Normalized Cuts

Shi & Malik, IEEE PAMI 2000

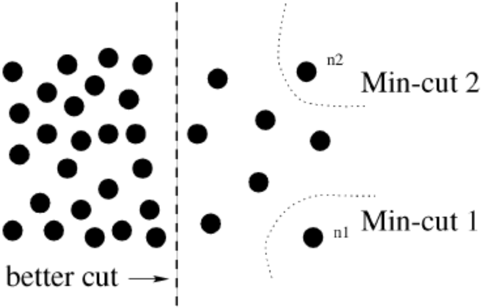
Jan Kybic

2020

Unsupervised segmentation - graph partitioning

- Pixels = vertices
- Edges = neighbors
- Edge weights = similarities
- Segmentation = finding a cut (partitioning)
- Classes = graph components

Minimum cut



$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v).$$

Normalized cut

Relative cost of the cut

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)},$$

$$assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$$

Computing the normalized cut

Indicator $x_i = 1$if $i \in A$, otherwise $x_i = -1$

Connection weight : $d(i) = \sum_j w(i,j)$

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(B,A)}{assoc(B,V)}$$
$$= \frac{\sum_{(x_i>0,x_j<0)} -w_{ij}x_ix_j}{\sum_{x_i>0} d_i} + \frac{\sum_{(x_i<0,x_j>0)} -w_{ij}x_ix_j}{\sum_{x_i<0} d_i}.$$

Rayleigh quotient

$$k = \frac{\sum_{x_i>0} \boldsymbol{d}_i}{\sum_i \boldsymbol{d}_i} \qquad b = \frac{k}{1-k}$$

$$\boldsymbol{y} = (\mathbf{1} + \boldsymbol{x}) - b(\mathbf{1} - \boldsymbol{x}),$$

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\mathbf{D} - \mathbf{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathbf{D}\boldsymbol{y}}$$

with $\boldsymbol{y}^T\mathbf{D}\mathbf{1} = 0$ and $y(i) \in \{2, -2b\}$

Motivation

Solving

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\mathbf{D}-\mathbf{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathbf{D}\boldsymbol{y}}$$

is equivalent to

$$\inf_{\boldsymbol{y}^T\mathbf{D}\mathbf{1}=0} \frac{\sum_i \sum_j (\boldsymbol{y}(i)-\boldsymbol{y}(j))^2 w_{ij}}{\sum_i \boldsymbol{y}(i)^2 \mathbf{d}(i)}$$

spring analogy - oscillatory modes

Eigenvalue solution

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\mathbf{D}-\mathbf{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathbf{D}\boldsymbol{y}}$$

Relaxation solved by the generalized eigenvalue problem

$$(\mathbf{D}-\mathbf{W})\boldsymbol{y} = \lambda \mathbf{D}\boldsymbol{y}.$$

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}\boldsymbol{x} = \lambda \boldsymbol{x}$$

$\rightarrow$

$y_0 = 1$ is an eigenvector with $\lambda = 0$. 2nd largest eigenvalue $\rightarrow$ eigenvector y_1 = **solution** satisfying $\boldsymbol{y}^T\mathbf{D1} = 0$

Graph construction

- Construct a weighted graph

$$w_{ij} = e^{\frac{-\|\boldsymbol{F}_{(i)} - \boldsymbol{F}_{(j)}\|_2^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-\|\boldsymbol{X}_{(i)} - \boldsymbol{X}_{(j)}\|_2^2}{\sigma_X^2}} & \text{if } \|X(i) - X(j)\|_2 < r \\ 0 & \text{otherwise.} \end{cases}$$

- Solve $(\mathbf{D} - \mathbf{W})\boldsymbol{y} = \lambda \mathbf{D}\boldsymbol{y}.$
  Complexity
  $O(N)$ thanks to sparsity and low accuracy requirement

Other pixel features

- $\boldsymbol{F}(i) = \boldsymbol{I}(i)$, the intensity value, for segmenting brightness images,
- $\boldsymbol{F}(i) = [v, v \cdot s \cdot sin(h), v \cdot s \cdot cos(h)](i)$, where h, s, v are the HSV values, for color segmentation,
- $\boldsymbol{F}(i) = [|\boldsymbol{I} * f_1|, \ldots, |\boldsymbol{I} * f_n|](i)$, where the f_i are DOOG filters at various scales and orientations as used in [16], in the case of texture segmentation.

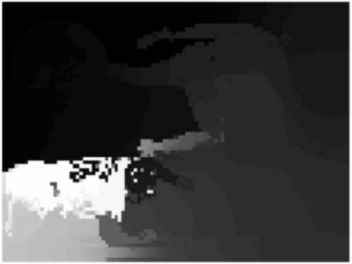
Extended to spatiotemporal data, motion profiles

Finalizing the segmentation

- Thresholding y_i- minimize *Ncut*, l thresholds
- Partition recursively if desired
  - Stop if *Ncut* too high
  - Stop if eigenspectrum too smooth

Example

Example - eigenvectors



Simulataneous k-way cut

- Recursive 2-way cut computationally wasteful
- Use n top eigenvectors for further partitioning as labels
- Numerical inaccuracies $\rightarrow$use $\boldsymbol{k}$-**means** to cluster pixel labels
- Postprocessing
  - Greeedy pruning (merging)
  - Global cut at segment level - eigenvalue formulation or exhaustive
- *Not used in the presented experiments*

Examples

Example (2)

Other eigenvalue formulations

Finding clumps ⟷ Finding splits

	Average association	Normalized Cut	Average cut
Discrete formulation	$\frac{asso(A,A)}{\lvert A\rvert} + \frac{asso(B,B)}{\lvert B\rvert}$	$\frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$ or $2 - \left(\frac{asso(A,A)}{asso(A,V)} + \frac{asso(B,B)}{asso(B,V)}\right)$	$\frac{cut(A,B)}{\lvert A\rvert} + \frac{cut(A,B)}{\lvert B\rvert}$
Continuous solution	$Wx = \bar{\lambda} x$	$(D-W)x = \bar{\lambda} D x$ or $W x = (1 - \bar{\lambda}) D x$	$(D-W)x = \bar{\lambda} x$

Conclusions

- Unsupervised algorithm based on graph clustering
- Penalizes small classes
- Eigenvalue (spectral) formulation
- Lot of theory, short experimental evaluation