = \lambda_c = 0$.
    - Eigenvectors corresponding to these zero eigenvalues are indicators of the components
Discretization (Grouping)

- Are partitions \( A, B \) determined by: \( A = \{i \mid q_i < 0\}, \ B = \{i \mid q_i \geq 0\} \)?
  - \( f(q) \) is insensitive to additive constant \( c \) to \( q \)
  - Trivial approach: sort \( q \) to increasing order, and cut in the middle point.
  - Clustering approach: cluster data into two groups based on the values of \( q \)
Example – 2 Spirals

In the embedded space given by the eigen-vector corresponding to the second smallest eigen-value \((q)\), all data points of a cluster have been lied in a point (have equal \(q\)).

K-NN graph (for \(k = 3\)) has two connected component (\(q\) for each data point indicates in which connected component it lies).
Example

k-NN Graph

$k = 3$

$k = 7$
There are two parameters to set: $k$ (or $\varepsilon$) and $\sigma$

A small $k$ leads to a sparse graph and serves to limit the comparisons between points to those that are close.

This is advantageous since the Euclidean distance is unlikely to be reasonable for points far away.

$\sigma$ serves a similar role but, unlike $k$, is tied to the actual scale of the points (their distances).

Sometimes unweighted knn-graph is used to avoid setting this parameter ($\sigma = +\infty$)
Some Cut Definitions

- **Normalized Cut**
  \[
  \text{NCut}(A, \bar{A}) = \frac{\text{cut}(A, \bar{A})}{\text{vol}(A)} + \frac{\text{cut}(A, \bar{A})}{\text{vol}(\bar{A})} = \frac{\text{cut}(A, \bar{A})}{\text{cut}(A, \bar{A}) + \text{cut}(A, A)} + \frac{\text{cut}(A, \bar{A})}{\text{cut}(\bar{A}, A) + \text{cut}(\bar{A}, \bar{A})}
  \]

- **Ratio Cut**
  \[
  \text{RCut}(A, \bar{A}) = \frac{\text{cut}(A, \bar{A})}{|A|} + \frac{\text{cut}(A, \bar{A})}{|\bar{A}|}
  \]

- **Min-Max Cut**
  \[
  \text{MMCut}(A, \bar{A}) = \frac{\text{cut}(A, \bar{A})}{\text{cut}(A, A)} + \frac{\text{cut}(A, \bar{A})}{\text{cut}(\bar{A}, \bar{A})}
  \]
Relaxed Optimization Problems

- **NormalizedCut** (Shi & Malik, 1997):
  \[
  \min_q q^T Lq \\
  \text{s.t. } q^T Dq = 1, q^T D1 = 0
  \]
  The solution is the second eigenvector of \( Lq = \lambda Dq \)

- **RatioCut** (Hagen & Kahng, 1992):
  \[
  \min q^T Lq \\
  \text{s.t. } q^T q = 1, q^T 1 = 0
  \]
  The solution is the second eigenvector of \( L \)

- **MinMaxCut** (Ding et al. 2001):
  \[
  \min_q \frac{q^T W q}{q^T D q}
  \]
  The solution is the second eigenvector of \( Lq = \lambda Dq \)
NCut Optimization Problem: Details

- Condition for exactly balancing clusters: $p^T D 1 = 0$ ($p$ binary labels)
  - Equivalent to: $\text{cut}(A, \bar{A}) + \text{cut}(A, A) = \text{cut}(\bar{A}, A) + \text{cut}(\bar{A}, \bar{A})$.
  - We instead use a relaxed criterion $q^T D 1 = 0$ ($q$ can be continuous)
  - Then, small changes in $q_i$ for nodes that are strongly connected to others ($D_{ii}$ is large) require larger compensating changes at nodes that are only weakly coupled to others.
    - Isolated nodes become “followers”.

- $q^T D q = 1$
  - Normalize $q$ so as to avoid $q \to 0$. 
NCut Optimization Problem: Solution

\[ \min_{q} q^{T} Lq \]
\[ \text{s.t. } q^{T} D q = 1, q^{T} D \mathbf{1} = 0 \]

- **Solution:**

\[ \frac{\partial q^{T} Lq + \lambda (q^{T} D q - 1)}{\partial q} = 0 \Rightarrow Lq = \lambda Dq \]

- The eigenvector with the smallest eigenvalue (\( \lambda = 0 \)) is \( q = 1^{T} \) (for graph with one connected component).
  - This would not satisfy \( q^{T} D \mathbf{1} = 0 \) but the second smallest eigenvector does.

- **Solution:** eigen vector corresponding to the second smallest eigenvalue from

\[ (D - W)q = \lambda Dq \]
\[ (I - D^{-1/2}WD^{-1/2})q = \lambda q \]
Normalized Laplacian Matrix

\[ L_{sym} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2} \]

\[ L_{rw} = D^{-1}L \]
Multi-way Spectral Clustering

- How do we partition a graph into $k$ clusters?

- Two basic approaches:
  - Recursive bi-partitioning (Hagen et al., 1991)
    - Recursively apply bi-partitioning in a hierarchical divisive manner.
    - Disadvantages: Inefficient, unstable
  
  - Cluster multiple eigenvectors (Shi & Malik, 2000)
    - Build a reduced space from multiple eigenvectors.
      - K-means is applied on the reduced $N \times k$ space to produce $k$ clusters
Multi-way Spectral Clustering: NCut
(Shi & Malik, 2000)

\[ G = C_1 \cup \cdots \cup C_k \]

\[ ncut(C_1, \ldots, C_k) = \sum_{i,j} \frac{cut(C_i, C_j)}{vol(C_i)} + \frac{cut(C_j, C_i)}{vol(C_j)} \]

\[ = \sum_i \frac{cut(C_i, G - C_i)}{vol(C_i)} \]

- Indicator of the i-th cluster: \( p_i(j) = \begin{cases} 1, & x^{(j)} \in C_i \\ 0, & x^{(j)} \notin C_i \end{cases} \)

\[ ncut(C_1, \ldots, C_k) = \sum_{i=1}^{k} \frac{p_i^T (D - W) p_i}{p_i^T D p_i} \]
Multi-way Spectral Clustering: NCut

- $p'_i = p_i/|C_i|, P' = [p'_1, ..., p'_k]$

\[
n\text{cut}(C_1, ..., C_k) = \text{tr}(P'^T(I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}})P')
\]

- Relaxation: min $\text{tr}(Q^T(I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}})Q)$ \hspace{1cm} s. t. $Q^TQ = I$

  - Solution can be found using an eigendecomposition
Example 1

\(\varepsilon\)-graph:
\(\varepsilon = \sqrt{5}\)

Ideal graph for clustering:
3 connected components

Connected components are no longer disconnected, but only few edges with low weight between them.
Example 1

\[ Q = \begin{bmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 1
\end{bmatrix} \]

before normalization of columns
Example 1

- Using $q_2$ and $q_3$ to group data:

  - $\epsilon$-graph:
    - $\epsilon = 3$
    - $\sigma = 1$

The points can be easily clustered by k-means
Example 2

Histogram of the sample

Eigenvalues

Eigenvector 1  Eigenvector 2  Eigenvector 3  Eigenvector 4  Eigenvector 5

[Luxberg, 2007]
Aside: How to select $k$?

- Eigengap: difference between two consecutive eigenvalues.
- Most stable clustering is generally given by the value $k$ that maximises the expression

$$k^* = \arg \max_k |\lambda_{k+1} - \lambda_k|$$
Spectral Clustering Methods: Summary

- **Main steps:**
  - **Pre-processing:**
    - Graph construction and matrix representation (e.g. Laplacian)
  
  - **Decomposition:**
    - Compute eigenvalues and eigenvectors of the matrix.
    - Map data points to a lower-dimensional embedded space based on one (or more eigenvectors).
      - embedding data in a space in which clusters are more obvious

  - **Grouping**
    - Assign points to clusters, based on the new representation.
      - E.g., applying a classical clustering algorithm such as k-means
Spectral Clustering: Summary

- **Advantages:**
  - Find quality of a partition using graph cut criteria.
  - Efficient approach to calculate near-optimal partitions.
  - Relaxed optimization problem is convex and can be solved as an eigenvalue problem.
  - Do not restrict the shapes of clusters to predefined ones.

- **Disadvantages:**
  - Sensitive to the parameters used for similarity graph construction.
References


- C. Ding, A Tutorial on Spectral Clustering, ICML 2004.