Computer Vision

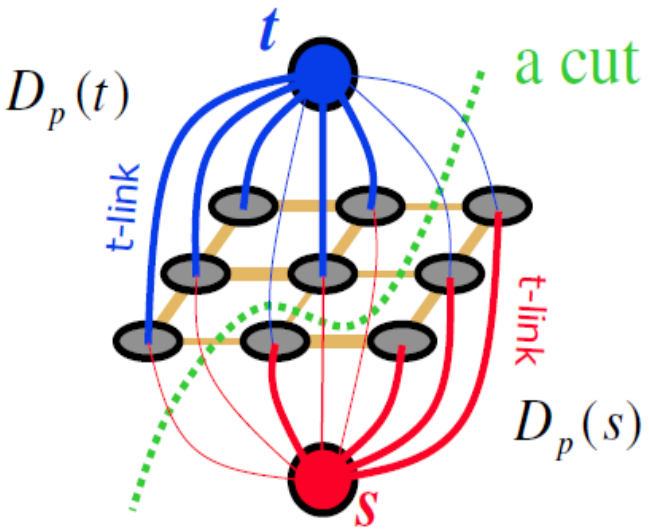
Class 3

Spectral clustering

Graph cuts with only pair-wise costs

- Minimum cut (→ Image Processing, class 9):

$$E(u) = \sum_i D(u_i) + \sum_{i,j \in \mathcal{N}(i)} w_{ij} \delta(u_i \neq u_j)$$



- When removing the unary term (the t-links), the optimal solution will assign all nodes to either source or target (no costs)
  → trivial solution

- This is also known as the **shrinking bias** of the minimum cut.

- To avoid the shrinking bias, the volume/connectivity within the resulting region must be considered

  → ratio cut, average cut, normalized cut

- Ratio cut: $\frac{\text{cut}(A, \Omega - A)}{|A|}$

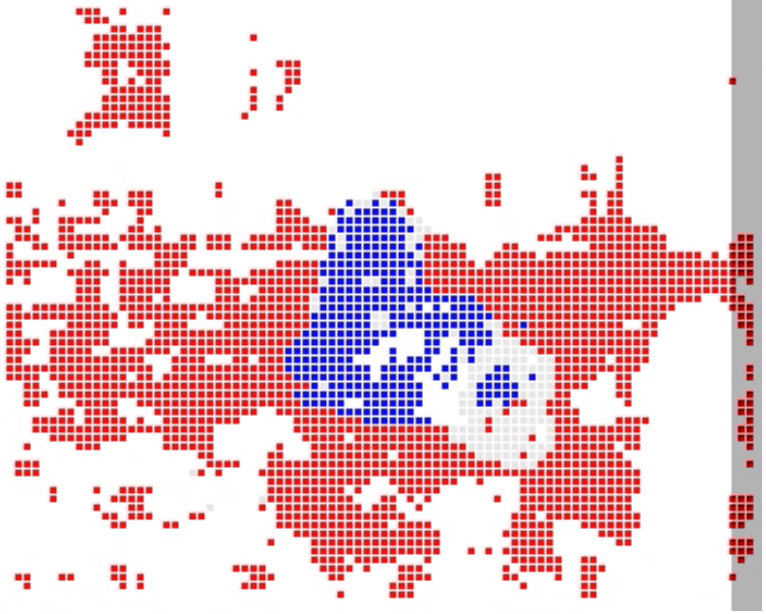
Examples for problems without unary costs

Motion segmentation based on point trajectories

- Hard to build a motion model (unary cost) for each object
- Easy to define pairwise cost between point trajectories

3D surface reconstruction

- Hard to build a model for the interior of the object volume
- Easy to define costs for potential surface positions (pairwise costs)



Agglomerative clustering

- Compute pairwise distances d_{ij} between all data points (or neighboring data points)

- Greedy heuristic:
  - Merge the two points/clusters with the smallest distance to a single cluster
  - Recompute the affected distances
  - Iterate until remaining distances reach a certain threshold

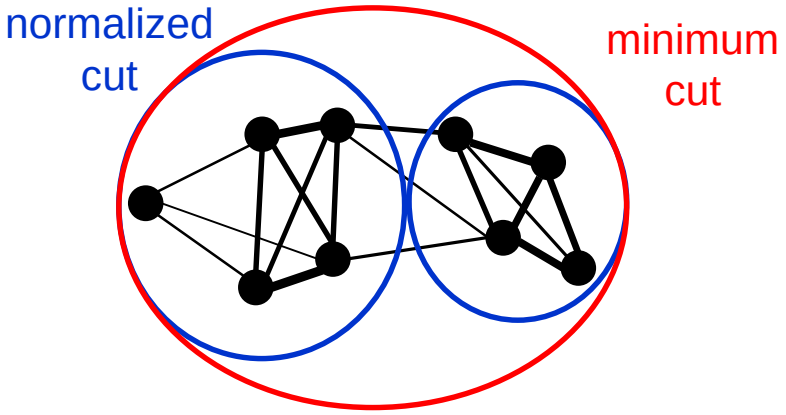
- Early decisions cannot be corrected
  → does not optimize a global criterion

- Very sensitive to chosen thresholds

Author: P. Felzenszwalb

Normalized cut

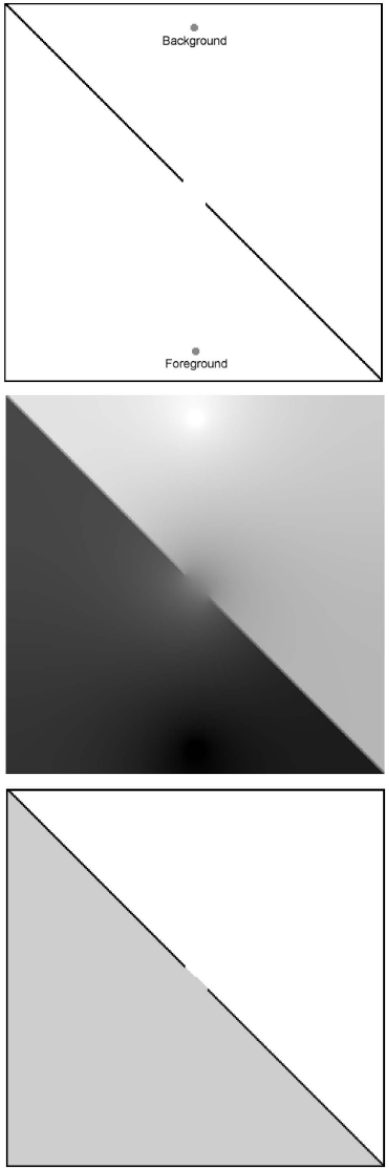
- Like agglomerative clustering based on pair-wise distances d_{ij}^2
- Turned into pair-wise affinities $w_{ij} = \exp\left(-\lambda d_{ij}^2\right)$ with a parameter $\lambda > 0$
- Symmetric $N \times N$ **affinity matrix** W (N: number of data points/pixels)
- Affinity matrix can also be regarded as a fully connected graph with N nodes
- Groups of points with strong internal connections generate clusters
- Find cut(s) of the graph such that each part is strongly connected (normalized cut vs. minimum cut)



Affinities become edge weights; weakest edges not shown here

Towards the normalized cut: random walk

- Related segmentation approach for supervised segmentation
- Given a graph and some seed nodes with cluster labels
- At each seed let a particle start and move randomly through the graph.
- Where does it end after infinite time? Repeat the experiment and measure for each node the probability of each particle ending there
- Assign to each node the label with the largest probability (rounding)
- Can be formulated as weighted linear diffusion on the graph (→ efficient computation)



Author: Leo Grady

Formal writing as a multi-label optimization problem

- Introduce K labeling vectors $\mathbf{u}^k$. Each vector has N components (= number of nodes in the graph)

- Integer problem with $u_i^k \in \{0, 1\}$
  Relaxed version with $u_i^k \in [0, 1]$ (actual random walk)

- Minimize

$$E(\mathbf{u}) = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} w_{ij} (u_i^k - u_j^k)^2$$

(gradient descent leads to linear diffusion with diffusivity w_{ij})

subject to

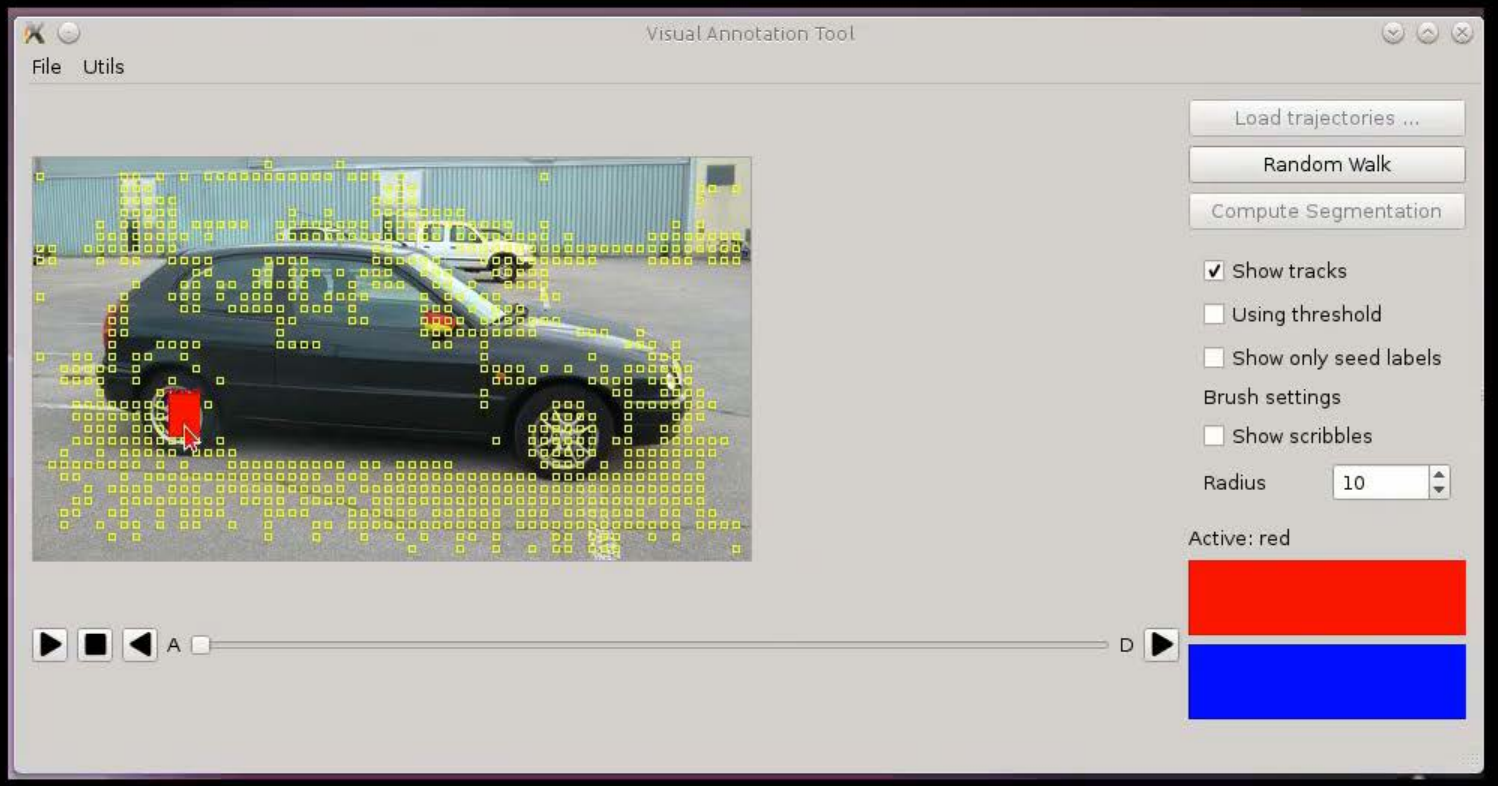
$$u_i^k = 1 \; \forall i \in \mathcal{S}^k$$

(keep seed labels $\mathcal{S}^k$ untouched)

$$\sum_k u_i^k = 1 \; \forall i$$

(labels must sum to 1)

Random walk example



Video Annotation Tool (Live Demo)

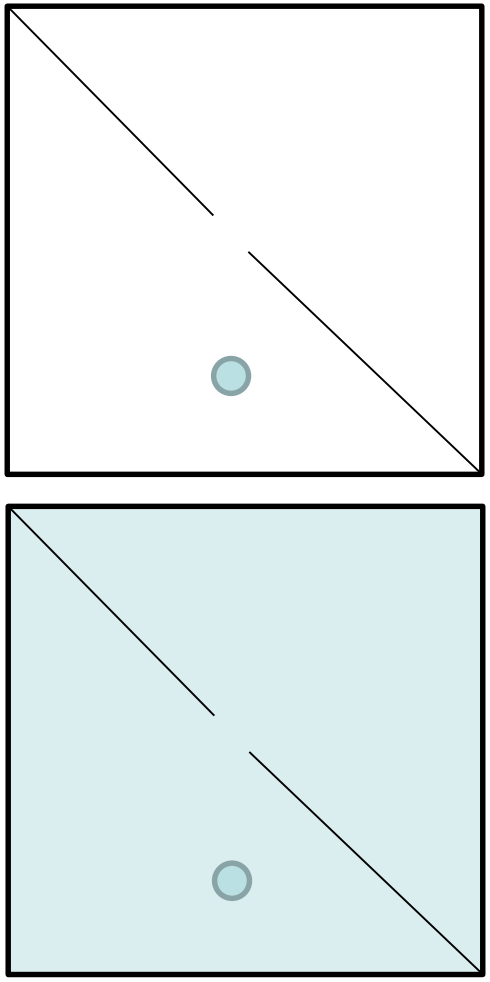
Nagaraja et al. 2015

Case without any seeds

- Back to the original problem, where we wanted to optimally separate the graph into parts that are themselves well connected
- Two cluster case for the beginning
- Imagine we put a seed at an arbitrary node in the graph
- Running diffusion leads to the equilibrium: all nodes get the seed value → trivial solution
- No surprise: this is the global optimum of

$$E(\mathbf{u}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} w_{ij}(u_i - u_j)^2$$

- Do not give up too early….



Adding orthonormality constraints

- What is the global optimum that is different from the trivial solution?

$$E(\mathbf{u}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} w_{ij}(u_i - u_j)^2 \qquad \mathbf{u}^\top \mathbf{1} = 0$$

maximally smooth

Orthogonal to the trivial solution

- We get a non-trivial solution

- To make this solution independent of the choice of the seed value, enforce a unit length vector:

$$|\mathbf{u}| = 1$$

- Soft separation of the graph

- Can be formulated as an eigenvalue problem

Graph Laplacian

- Define a diagonal matrix D , where

$$d_i = \sum_j w_{ij}$$

(the sum of all edge weights connected to node i)

- Graph Laplacian $D - W$

- Eigenvalue problem on the Laplacian:

$$\mathbf{u}^* = \text{argmin}_{\mathbf{u}} \frac{\mathbf{u}^\top (D - W)\mathbf{u}}{\mathbf{u}^\top \mathbf{u}}$$

- Corresponds to minimizing

$$E(\mathbf{u}) = \sum_{i=1}^{N} \sum_j w_{ij}(u_i - u_j)^2$$

subject to $\|\mathbf{u}\| = 1$

Eigendecomposition of the Graph Laplacian

- Eigenvalue problem:

$$\mathbf{u}^* = \text{argmin}_{\mathbf{u}} \frac{\mathbf{u}^\top (D - W)\mathbf{u}}{\mathbf{u}^\top \mathbf{u}}$$

- Minimum value is the smallest eigenvalue of $D - W$
- Corresponding argument is the corresponding eigenvector
- Trivial solution: constant vector with unit length leads to the minimum 0
- More interesting: eigenvector corresponding to the second smallest eigenvalue (by definition orthogonal to the first eigenvector)
- "The smoothest $\mathbf{u}$ that is orthogonal to the constant vector"

Graph theoretic interpretation

- Computing the eigenvector corresponding to the second smallest eigenvalue of $D - W$ is a relaxed minimization of the **average cut**

$$\frac{\text{cut}(A, \Omega - A)}{|A|} + \frac{\text{cut}(A, \Omega - A)}{|\Omega - A|}$$

which is a symmetrized version of the **ratio cut**

$$\frac{\text{cut}(A, \Omega - A)}{|A|}$$

- In contrast to the minimum cut, the average cut has a non-trivial global minimum

- This is thanks to the normalization by the size of the emerging regions A and $\Omega - A$

Normalized graph Laplacian and the normalized cut

- The graph Laplacian is often normalized

$$D - W \quad \rightarrow \quad D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$$

(weights divided by the geometric mean of the row and column sums)

$$w_{ij} \rightarrow \frac{w_{ij}}{\sqrt{d_i d_j}}$$

- The eigenvector $\mathbf{z}$ of the normalized problem must be rescaled:
$\mathbf{u} = D^{-\frac{1}{2}}\mathbf{z}$ (see Shi-Malik 2000 for details)

- This leads to the **normalized cut**

$$\frac{\mathrm{cut}(A, \Omega - A)}{\mathrm{assoc}(A, \Omega)} + \frac{\mathrm{cut}(A, \Omega - A)}{\mathrm{assoc}(\Omega - A, \Omega)}$$

Measures the removed edges (cut) relative to the total edge weight originating in each region (assoc)

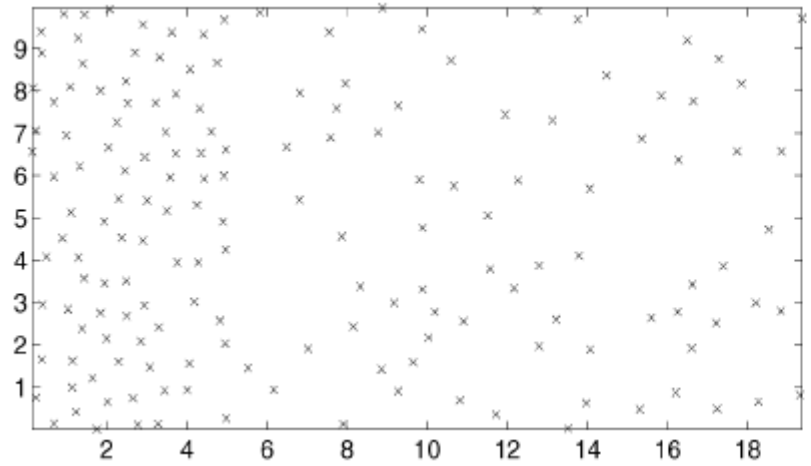
Normalized cut vs. average cut

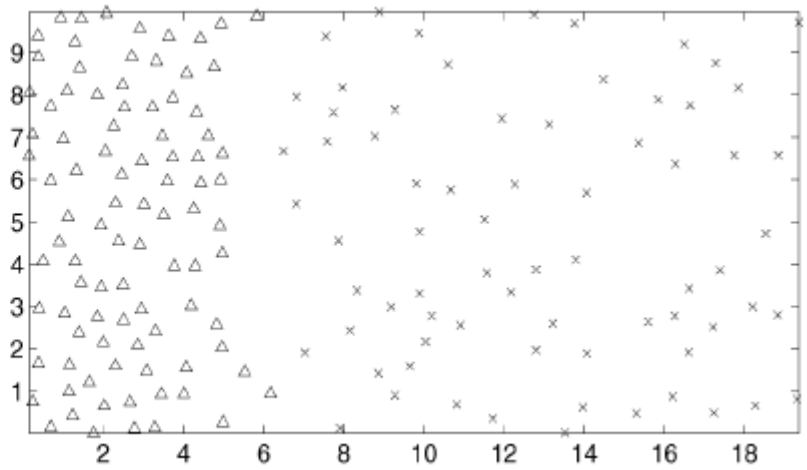
- Minimizing the normalized cut is equivalent to maximizing the **normalized association**

$$\frac{\text{assoc}(A, A)}{\text{assoc}(A, \Omega)} + \frac{\text{assoc}(\Omega - A, \Omega - A)}{\text{assoc}(\Omega - A, \Omega)}$$

- Advantage I: Limiting case where looking for similarities is the same as looking for differences

- Advantage II: Can separate areas with higher connectivity from areas with lower connectivity





Eigensolver

- In many applications, the affinity matrix is large
  → usual numerical methods for computing eigenvalues with cubic complexity $\mathcal{O}(N^3)$ are prohibitively slow

1. Some graphs may not be fully connected
   → matrix will be sparse (many zero entries)

2. We just require the smallest eigenvalues and their corresponding eigenvectors, not all of them

- Can we exploit this to have a faster solver?

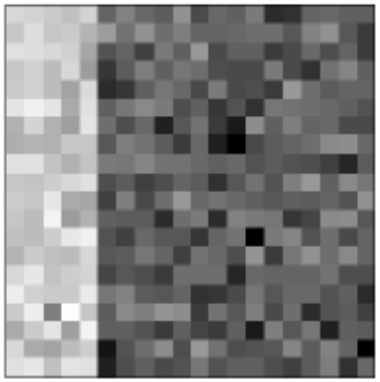
- Indeed there is the **Lanczos method**, which efficiently computes just the smallest (or largest) eigenvalues of a matrix in $\mathcal{O}(N^2)$

- Available via the ARPACK numerical library (wrapped by SciPy via “eigs”)

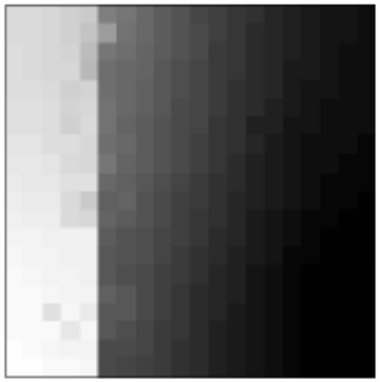
Binary segmentation

- Recall: We solve the real-valued relaxed problem (spectral relaxation)
  → generally no clear cut of the graph, just soft indicators

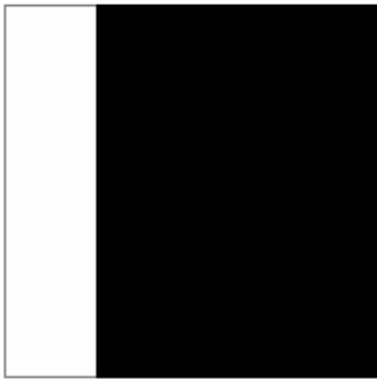
- We get an approximate solution of the binary segmentation problem (average cut or normalized cut) by thresholding the eigenvector (rounding)



Input image



2nd Eigenvector

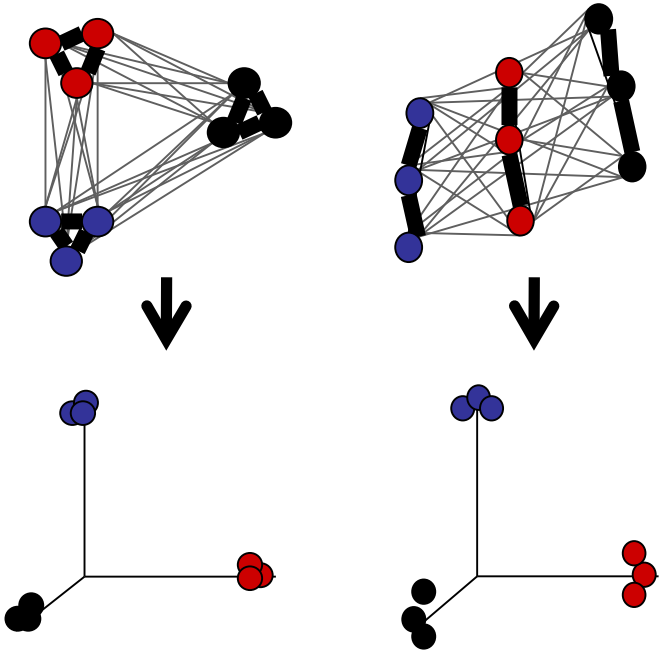


After thresholding

- Average cut and normalized cut are NP hard problems

More than two clusters, Laplacian eigenmaps

- The concept can be extended to more than two clusters
- The eigenvector to the third smallest eigenvalue indicates an alternative partitioning of the graph
- In the general case we compute the K smallest eigenvalues
- The corresponding eigenvectors span a K- dimensional subspace
  - Each node (data point) maps to a K-dimensional vector
  - Mapping is called **Laplacian eigenmap**
- We can run standard clustering techniques (k-means) to convert the real-valued vectors into integer labels
  → **spectral clustering**



Mappings from a graph to its eigenspace, clustering in the eigenspace

Balancing property of spectral clustering

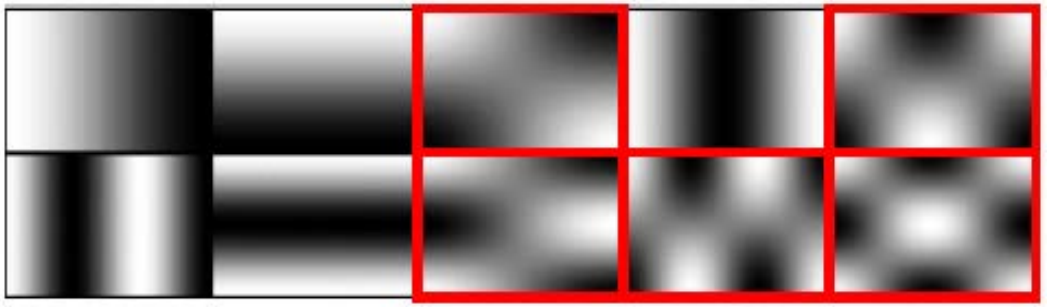
- The balancing property of the average and normalized cut transfers to spectral clustering, too.

$$\frac{\text{cut}(A, \Omega - A)}{|A|} + \frac{\text{cut}(A, \Omega - A)}{|\Omega - A|}$$

(average cut)

- Consequence: When affinities are not informative, spectral clustering tends to create equally sized regions

- With constant affinities, the eigenvectors corresponding to the smallest eigenvalues yield the Fourier basis



Eigenvectors of constant affinity matrix

- Consider balancing when applying spectral clustering on non-uniform grids

After thresholding

Finding coherent subsets

- Another graph based technique is to find the most connected subset among the nodes:

$$\max_A \left(\frac{\text{assoc}(A, A)}{|A|} \right)$$

- Can be used to find outliers in a dataset
- Relaxed problem: find the largest eigenvector of the affinity matrix W
- Nodes of the most connected subset obtain non-zero values
- Largest eigenvector can be found also by **power iteration** if $\lambda_1 > \lambda_2$

$$v^{k+1} = \frac{Wv^k}{\|Wv^k\|}$$ (very simple algorithm)

- Major part of the PageRank algorithm that initiated Google's success

References

- L. Grady: Random walks for image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768-1783, 2006.

- J. Shi, J. Malik: Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888-905, 2000.

- F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.

- A. Ng, M. Jordan, Y. Weiss: On spectral clustering: analysis and an algorithm, *Advances in Neural Information Processing Systems (NIPS)*, 2001.

- P. Perona, W. Freeman: A factorization approach to grouping, *European Conference on Computer Vision*, pp. 655-670,1998.