Unsupervised Learning (a.k.a Clustering)

Marcello Pelillo

University of Venice, Italy

Artificial Intelligence

a.y. 2018/19

Ca' Foscari University of Venice

eclt European Centre for Living Technology

The "classical" clustering problem

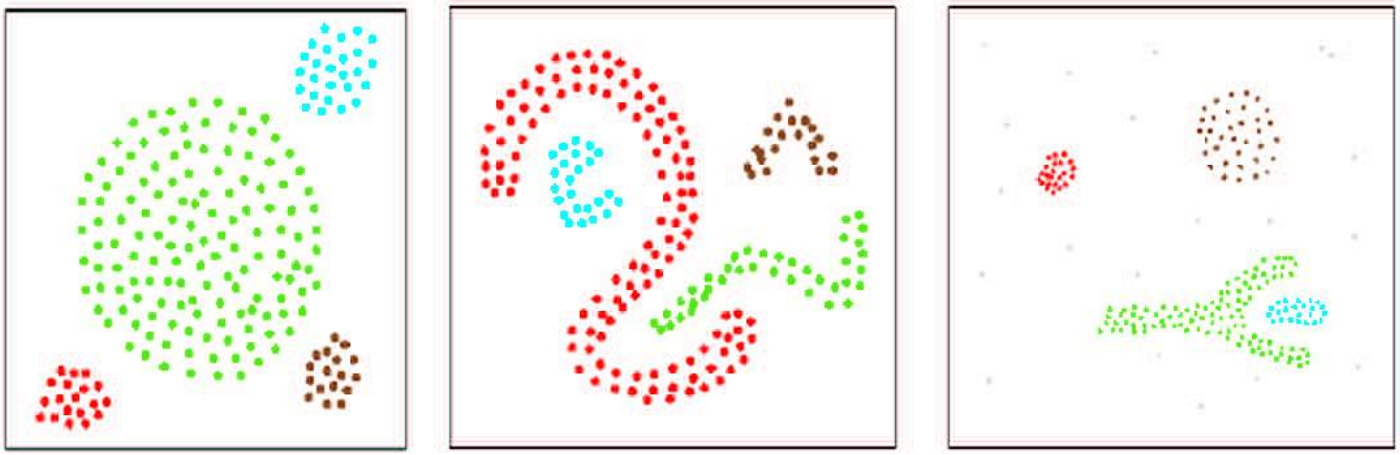
Given:

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

} = an edge-weighted graph G

Goal: *Partition* the vertices of the G into maximally homogeneous groups (i.e., clusters).

Usual assumption: symmetric and pairwise similarities (G is an undirected graph)



Applications

Clustering problems abound in many areas of computer science and engineering.

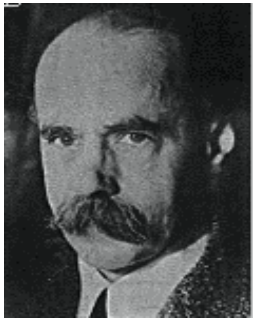
A short list of applications domains:

Image processing and computer vision
Computational biology and bioinformatics
Information retrieval
Document analysis
Medical image analysis
Data mining
Signal processing
…

For a review see, e.g., A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters 31(8):651-666, 2010.

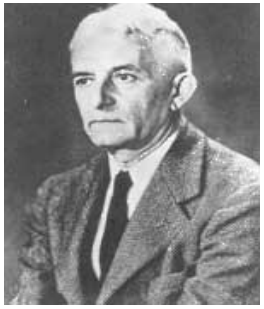
Basic ideas of grouping in humans: The Gestalt school

Wertheimer



Koehler

Koffka



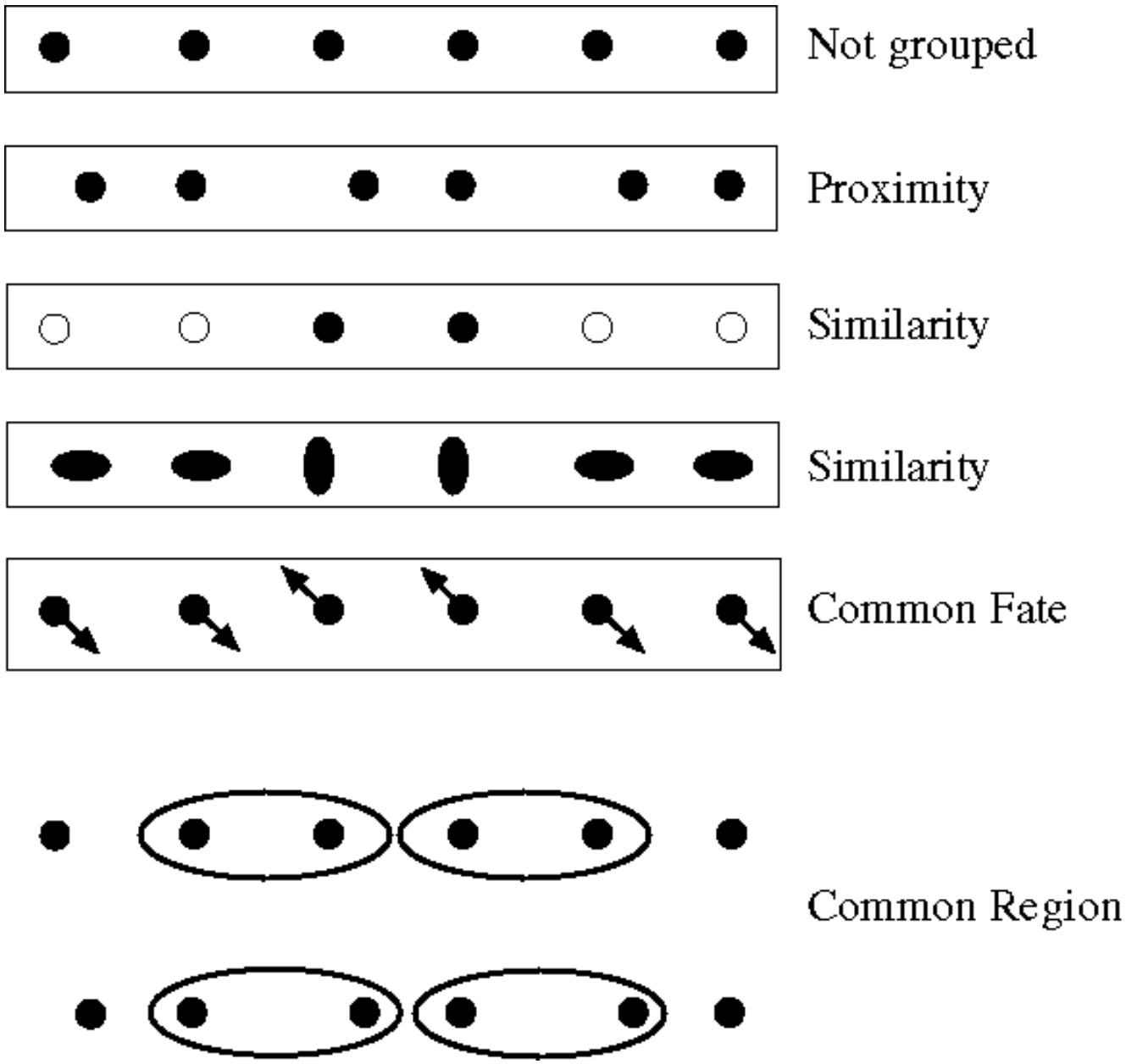
Gestalt properties

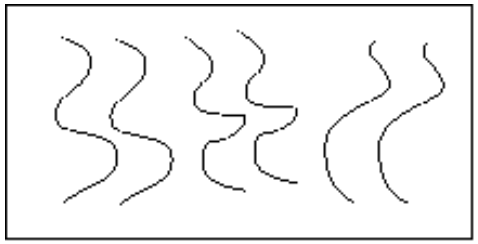
elements in a collection of elements can have properties that result from relationships

- Gestaltqualitat

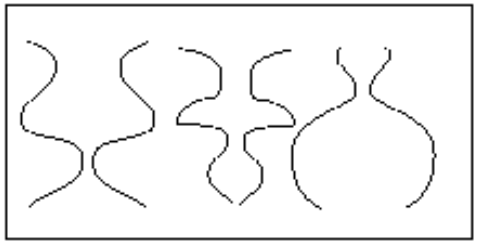
A series of factors affect whether elements should be grouped together

- Gestalt factors

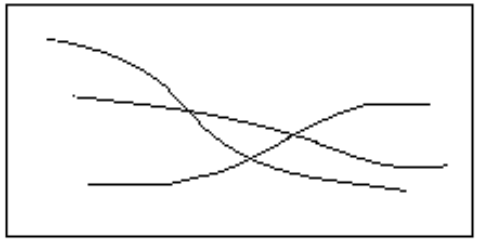

Not grouped
Proximity
Similarity
Similarity
Common Fate
Common Region



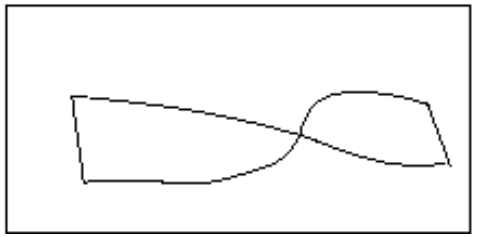
Parallelism



Symmetry



Continuity



Closure

Clustering

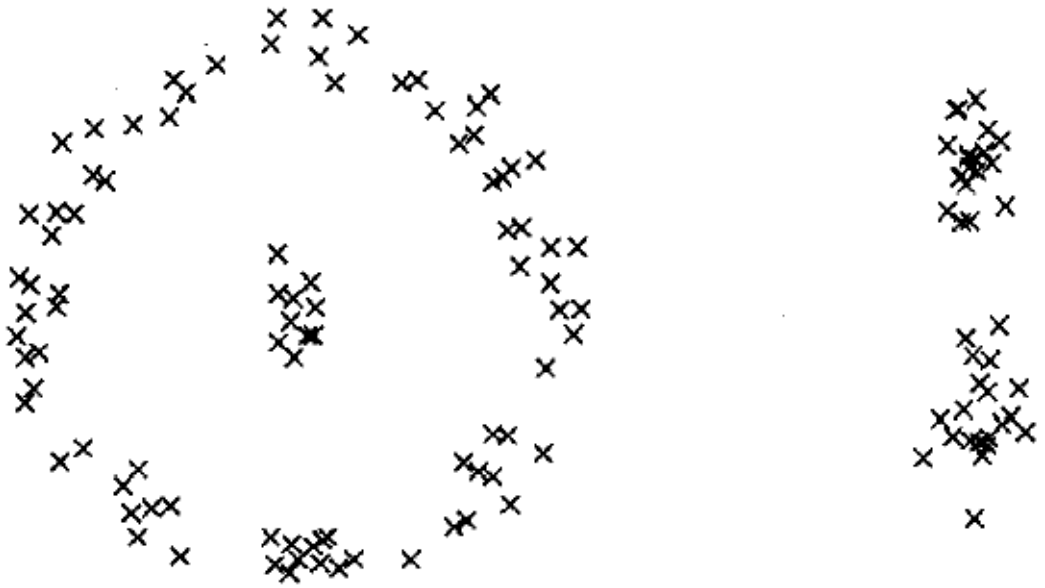
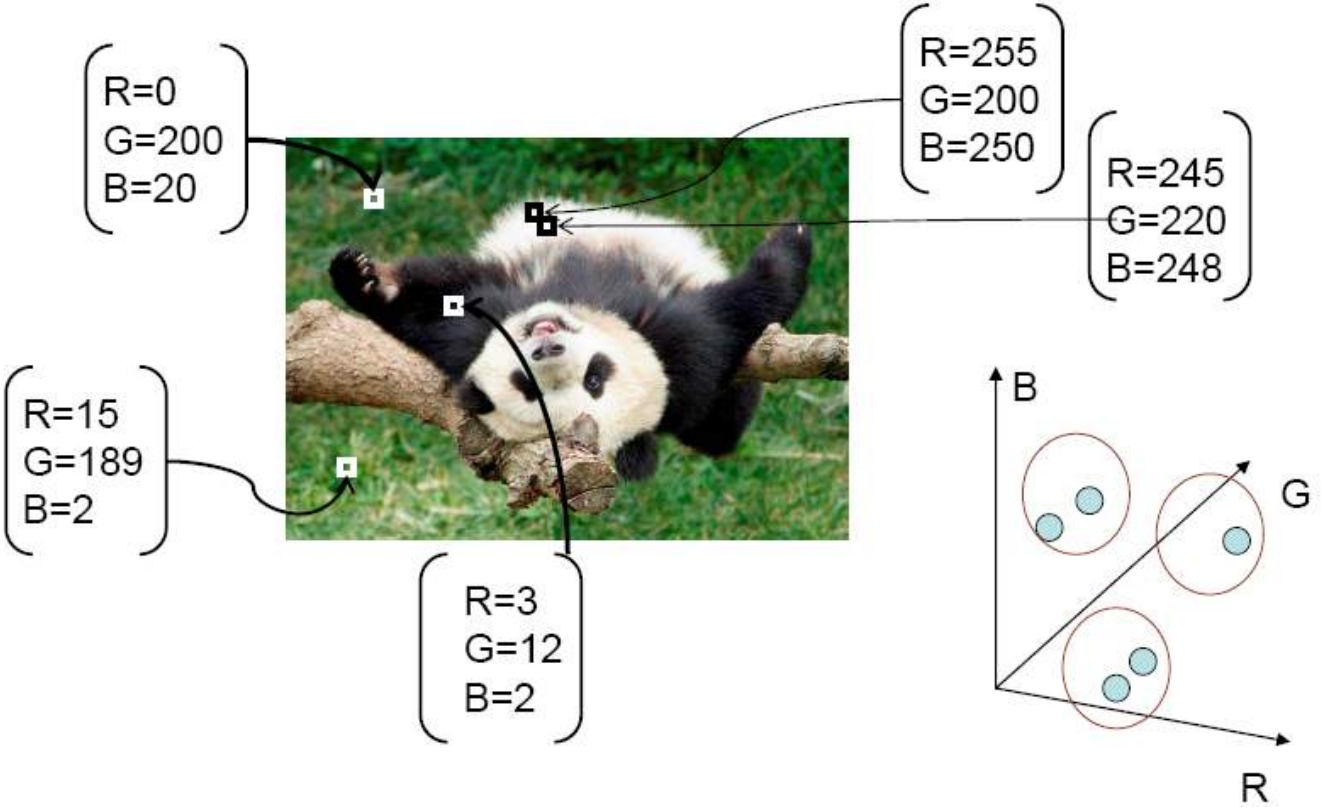


Figure 1: How many groups?

Segmentation as clustering

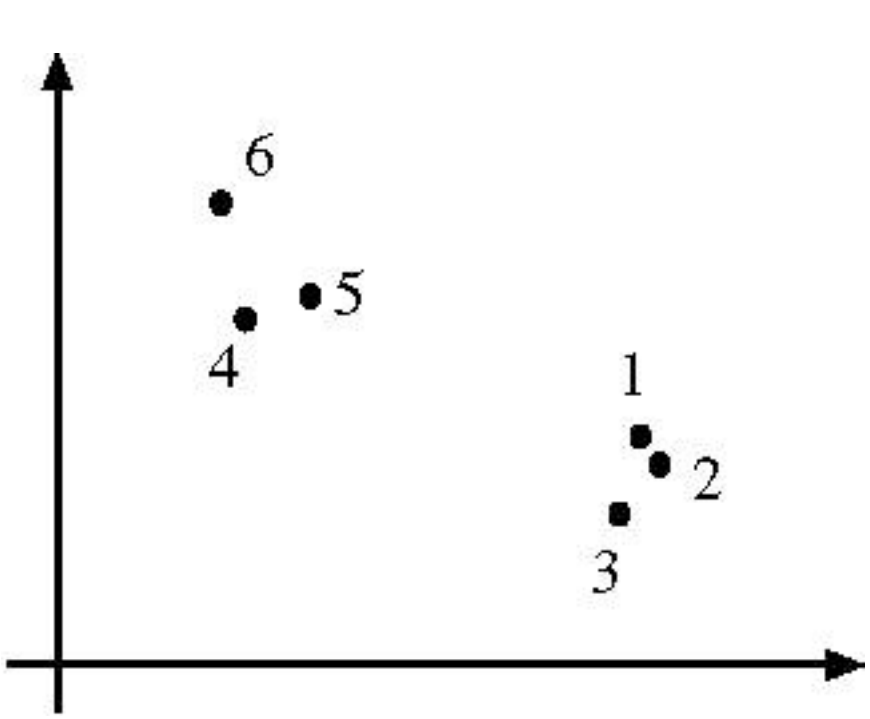
- Cluster similar pixels (features) together

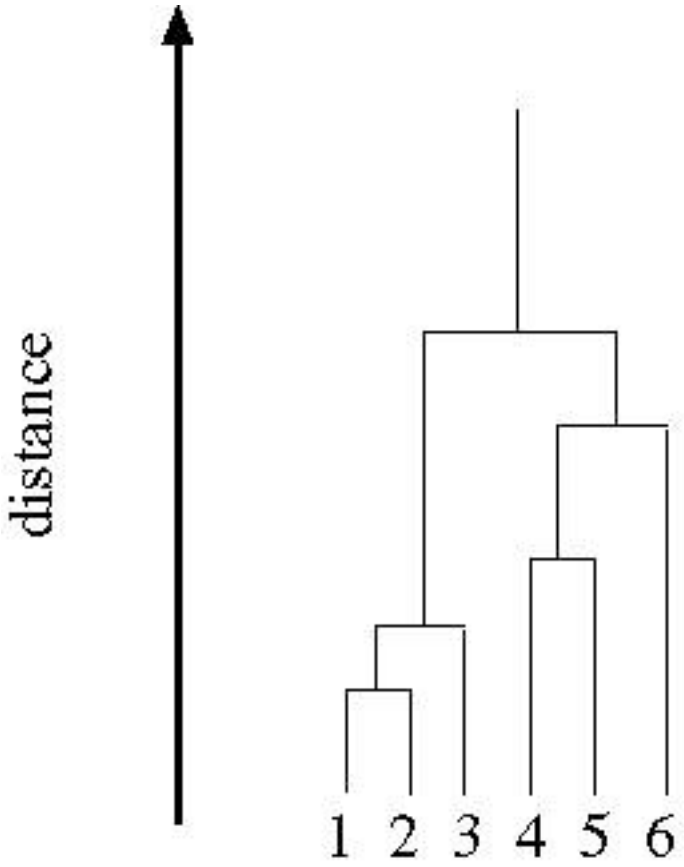


Source: K. Grauman

Segmentation as clustering

- Cluster together (pixels, tokens, etc.) that belong together
- Agglomerative clustering
  - attach closest to cluster it is closest to
  - repeat
- Divisive clustering
  - split cluster along best boundary
  - repeat
- Point-Cluster distance
  - single-link clustering
  - complete-link clustering
  - group-average clustering
- Dendrograms
  - yield a picture of output as clustering process continues


6
5
4
1
2
3


distance
1 2 3 4 5 6

K-Means

An iterative clustering algorithm

– **Initialize:**

Pick *K* random points as cluster centers

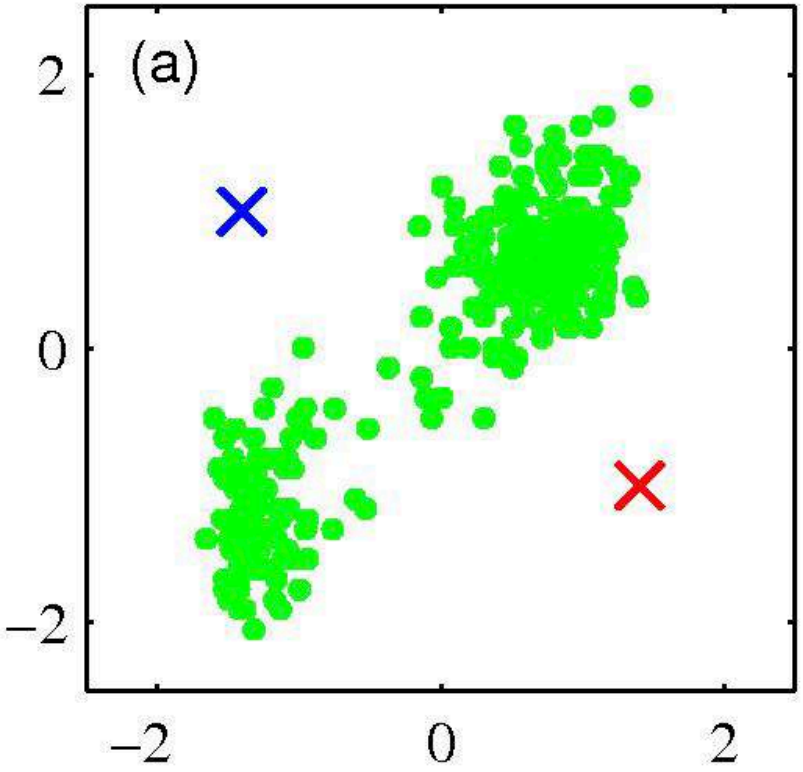
– **Alternate:**

1. Assign data points to closest cluster center
2. Change the cluster center to the average of its assigned points

– **Stop** when no points' assignments change

Note: Ensure that every cluster has at least one data point. Possible techniques for doing this include supplying empty clusters with a point chosen at random from points far from their cluster centers.

K-means clustering: Example

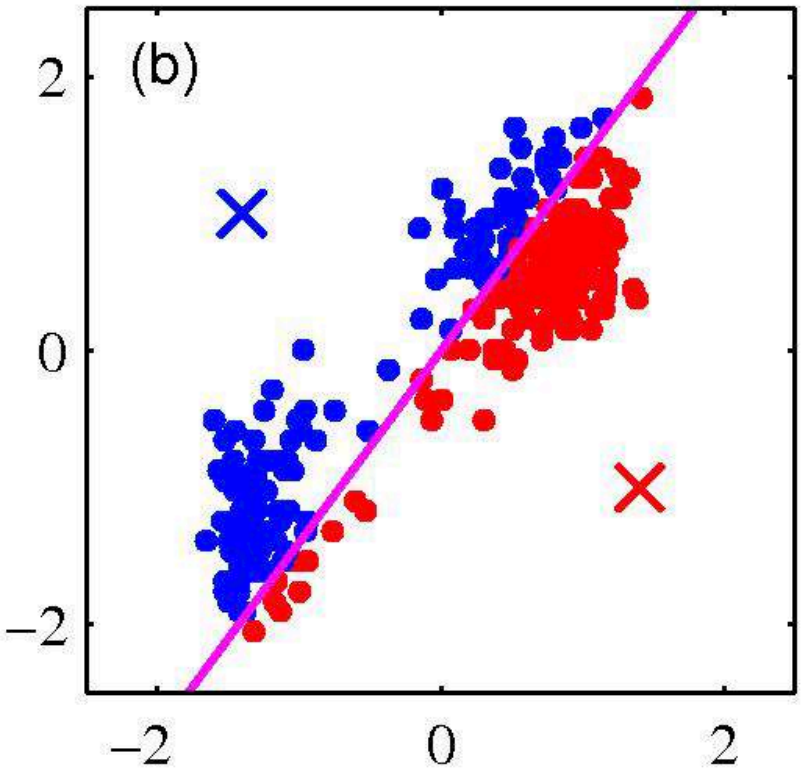


Initialization:
Pick K random points as cluster centers

Shown here for K=2

Adapted from D. Sontag

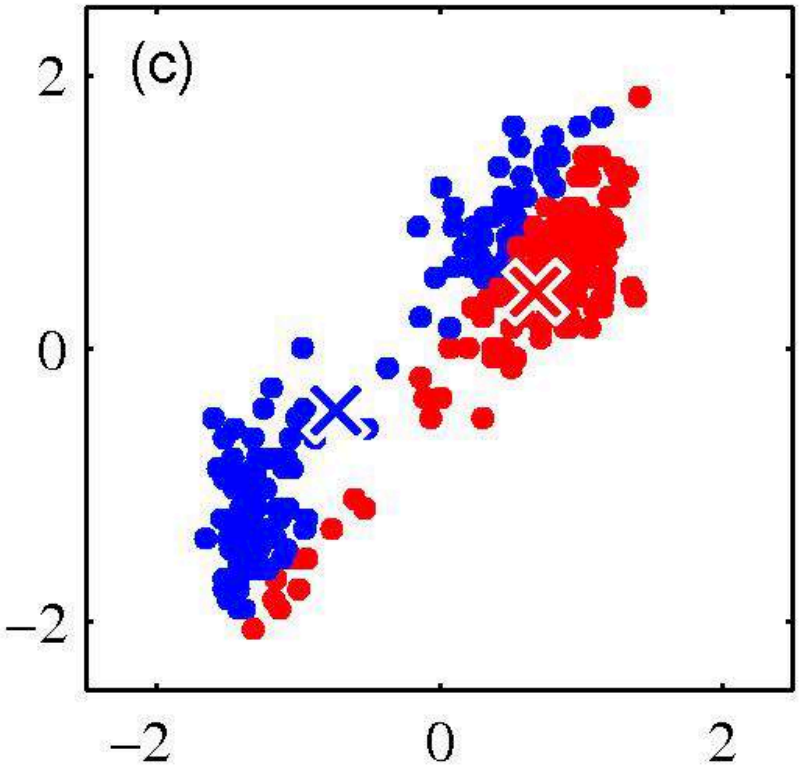
K-means clustering: Example



Iterative Step 1:
Assign data points to closest cluster center

Adapted from D. Sontag

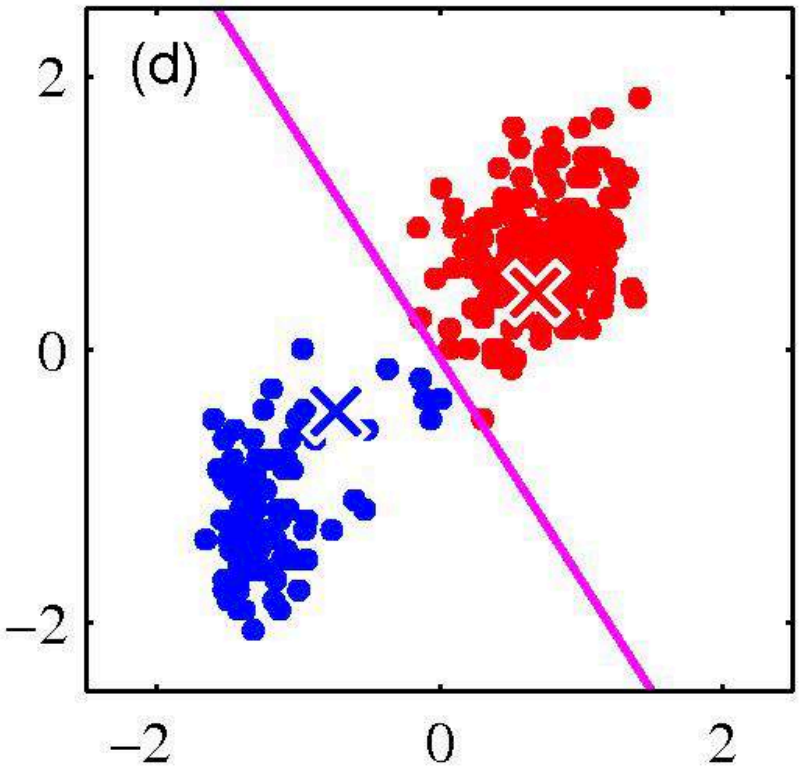
K-means clustering: Example



Iterative Step 2:
Change the cluster center to the average of the assigned points

Adapted from D. Sontag

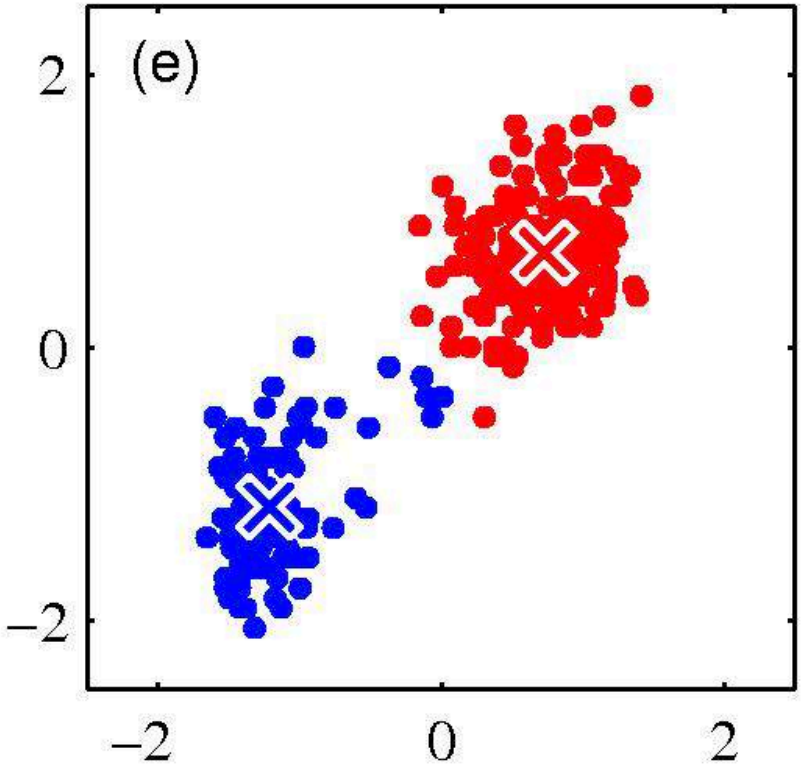
K-means clustering: Example



Repeat until convergence

Adapted from D. Sontag

K-means clustering: Example



Final output

Adapted from D. Sontag

Image

Clusters on intensity

Clusters on color

K-means clustering using intensity alone and color alone

Properties of K-means

Guaranteed to converge in a finite number of steps.

Minimizes an objective function (compactness of clusters):

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

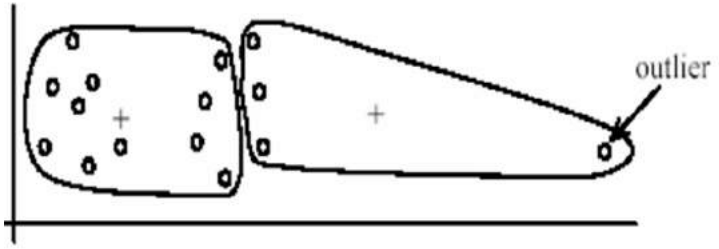
where μ_i is the center of cluster i.

Running time per iteration:

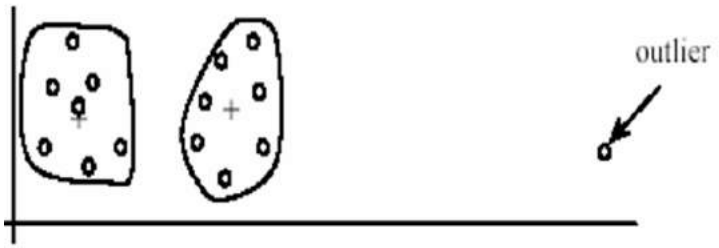
- Assign data points to closest cluster center: $O(Kn)$ time
- Change the cluster center to the average of its points: $O(n)$ time

Properties of K-means

- Pros
  - Very simple method
  - Efficient
- Cons
  - Converges to a *local* minimum of the error function
  - Need to pick K
  - Sensitive to initialization
  - Sensitive to outliers
  - Only finds "spherical" clusters

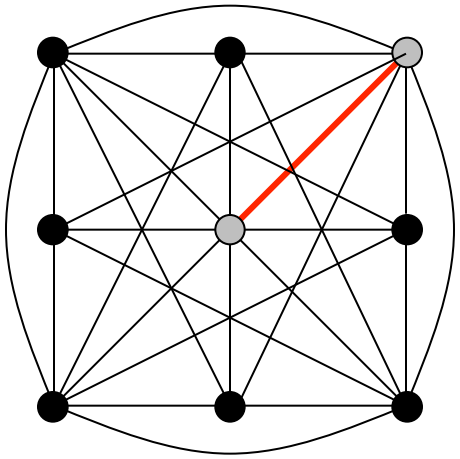


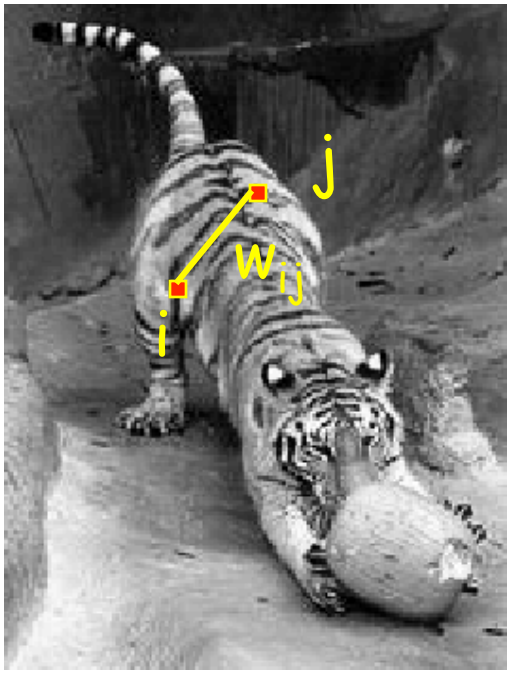
(A): Undesirable clusters



(B): Ideal clusters

Images as graphs





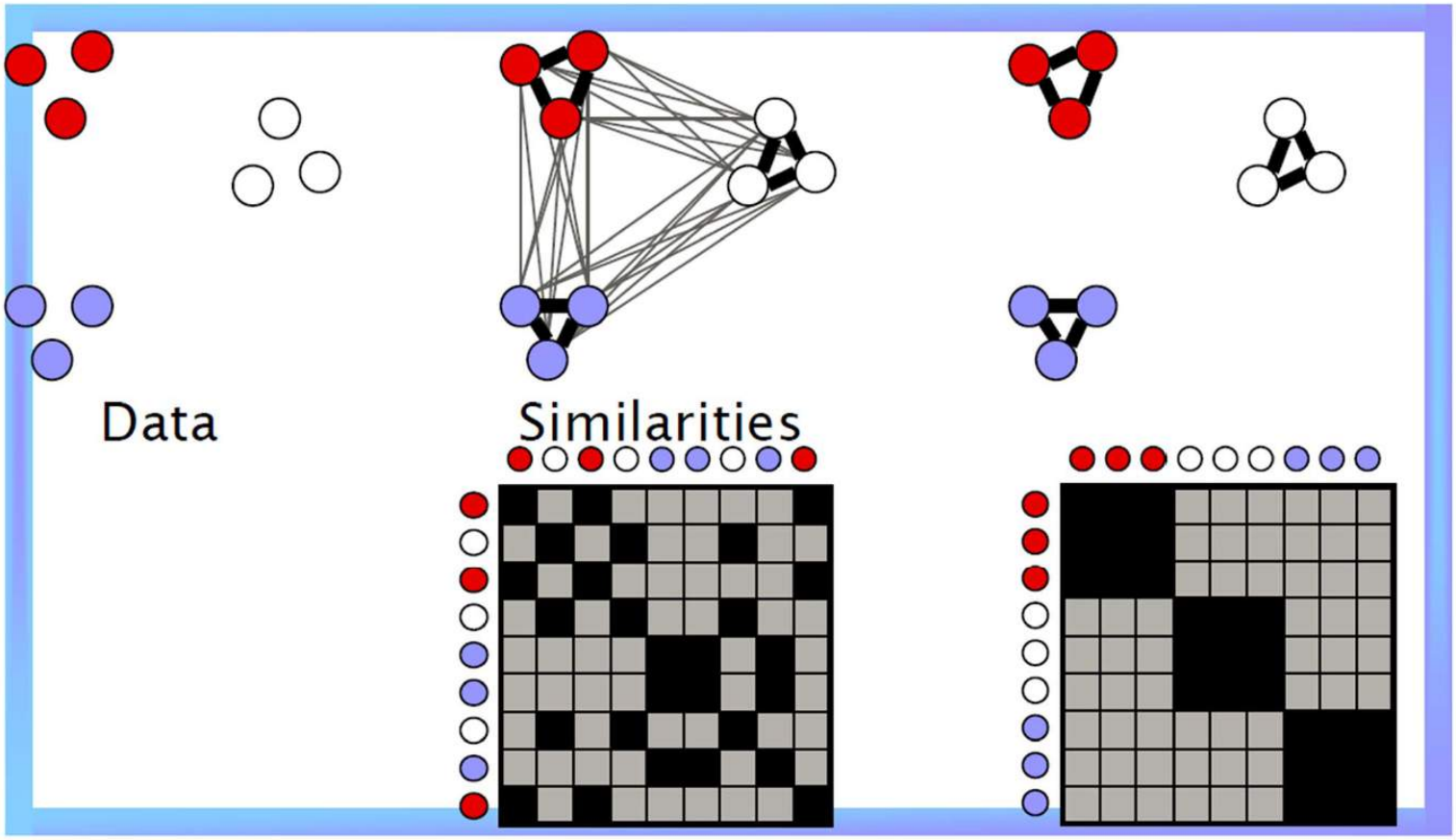
- Node for every pixel
- Edge between every pair of pixels (or every pair of "sufficiently close" pixels)
- Each edge is weighted by the *affinity* or similarity of the two nodes

Source: S. Seitz

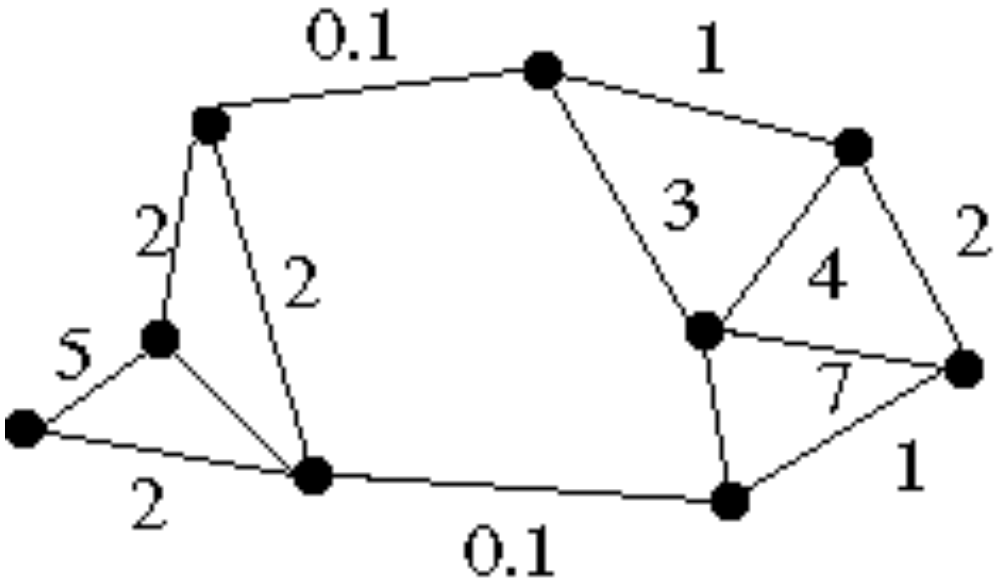
Graph-theoretic (pairwise) clustering

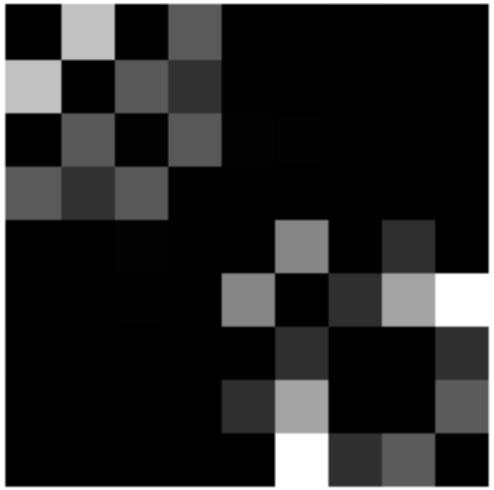
- Represent tokens using a weighted graph.
  - affinity matrix
- Cut up this graph to get subgraphs with strong interior links

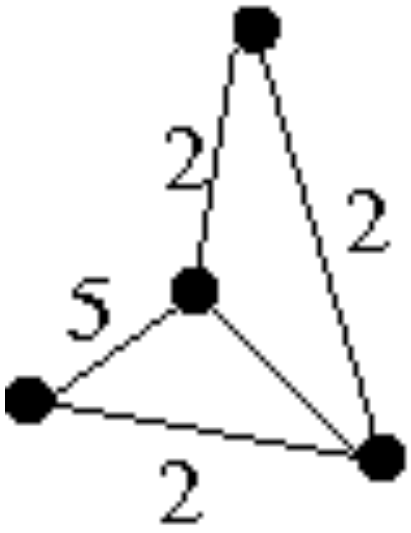
Graphs and matrices

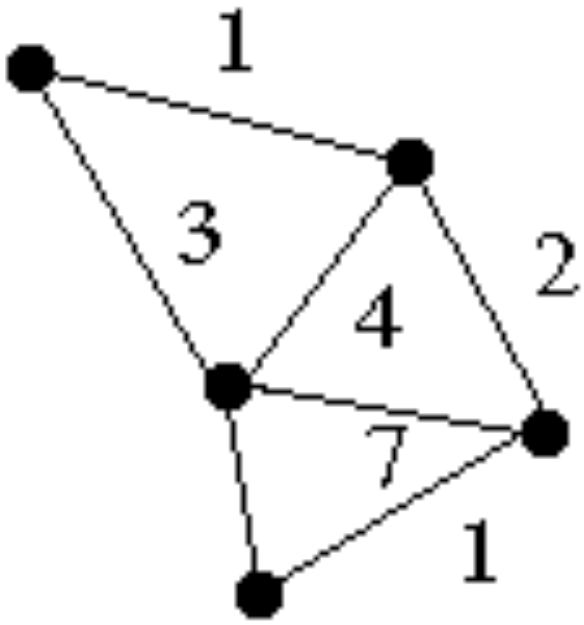


Source: D. Sontag


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




2
2
5
2


1
3
2
4
7
1

Measuring affinity

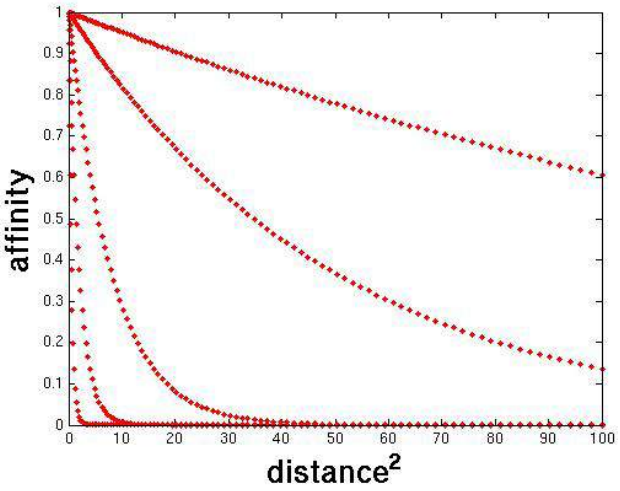
- Suppose we represent each pixel by a feature vector **x**, and define a distance function appropriate for this feature representation

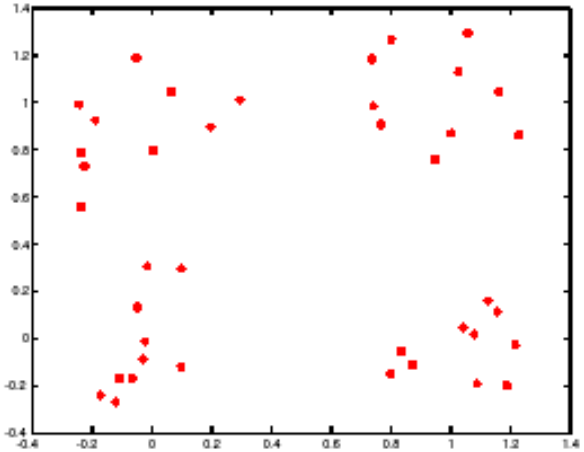
- Then we can convert the distance between two feature vectors into an affinity with the help of a Gaussian kernel:

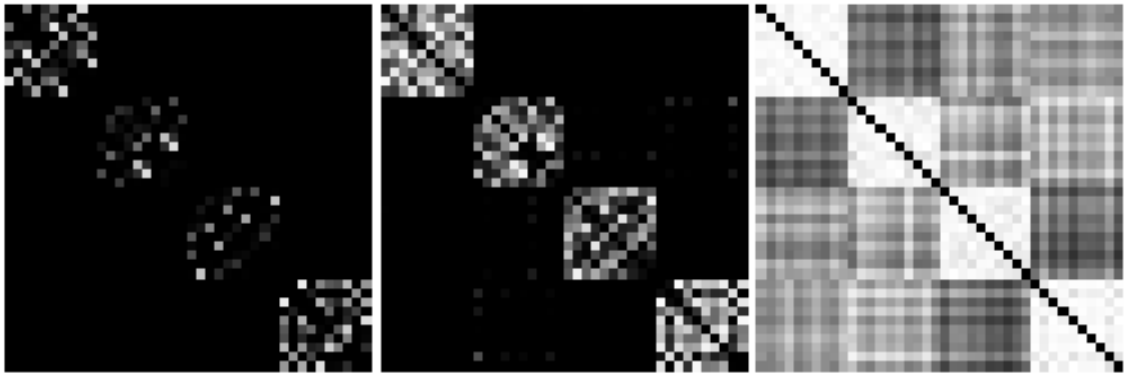
$$\exp\left(-\frac{1}{2\sigma^2}\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j)^2\right)$$

Scale affects affinity

- Small σ: group only nearby points
- Large σ: group far-away points







Eigenvector-based clustering

Let us represent a cluster using a vector $\mathbf{x}$ whose k-th entry captures the participation of node k in that cluster. If a node does not participate in a cluster, the corresponding entry is zero.

We also impose the restriction that $\mathbf{x}^T\mathbf{x} = 1$

We want to maximize:

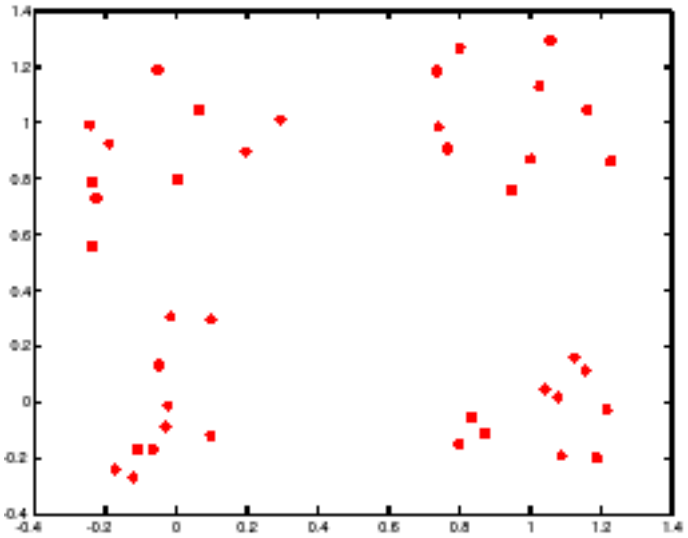
$$\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_i x_j = \mathbf{x}^T\mathbf{A}\mathbf{x}$$

which is a measure for the cluster's cohesiveness.

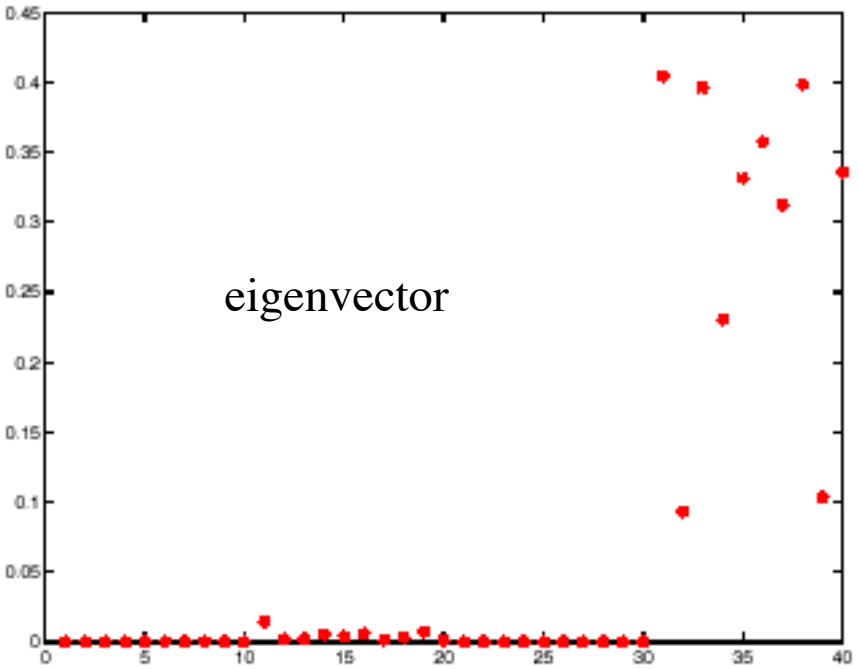
This is an **eigenvalue problem**!

Choose the eigenvector of A with largest eigenvalue

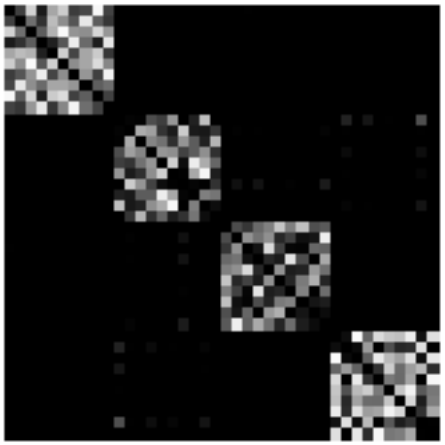
Example eigenvector



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

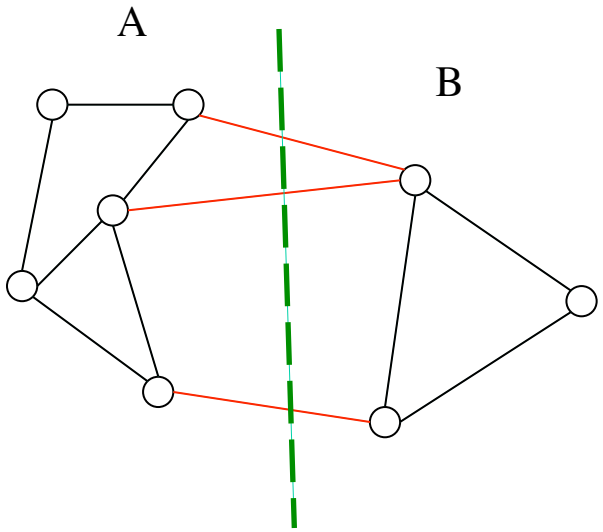
1. Construct (or take as input) the affinity matrix A
2. Compute the eigenvalues and eigenvectors of A
3. Repeat
4. Take the eigenvector corresponding to the largest unprocessed eigenvalue
5. Zero all components corresponding to elements that have already been clustered
6. Threshold the remaining components 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

Let $G=(V, E, w)$ a weighted graph.

Given a "cut" (A, B), with $B = V \setminus A$, define:

$$cut(A,B) = \sum_{i \in A} \sum_{j \in B} w(i,j)$$



Minimum Cut Problem

Among all possible cuts (A, B),
find the one which minimizes $cut(A, B)$

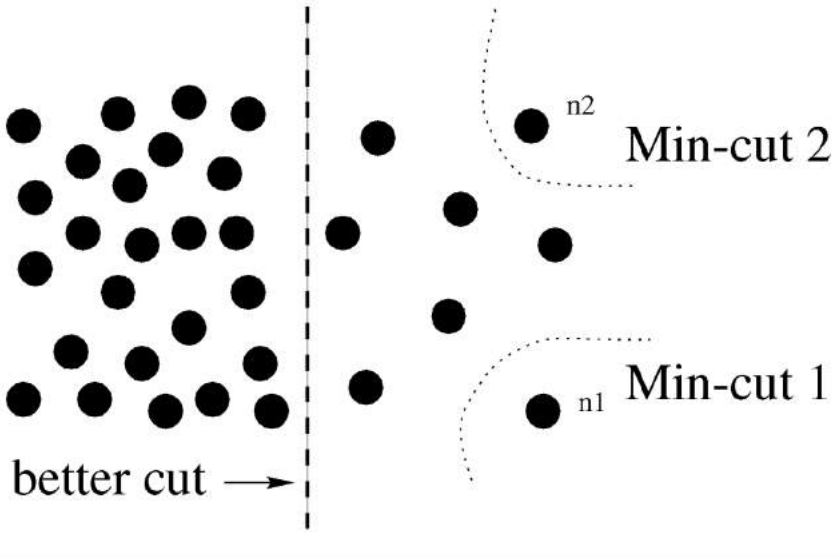
MinCut clustering

Good news

Solvable in polynomial time

Bad news

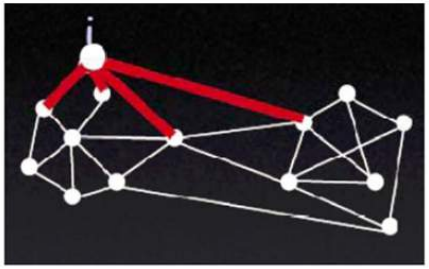
Favors highly unbalanced clusters (often with isolated vertices)

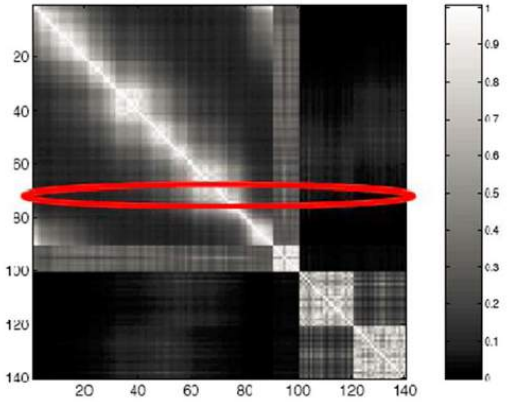


Graph terminology

Degree of nodes

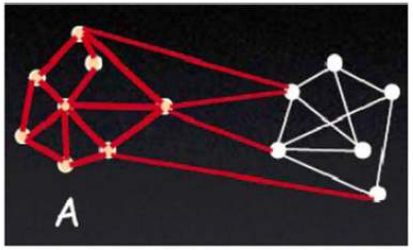
$$d_i = \sum_j w_{i,j}$$

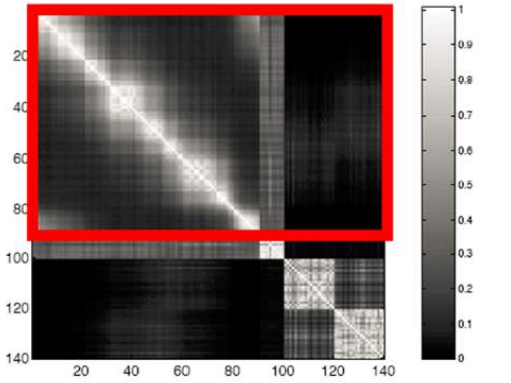




Volume of a set

$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$

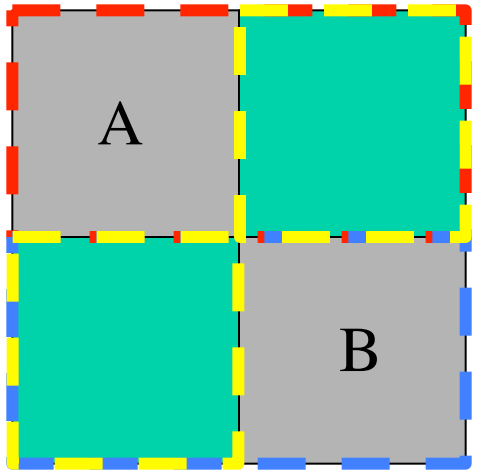




Adapted from D. Sontag

Normalized Cut

$$Ncut(A,B) = cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$$



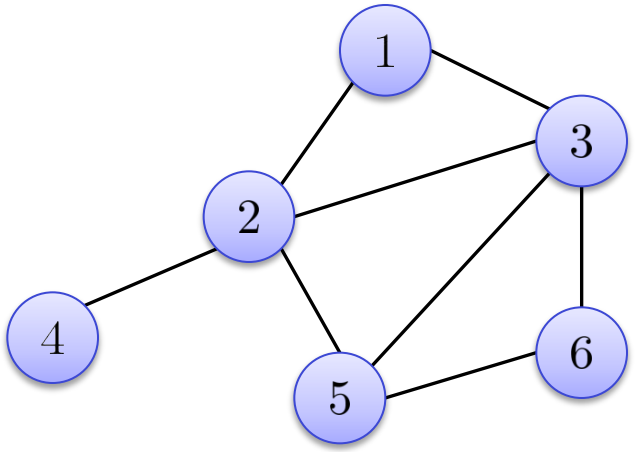
Adapted from D. Sontag

Graph Laplacian (unnormalized)

Defined as

$$L = D - W$$

Example:



$$L = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 \\ -1 & -1 & 4 & 0 & -1 & -1 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 3 & -1 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{pmatrix}$$

Assume the weights of edges are 1

Key fact

For all vectors f in $\mathbf{R}^n$, we have:

$$f^\top L f = \frac{1}{2} \sum_{ij=1}^{n} w_{ij}(f_i - f_j)^2$$

Indeed:

$$\begin{aligned} f^\top L f &= f^\top D f - f^\top W f \\ &= \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} \\ &= \frac{1}{2}\left(\sum_i (\sum_j w_{ij}) f_i^2 - 2 \sum_{ij} f_i f_j w_{ij} + \sum_j (\sum_i w_{ij}) f_j^2 \right) \\ &= \frac{1}{2} \sum_{ij} w_{ij}(f_i - f_j)^2 \end{aligned}$$

Properties

- L is symmetric (by assumption) and positive semi-definite:

  $$f' L f \geq 0$$

  for all vectors f (by "key fact")
- Smallest eigenvalue of L is 0; corresponding eigenvector is **1**
- Thus eigenvalues are: $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

First relation between spectrum and clusters:

- Multiplicity of eigenvalue $\lambda_1 = 0$ is the number of connected components of the graph
- eigenspace is spanned by the characteristic functions of these components (so all eigenvectors are piecewise constant)

Normalized graph Laplacians

- Row sum (random walk) normalization:

$$L_{\text{rw}} = D^{-1} L$$
$$= I - D^{-1} W$$

- Symmetric normalization:

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2}$$
$$= I - D^{-1} W D^{-1/2}$$

Spectral properties of both matrices similar to the ones of L.

Solving Ncut

Any cut (A, B) can be represented by a binary indicator vector x:

$$x_i = \begin{cases} +1 & \text{if } i \in A \\ -1 & \text{if } i \in B \end{cases}$$

It can be shown that:

Rayleigh quotient

$$\min_x Ncut(x) = \min_y \frac{y'(D-W)y}{y'Dy}$$

subject to the constraint that $y'D\mathbf{1} = \sum_i y_i d_i = 0$ (with $y_i \in \{1, -b\}$).

This is **NP-hard!**

Ncut as an eigensystem

If we *relax* the constraint that y be a discrete-valued vector and allow it to take on real values, the problem

$$\min_y \frac{y'(D-W)y}{y'Dy}$$

is equivalent to:

$$\min_y y'(D-W)y \quad \text{subject to} \quad y'Dy = 1$$

This amounts to solving a *generalized* eigenvalue problem:

Laplacian $\leftarrow$ $(D-W)y = \lambda Dy$

Note: Equivalent to a *standard* eigenvalue problem using the normalized Laplacian: $L_{\text{rw}} = D^{-1}L = I - D^{-1}W$.

2-way Ncut

1. Compute the affinity matrix W, compute the degree matrix D

2. Solve the generalized eigenvalue problem $(D - W)y = \lambda Dy$

3. Use the eigenvector associated to the second smallest eigenvalue to bipartition the graph into two parts.

Why the *second* smallest eigenvalue?

Remember, the smallest eigenvalue of Laplacians is always 0
(corresponds to the trivial partition $A = V, B = \{\}$)

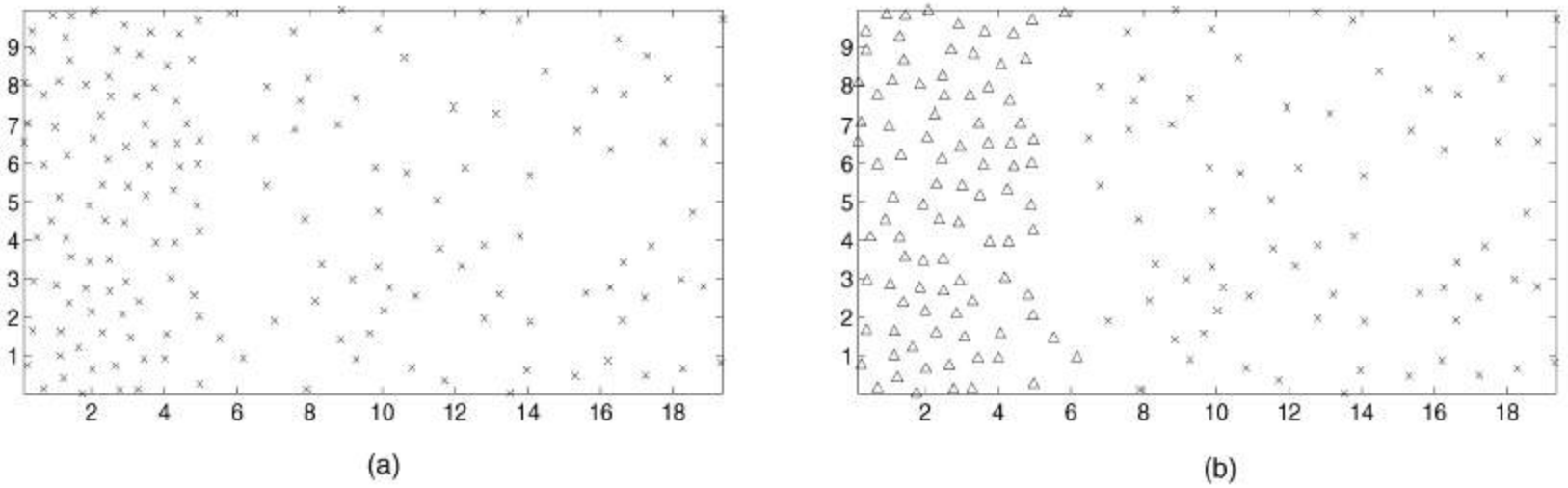
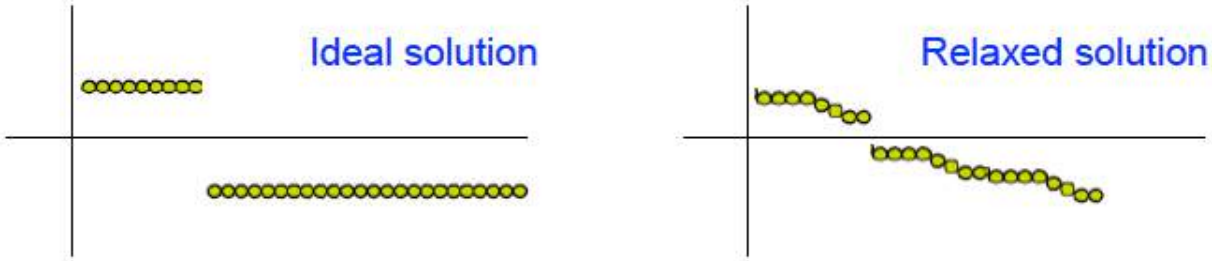
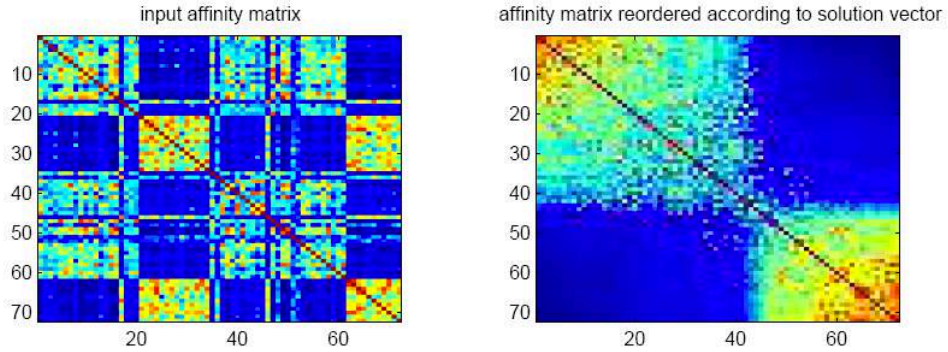
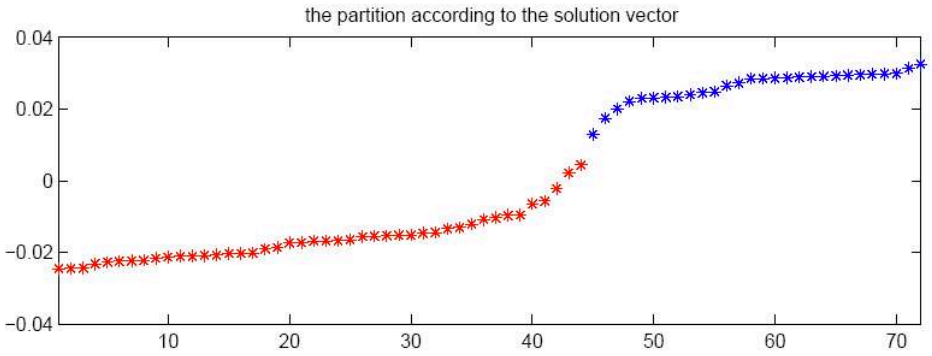


Fig. 5. (a) Point set generated by two Poisson processes, with densities of 2.5 and 1.0 on the left and right clusters respectively, (b) $\triangle$ and $\times$ indicate the partition of point set in (a). Parameter settings: $\sigma_X = 5$, $r = 3$.

The effect of relaxation







How to choose the splitting point?

- Pick a constant value (0 or 0.5)
- Pick the median value as splitting point
- Look for the splitting point that has minimum Ncut value:
  1. Choose *n* possible splitting points
  2. Compute Ncut value
  3. Pick minimum

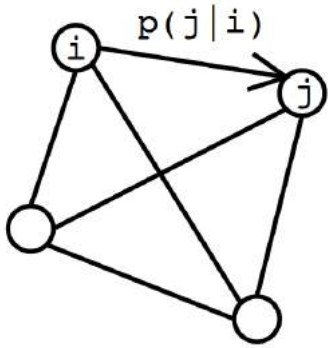
Random walk intepretation

Construct a Markov chain where each data point is a state, connected to all other states with some probability.

With our affinity W and degree D, the stochastic matrix is:

$$P = D^{-1} W$$

which is the row-normalized version of W, so each entry $P(i, j)$ is a probability of "walking" to state j from state i.



Adapted from Y. Weiss

Random walk intepretation

Problem: Finding a cut (A, B) in a graph G such that a random walk does not have many opportunities to jump between the two clusters.

This is equivalent to the *Ncut* problem due to the following relation:

$$Ncut(A, B) = P(A \mid B) + P(B \mid A)$$

(Meila and Shi, 2001)

Ncut: More than 2 clusters

Approach #1: Recursive two-way cuts

1. Given a weighted graph $G = (V, E, w)$, summarize the information into matrices W and D

2. Solve $(D - W)y = \lambda Dy$ for eigenvectors with the smallest eigenvalues

3. Use the eigenvector with the second smallest eigenvalue to bipartition the graph by finding the splitting point such that Ncut is minimized

4. Decide if the current partition should be subdivided by checking the stability of the cut, and make sure Ncut is below the prespecified value

5. Recursively repartition the segmented parts if necessary

Note. The approach is computationally wasteful; only the second eigenvector is used, whereas the next few small eigenvectors also contain useful partitioning information.

Ncut: More than 2 clusters

Approach #2: Using first *k* eigenvectors

1. Construct a similarity graph and compute the unnormalized graph Laplacian L.
2. Compute the k smallest **generalized** eigenvectors $u_1, u_2, \cdots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$.
3. Let $U = [\, u_1\ u_2\ \cdots\ u_k\,] \in \mathbb{R}^{n \times k}$.
4. Let $y_i \in \mathbb{R}^k$ be the vector corresponding to the ith row of U.

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1k} \\ u_{21} & u_{22} & \cdots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nk} \end{bmatrix} = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{bmatrix}$$

5. Thinking of y_i's as points in $\mathbb{R}^k$, cluster them with k-means algorithms.

Fig. 2. A gray level image of a baseball game.

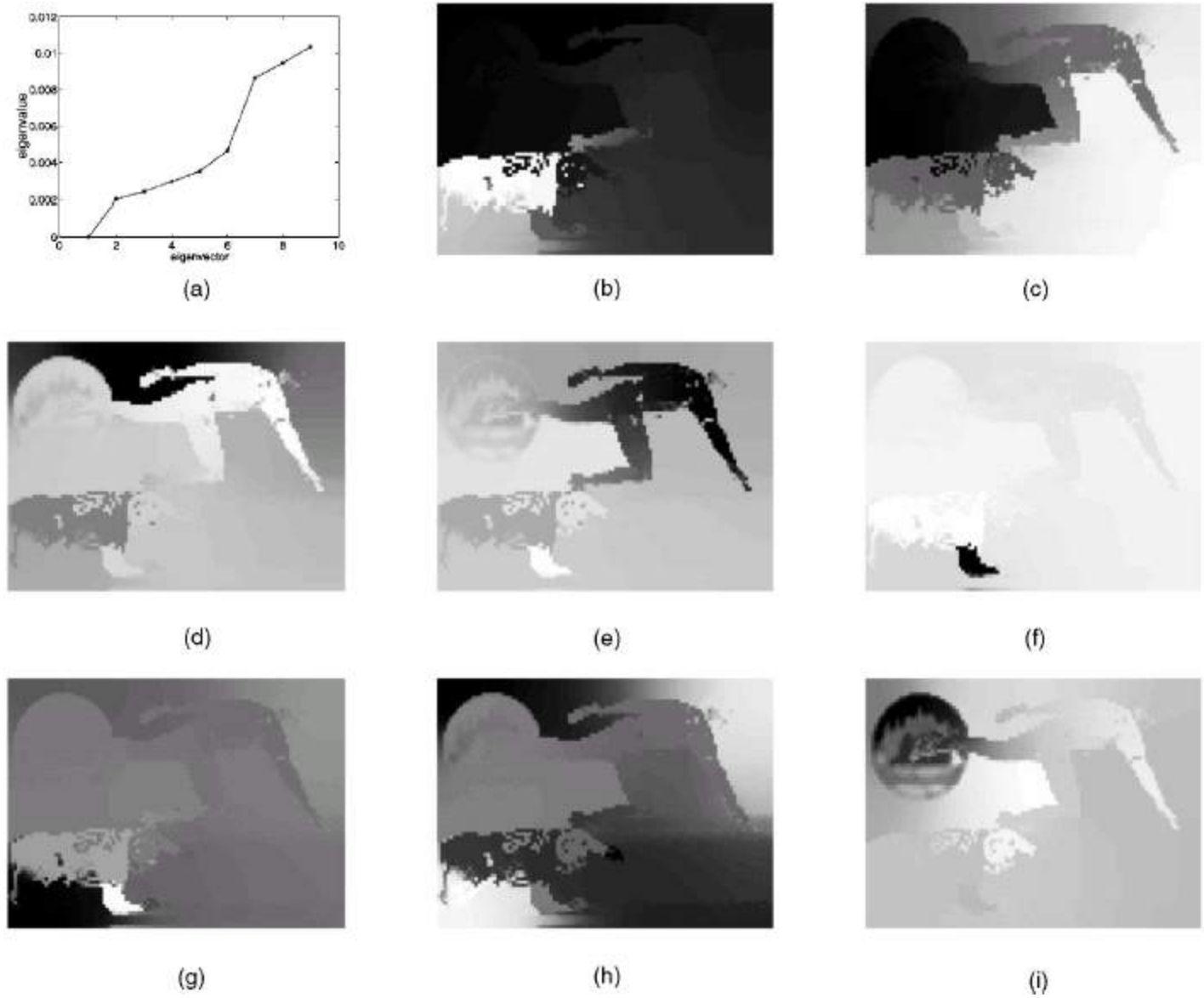


Fig. 3. Subplot (a) plots the smallest eigenvectors of the generalized eigenvalue system (11). Subplots (b)-(i) show the eigenvectors corresponding the second smallest to the ninth smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

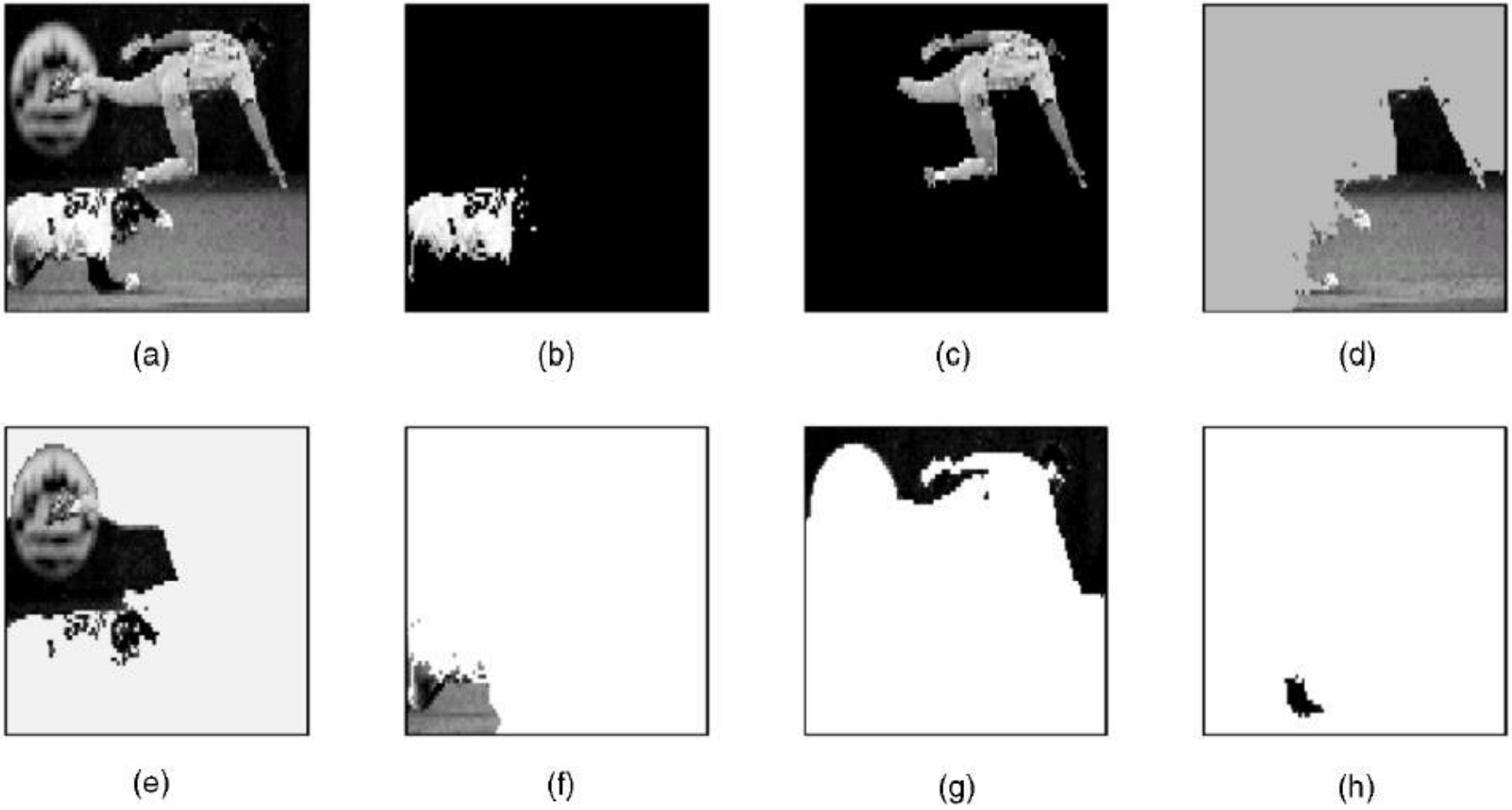


Fig. 4. (a) shows the original image of size 80×100. Image intensity is normalized to lie within 0 and 1. Subplots (b)-(h) show the components of the partition with $Ncut$ value less than 0.04. Parameter setting: $\sigma_I = 0.1$, $\sigma_X = 4.0$, $r = 5$.

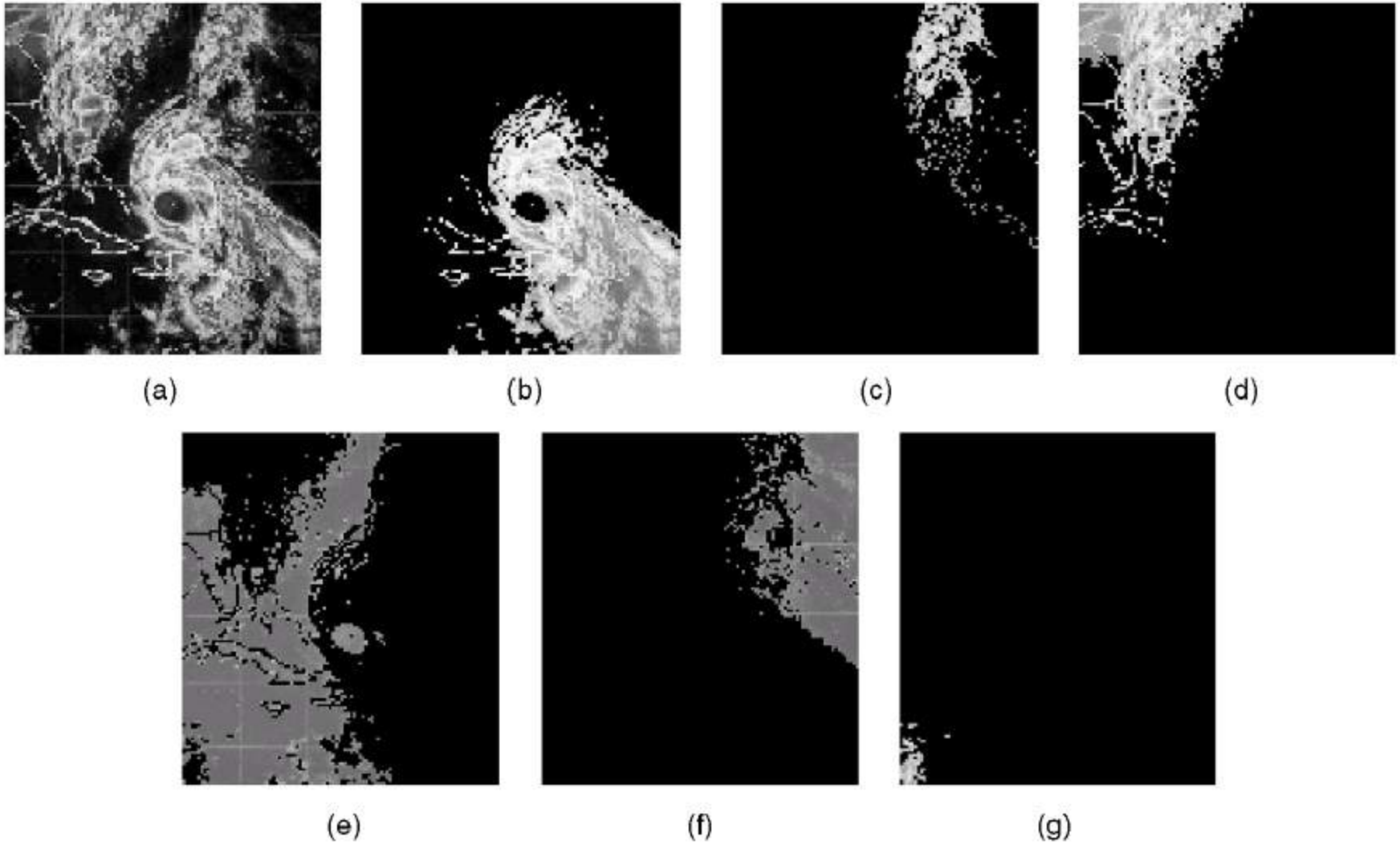


Fig. 8. (a) shows a 126×106 weather radar image. (b)-(g) show the components of the partition with $Ncut$ value less than 0.08. Parameter setting: $\sigma_I = 0.007$, $\sigma_x = 15.0$, $r = 10$.

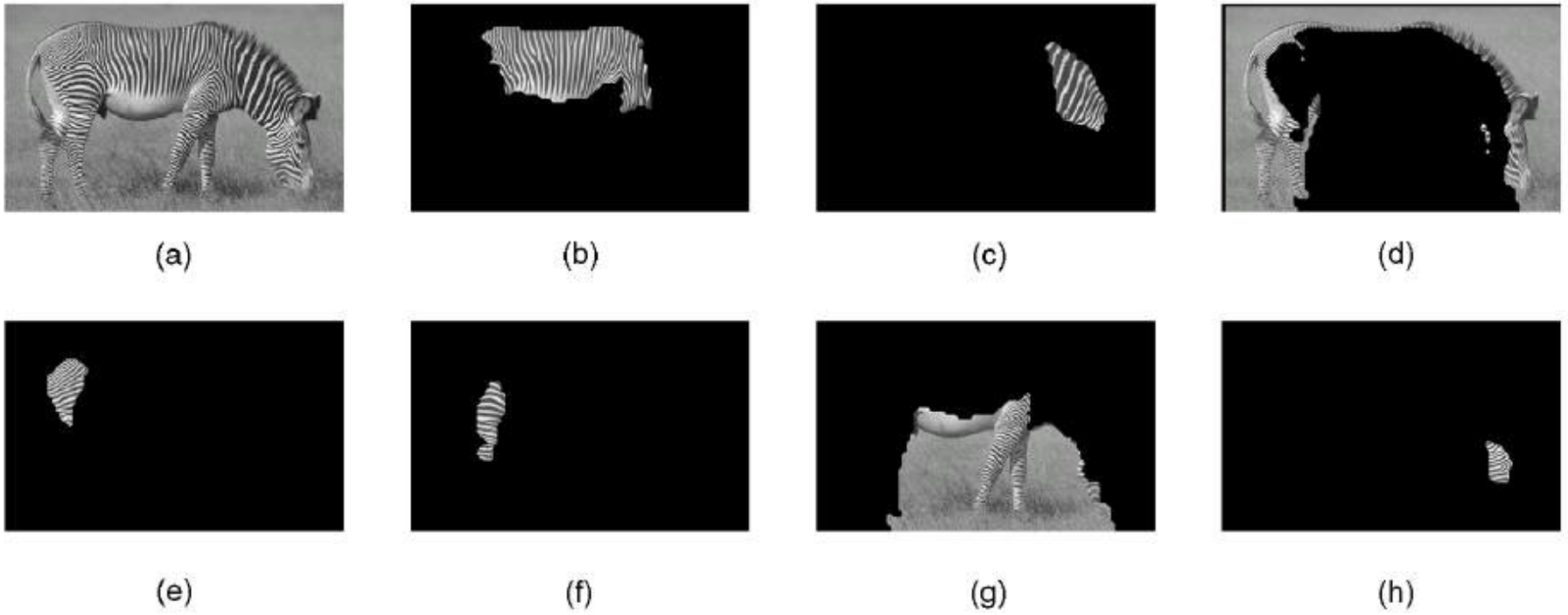


Fig. 10. (a) shows an image of a zebra. The remaining images show the major components of the partition. The texture features used correspond to convolutions with DOOG filters [16] at six orientations and five scales.

Spectral clustering

1. Construct a similarity graph and compute the normalized graph Laplacian L_{sym}.
2. Compute the k smallest eigenvectors $u_1, u_2, \cdots, u_k$ of L_{sym}.
3. Let $U = [\, u_1\, u_2\, \cdots\, u_k\,] \in \mathbb{R}^{n\times k}$.
4. Normalized the rows of U to norm 1.

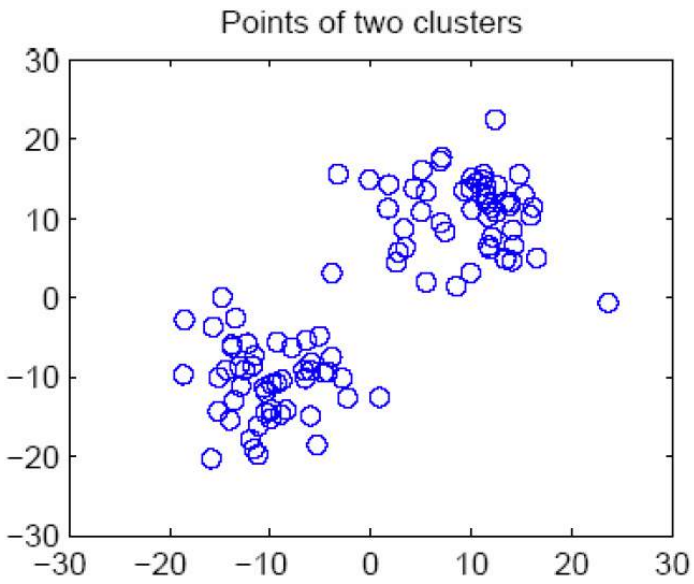
$$U_{ij} \leftarrow \frac{U_{ij}}{(\sum_k U_{ik}^2)^{1/2}}$$

5. Let $y_i \in \mathbb{R}^k$ be the vector corresponding to the ith row of U.
6. Thinking of y_i's as points in $\mathbb{R}^k$, cluster them with k-means algorithms.

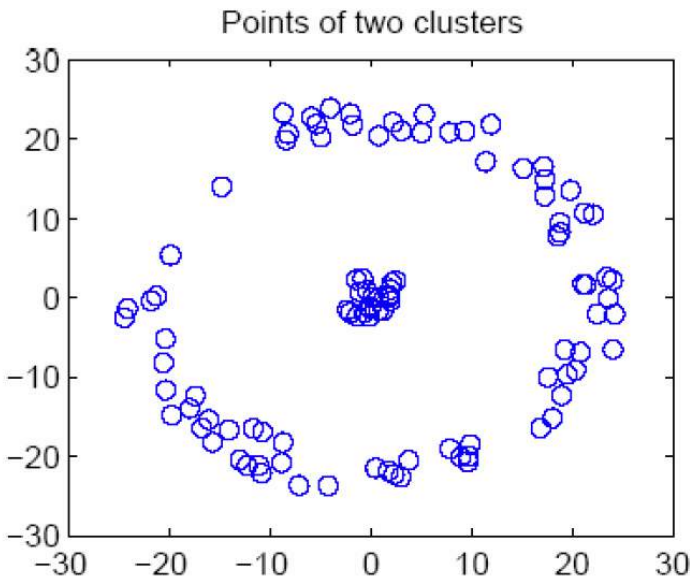
Ng, Jordan and Weiss (2002)

K-means *vs* Spectral clustering

Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.



Both perform same

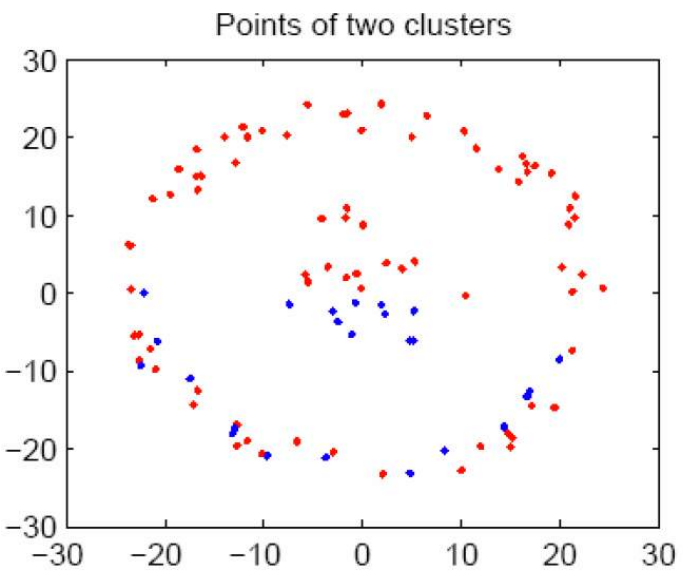


Spectral clustering is superior

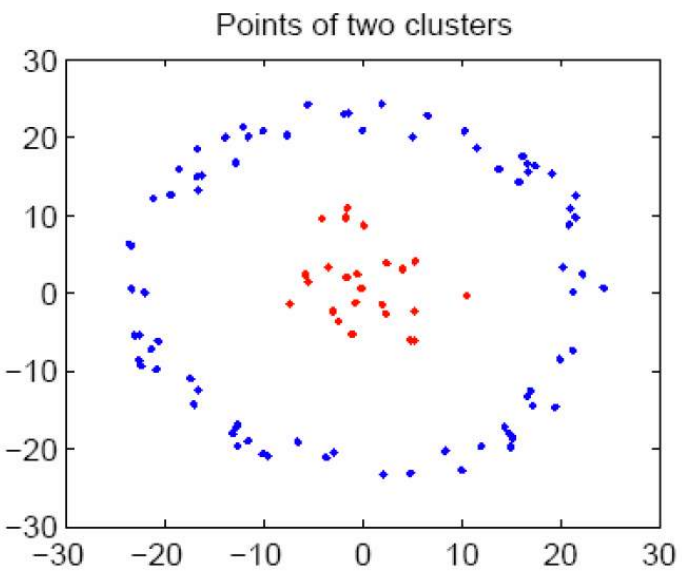
Adapted from A. Singh

K-means *vs* Spectral clustering

Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.



k-means output

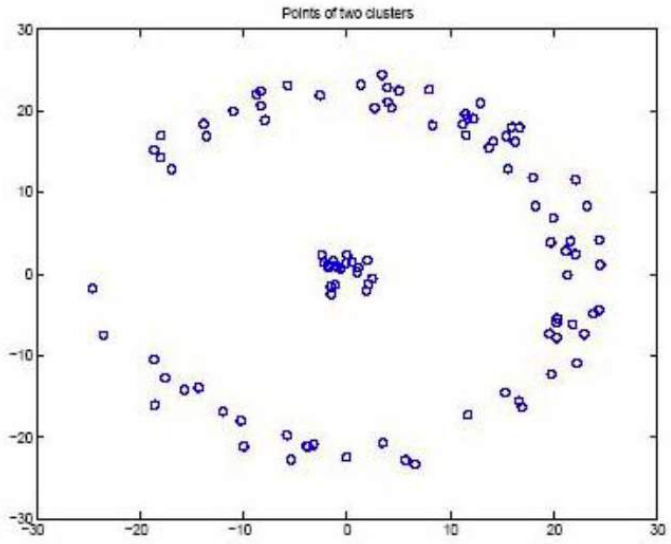


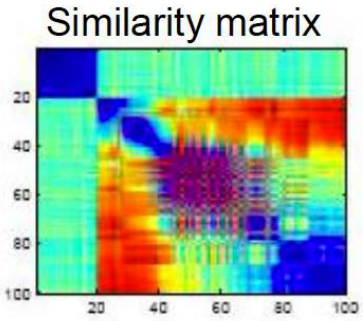
Spectral clustering output

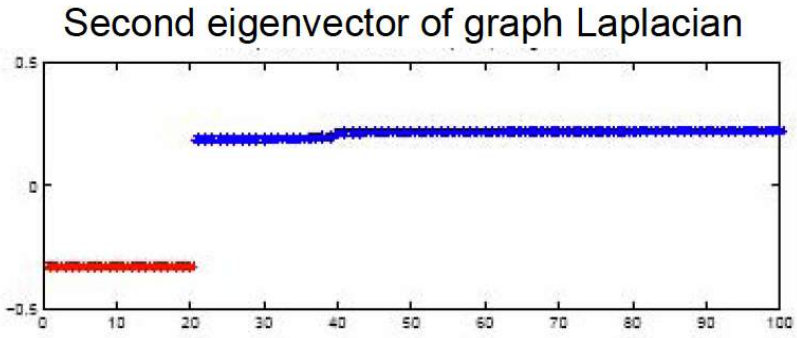
Adapted from A. Singh

K-means *vs* Spectral clustering

Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.

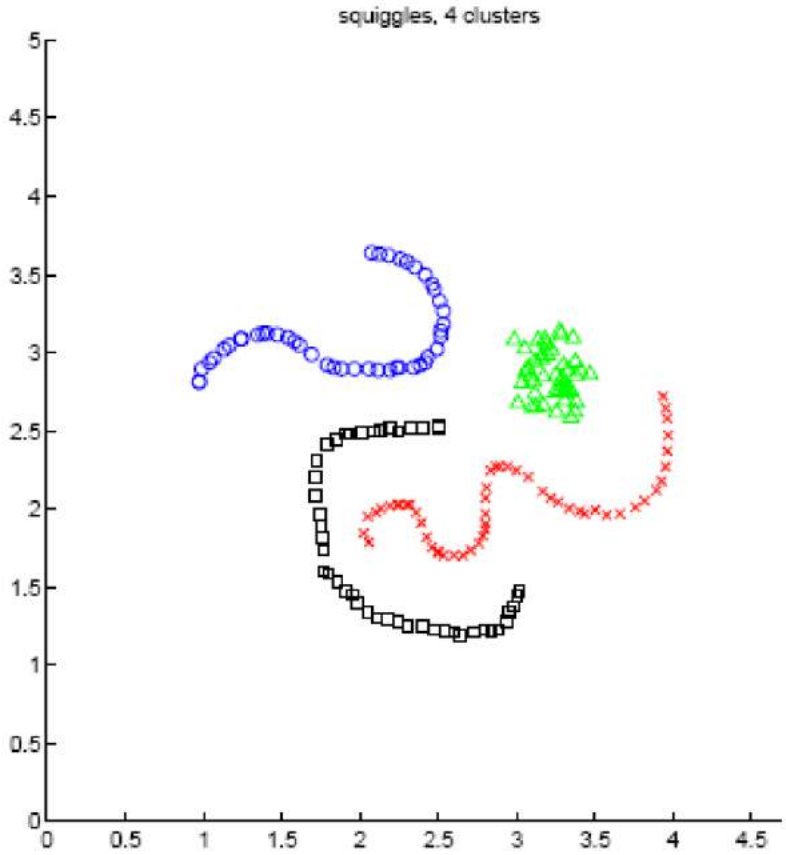


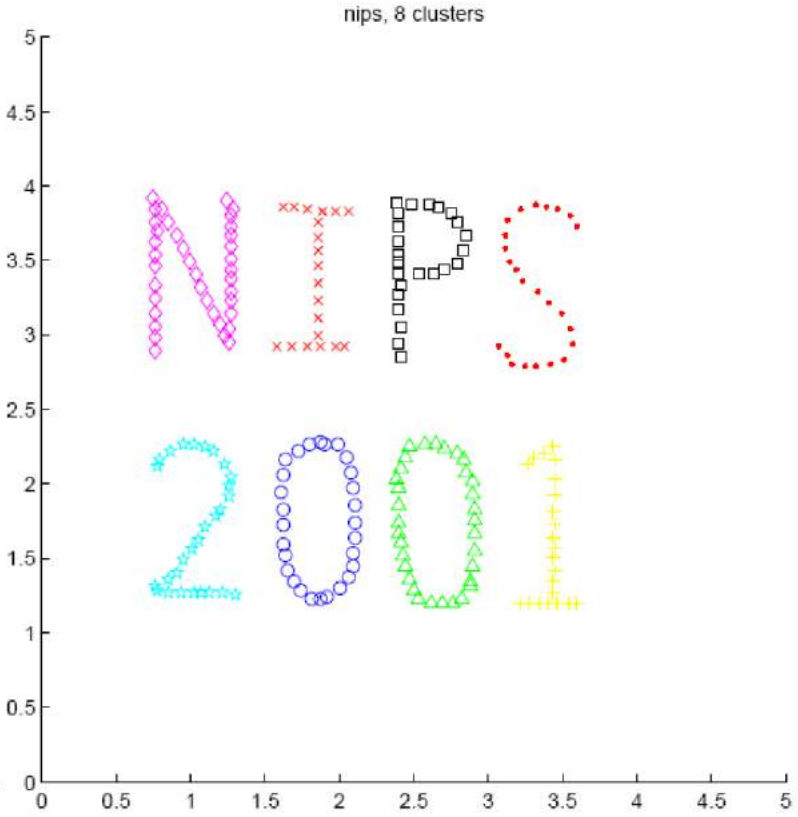




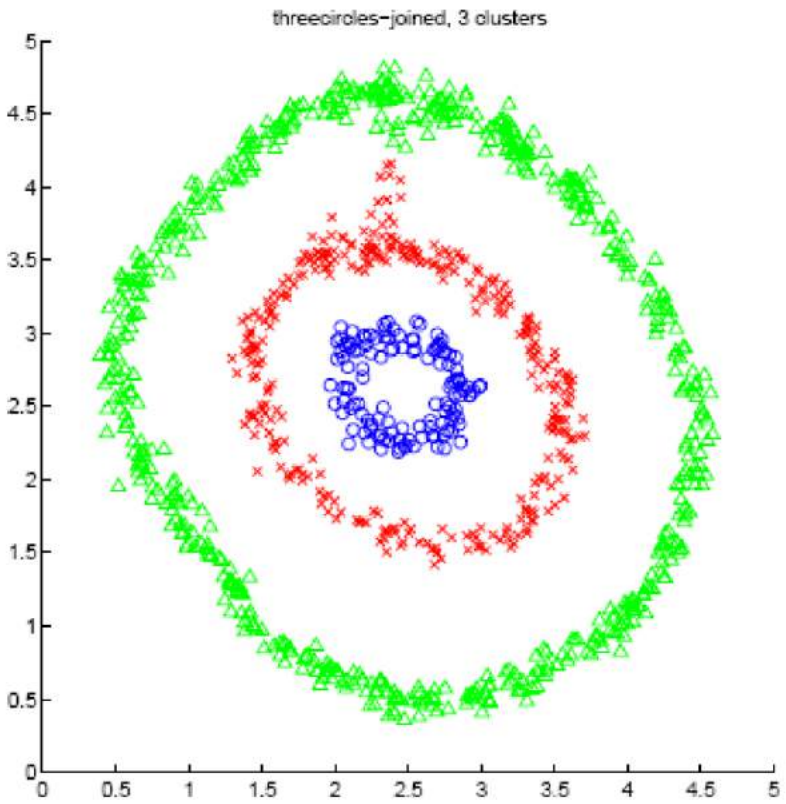
Adapted from A. Singh

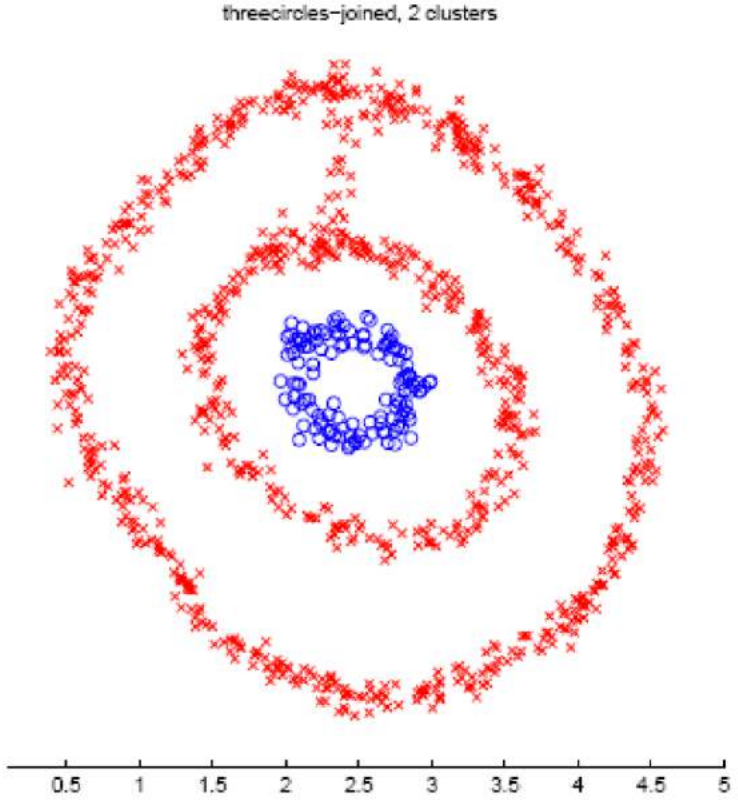
Examples


squiggles, 4 clusters
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


nips, 8 clusters
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

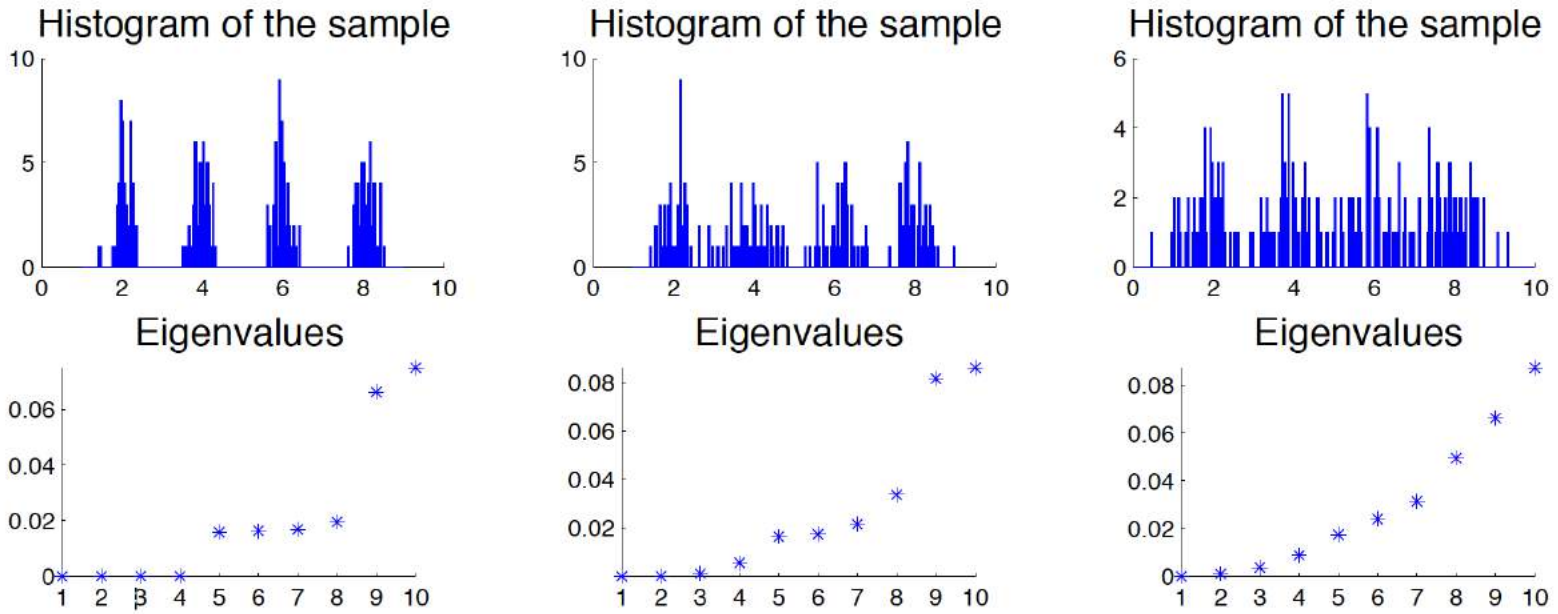
Examples (choice of k)





Choosing k

The eigengap heuristic: Choose k such that all eigenvalues $\lambda_1, \cdots, \lambda_k$ are very small, but λ_{k+1} is relatively large



Four 1D Gaussian clusters with increasing variance and corresponding eigevalues of L_{rw} (von Luxburg, 2007).

References

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

- M. Meila and J. Shi. A random walks view of spectral segmentation. *AISTATS* (2001).

- A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and analgorithm. *NIPS 14* (2002).

- U. von Luxburg, A tutorial on spectral clustering. *Statistics and Computing* 17(4) 395-4*16* (2007).

- A. K. Jain, Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 31(8):651-666 (2010).

Dominant Sets

The Need for Non-exhaustive Clusterings

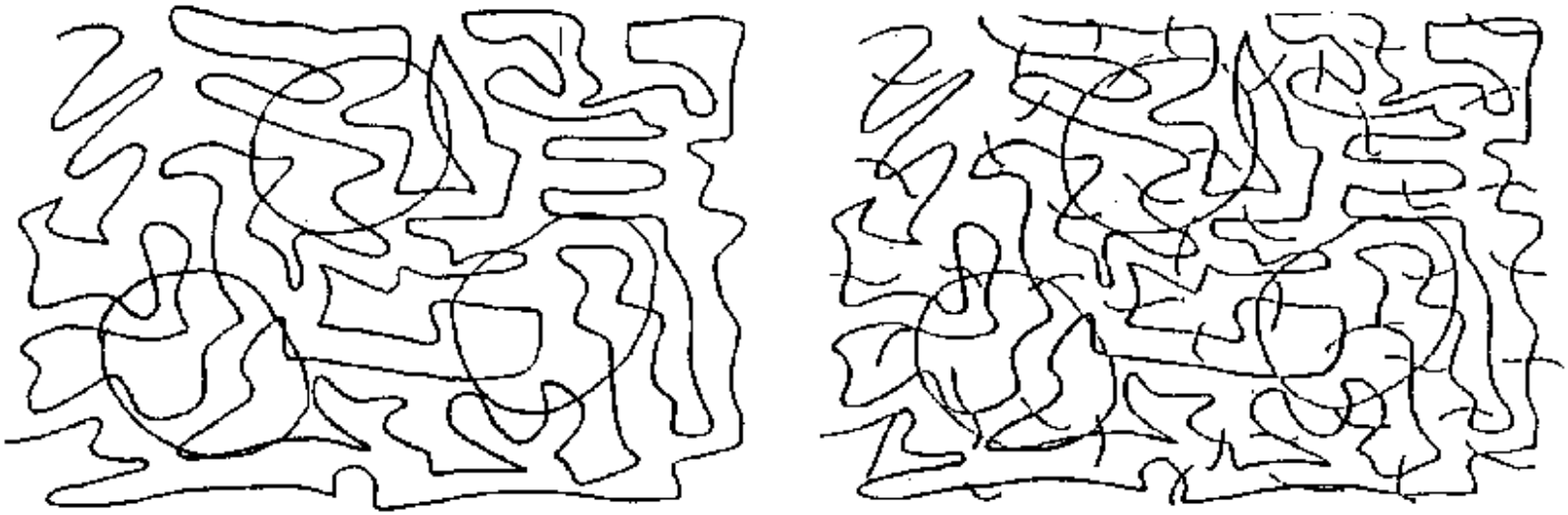


Figure 1a. Three prominent blobs are perceived immediately and with little effort. Locally, the blobs are similar to the background contours. (adopted from Mahoney (1986)

Figure 1b. Intersections were added to illustrate that the blobs are not distinguished by virtue of their intersections with the background curves.

Separating Structure from Clutter

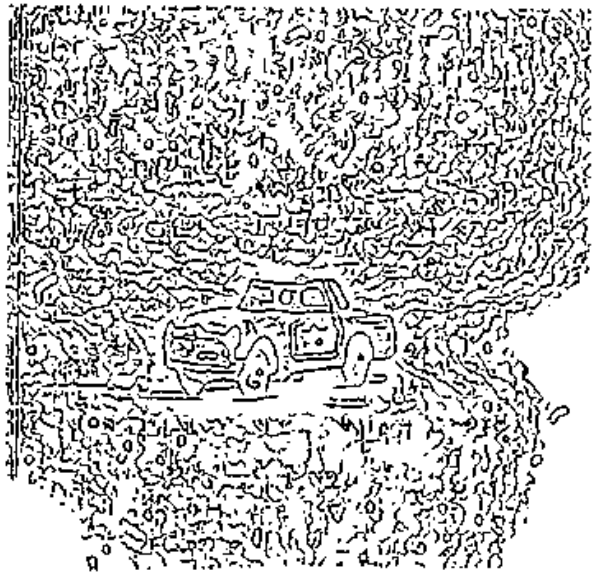
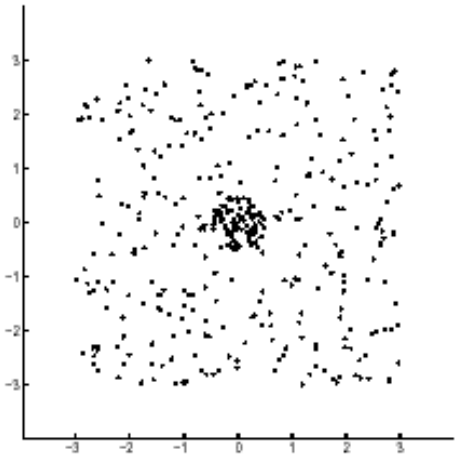


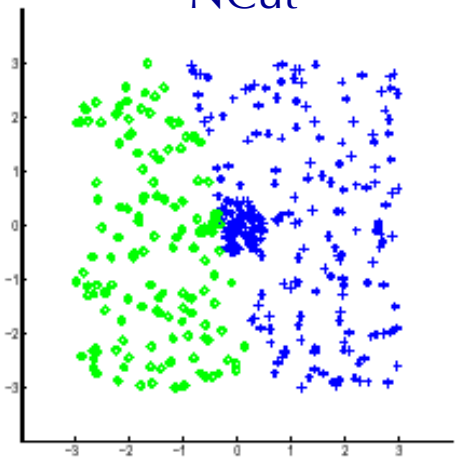
Figure 2. A circle in a background of 200 randomly placed and oriented segments The circle is still perceived immediately although its contour is fragmented.

Figure 3. An edge image of a car in a cluttered background. Our attention is drawn immediately to the region of interest. It seems that the car need not be recognized to attract our attention. The car also remains salient when parallel lines and small blobs are removed, and when the less textured region surrounding parts of the car is filled in with more texture.

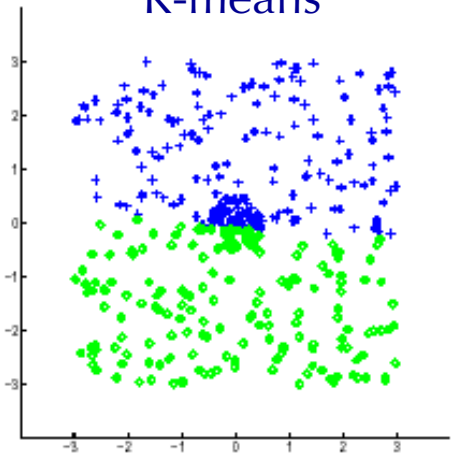
Separating Structure from Clutter



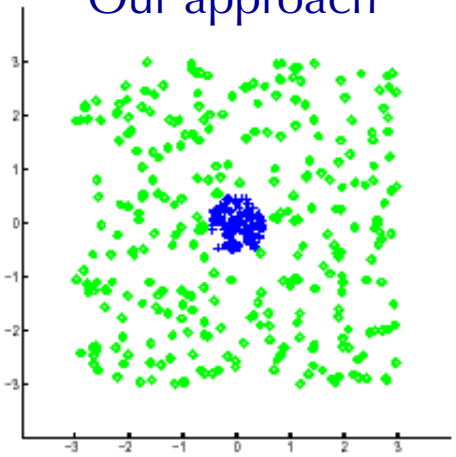
NCut



K-means



Our approach



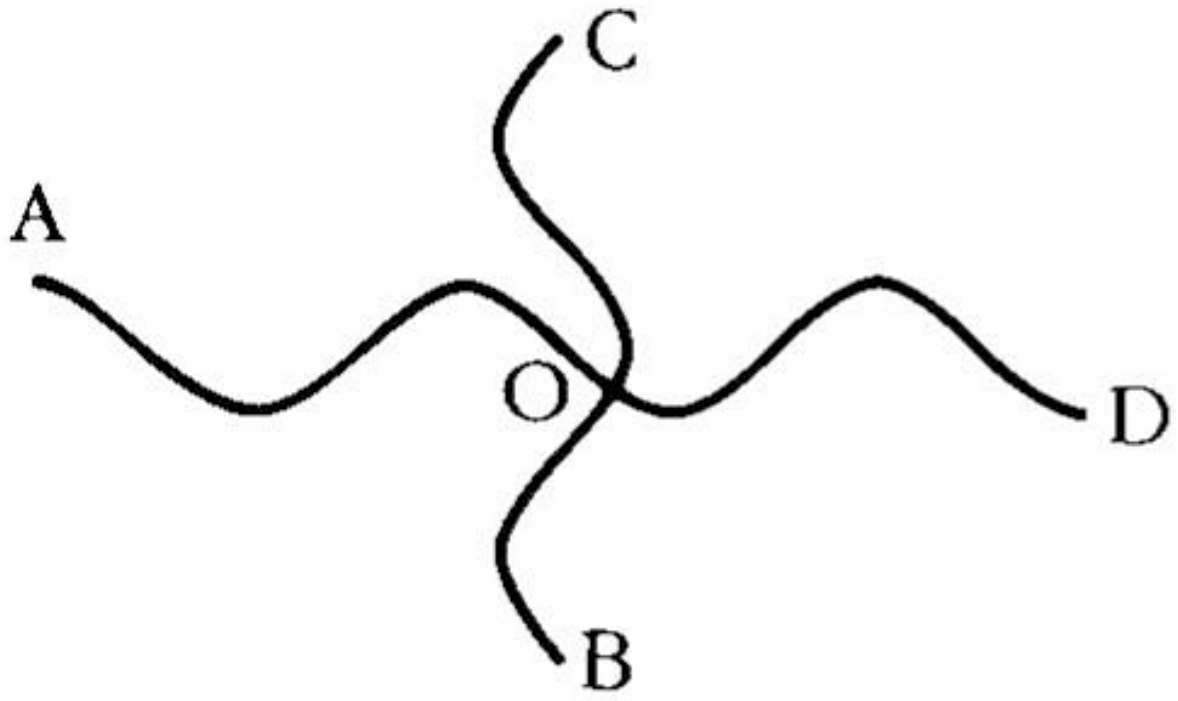
One-class Clustering

"[…] in certain real-world problems, natural groupings are found among only on a small subset of the data, while the rest of the data shows little or no clustering tendencies.

In such situations it is often more important to cluster a small subset of the data very well, rather than optimizing a clustering criterion over all the data points, particularly in application scenarios where a large amount of noisy data is encountered."

G. Gupta and J. Ghosh. Bregman bubble clustering: A robust framework for mining dense cluster. *ACM Trans. Knowl. Discov. Data* (2008).

When Groups Overlap



Does *O* belong to *AD* or to *BC* (or to none)?

The Need for Overlapping Clusters

Partitional approaches impose that each element cannot belong to more than one cluster. There are a variety of important applications, however, where this requirement is too restrictive.

Examples:

- ✓ clustering micro-array gene expression data
- ✓ clustering documents into topic categories
- ✓ perceptual grouping
- ✓ segmentation of images with transparent surfaces

References:

- ✓ N. Jardine and R. Sibson. The construction of hierarchic and non-hierarchic classifications. *Computer Journal*, 11:177–184, 1968
- ✓ A. Banerjee, C. Krumpelman, S. Basu, R. J. Mooney, and J. Ghosh. Model-based overlapping clustering. *KDD 2005*.
- ✓ K. A. Heller and Z. Ghahramani. A nonparametric Bayesian approach to modeling overlapping clusters. *AISTATS 2007*.

The Symmetry Assumption

«Similarity has been viewed by both philosophers and psychologists as a prime example of a symmetric relation. Indeed, the assumption of symmetry underlies essentially all theoretical treatments of similarity.

Contrary to this tradition, the present paper provides empirical evidence for asymmetric similarities and argues that **similarity should not be treated as a symmetric relation**.»

Amos Tversky

"Features of similarities," *Psychol. Rev.* (1977)

Examples of asymmetric (dis)similarities

- ✓ Kullback-Leibler divergence
- ✓ Directed Hausdorff distance
- ✓ Tversky's contrast model

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

The Notion of "Gestalt"

«In most visual fields the contents of particular areas "belong together" as circumscribed units from which their surrounding are excluded.»

W. Köhler, *Gestalt Psychology* (1947)

«In gestalt theory the word "Gestalt" means any segregated whole.»

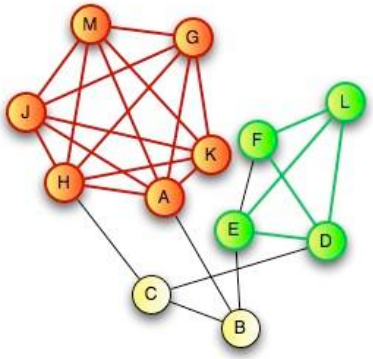
W. Köhler (1929)

Data Clustering: Old *vs.* New

By answering the question "what is a cluster?" we get a novel way of looking at the clustering problem.

```
Clustering_old(V,A,k)
   V1,V2,...,Vk <- My_favorite_partitioning_algorithm(V,A,k)
   return V1,V2,...,Vk

------

Clustering_new(V,A)
   V1,V2,...,Vk <- Enumerate_all_clusters(V,A)
   return V1,V2,...,Vk

Enumerate_all_clusters(V,A)
   repeat
      Extract_a_cluster(V,A)
   until all clusters have been found
   return the clusters found
```

A Special Case: Binary Symmetric Affinities

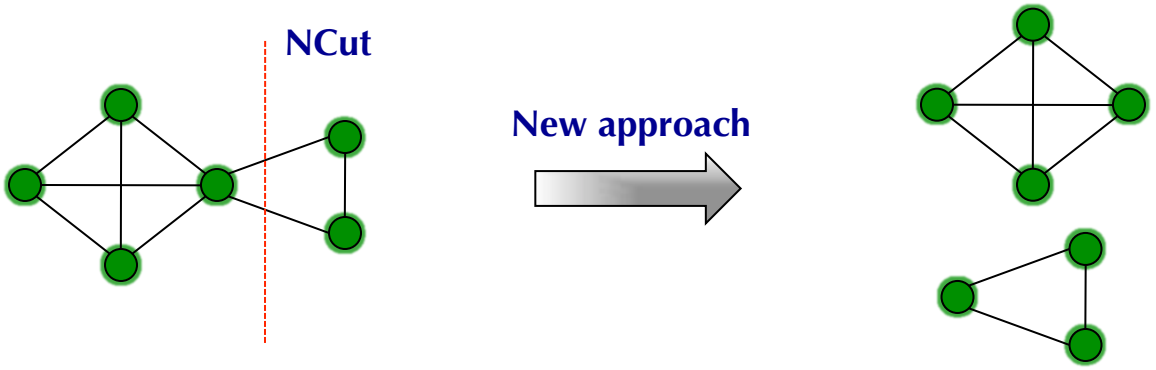


Suppose the similarity matrix is a binary (0/1) matrix.

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

In the 0/1 case, a meaningful notion of a cluster is that of a *maximal clique*.



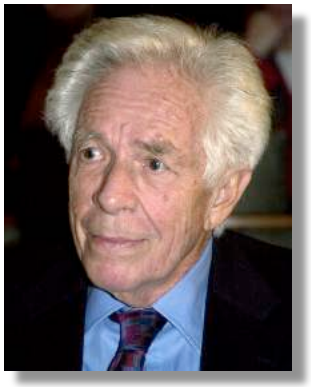
Advantages of the New Approach

- ✓ No need to know the number of clusters in advance (since we extract them sequentially)
- ✓ Leaves clutter elements unassigned (useful, e.g., in figure/ground separation or one-class clustering problems)
- ✓ Allows extracting overlapping clusters

Need a partition?

```
Partition_into_clusters(V,A)
  repeat
     Extract_a_cluster
     remove it from V
  until all vertices have been clustered
```

What is Game Theory?



"The central problem of game theory was posed by von Neumann as early as 1926 in Göttingen. It is the following: If n players, $P_1,\ldots, P_n$, play a given game Γ, how must the i^{th} player, P_i, play to achieve the most favorable result for himself?"

Harold W. Kuhn

Lectures on the Theory of Games (1953)

A few cornerstones in game theory

1921–1928: Emile Borel and John von Neumann give the first modern formulation of a mixed strategy along with the idea of finding minimax solutions of normal-form games.

1944, 1947: John von Neumann and Oskar Morgenstern publish *Theory of Games and Economic Behavior*.

1950–1953: In four papers John Nash made seminal contributions to both non-cooperative game theory and to bargaining theory.

1972–1982: John Maynard Smith applies game theory to biological problems thereby founding "evolutionary game theory."

late 1990's –: Development of algorithmic game theory…

"Solving" a Game

		Player 2		
		Left	Middle	Right
Player 1	Top	3 , 1	2 , 3	10 , 2
	High	4 , 5	3 , 0	, 4
	Low	2 , 2	5 , 4	12 , 3
	Bottom	5 , 6	4 , 5	9 , 7

Nash equilibrium!

Basics of (Two-Player, Symmetric) Game Theory

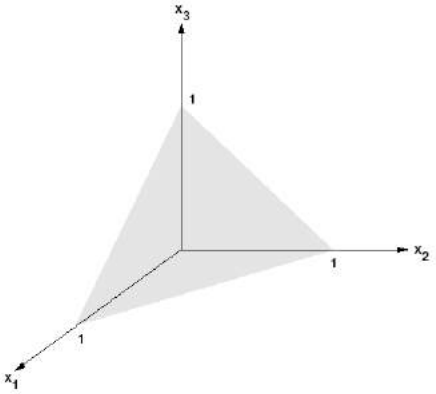
Assume:

- a (symmetric) game between two players
- complete knowledge
- a pre-existing set of **pure strategies** (actions) $O=\{o_1,\ldots,o_n\}$ available to the players.

Each player receives a payoff depending on the strategies selected by him and by the adversary. Players' goal is to maximize their own returns.

A **mixed strategy** is a probability distribution $\mathbf{x}=(x_1,\ldots,x_n)^T$ over the strategies.

$$\Delta = \left\{ x \in R^n : \ \forall i = 1 \ldots n : x_i \geq 0, \text{ and } \sum_{i=1}^{n} x_i = 1 \right\}$$



Nash Equilibria

- Let A be an arbitrary **payoff** matrix: a_{ij} is the payoff obtained by playing i while the opponent plays j.

- The average payoff obtained by playing mixed strategy **y** while the opponent plays **x**, is:

$$\mathbf{y}'\mathbf{A}\mathbf{x} = \sum_i \sum_j a_{ij} y_i x_j$$

- A mixed strategy **x** is a (symmetric) **Nash equilibrium** if

$$\mathbf{x}'\mathbf{A}\mathbf{x} \geq \mathbf{y}'\mathbf{A}\mathbf{x}$$

for all strategies **y**. (Best reply to itself.)

Theorem (Nash, 1951). Every finite normal-form game admits a mixed-strategy Nash equilibrium.

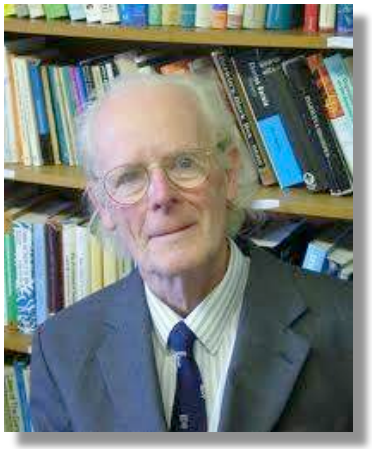
Evolution and the Theory of Games

"We repeat most emphatically that our theory is thoroughly static. A dynamic theory would unquestionably be more complete and therefore preferable.

But there is ample evidence from other branches of science that it is futile to try to build one as long as the static side is not thoroughly understood."

John von Neumann and Oskar Morgenstern

Theory of Games and Economic Behavior (1944)



"Paradoxically, it has turned out that game theory is more readily applied to biology than to the field of economic behaviour for which it was originally designed."

John Maynard Smith

Evolution and the Theory of Games (1982)

Evolutionary Games and ESS's

Assumptions:

- ✓ A large population of individuals belonging to the same species which compete for a particular limited resource
- ✓ This kind of conflict is modeled as a symmetric two-player game, the players being pairs of randomly selected population members
- ✓ Players do not behave "rationally" but act according to a pre-programmed behavioral pattern (pure strategy)
- ✓ Reproduction is assumed to be asexual
- ✓ Utility is measured in terms of Darwinian fitness, or reproductive success

A Nash equilibrium **x** is an ***Evolutionary Stable Strategy*** (ESS) if, for all strategies y:

$$\mathbf{y}'A\mathbf{x} = \mathbf{x}'A\mathbf{x} \quad \Rightarrow \quad \mathbf{x}'A\mathbf{y} > \mathbf{y}'A\mathbf{y}$$

ESS's as Clusters

We claim that 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.

Basic Definitions

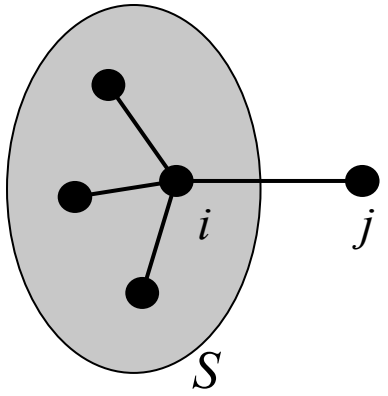
Let $S \subseteq V$ be a non-empty subset of vertices, and $i \in S$.

The **(average) weighted degree** of i w.r.t. S is defined as:

$$\text{awdeg}_S(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij}$$

Moreover, if $j \notin S$, we define:

$$\phi_S(i,j) = a_{ij} - \text{awdeg}_S(i)$$



Intuitively, $\phi_S(i,j)$ measures the similarity between vertices j and i, with respect to the (average) similarity between vertex i and its neighbors in S.

Assigning Weights to Vertices

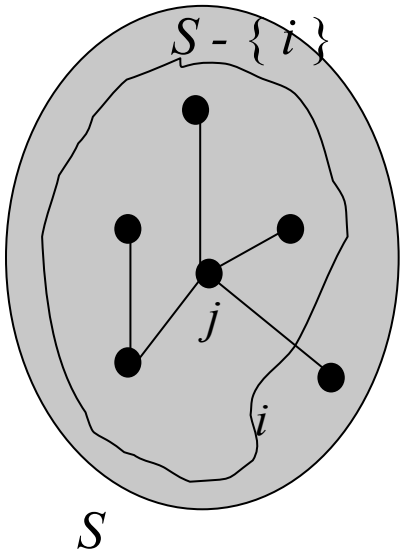
Let $S \subseteq V$ be a non-empty subset of vertices, and $i \in S$.

The **weight** of i w.r.t. S is defined as:

$$w_S(i) = \begin{cases} 1 & \text{if } |S| = 1 \\ \sum_{j \in S - \{i\}} \phi_{S-\{i\}}(j,i) w_{S-\{i\}}(j) & \text{otherwise} \end{cases}$$

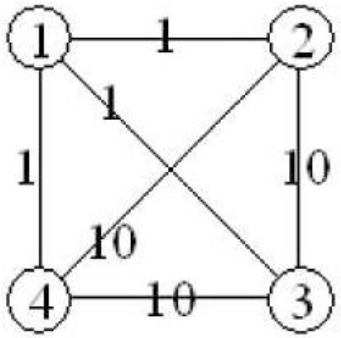
Further, the **total weight** of S is defined as:

$$W(S) = \sum_{i \in S} w_S(i)$$

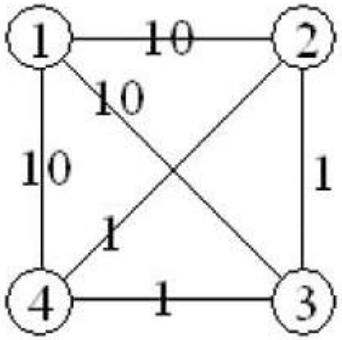


Interpretation

Intuitively, $w_S(i)$ gives us a measure of the overall (relative) similarity between vertex i and the vertices of S-{i} with respect to the overall similarity among the vertices in S-{i}.



$w_{\{1,2,3,4\}}(1) < 0$

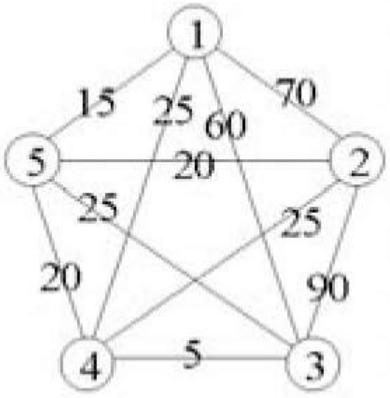


$w_{\{1,2,3,4\}}(1) > 0$

Dominant Sets

Definition (Pavan and Pelillo, 2003, 2007). A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be a **dominant set** if:

1. $w_S(i) > 0$, for all $i \in S$ (internal homogeneity)
2. $w_{S \cup \{i\}}(i) < 0$, for all $i \notin S$ (external homogeneity)



The set {1,2,3} is dominant.

Dominant sets ≡ clusters

The Clustering Game

Consider the following "clustering game."

- ✓ Assume a preexisting set of objects O and a (possibly asymmetric) matrix of affinities A between the elements of O.
- ✓ Two players play by simultaneously selecting an element of O.
- ✓ After both have shown their choice, each player receives a payoff proportional to the affinity that the chosen element has wrt the element chosen by the opponent.

Clearly, it is in each player's interest to pick an element that is strongly supported by the elements that the adversary is likely to choose.

Hence, in the (pairwise) clustering game:

- ✓ There are 2 players (because we have pairwise affinities)
- ✓ The objects to be clustered are the pure strategies
- ✓ The (null-diagonal) affinity matrix coincides with the similarity matrix

Dominant Sets are ESS's

Theorem (Torsello, Rota Bulò and Pelillo, 2006). Evolutionary stable strategies of the clustering game with affinity matrix A are in a one-to-one correspondence with dominant sets.

Note. Generalization of well-known Motzkin-Straus theorem from graph theory (1965).

Dominant-set clustering

- ✓ To get a single dominant-set cluster use, e.g., replicator dynamics (but see Rota Bulò, Pelillo and Bomze, *CVIU* 2011, for faster dynamics)
- ✓ To get a partition use a simple *peel-off* strategy: iteratively find a dominant set and remove it from the graph, until all vertices have been clustered
- ✓ To get overlapping clusters, enumerate dominant sets (see Bomze, 1992; Torsello, Rota Bulò and Pelillo, 2008)

Special Case: 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, 2003, 2007; Sarkar and Boyer, 1998; Perona and Freeman, 1998).

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

Note. In the 0/1 (symmetric) case, ESS's are in one-to-one correspondence to (strictly) **maximal cliques** (Motzkin-Straus theorem).

Replicator Dynamics

Let $x_i(t)$ the population share playing pure strategy i at *time* t. The **state** of the population at time t is: $x(t)= (x_1(t),\ldots,x_n(t)) \in \Delta$.

Replicator dynamics (Taylor and Jonker, 1978) are motivated by Darwin's principle of natural selection:

$$\frac{\dot{x}_i}{x_i} \propto \text{payoff of pure strategy i} - \text{average population payoff}$$

which yields:

$$\dot{x}_i = x_i\left[(Ax)_i - x^T Ax\right]$$

Theorem (Nachbar, 1990; Taylor and Jonker, 1978). A point $x \in \Delta$ is a Nash equilibrium if and only if x is the limit point of a replicator dynamics trajectory starting from the interior of Δ.

Furthermore, if $x \in \Delta$ is an ESS, then it is an asymptotically stable equilibrium point for the replicator dynamics.

Doubly Symmetric Games

In a doubly symmetric (or partnership) game, the payoff matrix A is symmetric ($A = A^T$).

Fundamental Theorem of Natural Selection (Losert and Akin, 1983).

For any doubly symmetric game, the average population payoff $f(x) = x^T Ax$ is strictly increasing along any non-constant trajectory of replicator dynamics, namely, $d/dt f(x(t)) \geq 0$ for all $t \geq 0$, with equality if and only if $x(t)$ is a stationary point.

Characterization of ESS's (Hofbauer and Sigmund, 1988)

For any doubly simmetric game with payoff matrix A, the following statements are equivalent:

a) $x \in \Delta^{ESS}$

b) $x \in \Delta$ is a strict local maximizer of $f(x) = x^T Ax$ over the standard simplex Δ

c) $x \in \Delta$ is asymptotically stable in the replicator dynamics

Discrete-time Replicator Dynamics

A well-known discretization of replicator dynamics, which assumes non-overlapping generations, is the following (assuming a non-negative A):

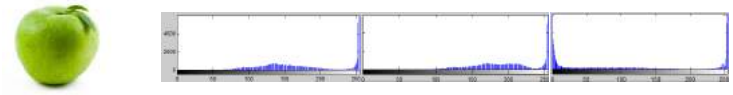
$$x_i(t+1) = x_i(t)\frac{A\big(x(t)\big)_i}{x(t)^T Ax(t)}$$

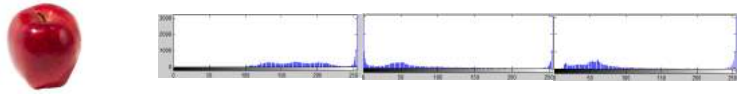
which inherits most of the dynamical properties of its continuous-time counterpart (*e.g.*, the fundamental theorem of natural selection).

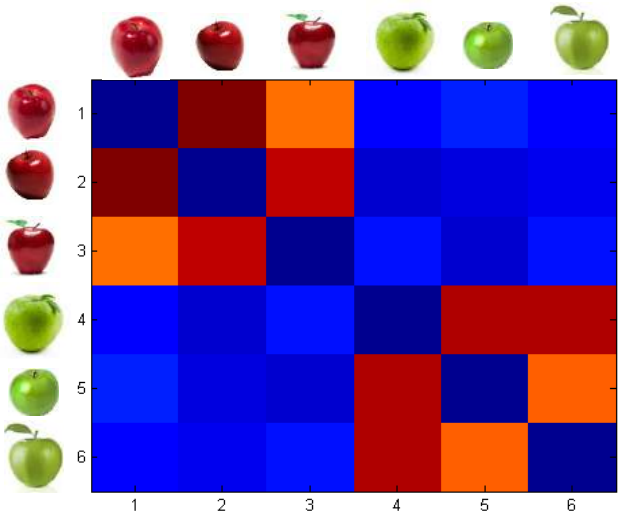
MATLAB implementation

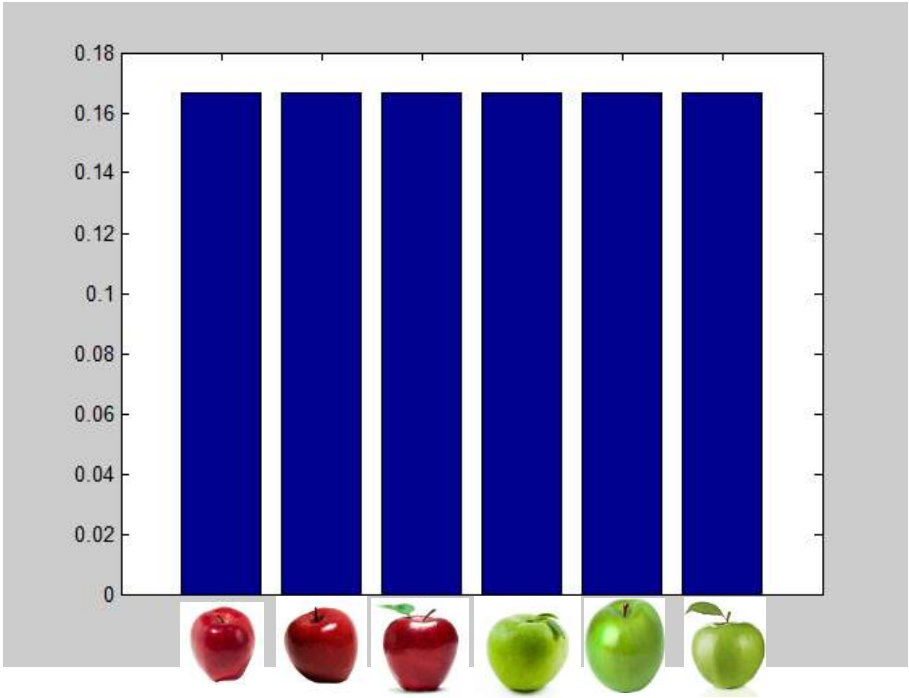
```
distance=inf;

while distance>epsilon

    old_x=x;

    x = x.*(A*x);

    x = x./sum(x);

    distance=pdist([x,old_x]');

end
```

A Toy Example



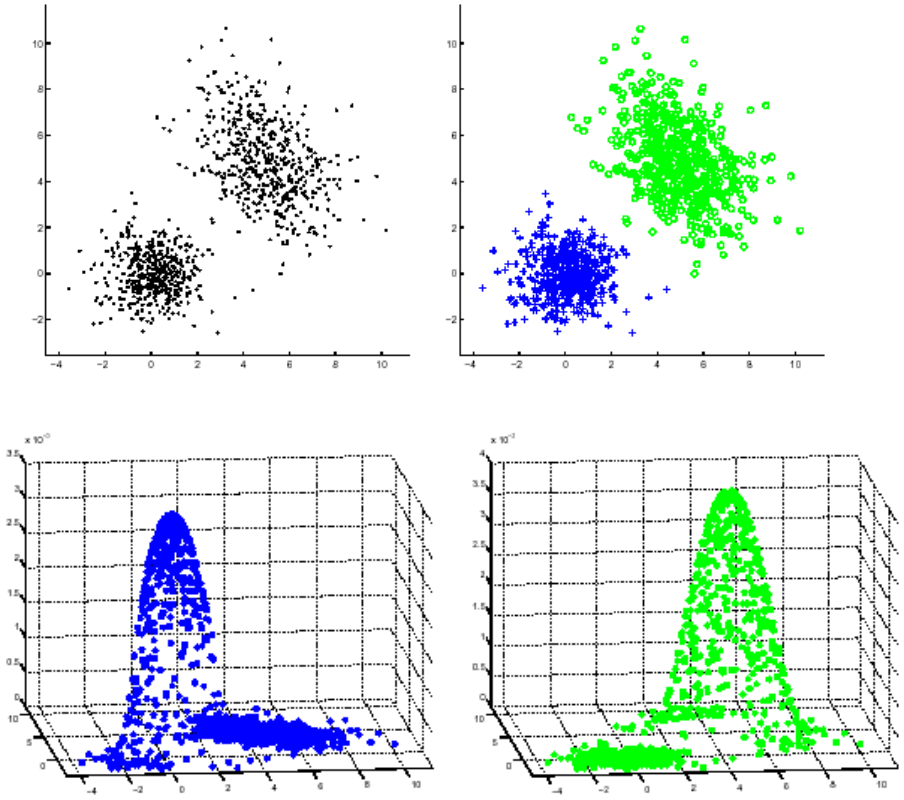






Measuring the Degree of Cluster Membership

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.



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.

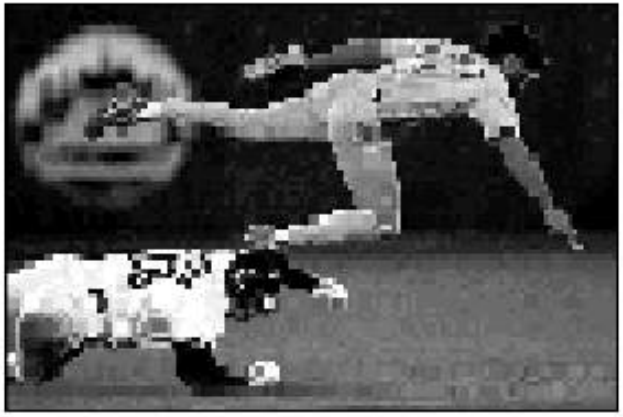
For the sake of comparison, in the experiments we used the same similarities used in Shi and Malik's normalized-cut paper (PAMI 2000).

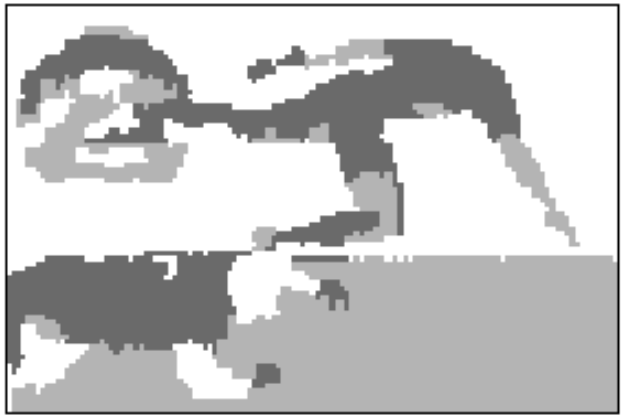
To find a hard partition, the following *peel-off* strategy was used:

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

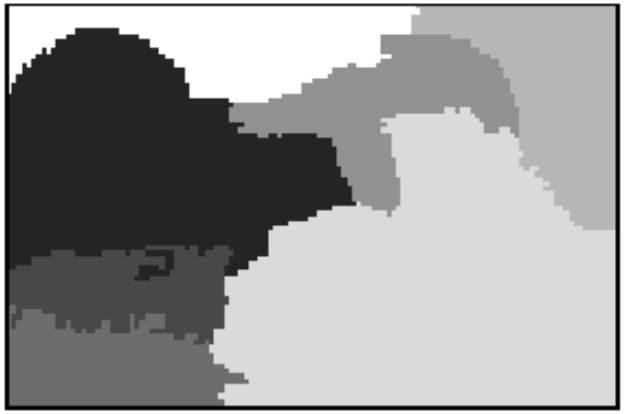
To find a single dominant set we used replicator dynamics (but see Rota Bulò, Pelillo and Bomze, *CVIU 2011*, for faster game dynamics).

Intensity Segmentation Results



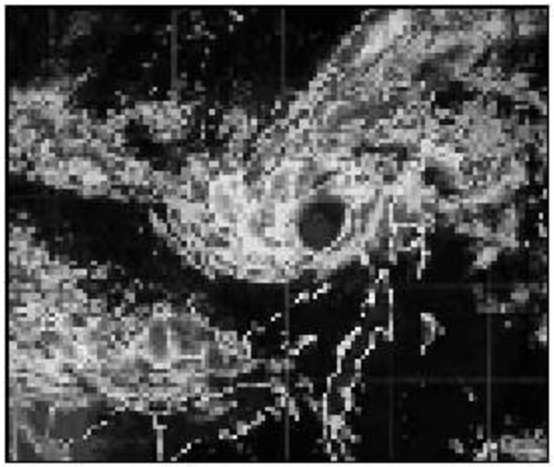


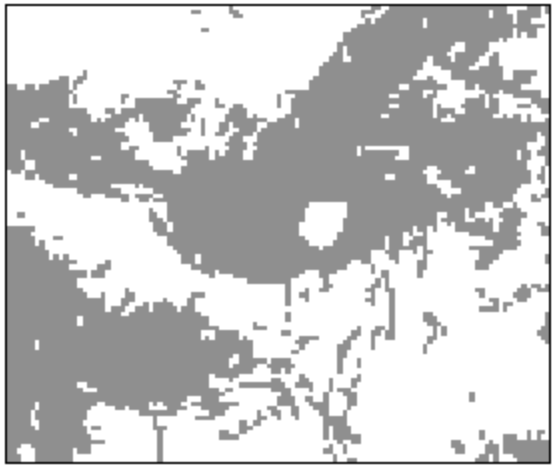
Dominant sets



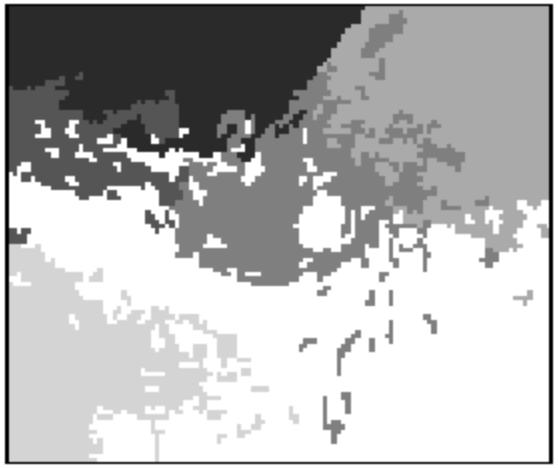
Ncut

Intensity Segmentation Results



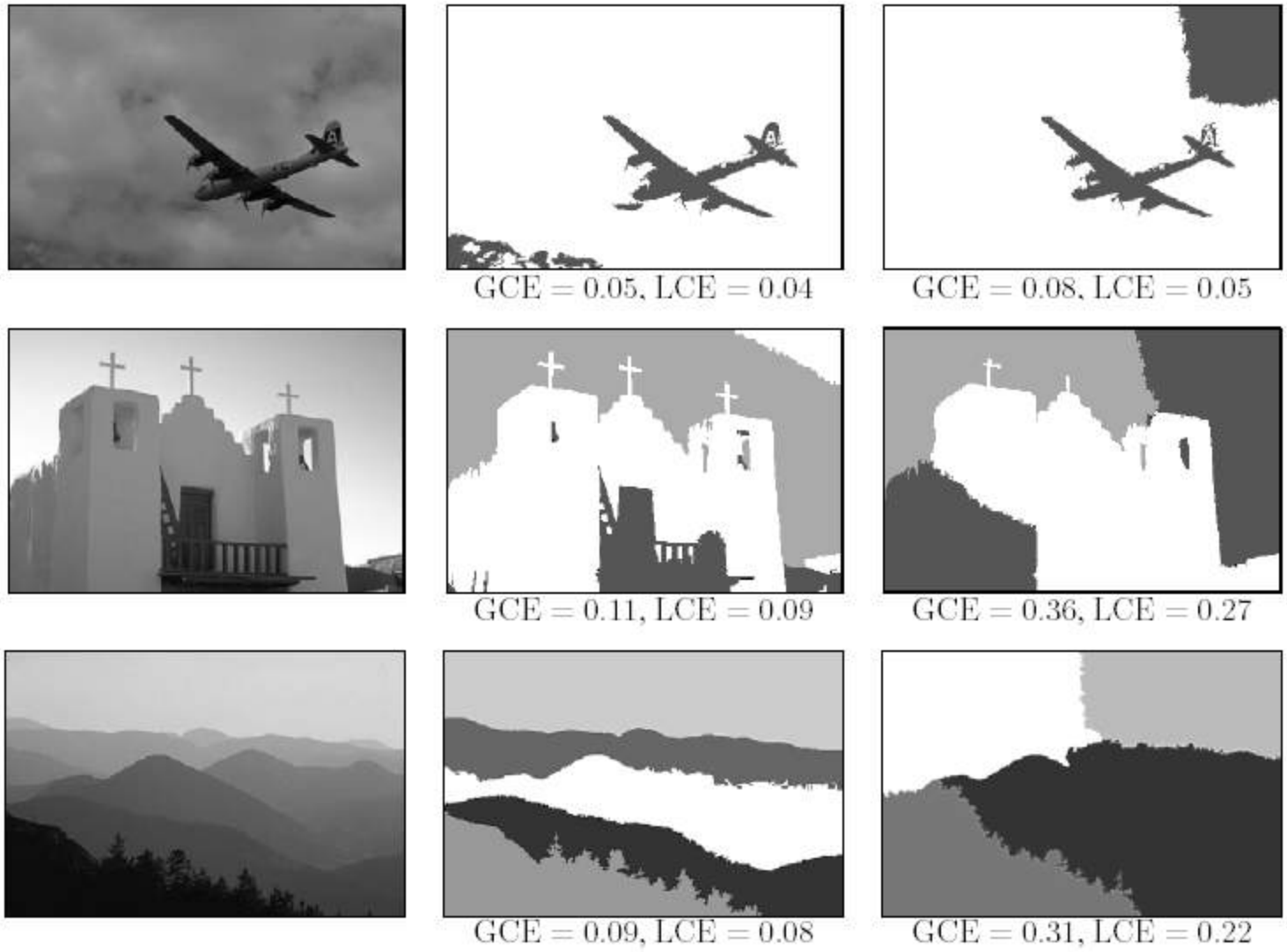


Dominant sets



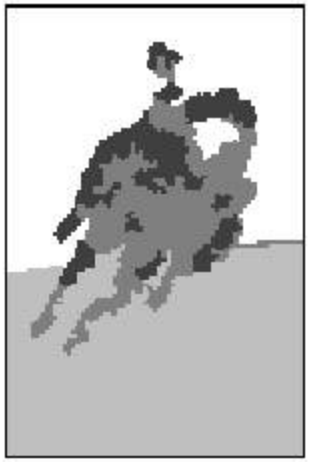
Ncut

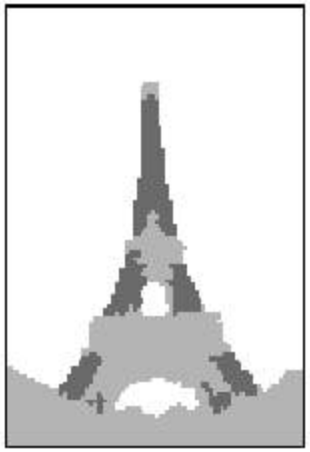
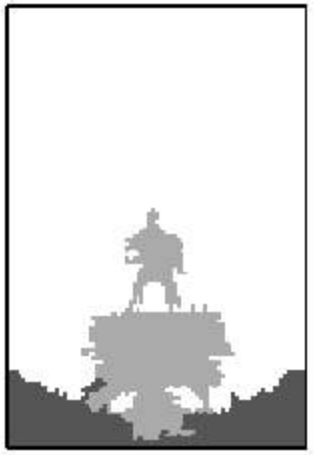
Results on the Berkeley Dataset



Color Segmentation Results

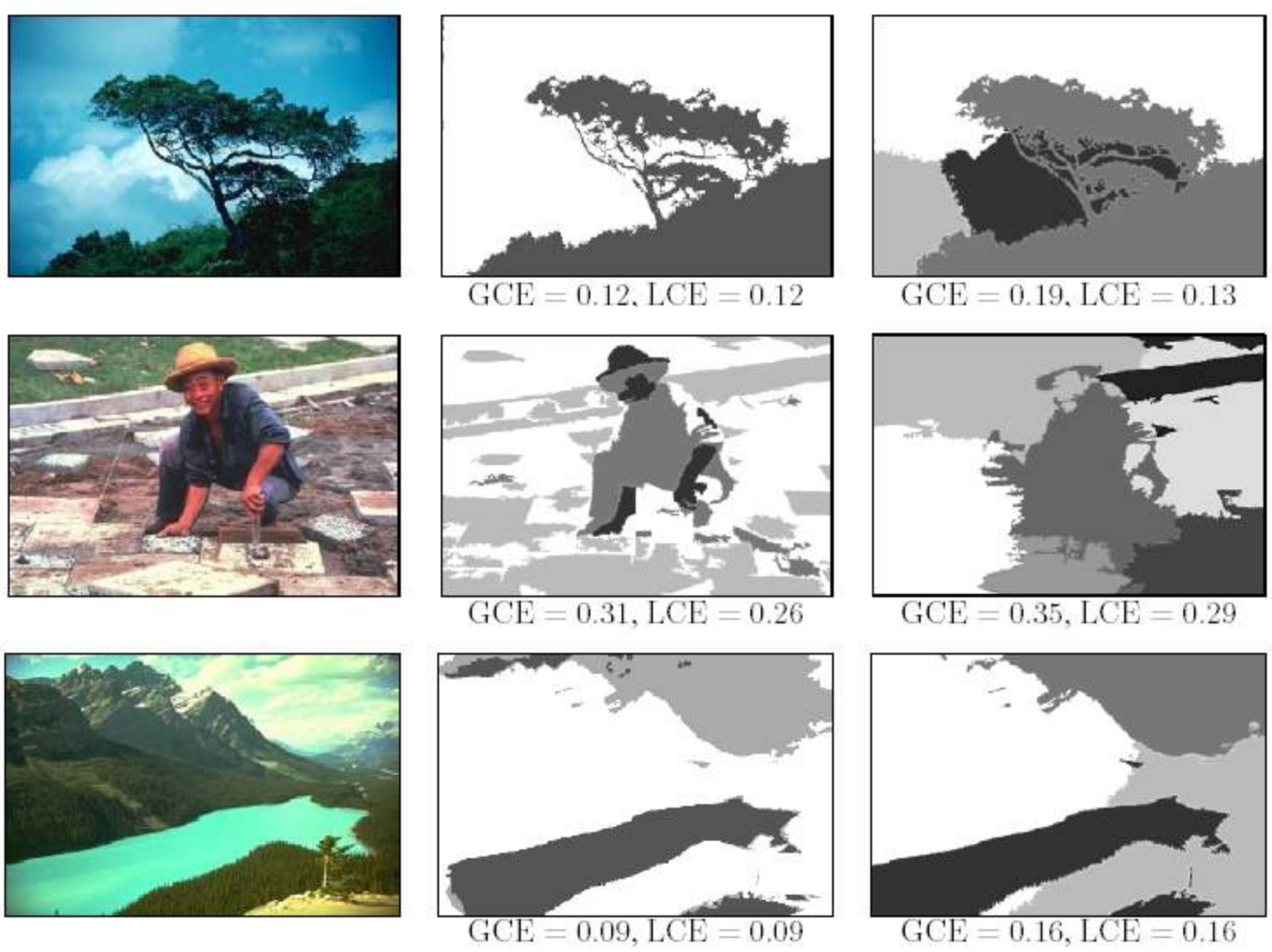




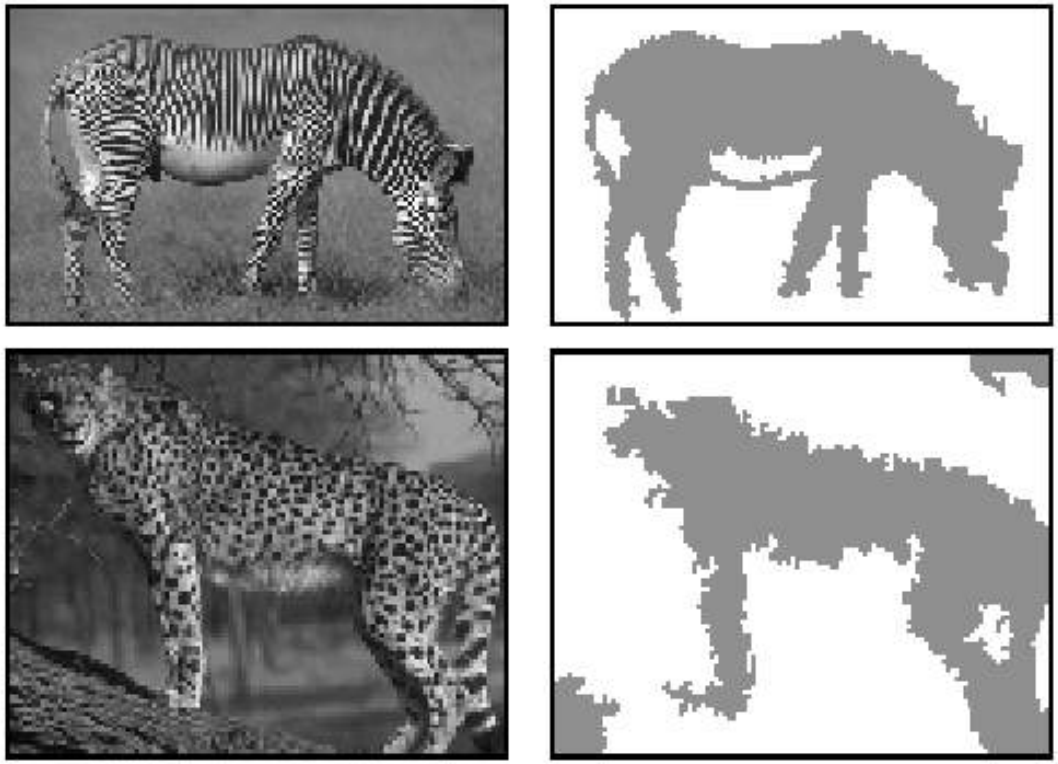



Original image | Dominant sets | Ncut

Results on the Berkeley Dataset

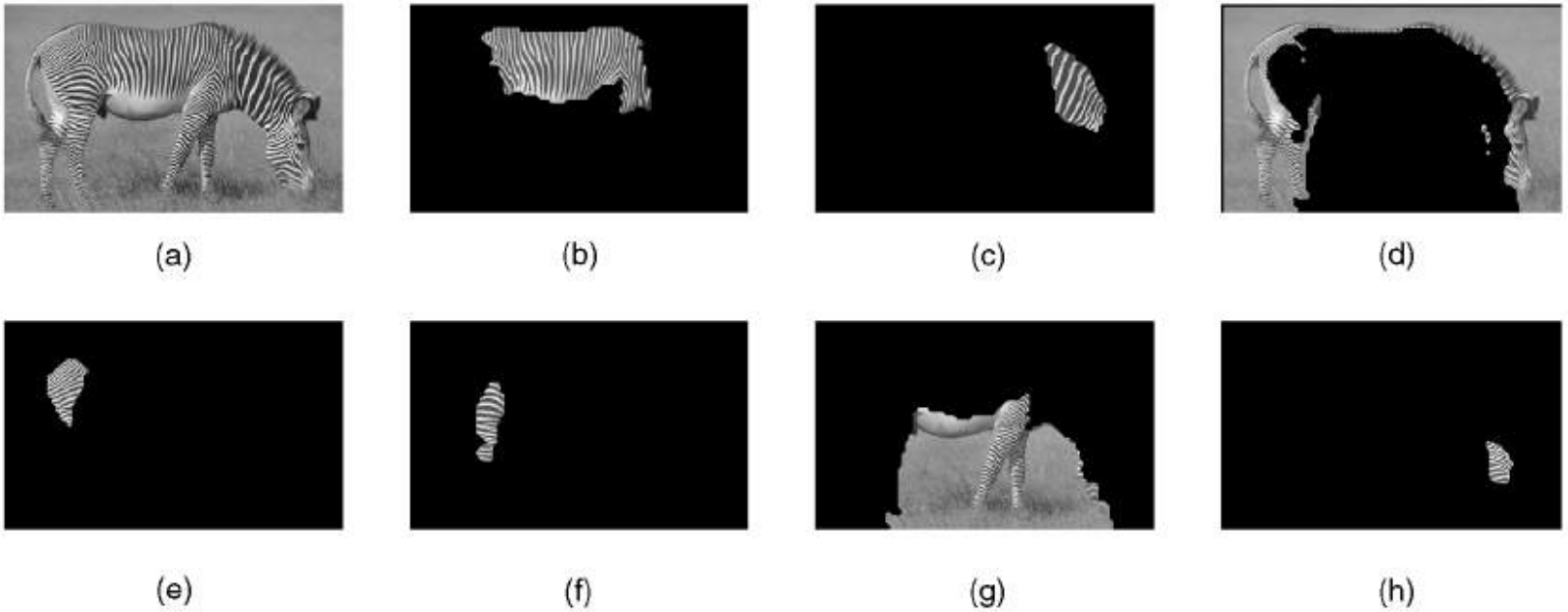


Texture Segmentation Results



Dominant sets

Texture Segmentation Results



NCut

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

S. Rota Bulò and M. Pelillo. Dominant-set clustering: A review. *EJOR* (2017).

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

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

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

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

M. Pelillo. What is a cluster? Perspectives from game theory. *NIPS 2009 Workshop on "Clustering: Science or Art?"* (talk available on videolectures.net).

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

S. Vascon et al., Detecting conversational groups in images and sequences: A robust game-theoretic approach. *CVIU* 2016.

E. Zemene and M. Pelillo, Interactive image segmentation using constrained dominant sets. *ECCV* 2016.