NETWORK SCIENCE

Graph Clustering and Partitioning

Prof. Marcello Pelillo

Ca' Foscari University of Venice

a.y. 2016/17

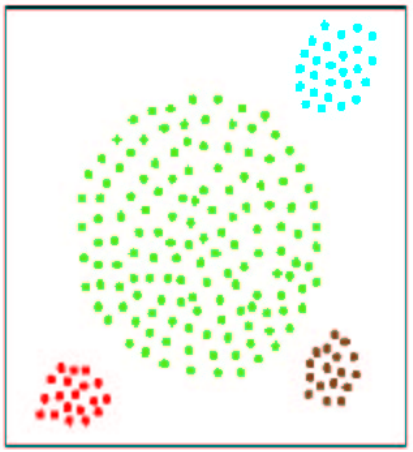
The Clustering Problem

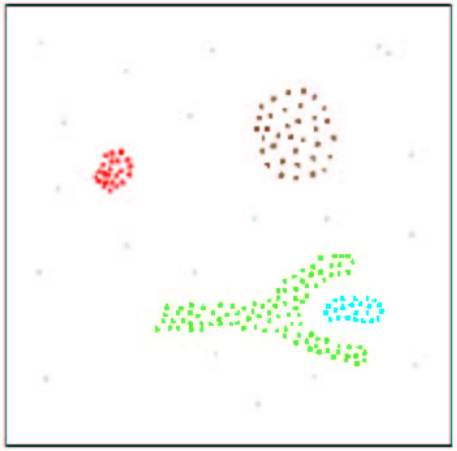
Given:

- a set of n "objects"
- an $n \times n$ matrix A of pairwise similarities

} = an edge-weighted graph

Goal: Group the the input objects (the vertices of the graph) into maximally homogeneous classes (i.e., clusters).





What Is a Cluster?

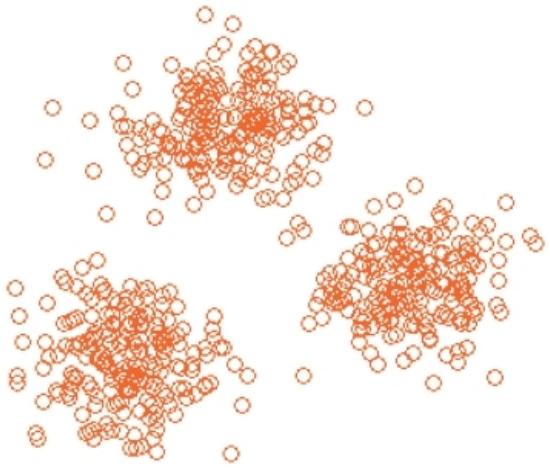
No universally accepted (formal) definition of a "cluster" but, informally, a cluster should satisfy two criteria:

Internal criterion

all "objects" *inside* a cluster should be highly similar to each other

External criterion

all "objects" *outside* a cluster should be highly dissimilar to the ones inside



An answer from game theoy

The classical notion of ESS equilibrium provides a general and elegant answer to the question above.

The Clustering Game

In the (pairwise) clustering game we have:

- ✓ Two players (because we have pairwise affinities)
- ✓ Pure strategies = objects to be clustered
- ✓ Payoff matrix = similarity matrix

It is in each player's interest to pick an element that is similar to the one that the adversary is likely to choose.

ESS's abstract well the main characteristics of a cluster:

- ✓ **Internal coherency**: High mutual support of all elements within the group
- ✓ **External incoherency**: Low support from elements of the group to elements outside the group

ESS's are equivalent to dominant sets (Pavan and Pelillo, 2007).

Special Case #1: Symmetric Affinities

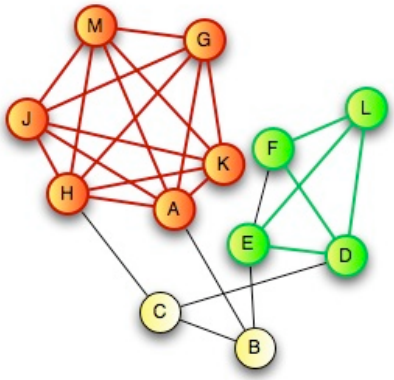
Given a symmetric real-valued matrix A (with null diagonal), consider the following Standard Quadratic Programming problem (StQP):

$$\begin{aligned} \text{maximize} \quad & f(x) = x^T A x \\ \text{subject to} \quad & x \in \Delta \end{aligned}$$

Note. The function $f(x)$ provides a measure of cohesiveness of a cluster (see Pavan and Pelillo, 2007; Sarkar and Boyer, 1998; Perona and Freeman, 1998).

ESS's are in one-to-one correspondence to (strict) local solutions of StQP

Special Case #2: Binary Symmetric Affinities



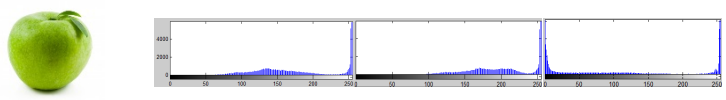
Suppose the similarity matrix is a binary (0/1) matrix.

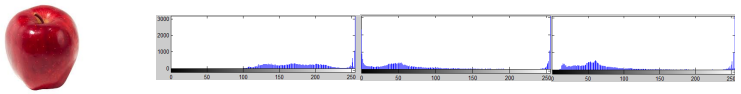
Given an unweighted undirected graph $G=(V,E)$:

A *clique* is a subset of mutually adjacent vertices
A *maximal clique* is a clique that is not contained in a larger one

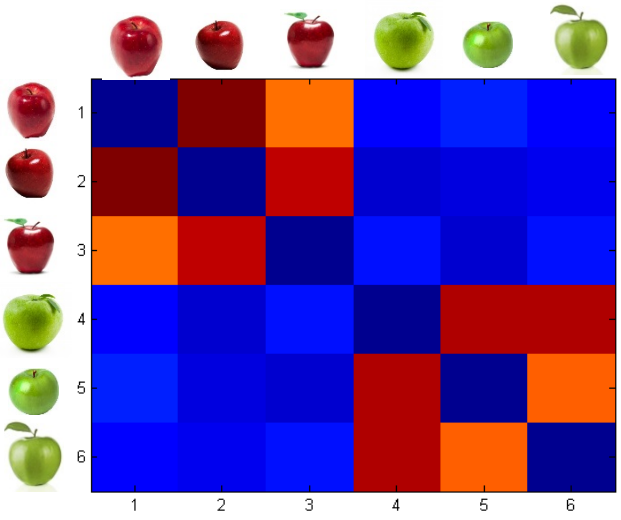
ESS's are in one-to-one correspondence to maximal cliques of G

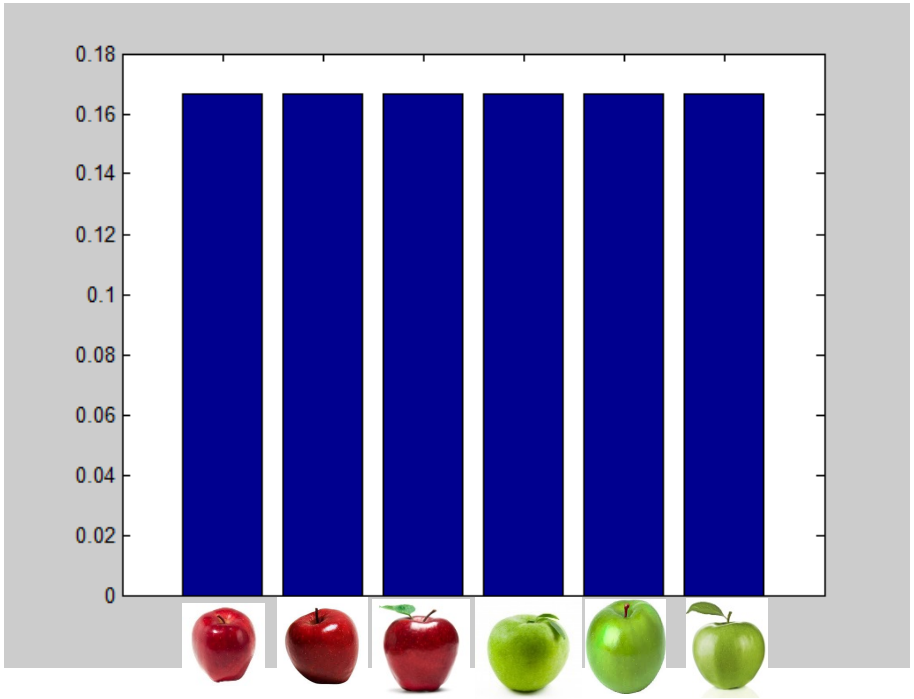
A Toy Example: Grouping Apples





Payoffs between apples is computed using a distance between RGB histograms

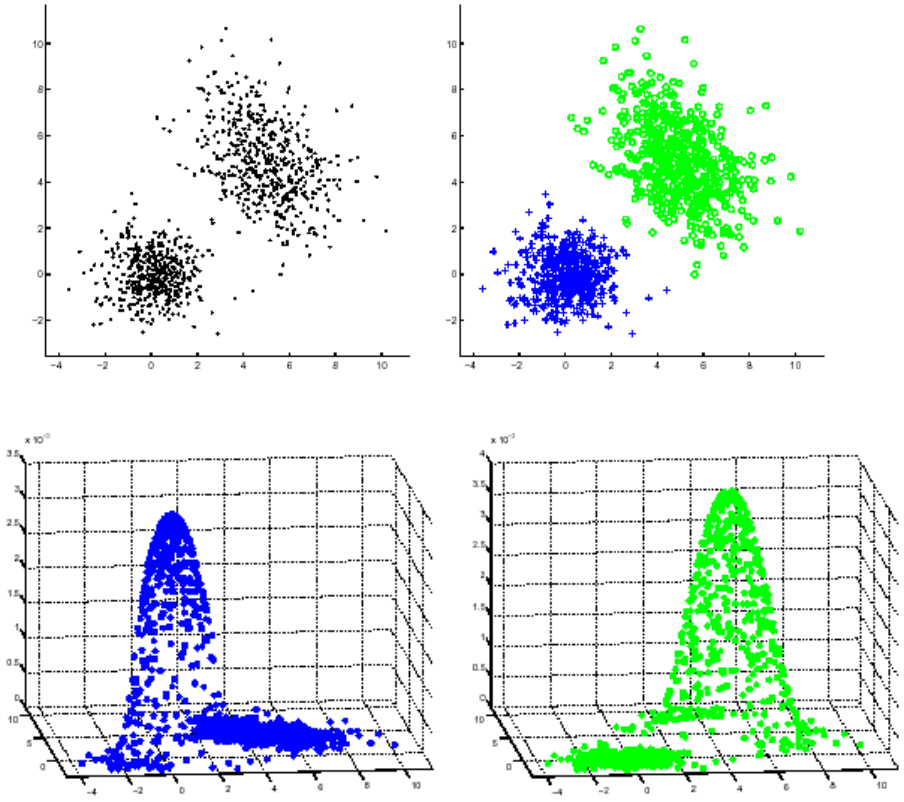




Measuring the Degree of Cluster Membership

The components of the converged vector give us a measure of the participation of the corresponding vertices in the cluster, while the value of the objective function provides of the cohesiveness of the cluster.

See Rosch's *prototype theory* of categorization (e.g., Lakoff, 1987).

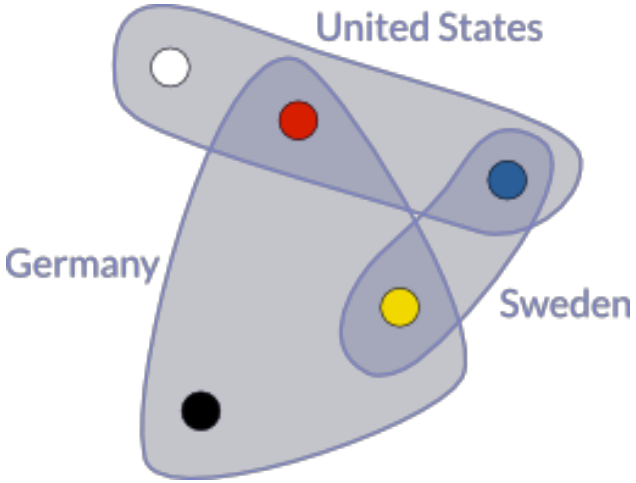


Dealing with High-Order Similarities

A (weighted) hypergraph is a triplet $H = (V, E, w)$, where

- V is a finite set of vertices
- $E \subseteq 2^V$ is the set of (hyper-)edges (where 2^V is the power set of V)
- $w : E \rightarrow R$ is a real-valued function assigning a weight to each edge

We will focus on a particular class of hypergraphs, called **k-graphs**, whose edges have fixed cardinality $k \geq 2$.



A hypergraph where the vertices are flag colors and the hyperedges are flags.

The Hypergraph Clustering Game

Given a weighted k-graph representing an instance of a hypergraph clustering problem, we cast it into a k-player (hypergraph) clustering game where:

- ✓ There are k players
- ✓ The objects to be clustered are the pure strategies
- ✓ The payoff function is proportional to the similarity of the objects/ strategies selected by the players

Definition (ESS-cluster). Given an instance of a hypergraph clustering problem H = (V,E,w), an ESS-cluster of H is an ESS of the corresponding hypergraph clustering game.

As in the k=2 case, ESS-clusters do incorporate both internal and external cluster criteria (see PAMI 2013)

An exampe: Line Clustering

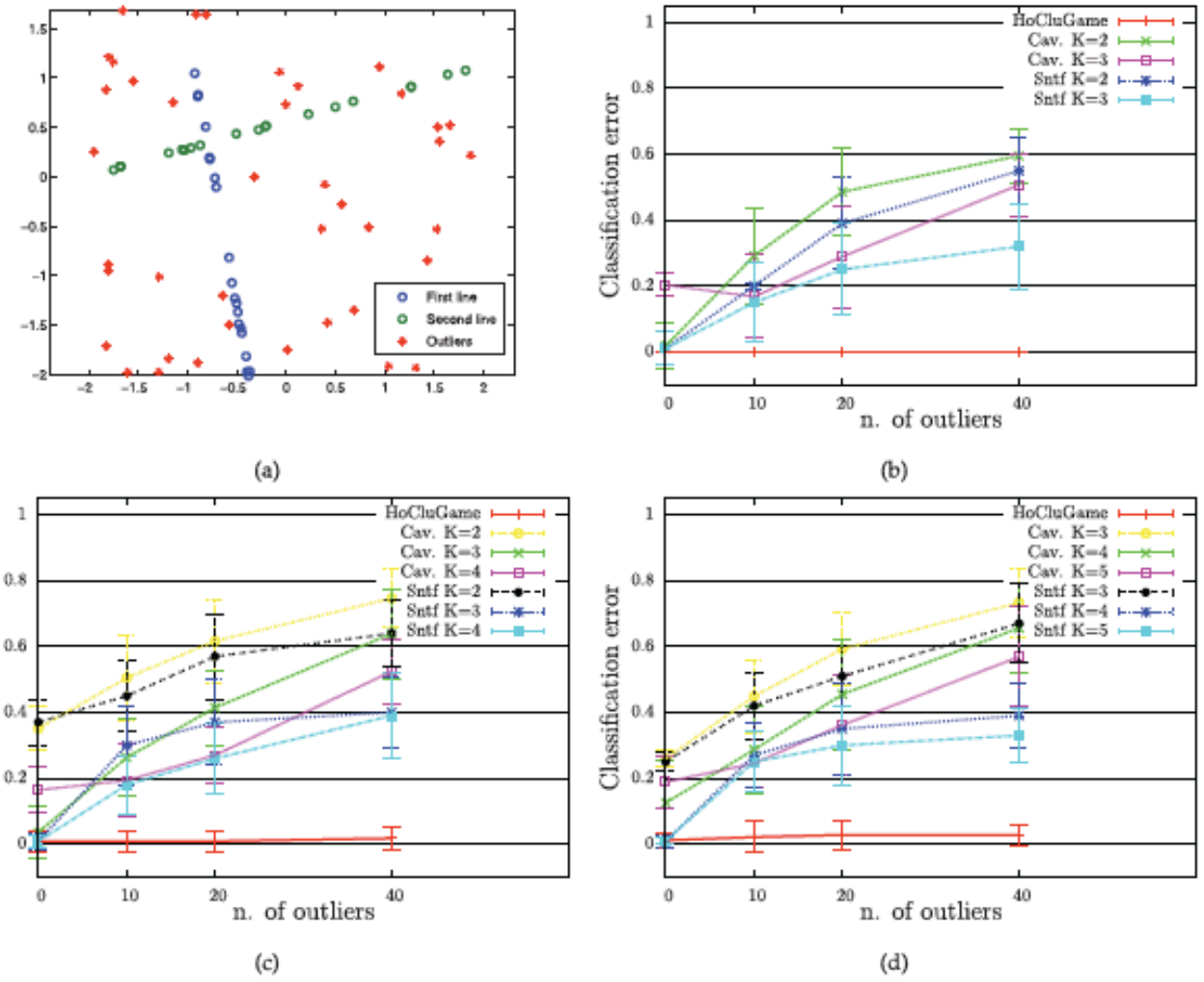


Fig. 3. Results of clustering two, three, and four lines with an increasing number of clutter points $(0, 10, 20, 40)$. (a) Example of two 5D lines (projected in 2D) with 40 clutter points. (b) Two lines. (c) Three lines. (d) Four lines.

In a nutshell…

The game-theoretic/dominant-set approach:

- ✓ makes no assumption on the structure of the affinity matrix, being it able to work with asymmetric and even negative similarity functions
- ✓ does not require *a priori* knowledge on the number of clusters (since it extracts them sequentially)
- ✓ leaves clutter elements unassigned (useful, e.g., in figure/ground separation or one-class clustering problems)
- ✓ allows principled ways of assigning out-of-sample items *(NIPS'04)*
- ✓ allows extracting overlapping clusters *(ICPR'08)*
- ✓ generalizes naturally to hypergraph clustering problems, i.e., in the presence of high-order affinities, in which case the clustering game is played by more than two players *(PAMI'13)*
- ✓ extends to hierarchical clustering (ICCV'03: EMMCVPR'09)
- ✓ allows using multiple affinity matrices using Pareto-Nash notion *(SIMBAD'15)*

References

Evolutionary game theory

J. Weibull. Evolutionary Game Theory. MIT Press (1995).

J. Hofbauer and K. Sigmund. Evolutionary Games and Population Dynamics. Cambridge University Press (1998).

Dominant sets

S. Rota Bulò and M. Pelillo. Dominant set clustering: A review. EJOR (under reveiw).

M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. *CVPR 2003*.

M. Pavan and M. Pelillo. Dominant sets and hierarchical clustering. *ICCV 2003*.

M. Pavan and M. Pelillo. Efficient out-of-sample extension of dominant-set clusters. *NIPS 2004*.

A. Torsello, S. Rota Bulò and M. Pelillo. Grouping with asymmetric affinities: A game-theoretic perspective. *CVPR 2006*.

M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *PAMI 2007*.

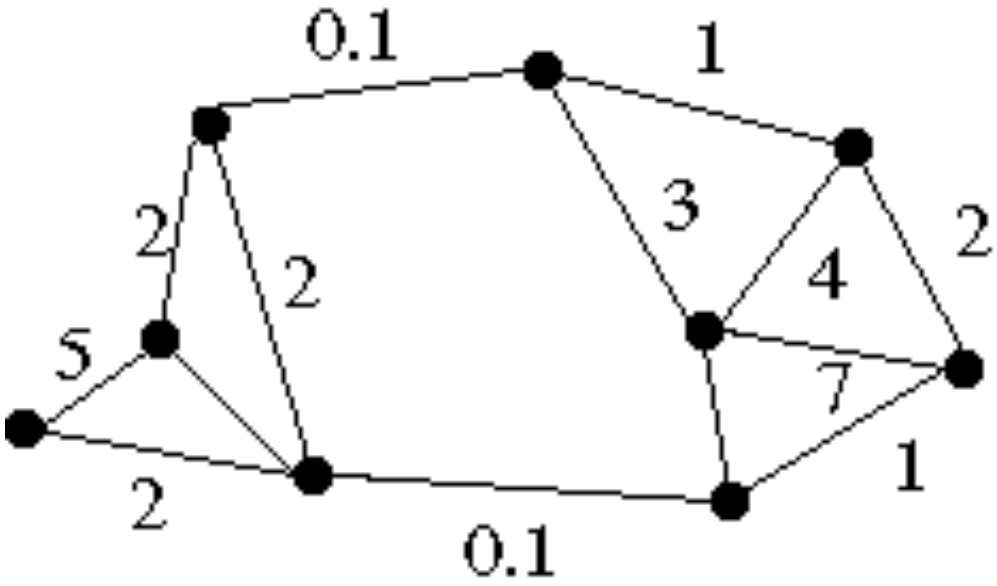
A. Torsello, S. Rota Bulò and M. Pelillo. Beyond partitions: Allowing overlapping groups in pairwise clustering. *ICPR 2008*.

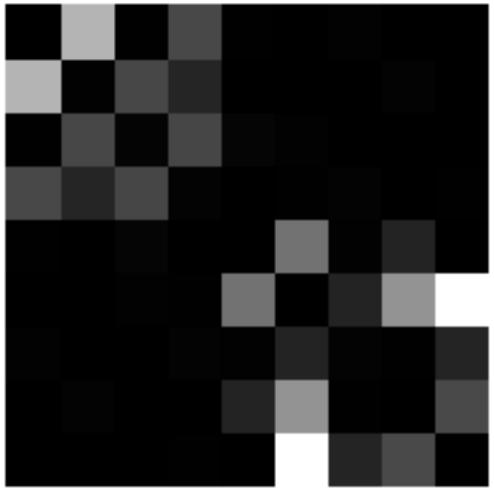
S. Rota Bulò and M. Pelillo. A game-theoretic approach to hypergraph clustering. *PAMI'13*.

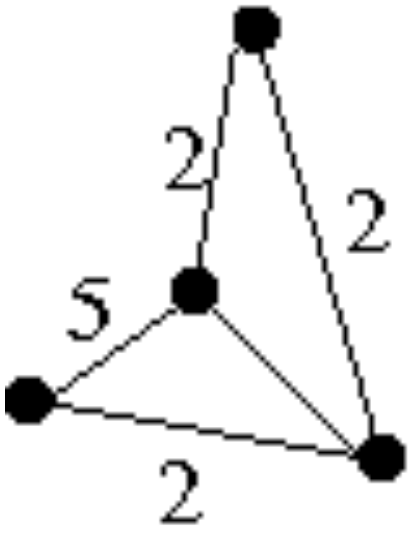
S. Rota Bulò, M. Pelillo and I. M. Bomze. Graph-based quadratic optimization: A fast evolutionary approach. CVIU 2011.

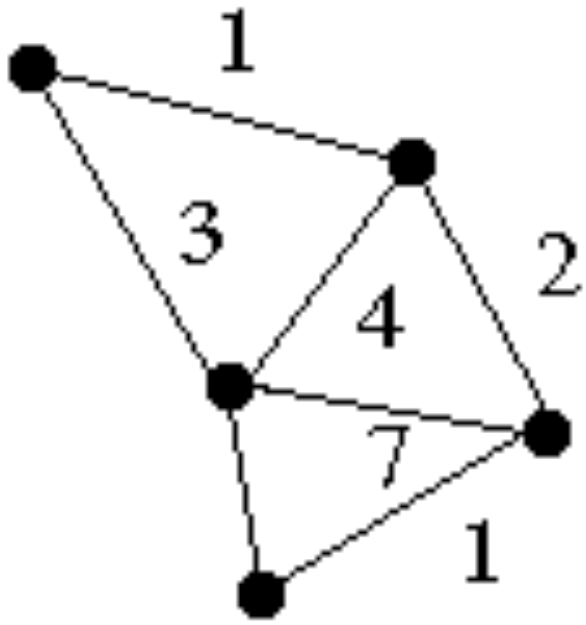
Graph Partitioning

- Represent tokens using a weighted graph.
  - affinity matrix
- Cut up this graph to get subgraphs with strong interior links


0.1
1
2
2
3
4
2
5
7
1
2
0.1




2
2
5
2


1
3
2
4
7
1

Eigenvector-based clustering

- Simplest idea: we want a vector a giving the association between each element and a cluster
- We want elements within this cluster to, on the whole, have strong affinity with one another
- We could maximize

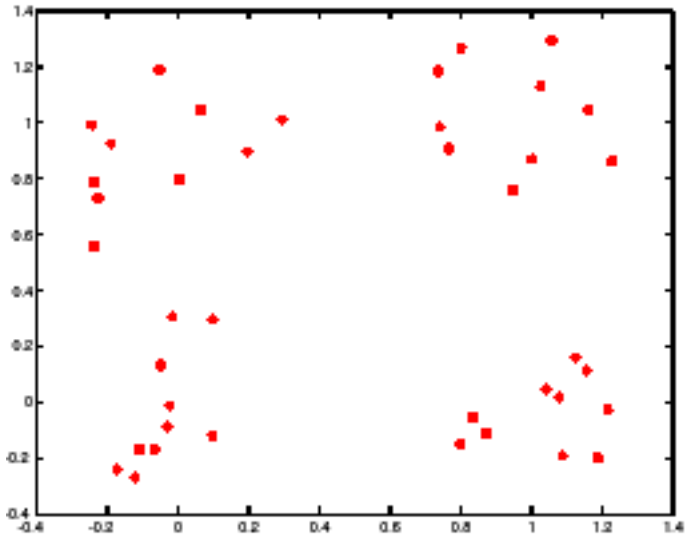
$$a^T A a$$

- But need the constraint

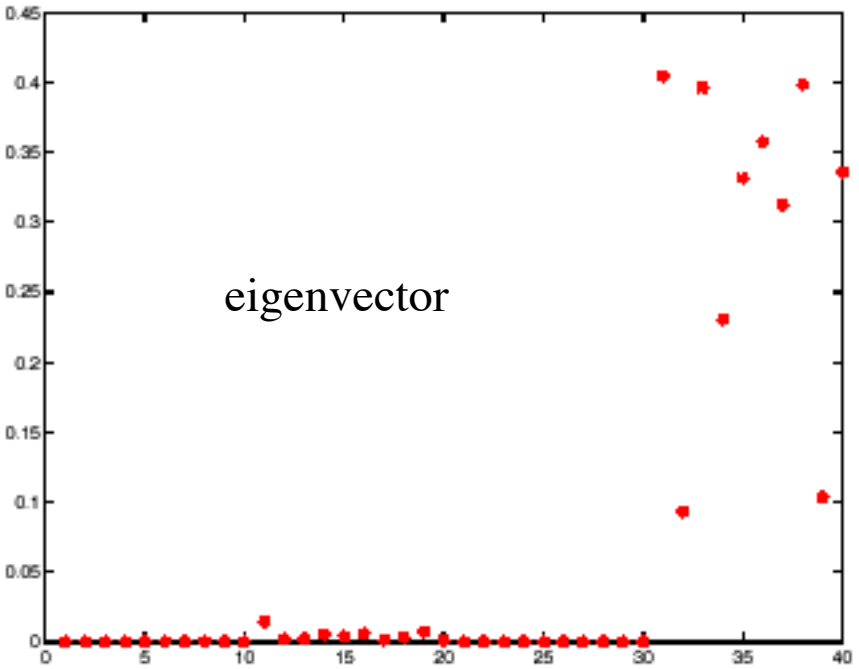
$$a^T a = 1$$

- This is an eigenvalue problem - choose the eigenvector of A with largest eigenvalue

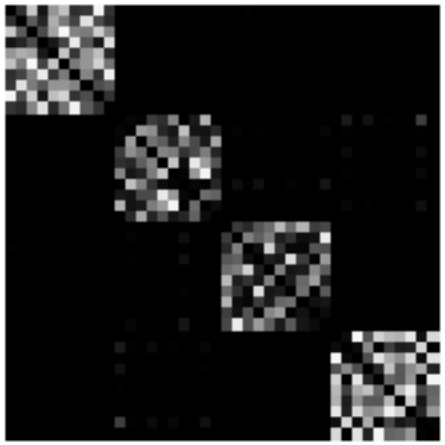
Example eigenvector



points



eigenvector



matrix

More than two segments

- Two options
  - Recursively split each side to get a tree, continuing till the eigenvalues are too small
  - Use the other eigenvectors

Clustering by eigenvectors: Algorithm

1. Construct (or take as input) the affinity matrix A
2. Compute the eigenvalues and eigenvectors of A
3. Repeat
4. Take the eigenvector corresponding to the largest unprocessed eigenvalue
5. Zero all components corresponding to elements that have already been clustered
6. Threshold the remaining components to determine which elements belong to this cluster
7. If all elements have been accounted for, there are sufficient clusters
8. Until there are sufficient clusters

Clustering as graph partitioning

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

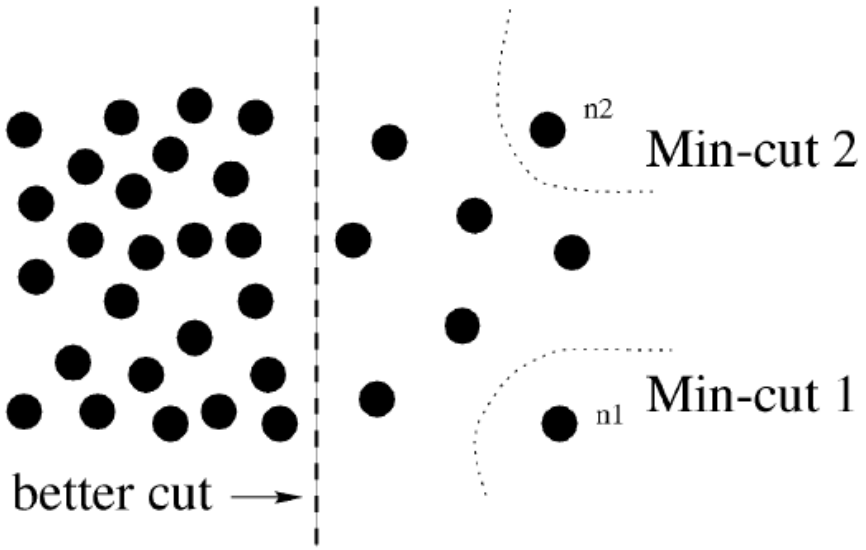


Fig. 1. A case where minimum cut gives a bad partition.

Normalized Cut

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

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

Cut vs Association

$$
\begin{aligned}
Ncut(A,B) &= \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \\
&= \frac{assoc(A,V) - assoc(A,A)}{assoc(A,V)} \\
&\quad + \frac{assoc(B,V) - assoc(B,B)}{assoc(B,V)} \\
&= 2 - \left(\frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)} \right) \\
&= 2 - Nassoc(A,B).
\end{aligned}
$$

Normalized cuts

- Current criterion evaluates within cluster similarity, but not across cluster difference
- Instead, wed like to maximize the within cluster similarity compared to the across cluster difference
- Write graph as V, one cluster as A and the other as B

- Maximize

$$\left(\frac{assoc(A,A)}{assoc(A,V)}\right)+\left(\frac{assoc(B,B)}{assoc(B,V)}\right)$$

- i.e. construct A, B such that their within cluster similarity is high compared to their association with the rest of the graph

- 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 x P diagonal matrix:

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & d_P \end{bmatrix}$$

"degree matrix"

- Let **A** = P x P symmetric matrix:

"affinity matrix"

$$\mathbf{A} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1P} \\ w_{21} & w_{22} & \cdots & w_{2P} \\ & & \cdots & \\ w_{P1} & w_{P2} & \cdots & w_{PP} \end{bmatrix}$$

- It can be shown that

  $\mathbf{y} = \arg\min_{\mathbf{x}} ncut(\mathbf{x})$

  $$= \arg\min_{\mathbf{y}} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{A})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} \text{ subject to } \mathbf{y}^T\mathbf{D}\mathbf{1} = 0$$

- 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: $(\mathbf{D} - \mathbf{A})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$

- 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 $\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}$, then $\mathbf{x}^T\mathbf{M}\mathbf{x} / \mathbf{x}^T\mathbf{x}$ is minimized by the next smallest eigenvector $\mathbf{x}_j$ and its minimum value is the eigenvalue λ_j

- Smallest eigenvector is always 0

  because $A=V$, $B=\{\}$ 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 3^{rd}, 4^{th}, 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

Normalized cuts algorithm

1. Given an image or image sequence, set up a weighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ and set the weight on the edge connecting two nodes to be a measure of the similarity between the two nodes.
2. Solve $(\mathbf{D} - \mathbf{W})\boldsymbol{x} = \lambda \mathbf{D}\boldsymbol{x}$ for eigenvectors with the smallest eigenvalues.
3. Use the eigenvector with the second smallest eigenvalue to bipartition the graph.
4. Decide if the current partition should be subdivided and recursively repartition the segmented parts if necessary.

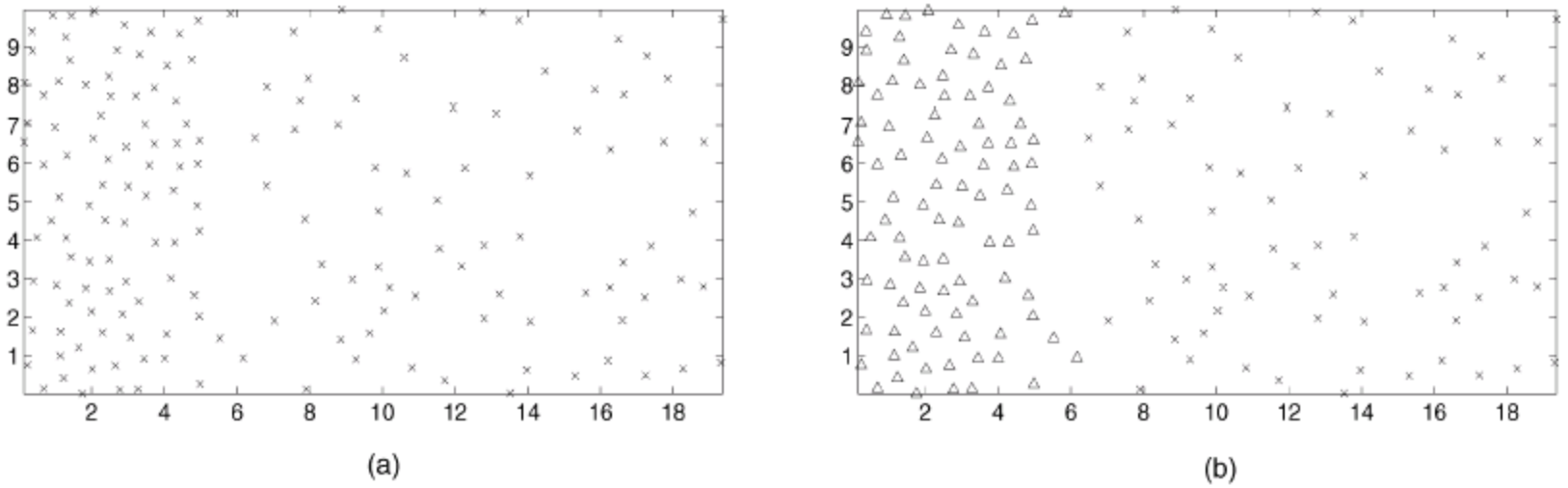


Fig. 5. (a) Point set generated by two Poisson processes, with densities of 2.5 and 1.0 on the left and right clusters respectively, (b) $\triangle$ and $\times$ indicate the partition of point set in (a). Parameter settings: $\sigma_X = 5$, $r = 3$.

Application to Image Segmentation

An image is represented as an edge-weighted undirected graph, where vertices correspond to individual pixels and edge-weights reflect the "similarity" between pairs of vertices.

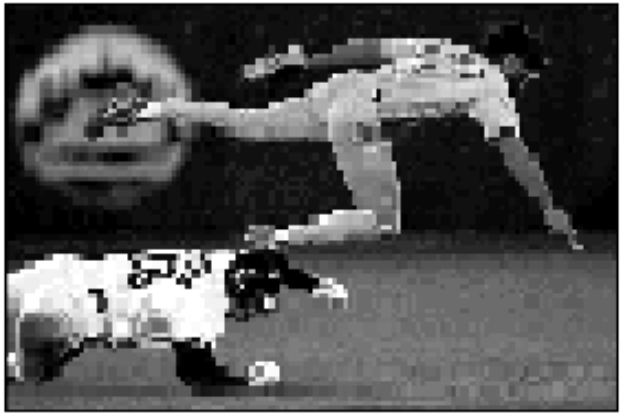
For the sake of comparison, in the experiments we used the same similarities used in Shi and Malik's normalized-cut paper (PAMI 2000).

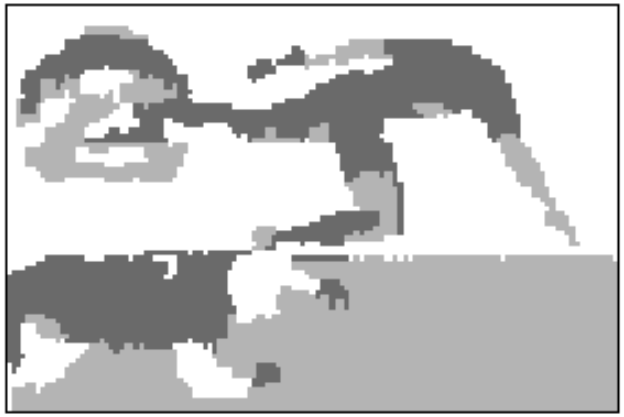
To find a hard partition, the following *peel-off* strategy was used:

```
Partition_into_dominant_sets(G)
Repeat
   find a dominant set
   remove it from graph
until all vertices have been clustered
```

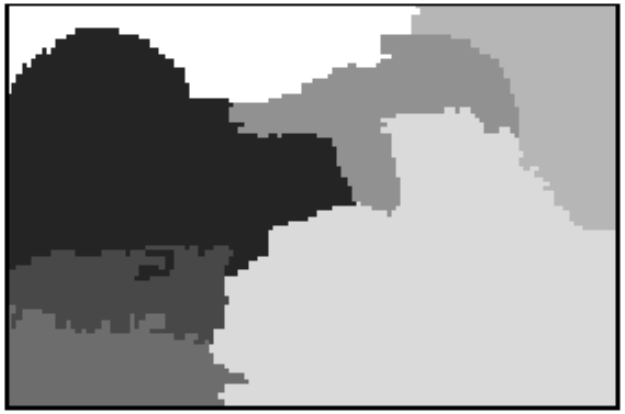
To find a single dominant set we used replicator dynamics (but see Rota Bulò, Pelillo and Bomze, *CVIU 2011*, for faster game dynamics).

Intensity Segmentation Results



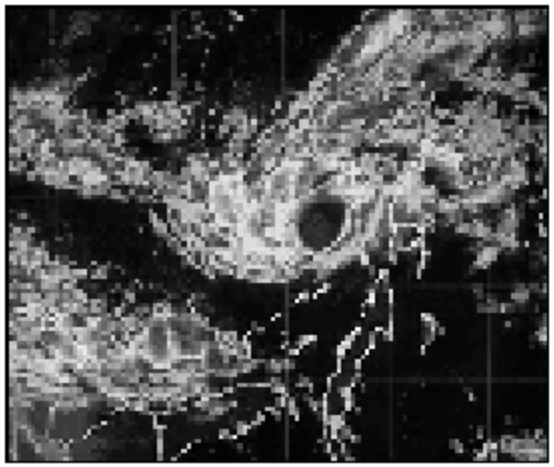


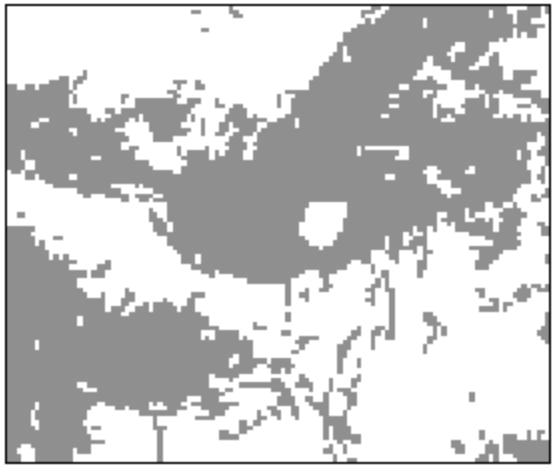
Dominant sets



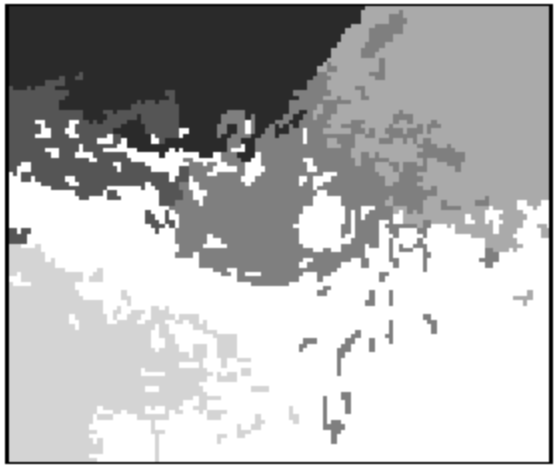
Ncut

Intensity Segmentation Results



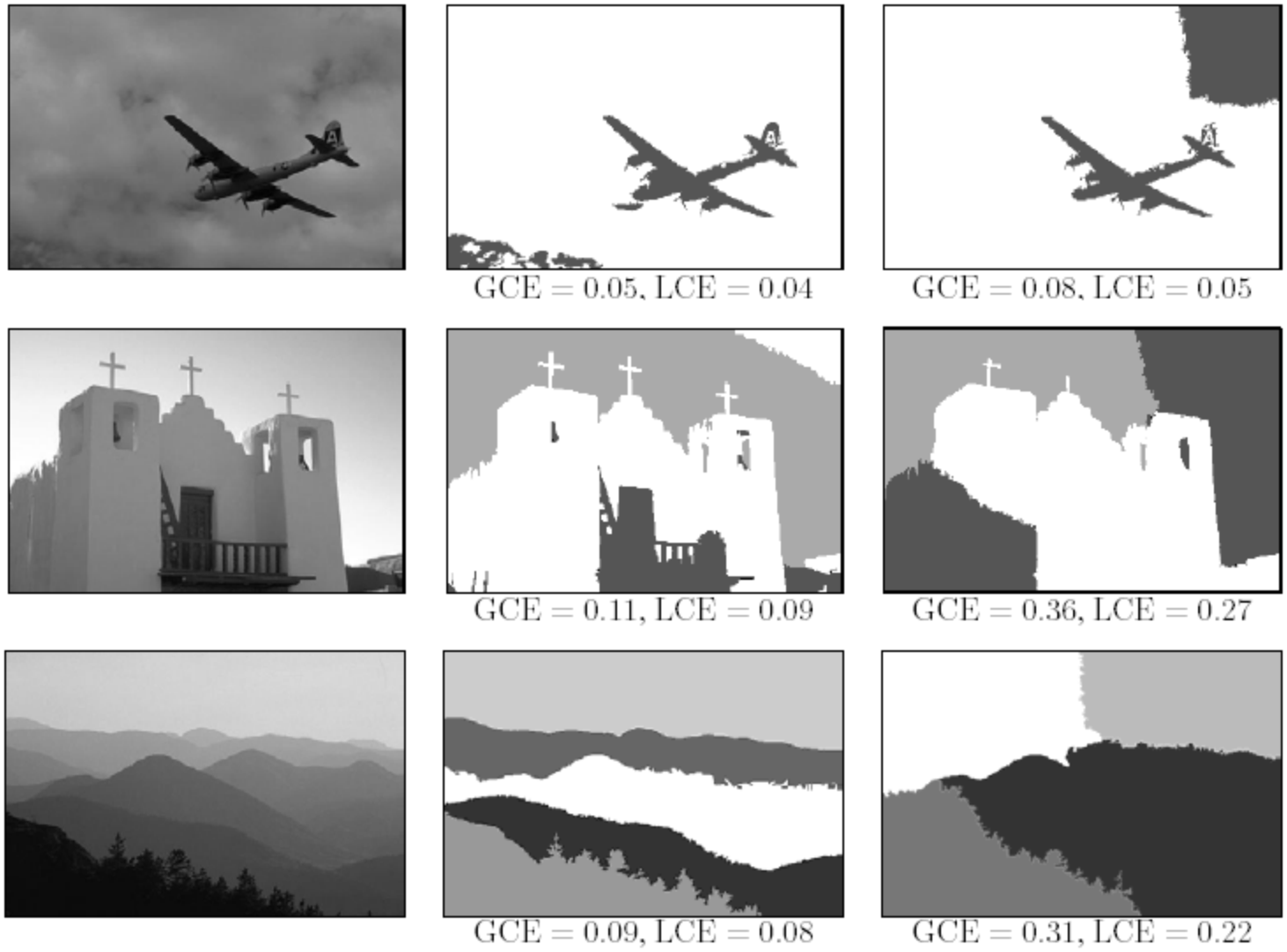


Dominant sets



Ncut

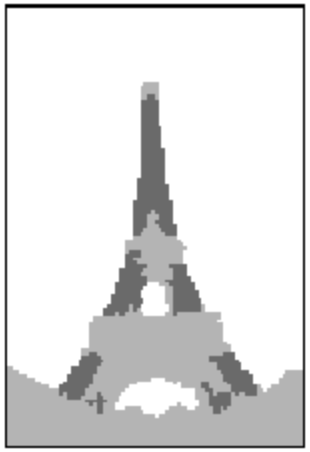
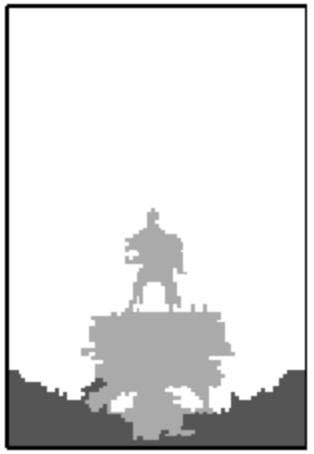
Results on the Berkeley Dataset



Color Segmentation Results

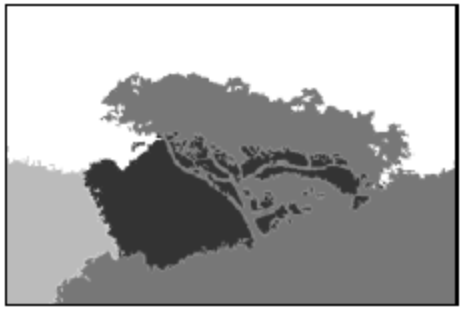
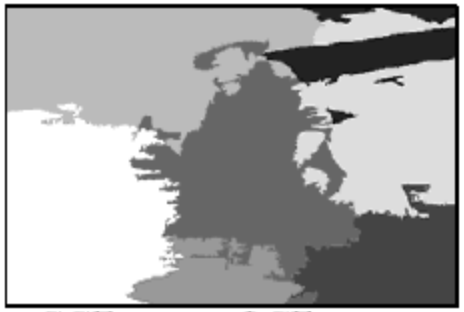
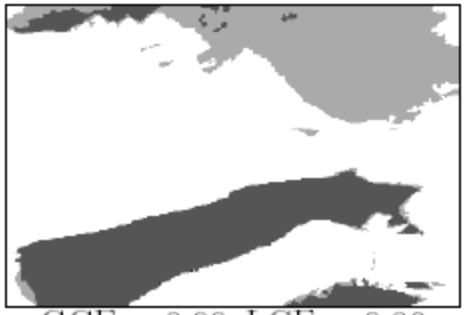
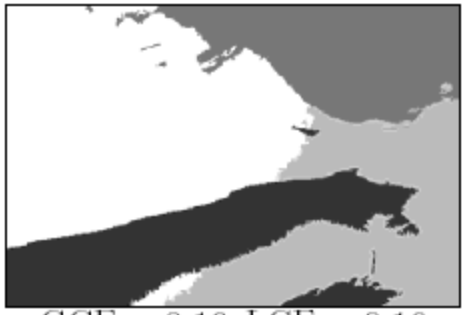




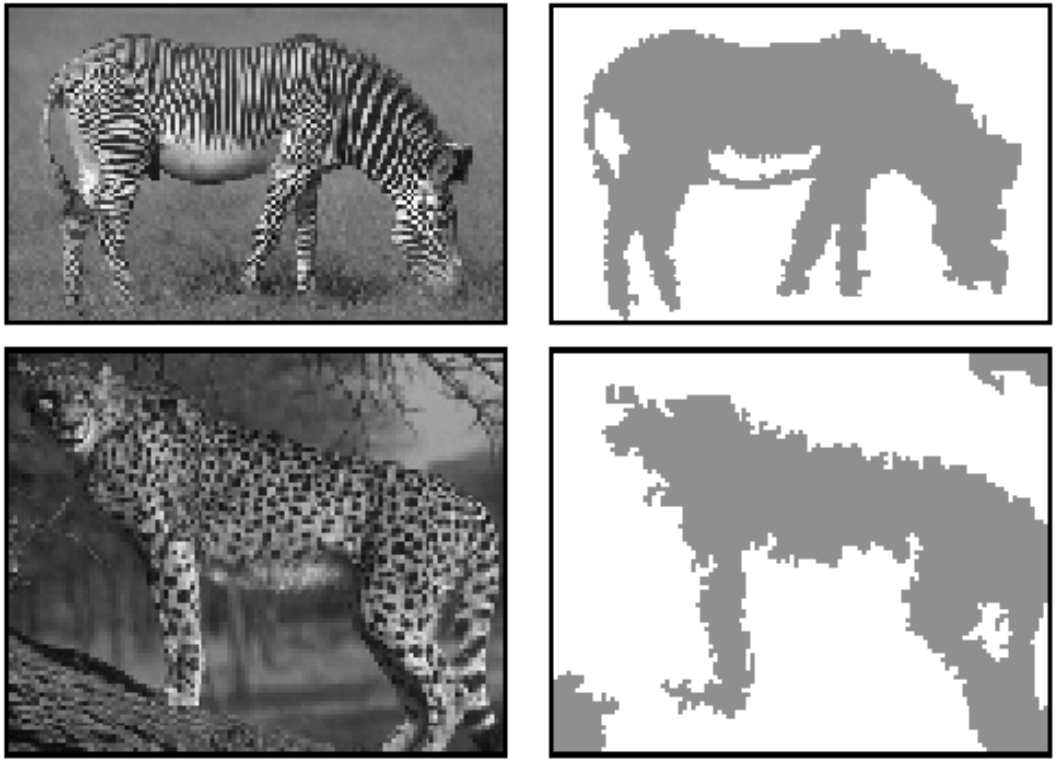



Original image | Dominant sets | Ncut

Results on the Berkeley Dataset

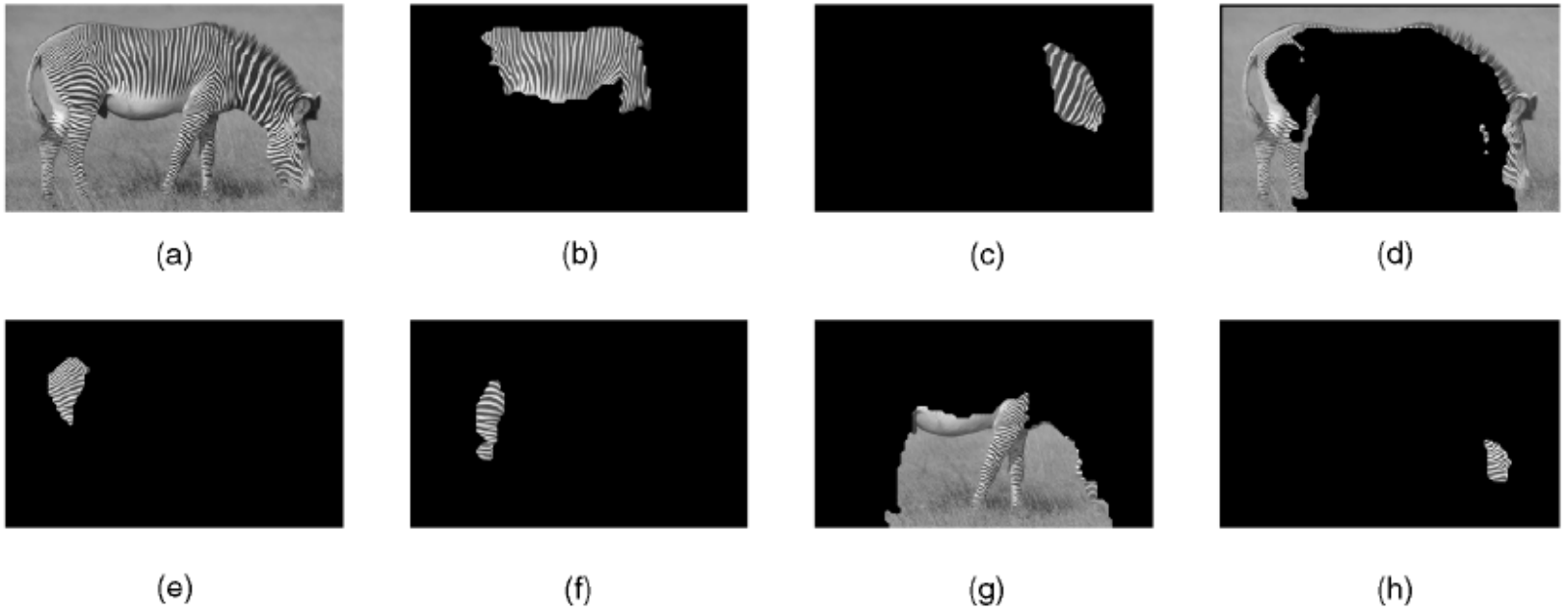
	Dominant sets	Ncut
	 GCE = 0.12, LCE = 0.12	 GCE = 0.19, LCE = 0.13
	 GCE = 0.31, LCE = 0.26	 GCE = 0.35, LCE = 0.29
	 GCE = 0.09, LCE = 0.09	 GCE = 0.16, LCE = 0.16

Texture Segmentation Results



Dominant sets

Texture Segmentation Results



NCut

Modularity

Define modularity to be

Q = (number of edges within groups) − (expected number within groups).

- Modularity is measured relative to a *null model*
  - Defined by P_{ij} = probability of an edge between vertices i and j
  - Examples:
    - $P_{ij} = p$ (Erdös-Rényi random graph)
    - $P_{ij} = k_i k_j / 2m$ ("configuration model")

Modularity

normalization

adjacency matrix

probability a random edge would go between i and j

$$Q = \frac{1}{4m} \sum_{\substack{i,j \\ \text{in same module}}} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$$

m = # edges in graph
k_i = degree(i)

Consider the case of only 2 modules.
Let $s_i = 1$ if node i is in module 1; -1 if node i is in module 2

$$\begin{aligned} Q &= \frac{1}{4m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) \\ &= \frac{1}{4m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j \end{aligned}$$

From: Carl Kingsford (2009)

Modularity

$$Q = \frac{1}{4m}\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2m}\right)s_i s_j$$

$$= \frac{1}{4m}\mathbf{s}^T B\mathbf{s}$$

s is a {-1,1} membership vector

"modularity" matrix

Let u_i ($i = 1,\ldots,n$) be the eigenvectors of matrix B with eigenvalue β_i for vector u_i. (Assume $\beta_1 \geq \beta_2 \geq \beta_3 \geq \beta_4 \geq \ldots \geq \beta_n$)

Write s as:

$$\mathbf{s} = \sum_i a_i u_i$$

where:

$$a_i = u_i^T \mathbf{s}$$

From: Carl Kingsford (2009)

Modularity

$$
\begin{aligned}
Q &= \frac{1}{4m}\mathbf{s}^T B\mathbf{s} \\
\text{drop the } (1/4\text{m}) \longrightarrow &= \left(\sum_i a_i u_i^T\right) B \left(\sum_j a_j u_j\right) \\
&= \left(\sum_i a_i u_i^T B\right) \left(\sum_j a_j u_j\right) \\
&= \sum_i \sum_j a_i a_j u_i^T B u_j
\end{aligned}
$$

Note:

1. $Bu_j = \beta_i u_j$
2. When $i \neq j$, $u_i^{\mathrm{T}} \mathrm{B} u_j = 0$ because $u_i \perp u_j$

$$Q = \sum_i (u_i^T \mathbf{s})^2 \beta_i$$

From: Carl Kingsford (2009)

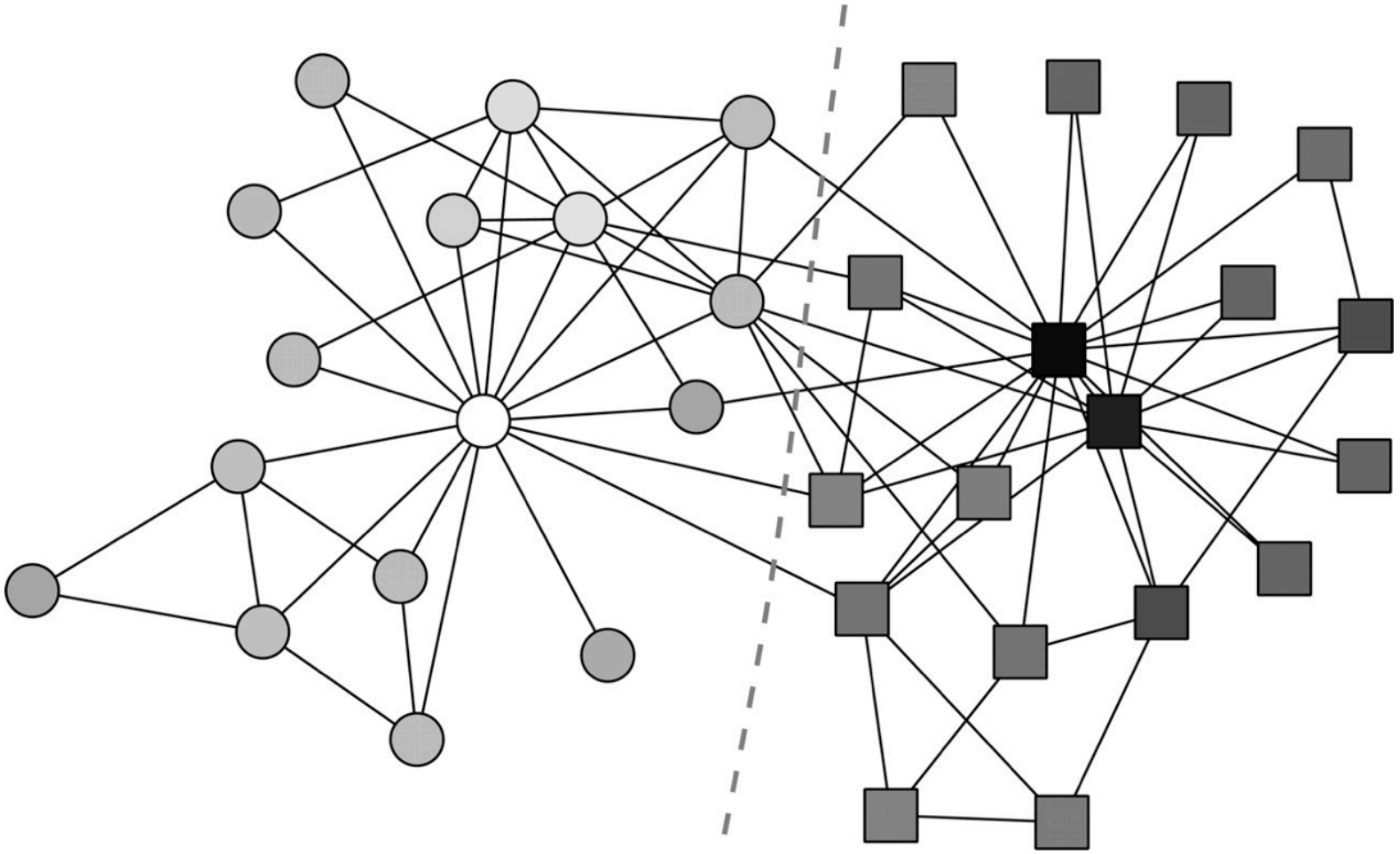
Modularity

Maximize $Q = \sum_i (u_i^T \mathbf{s})^2 \beta_i$

- If we were allowed to choose any **s** we'd pick the one that is parallel to u_1.
- **But:** s_i must be +1 or -1.
  This is a severe restriction.
- **So:** maximize $u_1 \cdot \mathbf{s}$, the projection of s along vector u_1.
- To do this: choose $s_i = 1$ if $u_1 > 0$, and $s_i = -1$ if $u_1 \leq 0$.

From: Carl Kingsford (2009)

Zachary's Karate Club



Newman, 2006