Data Clustering

- Two different criteria
  - Compactness, e.g., k-means, mixture models
  - Connectivity, e.g., spectral clustering
Spectral Clustering
Weighted Graph Partitioning

- Some graph terminology
  - Objects (e.g., pixels, data points)
    \( i \in I = \text{vertices of graph } G \)
  - Edges \((ij)\) = pixel pairs with \( W_{ij} > 0 \)
  - Similarity matrix \( W = [ W_{ij} ] \)
  - Degree
    \[
    d_i = \sum_{j \in G} W_{ij} \\
    d_A = \sum_{i \in A} d_i \quad \text{degree of } A \subseteq G
    \]
  - \( \text{Assoc}(A,B) = \sum_{i \in A} \sum_{j \in B} W_{ij} \)

Cuts in a Graph

- (edge) cut = set of edges whose removal makes a graph disconnected
- Weight of a cut:
  \[
  \text{cut}(A,B) = \sum_{i \in A} \sum_{j \in B} W_{ij} = \text{Assoc}(A,B)
  \]
- Normalized Cut criteria: minimum cut\((A,\overline{A})\)
  \[
  \text{Ncut}(A,\overline{A}) = \frac{\text{cut}(A,\overline{A})}{d_A} + \frac{\text{cut}(A,\overline{A})}{d_{\overline{A}}}
  \]
  More generally:
  \[
  \text{Ncut}(A_1, A_2, \ldots, A_k) = \sum_{i=1}^{k} \left( \frac{\sum_{j \in A_i, j \not\in A_i} W_{ij}}{\sum_{j \not\in A_i} W_{ij}} \right) = \sum_{i=1}^{k} \left( \frac{\text{cut}(A_i, \overline{A_i})}{d_A} \right)
  \]
Graph-based Clustering

- Data Grouping
- Image segmentation

Affinity Function

\[ W_{i,j} = e^{-\frac{\|X_i - X_j\|^2}{\sigma^2}} \]

- Affinities grow as \( \sigma \) grows \( \rightarrow \)
- How the choice of \( \sigma \) value affects the results?
- What would be the optimal choice for \( \sigma \)?
Clustering via Optimizing Normalized Cut

- The normalized cut:
  \[ \text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{d_A} + \frac{\text{cut}(A, B)}{d_B} \]

- Computing an optimal normalized cut over all possible \( y \) (i.e., partition) is NP hard

- Transform Ncut equation to a matrix form (Shi & Malik 2000):
  \[
  \min_y \text{Ncut}(x) = \min_y \frac{y^T (D - W) y}{y^T D y} \\
  \text{Subject to: } y \in \{0, 1\}^n \\
  y^T D y = 0
  \]

  - Still an NP hard problem

  \[
  \text{Rayleigh quotient theorem}
  \]

Relaxation

- Instead, relax into the continuous domain by solving generalized eigenvalue system:
  \[
  \min_y y^T (D - W) y, \quad \text{s.t. } y^T D y = 1
  \]

  - Which gives:
    \[ (D - W) y = \lambda D y \]  \text{Rayleigh quotient theorem}

  - Note that \( (D - W) 1 = 0 \) so, the first eigenvector is \( y_v = 1 \) with eigenvalue 0.

  - The second smallest eigenvector is the real valued solution to this problem!!
Algorithm

1. Define a similarity function between 2 nodes. i.e.:
   \[ w_{i,j} = e^{-\frac{d_{ij}}{\sigma_k^2}} \]
2. Compute affinity matrix (W) and degree matrix (D).
3. Solve \((D - W)y = \lambda Dy\)
   - Do singular value decomposition (SVD) of the graph Laplacian \(L = D - W\)
     \[ L = V^T \Lambda V \Rightarrow y^* \]
4. Use the eigenvector with the second smallest eigenvalue, \(y^*\), to bipartition the graph.
   - For each threshold \(k\), \(A_k = \{ i | y_i \text{ among } k \text{ largest element of } y^* \}\)
   - \(B_k = \{ i | y_i \text{ among } n-k \text{ smallest element of } y^* \}\)
   - Compute \(\text{Ncut}(A_k, B_k)\)
   - Output \(k^* = \arg \max_k \text{Ncut}(A_k, B_k)\) and \(A_{k^*}, B_{k^*}\)

Ideally ...

\[ \text{Ncut}(A, B) = \frac{y^T (D - S) y}{y^T D y}, \text{ with } y_i \in \{1, -1\}, y^T D 1 = 0. \]
Example (Xing et al, 2001)

Poor features can lead to poor outcome (Xing et al 2001)
Cluster vs. Block matrix

\[ \text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{d_A} + \frac{\text{cut}(A, B)}{d_B} \]

\[ \text{Degree}(A) = \sum_{i \in A, j \in V} W_{i,j} \]

Compare to Minimum cut

- Criterion for partition:

\[ \min \text{cut}(A, B) = \min_{A,B} \sum_{i \in A, j \in B} W_{i,j} \]

Problem!

Weight of cut is directly proportional to the number of edges in the cut.

Cuts with lesser weight than the ideal cut

First proposed by Wu and Leahy
K-means and Gaussian mixture methods are biased toward convex clusters.

Ncut is superior in certain cases.
Why?

General Spectral Clustering
Representation

- Partition matrix $X$:

$$X = [X_1, \ldots, X_K]$$

- Pair-wise similarity matrix $W$: $W(i, j) = \text{aff}(i, j)$

- Degree matrix $D$: $D(i, i) = \sum_j W_{i,j}$

- Laplacian matrix $L$: $L = D - W$

Eigenvectors and blocks

- Block matrices have block eigenvectors:

$$\begin{pmatrix}
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1
\end{pmatrix} \Rightarrow \begin{pmatrix}
.71 \\
.71 \\
0 \\
.71
\end{pmatrix}$$

- Near-block matrices have near-block eigenvectors:

$$\begin{pmatrix}
1 & 1 & 2 & 0 \\
1 & 1 & 0 & -.2 \\
.2 & 0 & 1 & 1 \\
0 & -.2 & 1 & 1
\end{pmatrix} \Rightarrow \begin{pmatrix}
.71 \\
.69 \\
.14 \\
0
\end{pmatrix}$$
Spectral Space

- Can put items into blocks by eigenvectors:

\[
\begin{pmatrix}
1 & 1 & .2 & 0 \\
1 & 1 & 0 & -.2 \\
.2 & 0 & 1 & 1 \\
0 & -.2 & 1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
.71 & 0 \\
.69 & -.14 \\
.14 & .69 \\
0 & .71
\end{pmatrix}
\]

- Clusters clear regardless of row ordering:

\[
\begin{pmatrix}
1 & .2 & 1 & 0 \\
.2 & 1 & 0 & 1 \\
1 & 0 & 1 & -.2 \\
0 & 1 & -.2 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
.71 & 0 \\
.14 & .69 \\
.69 & -.14 \\
0 & .71
\end{pmatrix}
\]

Spectral Clustering

- Algorithms that cluster points using eigenvectors of matrices derived from the data
- Obtain data representation in the low-dimensional space that can be easily clustered
- Variety of methods that use the eigenvectors differently (we have seen an example)
- Empirically very successful
- Authors disagree:
  - Which eigenvectors to use
  - How to derive clusters from these eigenvectors
- Two general methods
Method #1

- Partition using only one eigenvector at a time
- Use procedure recursively
- Example: Image Segmentation
  - Uses 2nd (smallest) eigenvector to define optimal cut
  - Recursively generates two clusters with each cut

Method #2

- Use k eigenvectors (k chosen by user)
- Directly compute k-way partitioning
- Experimentally has been seen to be “better”
Spectral Clustering Algorithm
Ng, Jordan, and Weiss 2003

- Given a set of points \( S = \{s_1, \ldots, s_n\} \)
- Form the affinity matrix \( w_{i,j} = e^{-\frac{\|s_i - s_j\|^2}{\sigma}} \), \( \forall i \neq j \), \( w_{i,j} = 0 \)
- Define diagonal matrix \( D_k = \sum_k a_{ik} \)
- Form the matrix \( L = D^{-1/2}W D^{-1/2} \)
- Stack the \( k \) largest eigenvectors of \( L \) to form the columns of the new matrix \( X \):

\[
X = \begin{bmatrix}
    x_1 & x_2 & \cdots & x_k
\end{bmatrix}
\]

- Renormalize each of \( X \)'s rows to have unit length and get new matrix \( Y \). Cluster rows of \( Y \) as points in \( \mathbb{R}^k \)

---

SC vs Kmeans

![SC vs Kmeans Diagram](image-url)
**Why it works?**

- K-means in the spectrum space!

**More formally ...**

- Recall generalized Ncut

\[
Ncut(A_1, A_2, \ldots, A_k) = \sum_{i=1}^{k-1} \left( \frac{\sum_{j \in A_i \cap A_{j+1}} W_{ij}}{\sum_{j \in A_i} W_{ij}} \right) \frac{\sum_{j \in A_i} \text{cut}(A_j, \overline{A_j})}{d_{A_i}}
\]

- Minimizing this is equivalent to spectral clustering

\[
\min \ Ncut(A_1, A_2, \ldots, A_k) = \sum_{i=1}^{k-1} \left( \frac{\sum_{j \in A_i \cap A_{j+1}} W_{ij}}{\sum_{j \in A_i} W_{ij}} \right) \frac{\sum_{j \in A_i} \text{cut}(A_j, \overline{A_j})}{d_{A_i}}
\]

\[
\min \ Y^T D^{-1/2} W D^{-1/2} Y
\]

\[
\text{s.t.} \quad Y^T Y = I
\]

**segments**

**Y**

\[
\begin{bmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 1
\end{bmatrix}
\]
Toy examples

Images from Matthew Brand (TR-2002-42)

User’s Prerogative

- Choice of $k$, the number of clusters

- Choice of scaling factor
  - Realistically, search over $\sigma^2$ and pick value that gives the tightest clusters

- Choice of clustering method: k-way or recursive bipartite

- Kernel affinity matrix

\[ w_{i,j} = K(S_i, S_j) \]
Conclusions

- **Good news:**
  - Simple and powerful methods to segment images.
  - Flexible and easy to apply to other clustering problems.

- **Bad news:**
  - High memory requirements (use sparse matrices).
  - Very dependant on the scale factor for a specific problem.

\[
W(i, j) = e^{-\frac{|\mathbf{Y}_i - \mathbf{Y}_j|^2}{\sigma^2}}
\]