Normalized Cut Method for Image Segmentation

- Criterion for measuring a candidate partitioning:
  Affinity measure between elements within each region is high, and the affinity between elements across regions is low
- Affinity: element × element → \( \mathbb{R}^+ \) Examples of components of an affinity function: spatial position, intensity, color, texture, motion. Defines the similarity of a pair of data elements.

Affinity (Similarity) Measures

- Intensity
  \[
  \text{aff}(x, y) = e^{-|I(x) - I(y)|^2 / 2\sigma_i^2}
  \]
- Distance
  \[
  \text{aff}(x, y) = e^{-\|x-y\|^2 / 2\sigma_i^2}
  \]
- Color
- Texture
- Motion

Problem Formulation

- Given an undirected graph \( G = (V, E) \), where \( V \) is a set of nodes, one for each data element (e.g., pixel), and \( E \) is a set of edges with weights representing the affinity between connected nodes
- Find the image partition that maximizes the “association” within each region and minimizes the “disassociation” between regions
- Finding the optimal partition is NP-complete
• Let A, B partition G. Therefore, \( A \cup B = V \), and \( A \cap B = \emptyset \)

• The affinity or similarity between A and B is defined as

\[
cut(A,B) = \sum_{i \in A, j \in B} W_{ij} = \text{total weight of edges removed}
\]

• The optimal bi-partition of G is the one that minimizes \( \text{cut} \)

• \( \text{Cut} \) is biased towards small regions

• Similarly, define the “normalized association:”

\[
nassoc(A,B) = \frac{\text{assoc}(A,A)}{\text{assoc}(A,V)} + \frac{\text{assoc}(B,B)}{\text{assoc}(B,V)}
\]

• \( \text{Nassoc} \) measures how similar, on average, nodes within the groups are to each other

• New goal: Find the bi-partition that minimizes \( ncut(A,B) \) and maximizes \( nassoc(A,B) \)

• But, it can be proved that \( ncut(A,B) = 2 - nassoc(A,B) \), so we can just minimize \( ncut \): \( y = \arg \min ncut \)

• So, instead define the \textbf{normalized} similarity, called the \textbf{normalized-cut}(A,B), as

\[
ncut(A,B) = \frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(B,A)}{\text{assoc}(B,V)}
\]

where \( \text{assoc}(A,V) = \sum_{i \in A, k \in V} W_{ik} \)

= total connection weight from nodes in A to all nodes in G

• \( \text{Ncut} \) measures the disimilarity between regions (“disassociation” measure)

• \( \text{Ncut} \) removes the bias based on region size (usually)

• Let \( y \) be a \( P = |V| \) dimensional vector where

\[
y_i = \begin{cases} 1, & \text{if node } i \in A \\ -1, & \text{otherwise} \end{cases}
\]

• Let \( d(i) = \sum_{j} W_{ij} \)

define the affinity of node \( i \) with all other nodes

• Let \( D = P \times P \) diagonal matrix:

\[
D = \begin{bmatrix} d_1 & 0 & \ldots & 0 \\ 0 & d_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & d_p \end{bmatrix}
\]

“degree matrix”
• Let $A = P \times P$ symmetric matrix:

```
<table>
<thead>
<tr>
<th>w_{1}</th>
<th>w_{2}</th>
<th>\ldots</th>
<th>w_{P}</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_{21}</td>
<td>w_{22}</td>
<td>\ldots</td>
<td>w_{2P}</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\ddots</td>
<td>\vdots</td>
</tr>
<tr>
<td>w_{P1}</td>
<td>w_{P2}</td>
<td>\ldots</td>
<td>w_{PP}</td>
</tr>
</tbody>
</table>
```

“affinity matrix”

• It can be shown that

\[
y = \arg \min_x ncut(x)
\]

\[
= \arg \min, \frac{y^T (D - A)y}{y^T Dy} \text{ subject to } y^T D 1 = 0
\]

• Relaxing the constraint on $y$ so as to allow it to have real values means that we can approximate the solution by solving an equation of the form:

\[
(D - A)y = \lambda Dy
\]

• The solution, $y$, is an eigenvector of $(D - A)$

• An eigenvector is a characteristic vector of a matrix and specifies a segmentation based on the values of its components; similar points will hopefully have similar eigenvector components.

• Theorem: If $M$ is any real, symmetric matrix and $x$ is orthogonal to the $j-1$ smallest eigenvectors $x_1, \ldots, x_{j-1}$, then $x^T M x / x^T x$ is minimized by the next smallest eigenvector $x_j$ and its minimum value is the eigenvalue $\lambda_j$

**NCUT Segmentation Algorithm**

1. Set up problem as $G = (V,E)$ and define affinity matrix $A$ and degree matrix $D$
2. Solve $(D - A)x = \lambda Dx$ for the eigenvectors with the smallest eigenvalues
3. Let $x_2$ = eigenvector with the 2nd smallest eigenvalue $\lambda_2$
4. Threshold $x_2$ to obtain the binary-valued vector $x_2'$ such that $ncut(x_2') \geq ncut(x'_2)$ for all possible thresholds $t$
5. For each of the two new regions, if $ncut <$ threshold $T$, then recurse on the region

• Smallest eigenvector is always $0$

  because $A=V, B=\emptyset$ means $ncut(A,B)=0$

• Second smallest eigenvector is the real-valued $y$ that minimizes $ncut$

• Third smallest eigenvector is the real-valued $y$ that optimally sub-partitions the first two regions

• Etc.

• Note: Converting from the real-valued $y$ to a binary-valued $y$ introduces errors that will propagate to each sub-partition
Comments on the Algorithm

- Recursively bi-partitions the graph instead of using the 3rd, 4th, etc. eigenvectors for robustness reasons (due to errors caused by the binarization of the real-valued eigenvectors)
- Solving standard eigenvalue problems takes $O(P^3)$ time
- Can speed up algorithm by exploiting the “locality” of affinity measures, which implies that $A$ is sparse (non-zero values only near the diagonal) and $(D - A)$ is sparse. This leads to a $O(P\sqrt{P})$ time algorithm

Example: 2D Point Set

Figure 3: A point set in the plane.

Eigenvalues and Eigenvectors

Figure 5: Select (1) plots the smallest 10 eigenvalues of the generalized eigenvector system.

Figure 6: Select (1) plots the second 10 eigenvalues of the generalized eigenvector system.

Figure 7: Select (1) plots the third 10 eigenvalues of the generalized eigenvector system.
Example 2: A Grayscale Image

Discretizing an Eigenvector
Partitioning stops when histogram is not bimodal

Some Example Results