

BBM406

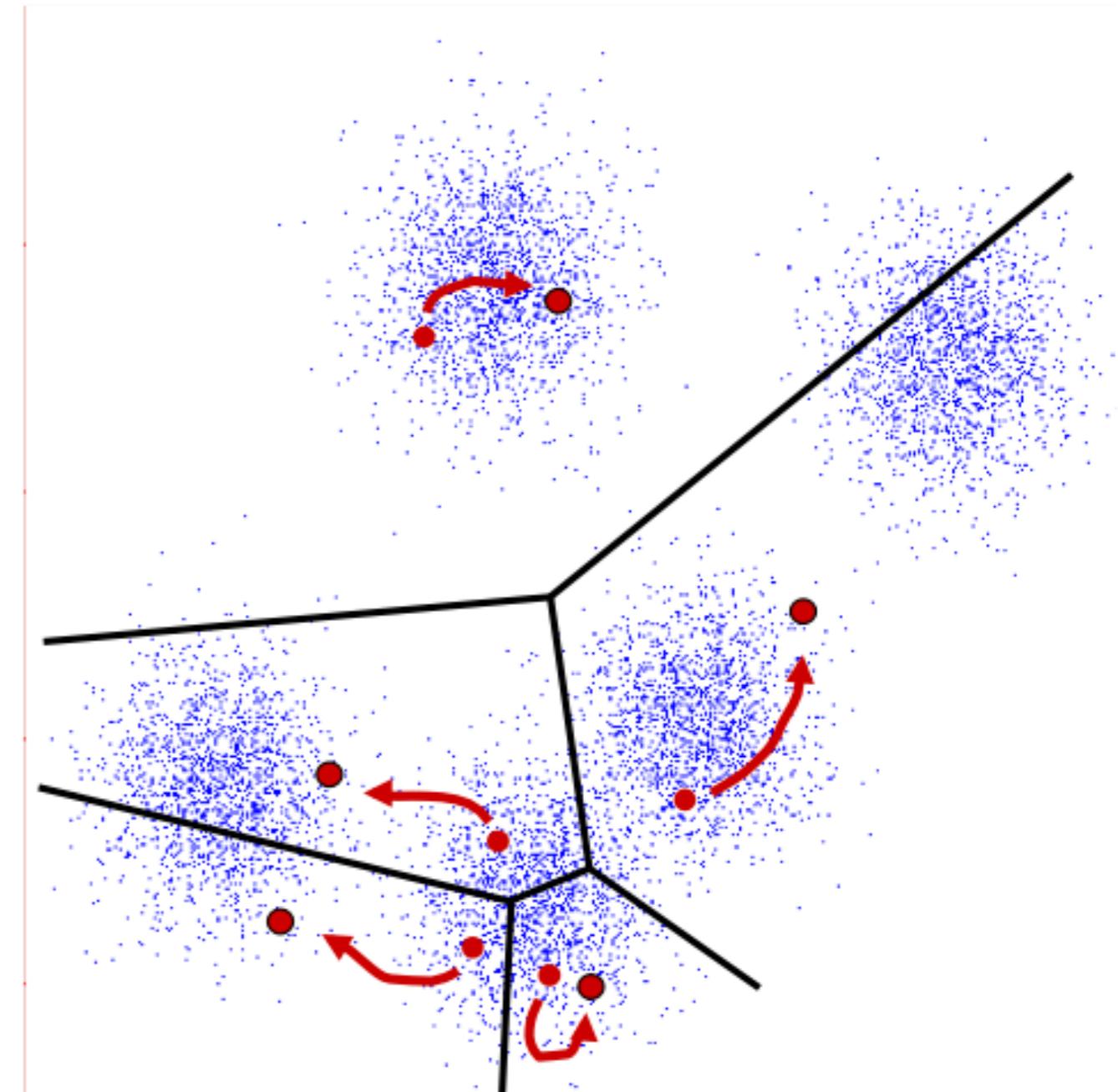
Fundamentals of Machine Learning

Lecture 22:

K-Means Example Applications
Spectral clustering
Hierarchical clustering
What is a good clustering?

Last time... K-Means

- An iterative clustering algorithm
 - Initialize: Pick K random points as cluster centers (means)
 - Alternate:
 - Assign data instances to closest mean
 - Assign each mean to the average of its assigned points
 - Stop when no points' assignments change



Today

- K-Means Example Applications
- Spectral clustering
- Hierarchical clustering
- What is a good clustering?

K-Means Example Applications

Example: K-Means for Segmentation

K=2



Original

Goal of Segmentation
is to partition an image
into regions each of
which has reasonably
homogenous visual
appearance.



Example: K-Means for Segmentation

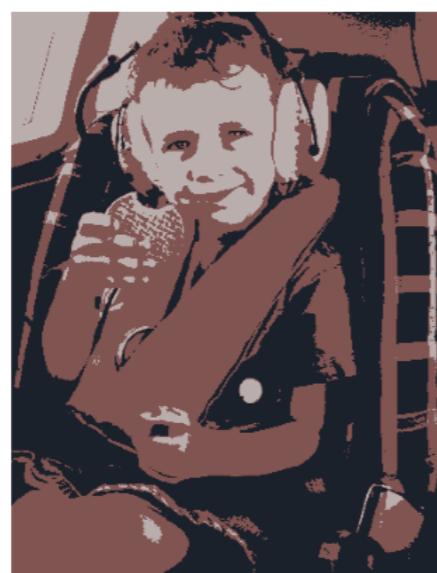
K=2



K=3



Original



Example: K-Means for Segmentation

K=2



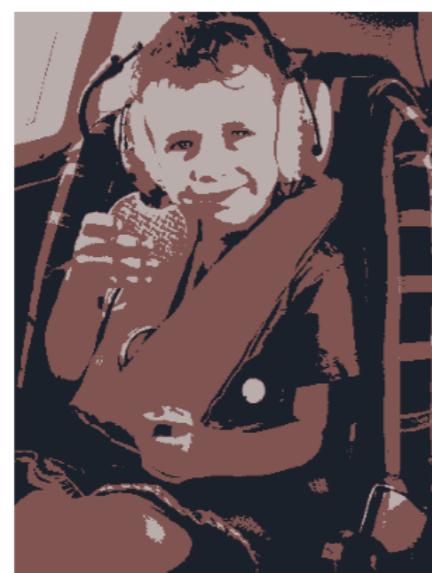
K=3



K=10



Original



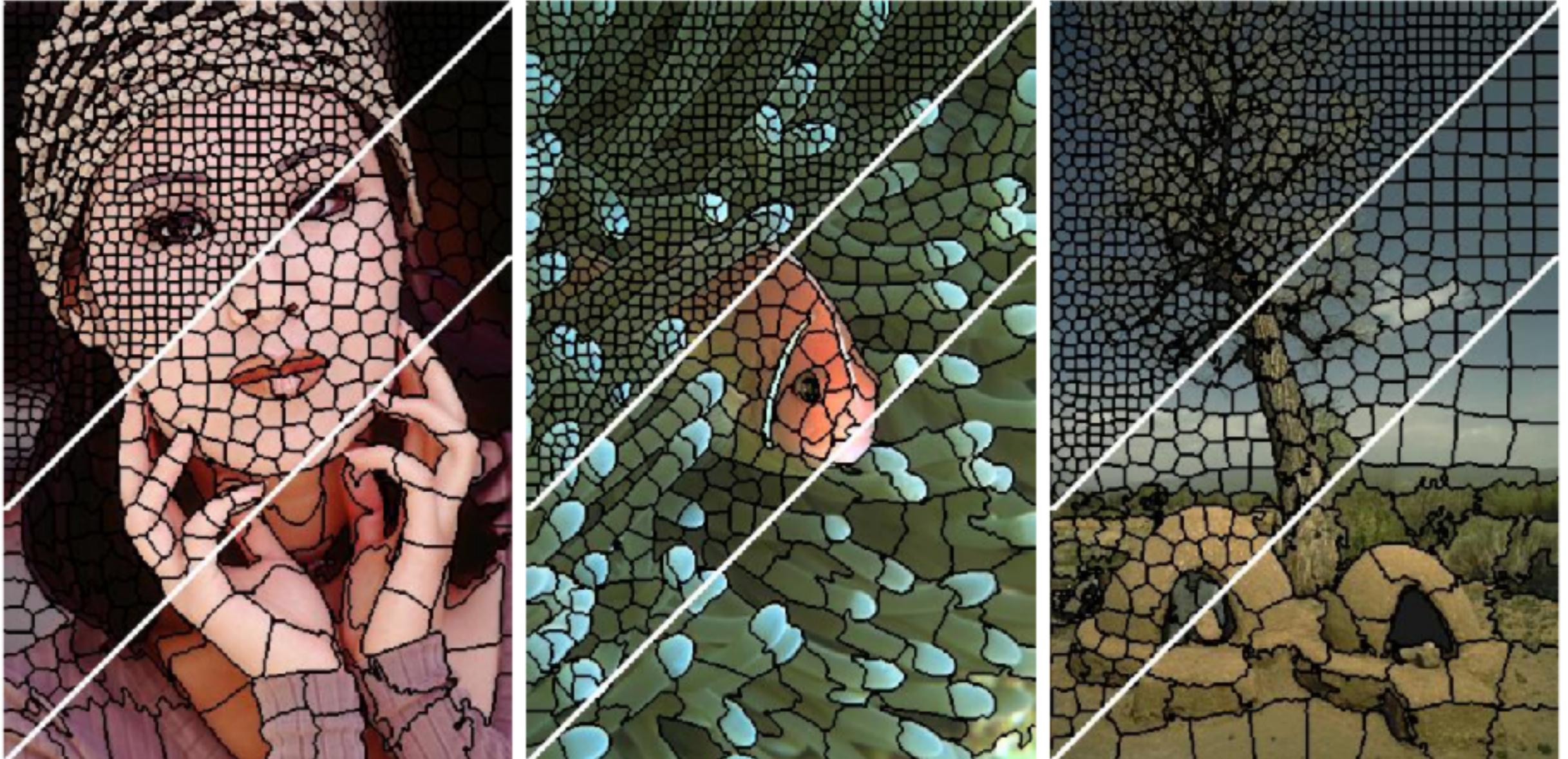
Example: Vector quantization



FIGURE 14.9. Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2×2 block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel

[Figure from Hastie et al. book]

Example: Simple Linear Iterative Clustering (SLIC) superpixels



$$\Psi(x, y) = \begin{bmatrix} \lambda x \\ \lambda y \\ I(x, y) \end{bmatrix} \quad \lambda: \text{spatial regularization parameter}$$

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE T-PAMI, 2012

Bag of Words model

the world of **TOTAL**



all about the company

Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

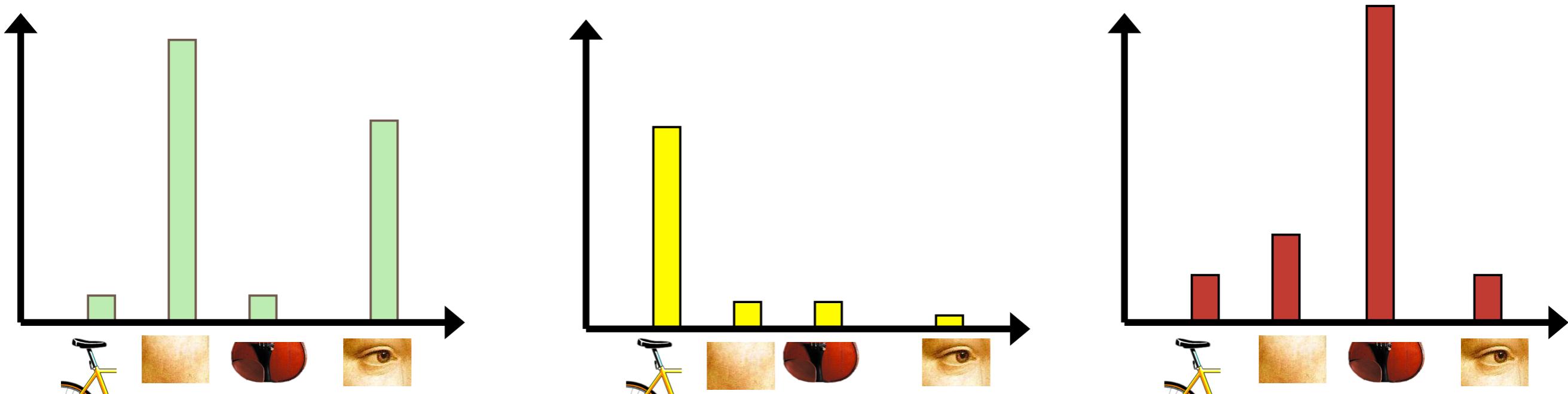
Our growing specialty chemicals sector adds balance and profit to the core energy business.

► All About The Company

- [Global Activities](#)
- [Corporate Structure](#)
- [TOTAL's Story](#)
- [Upstream Strategy](#)
- [Downstream Strategy](#)
- [Chemicals Strategy](#)
- [TOTAL Foundation](#)
- [Homepage](#)



aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
...	
Zaire	0



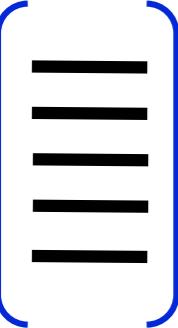
Object

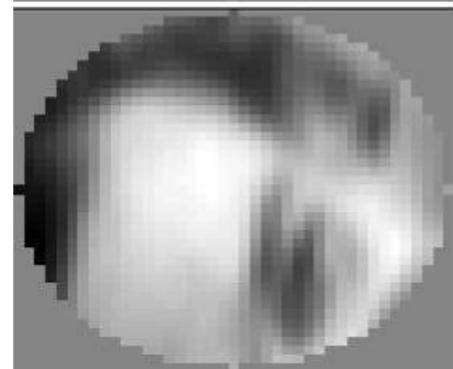


Bag of 'words'

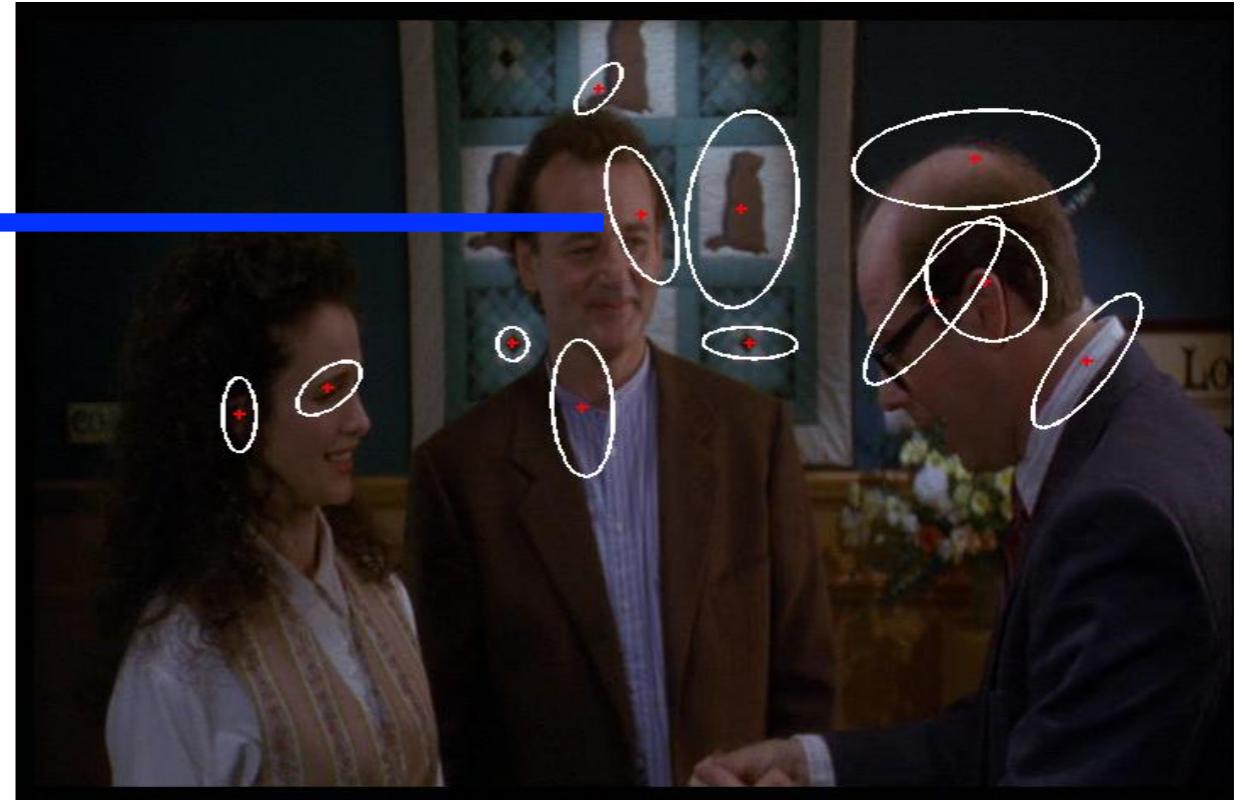


Interest Point Features

[]
Compute SIFT descriptor
[Lowe'99]



Normalize patch



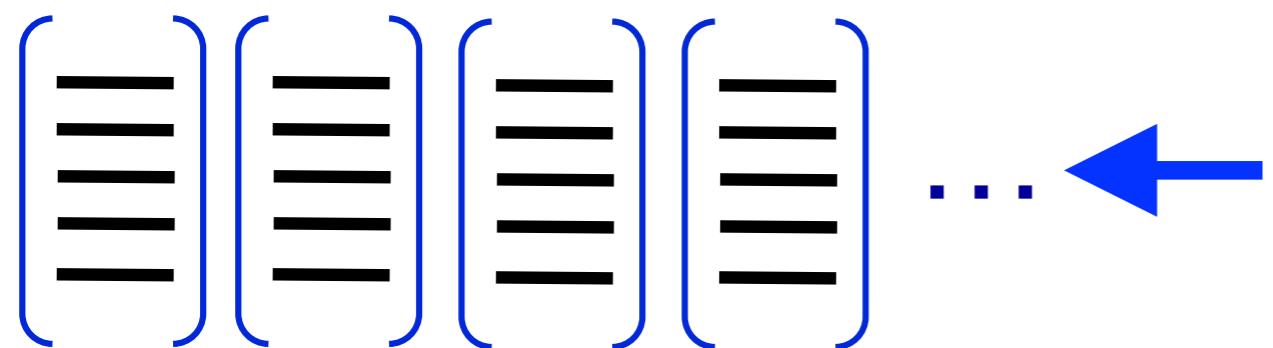
Detect patches

[Mikojaczyk and Schmid '02]

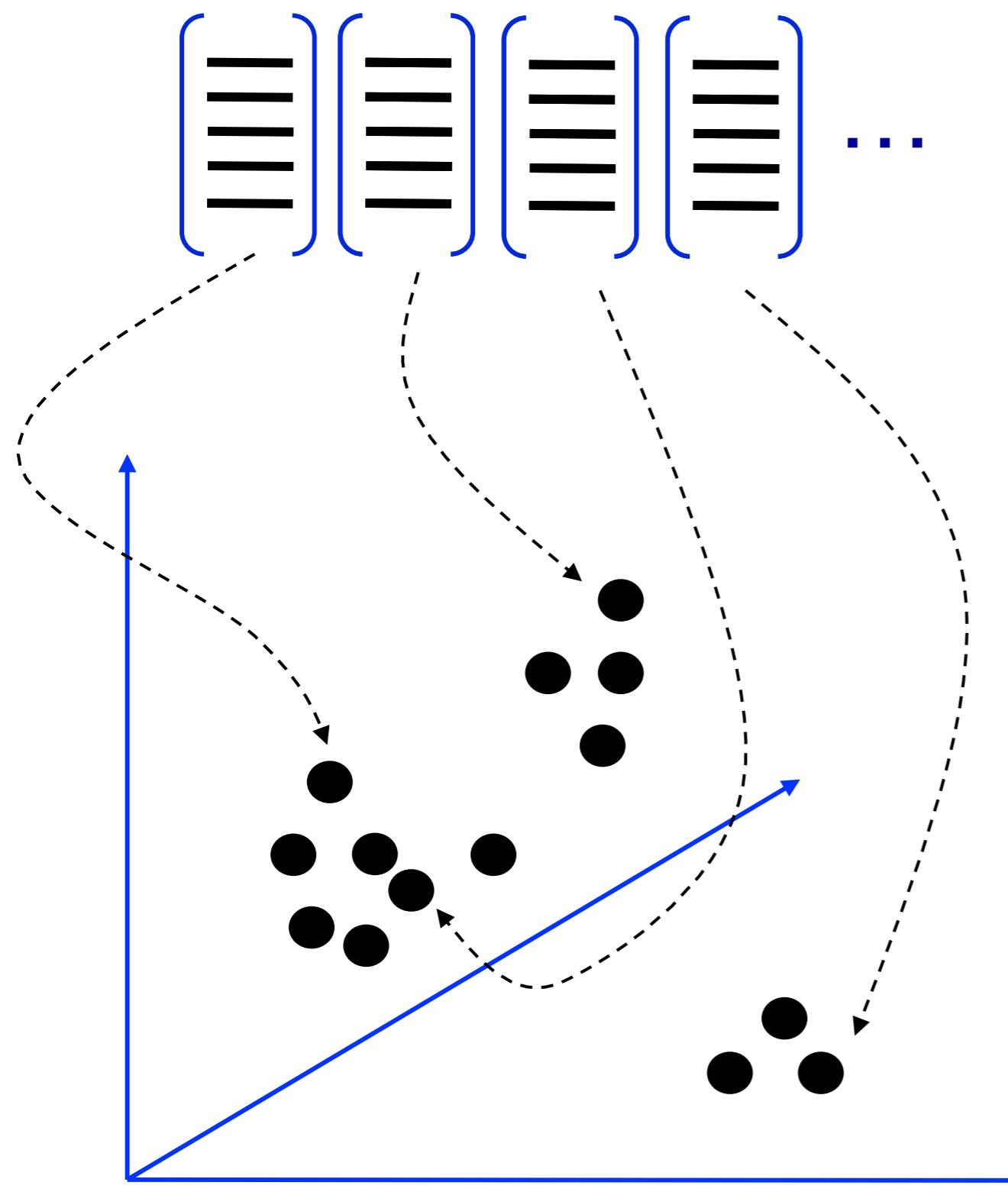
[Matas et al. '02]

[Sivic et al. '03]

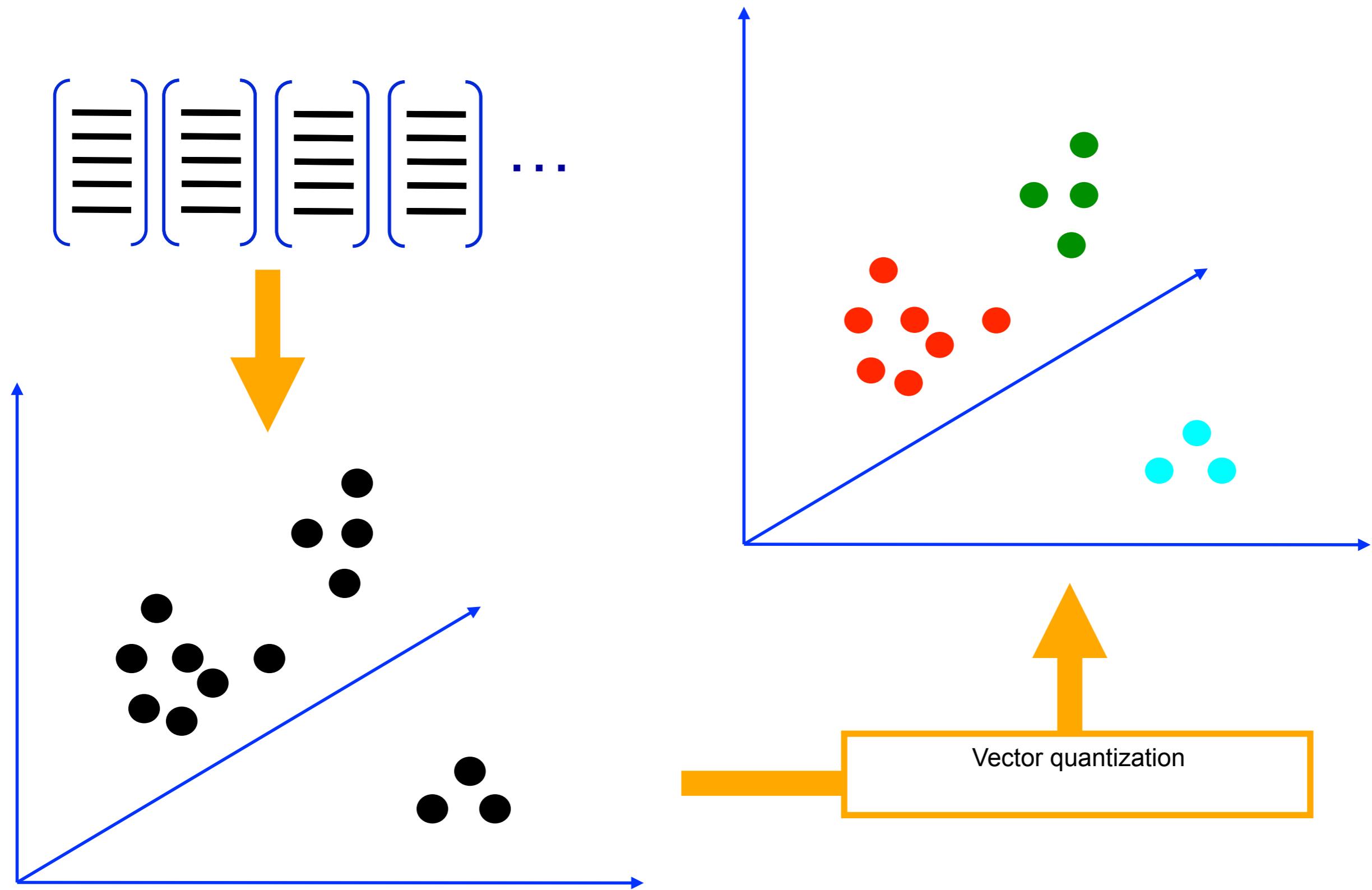
Patch Features



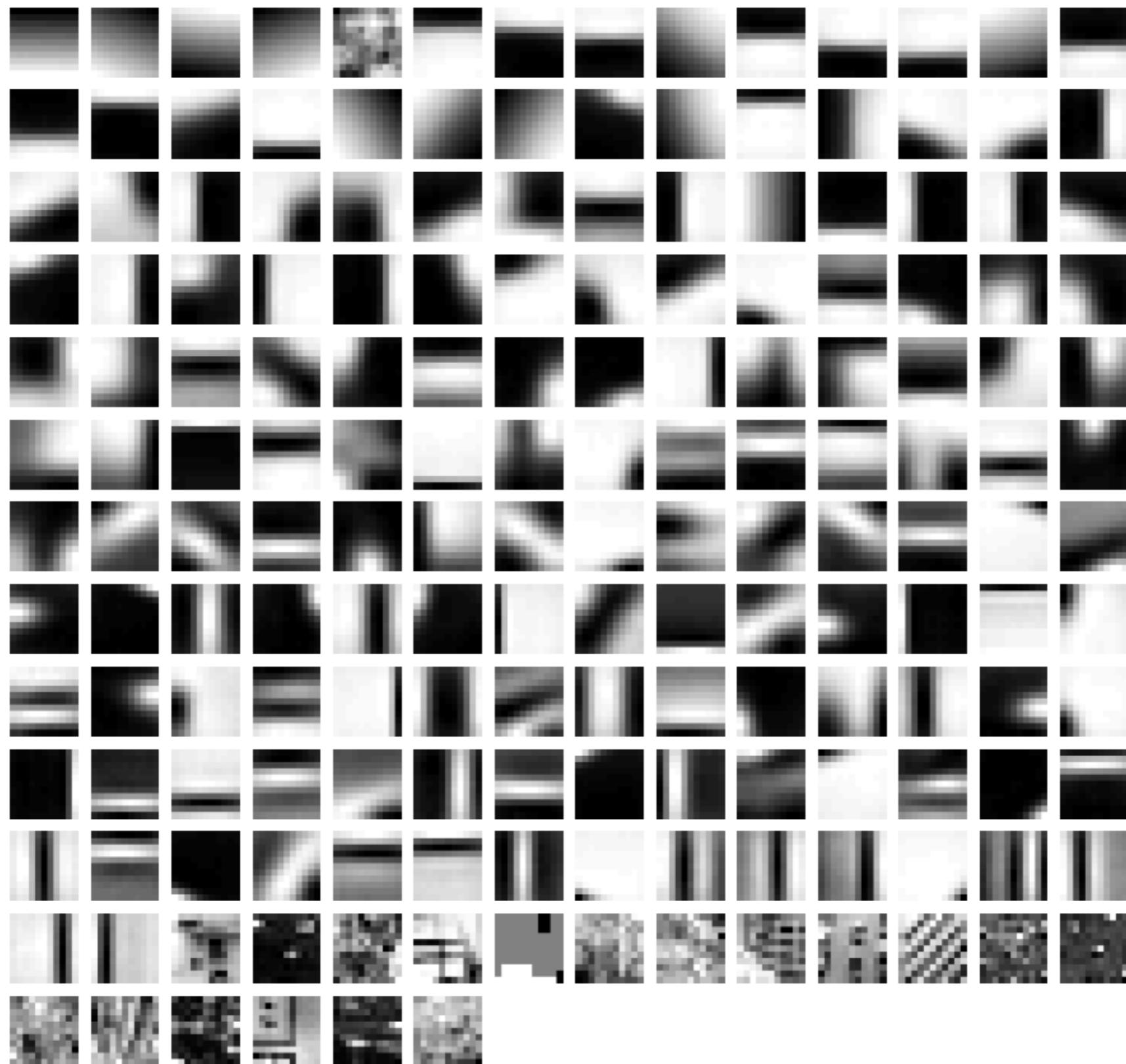
Dictionary Formation



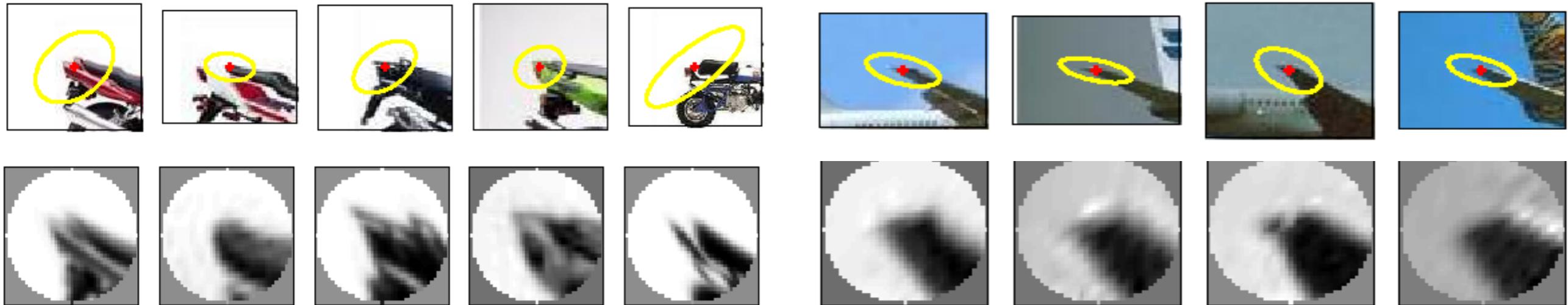
Clustering (usually K-means)



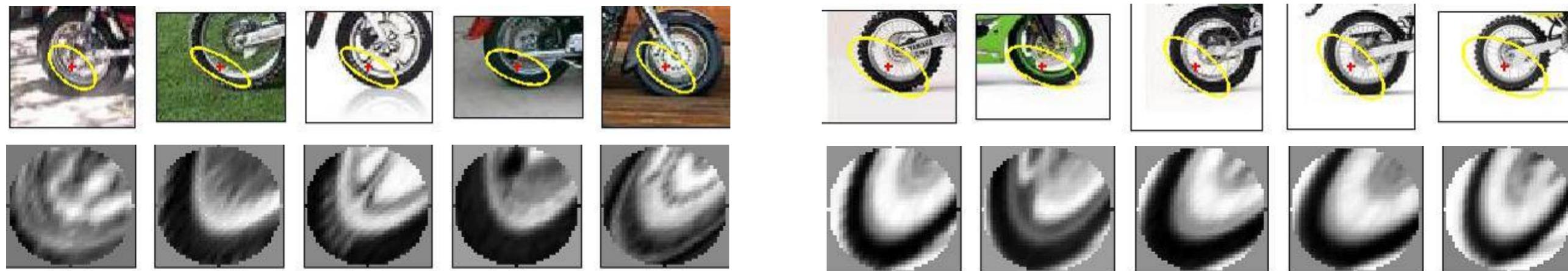
Clustered Image Patches



Visual synonyms and polysemy

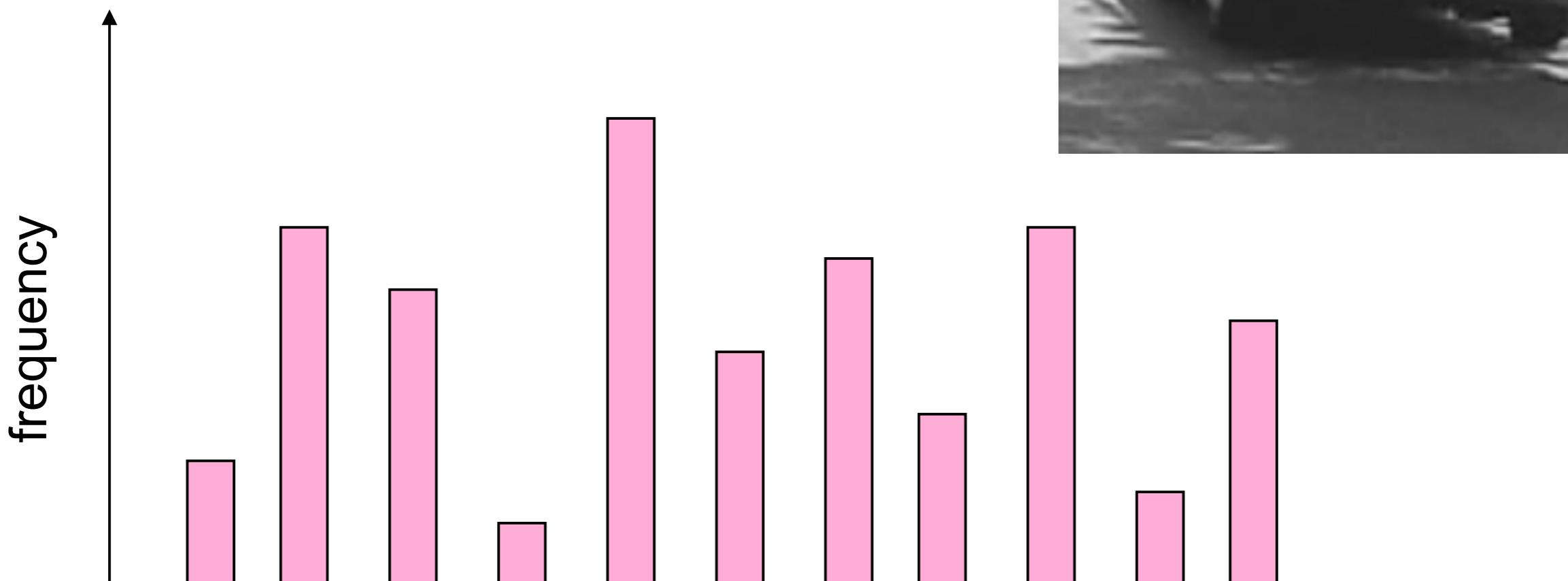


Visual Polysemy. Single visual word occurring on different (but locally similar) parts on different object categories.



Visual Synonyms. Two different visual words representing a similar part of an object (wheel of a motorbike).

Image Representation



...

codewords

Spectral clustering

Graph-Theoretic Clustering

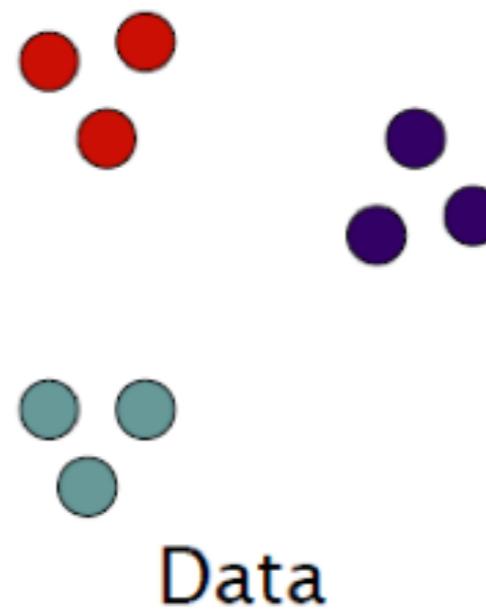
Goal: Given data points X_1, \dots, X_n and similarities $W(X_i, X_j)$, partition the data into groups so that points in a group are similar and points in different groups are dissimilar.

Similarity Graph: $G(V, E, W)$

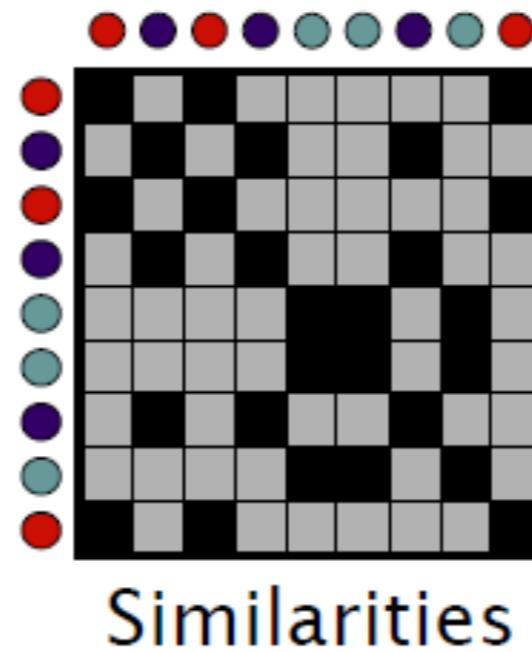
V – Vertices (Data points)

E – Edge if similarity > 0

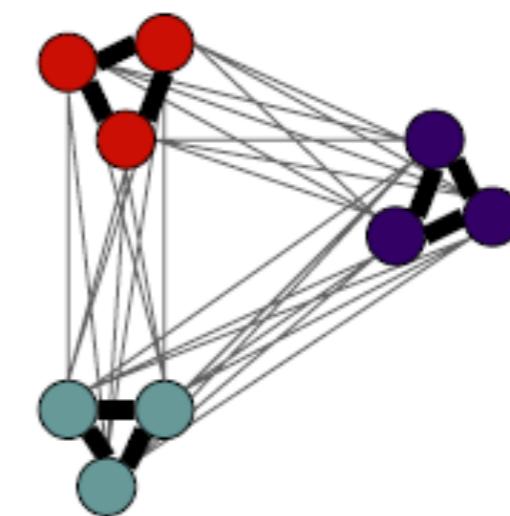
W - Edge weights (similarities)



Data



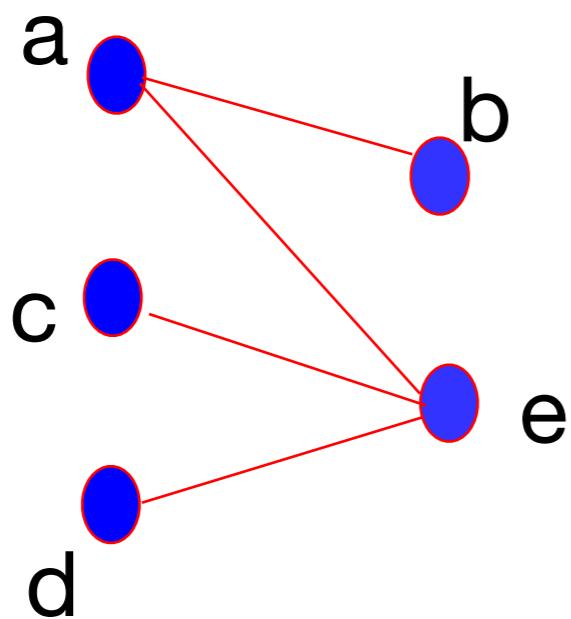
Similarities



Similarity graph

Partition the graph so that edges within a group have large weights and edges across groups have small weights.

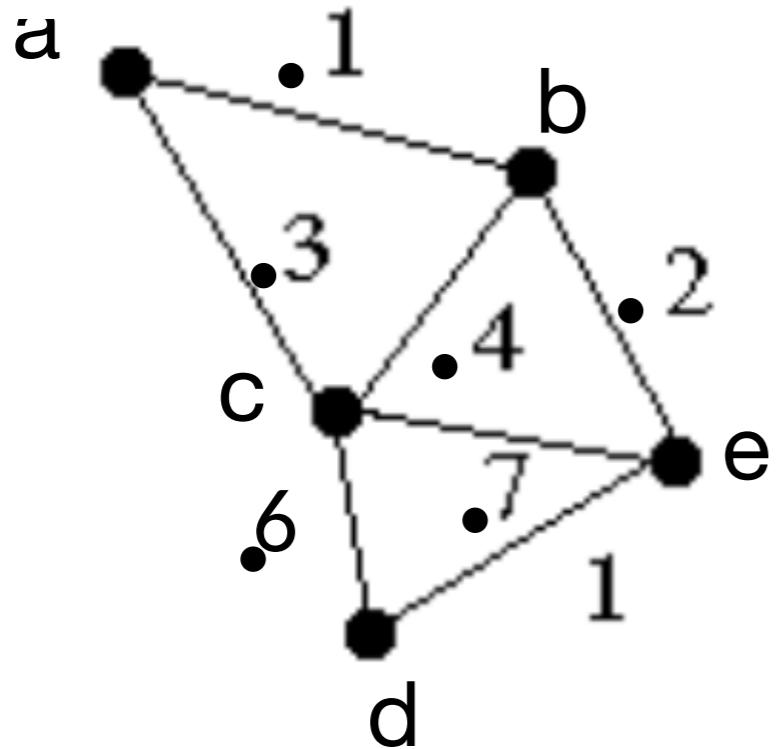
Graphs Representations



	a	b	c	d	e
a	0	1	0	0	1
b	1	0	0	0	0
c	0	0	0	0	1
d	0	0	0	0	1
e	1	0	1	1	0

Adjacency Matrix

A Weighted Graph and its Representation



Affinity Matrix

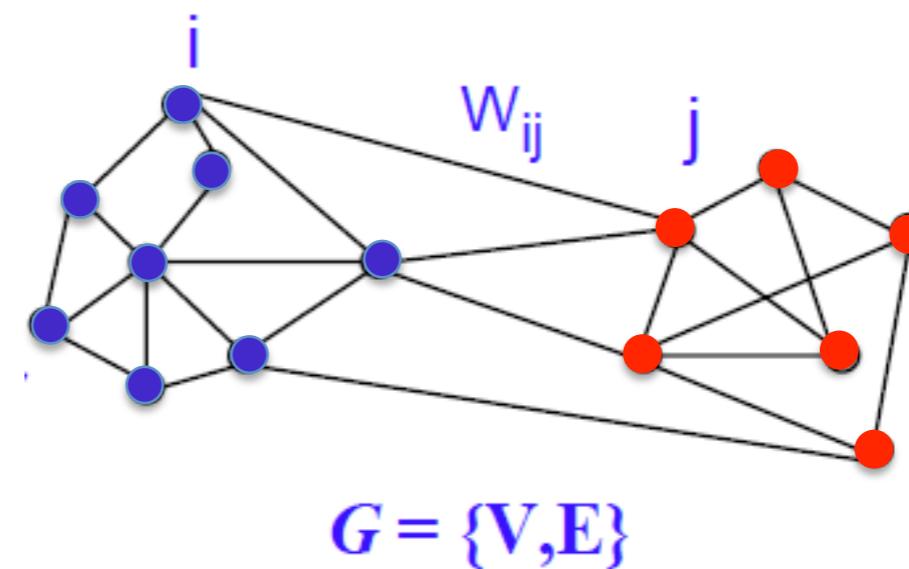
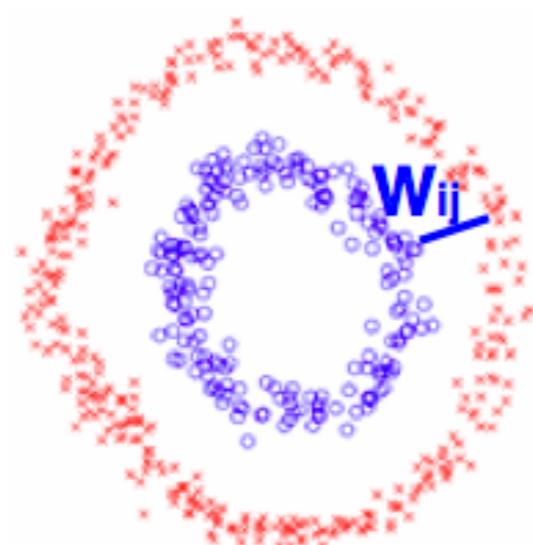
$$W = \begin{bmatrix} 1 & .1 & .3 & 0 & 0 \\ .1 & 1 & .4 & 0 & .2 \\ .3 & .4 & 1 & .6 & .7 \\ 0 & 0 & .6 & 1 & 1 \\ 0 & .2 & .7 & 1 & 1 \end{bmatrix}$$

W_{ij} : probability that i & j belong to the same cluster

Similarity graph construction

- Similarity Graphs: Model local neighborhood relations between data points
- E.g. epsilon-NN

$$W_{ij} = \begin{cases} 1 & \|x_i - x_j\| \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$
 → Controls size of neighborhood
or mutual k-NN graph ($W_{ij} = 1$ if x_i or x_j is k nearest neighbor of the other)

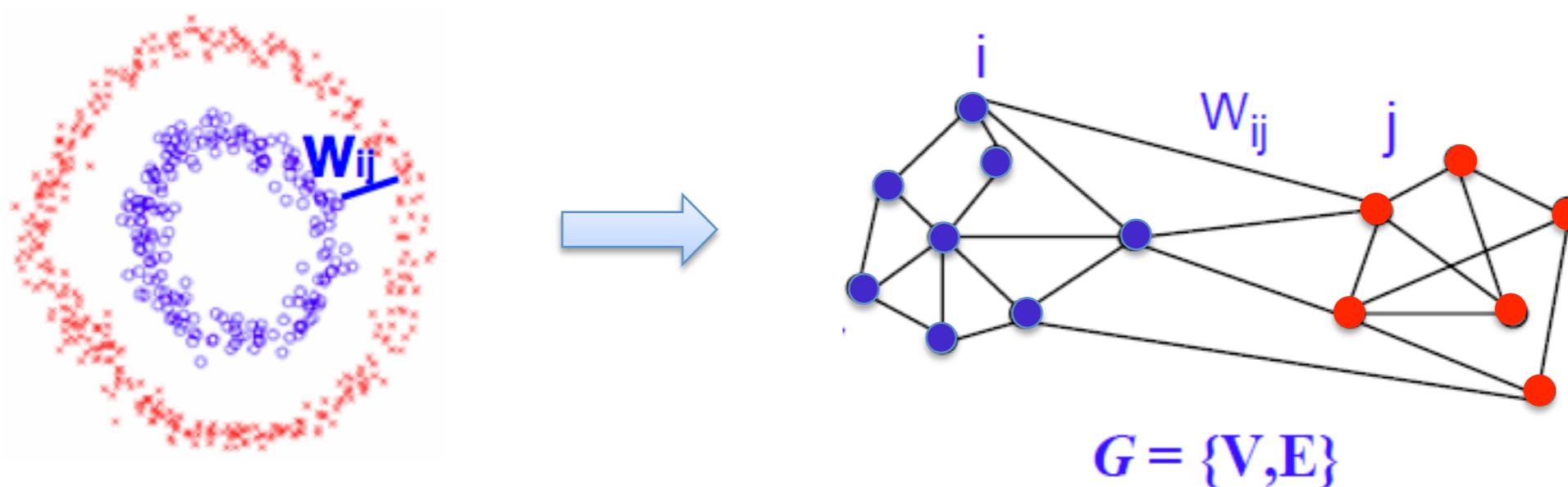


Similarity graph construction

- Similarity Graphs: Model local neighborhood relations between data points
- E.g. Gaussian kernel similarity function

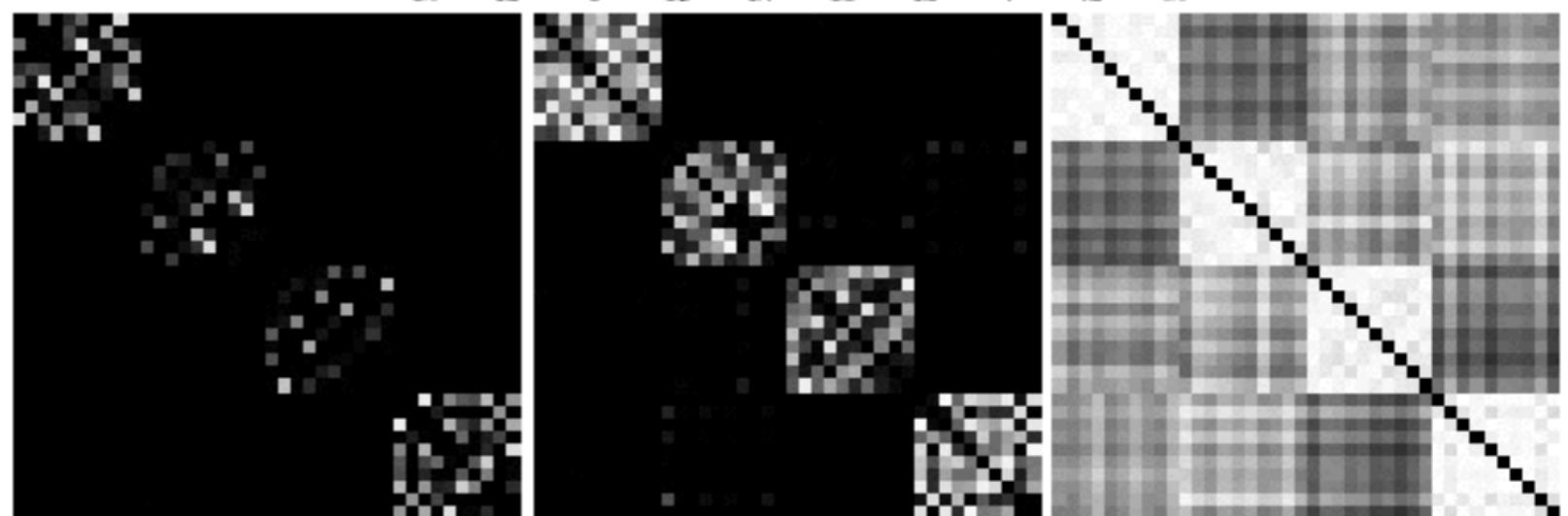
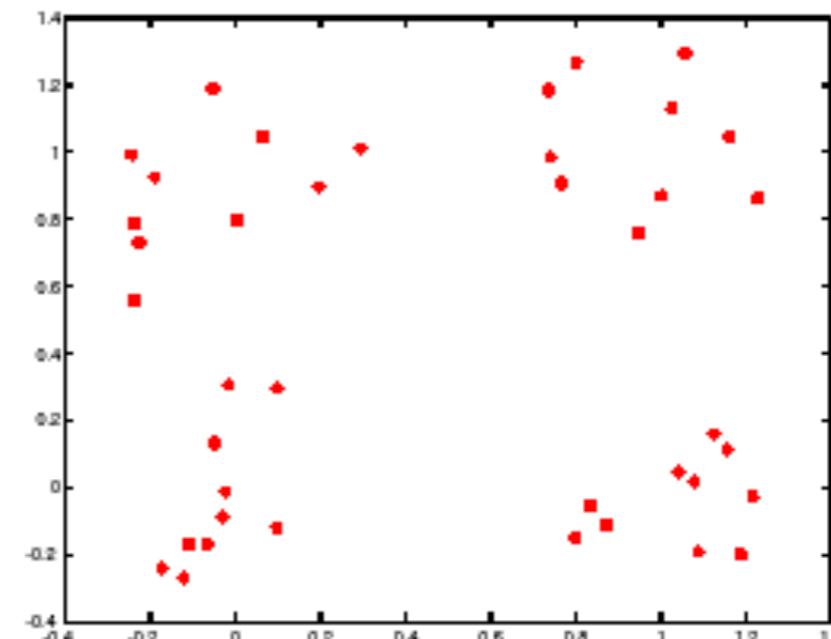
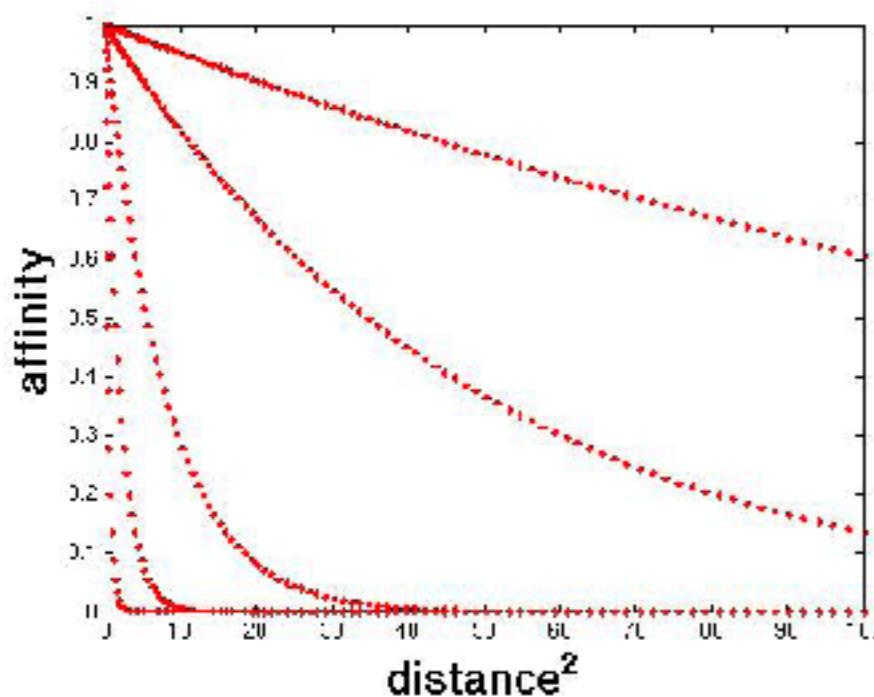
$$W_{ij} = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

→ Controls size of neighborhood



Scale affects affinity

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



Feature grouping by “relocalisation” of eigenvectors of the proximity matrix

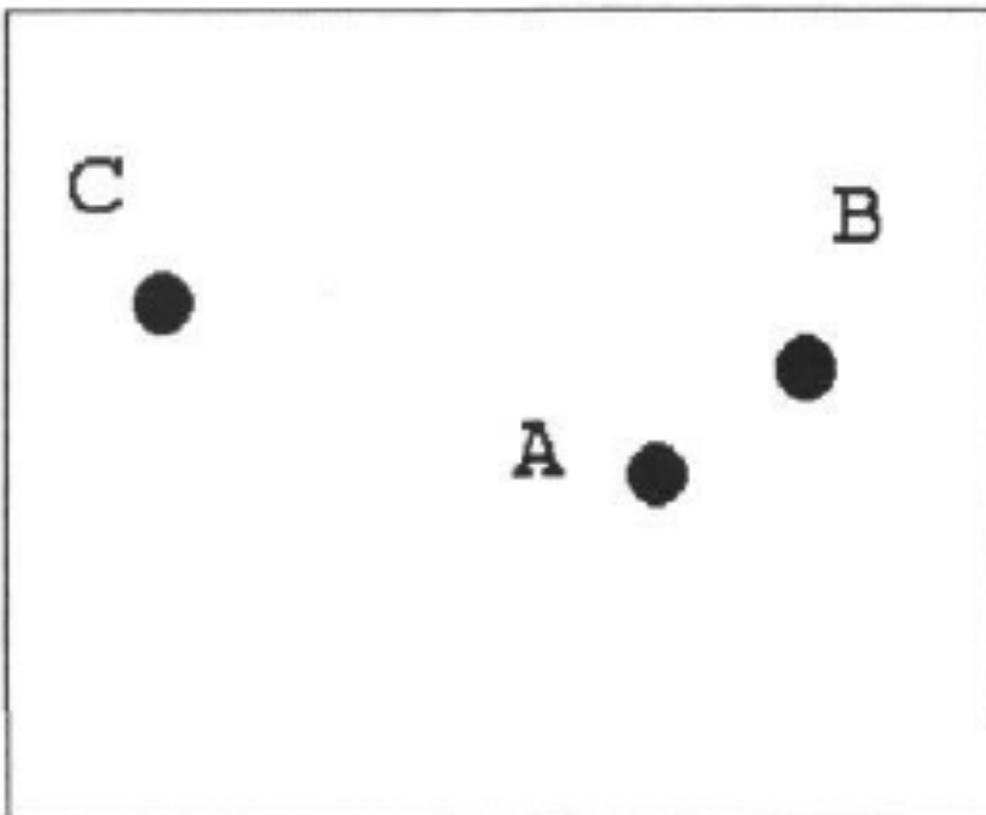
British Machine Vision Conference, pp. 103-108, 1990

Guy L. Scott

Robotics Research Group
Department of Engineering Science
University of Oxford

H. Christopher Longuet-Higgins

University of Sussex
Falmer
Brighton



Three points in feature space

$$W_{ij} = \exp(-\| z_i - z_j \|^2 / s^2)$$

With an appropriate s

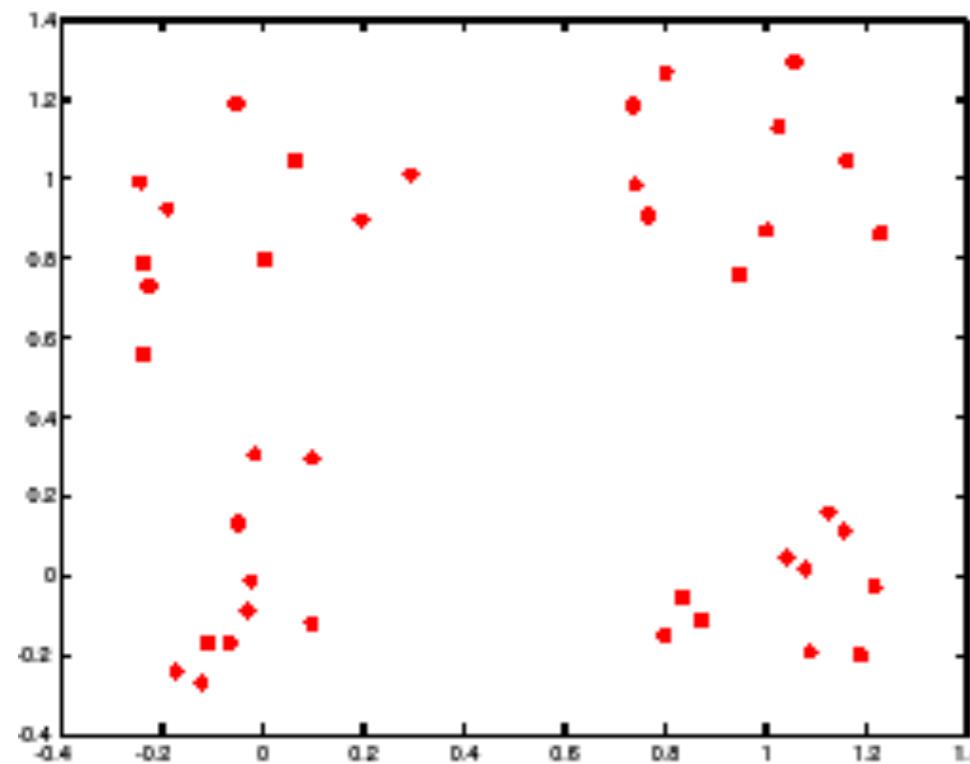
	A	B	C
A	1.00	0.63	0.03
B	0.63	1.00	0.0
C	0.03	0.0	1.00

The eigenvectors of W are:

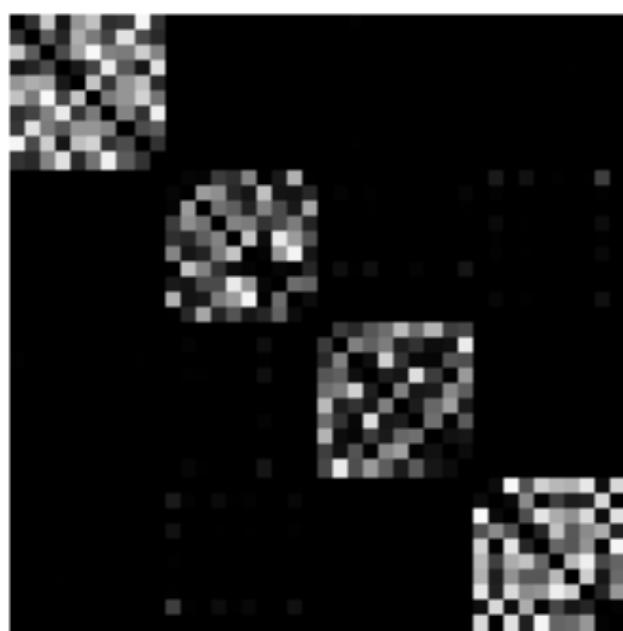
	E_1	E_2	E_3
Eigenvalues	1.63	1.00	0.37
A	-0.71	-0.01	0.71
B	-0.71	-0.05	-0.71
C	-0.04	1.00	-0.03

The first 2 eigenvectors group the points as desired...

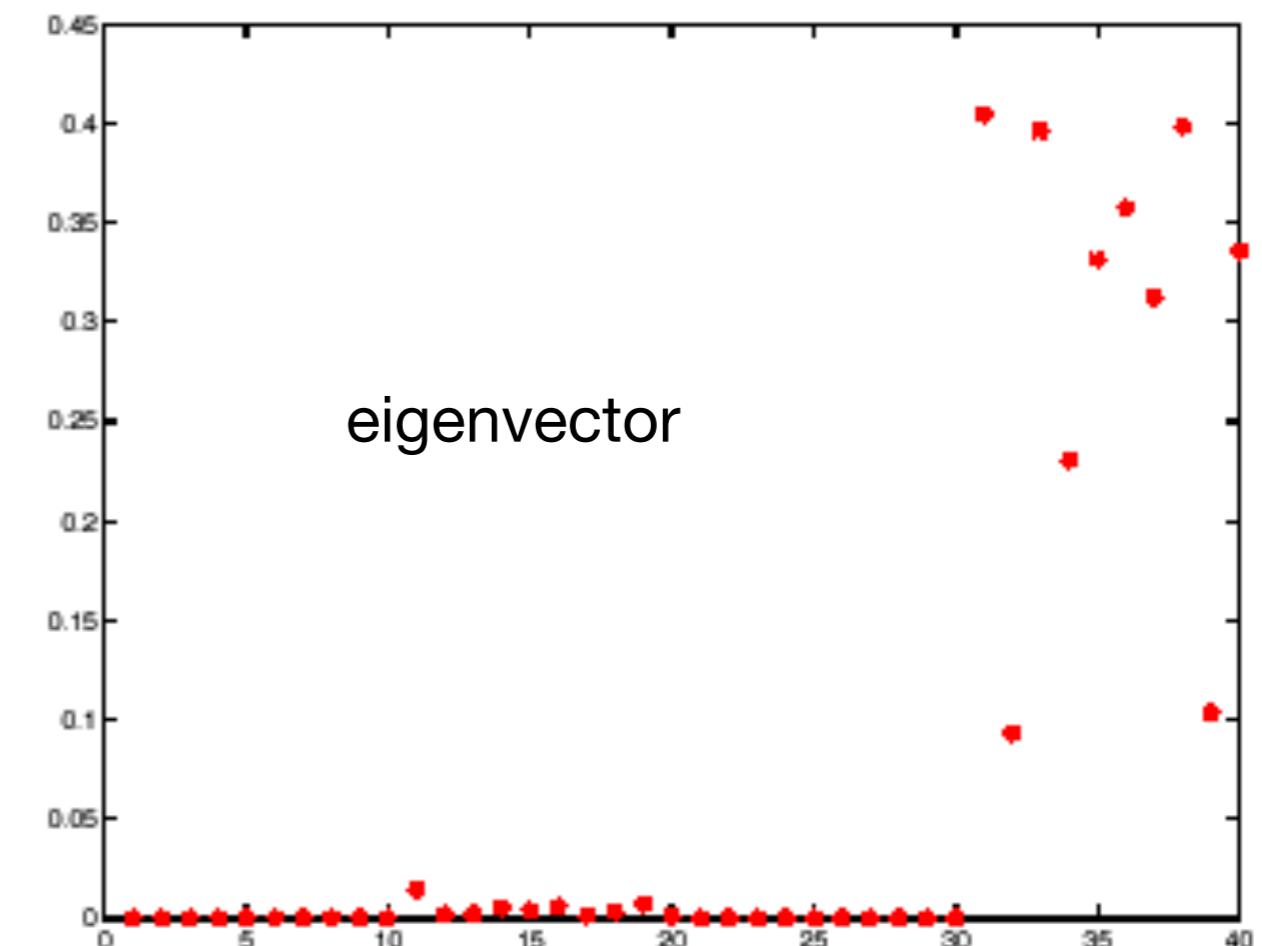
Example eigenvector



points

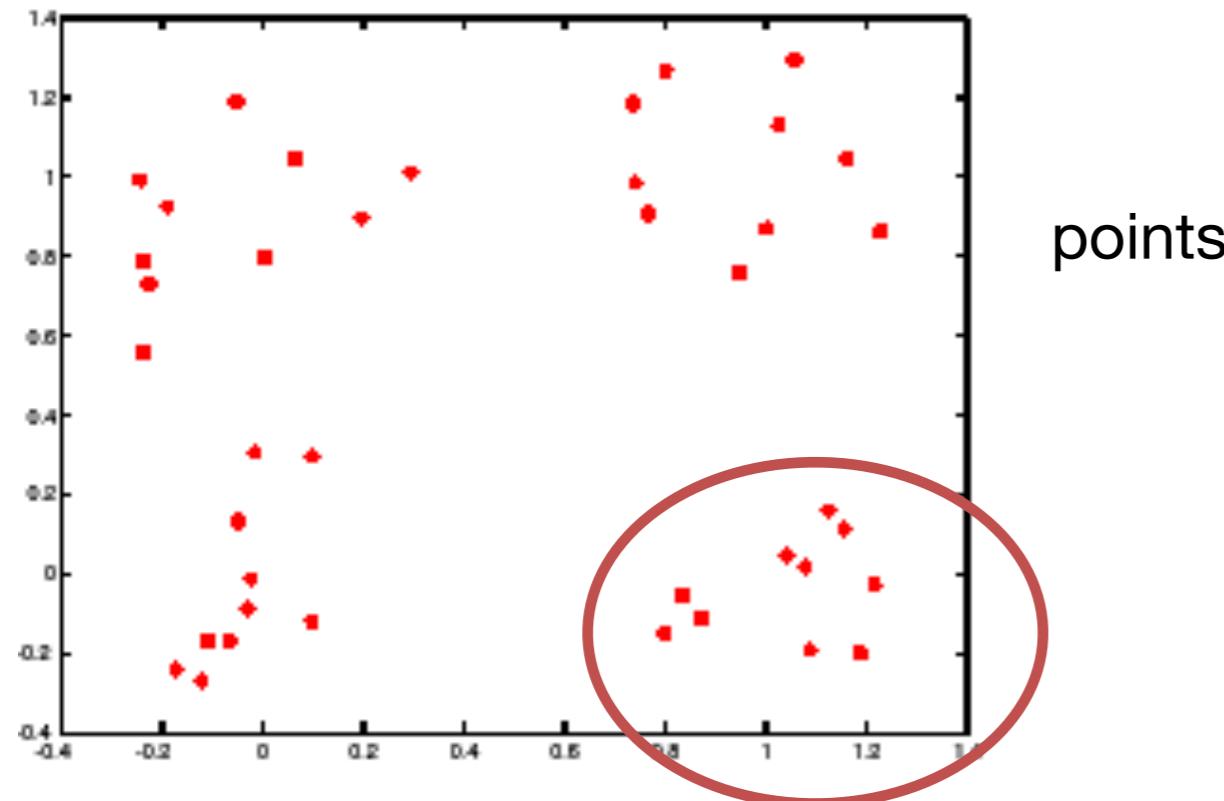


Affinity matrix

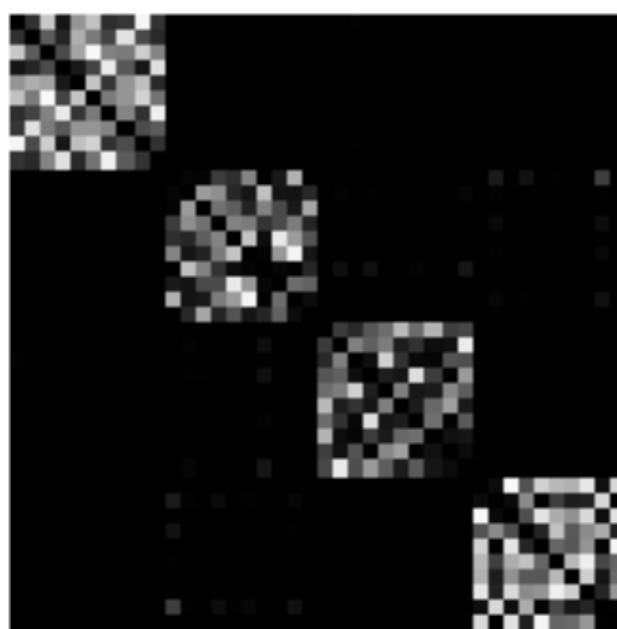


eigenvector

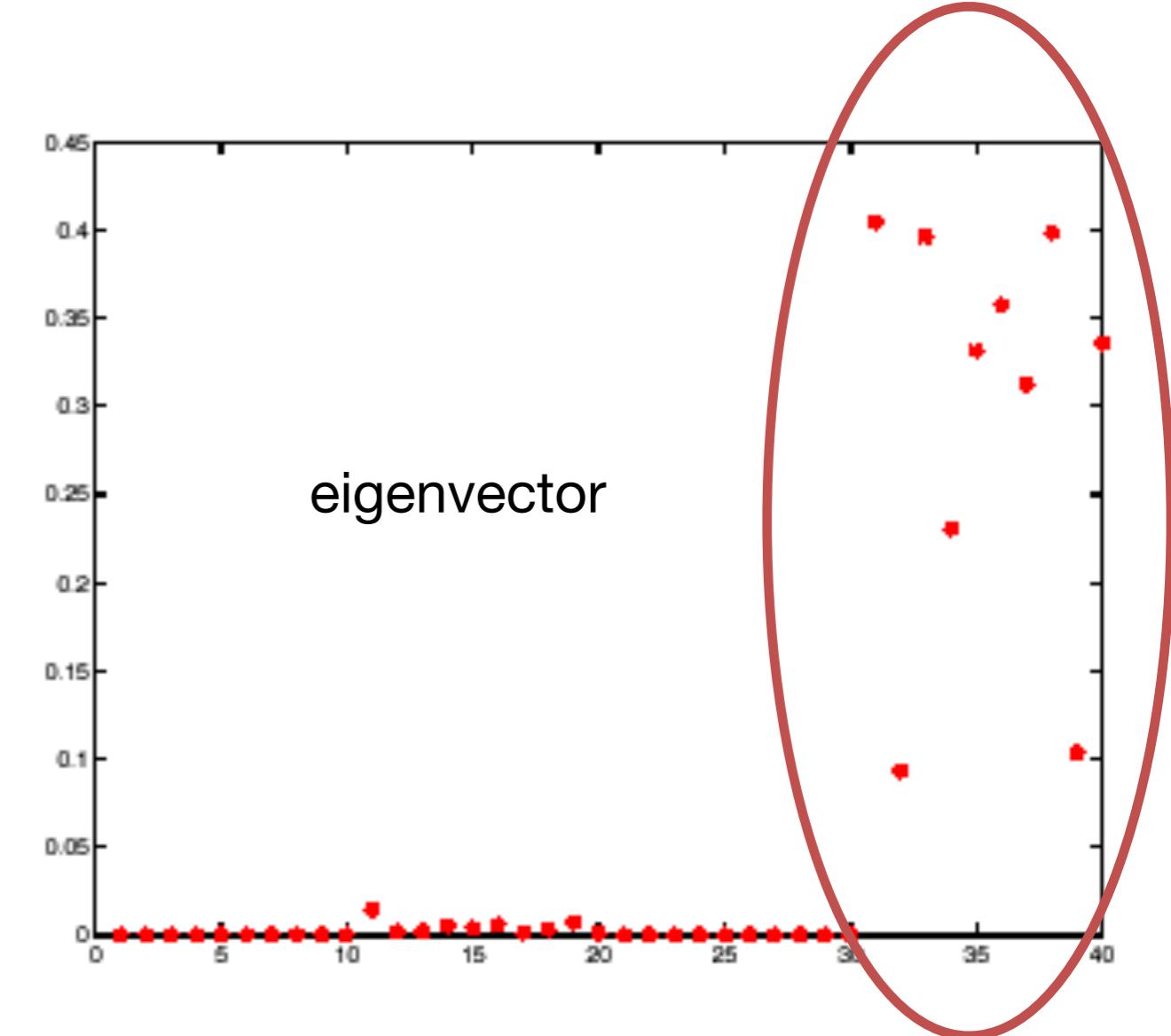
Example eigenvector



points

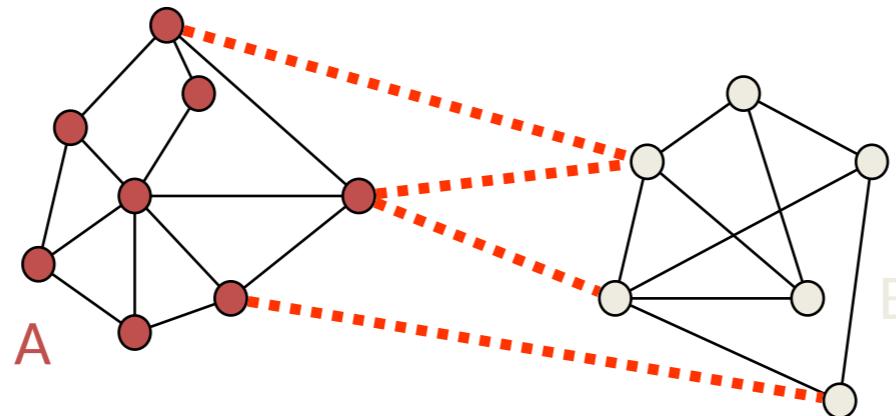


Affinity matrix



eigenvector

Graph cut

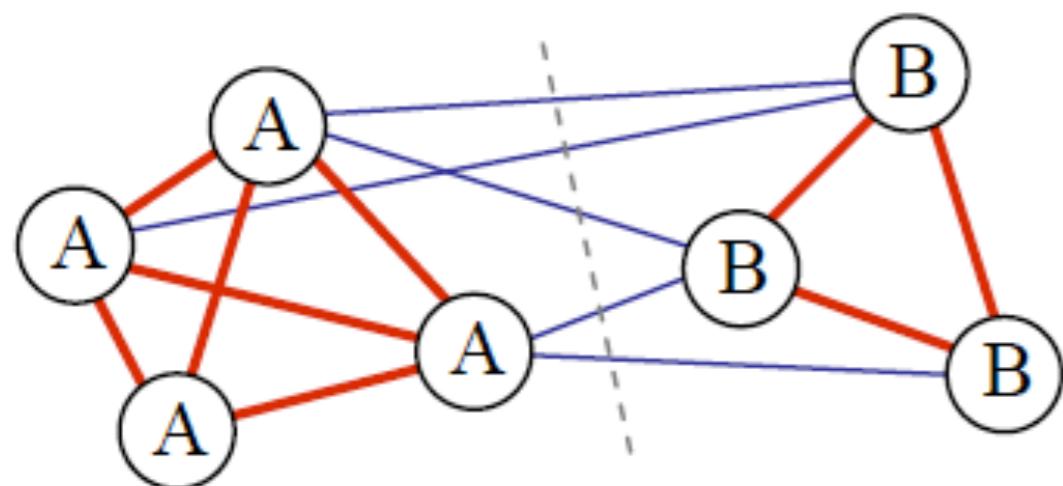


- Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges
- A graph cut gives us a partition (clustering)
 - What is a “good” graph cut and how do we find one?

Minimum cut

- A cut of a graph G is the set of edges S such that removal of S from G disconnects G .

Cut: sum of the weight of the cut edges:



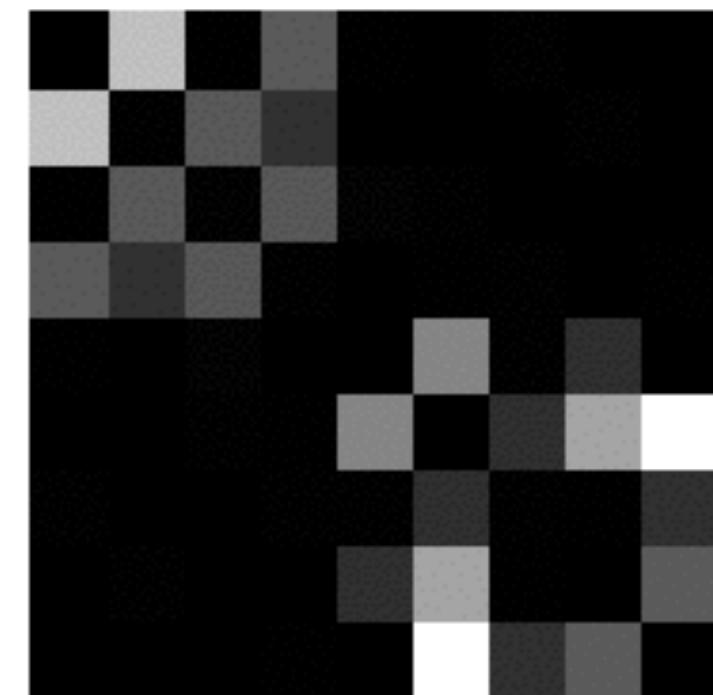
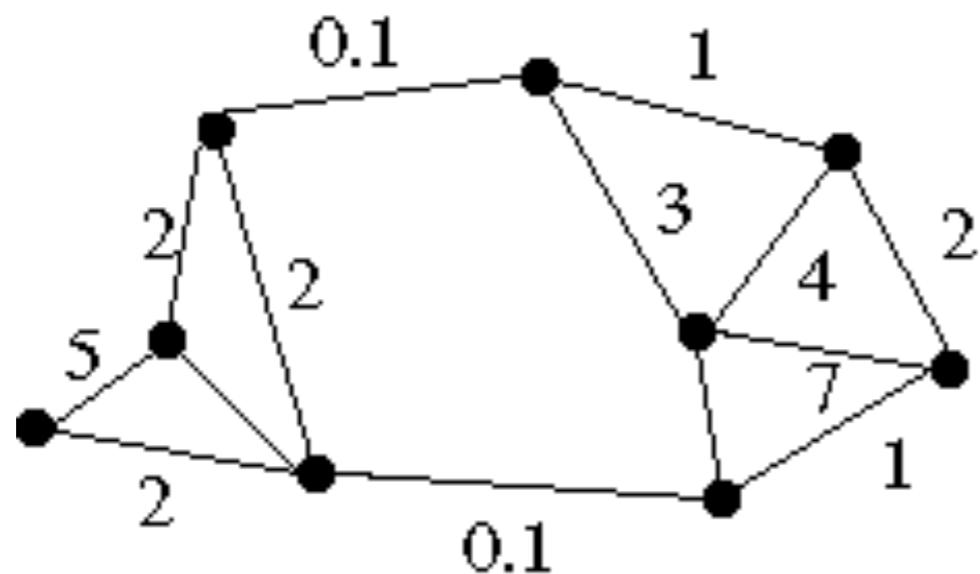
$$\text{cut}(A, B) = \sum_{u \in A, v \in B} W(u, v),$$

$$\text{with } A \cap B = \emptyset$$

Minimum cut

- We can do clustering by finding the **minimum cut** in a graph
 - Efficient algorithms exist for doing this

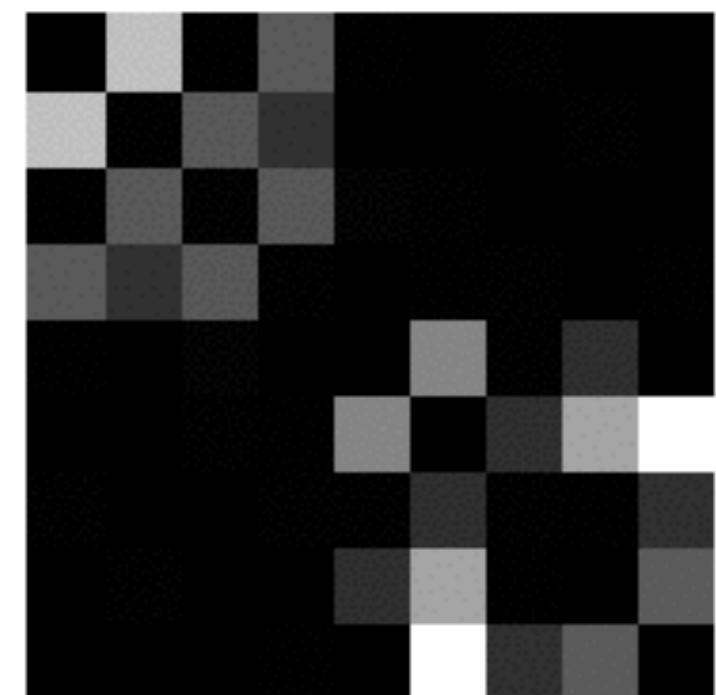
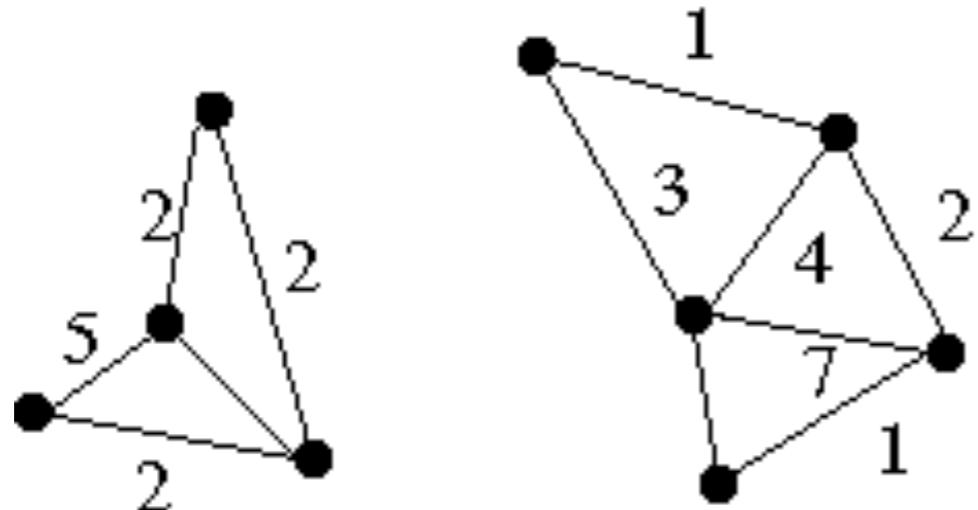
Minimum cut example



Minimum cut

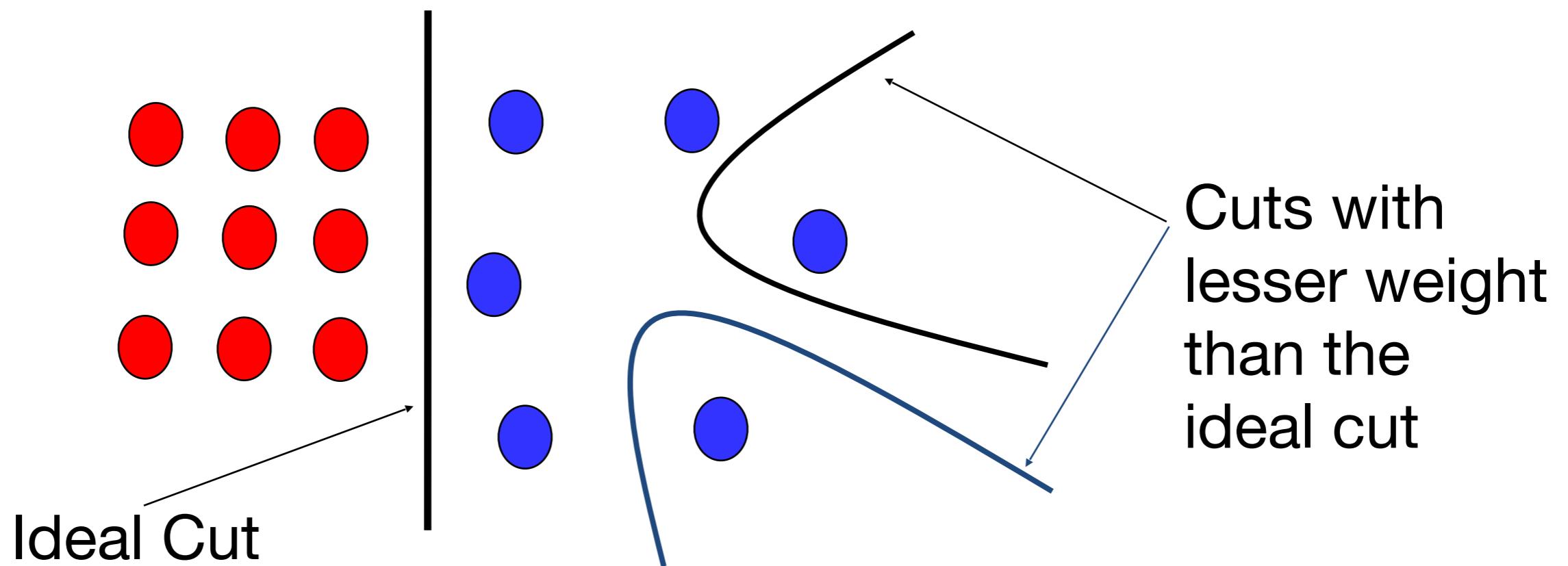
- We can do segmentation by finding the **minimum cut** in a graph
 - Efficient algorithms exist for doing this

Minimum cut example



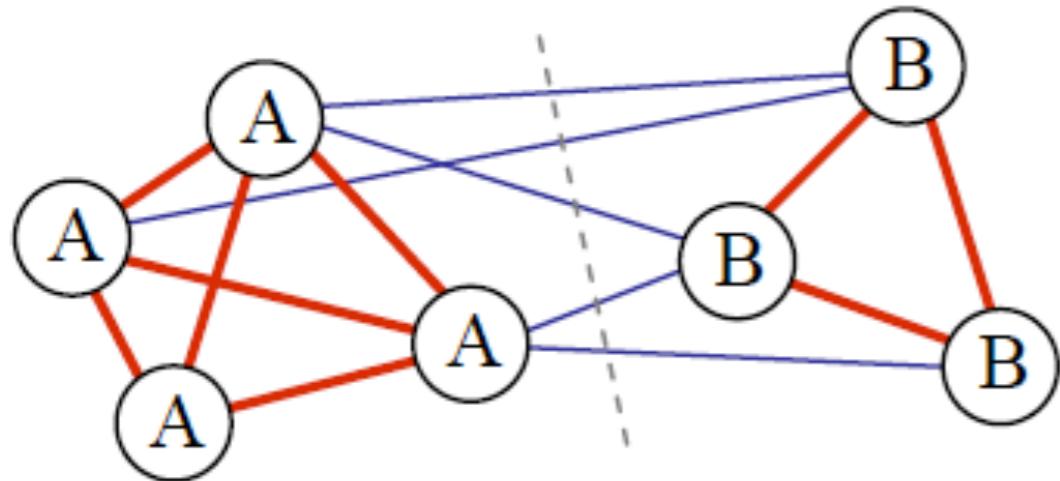
Drawbacks of Minimum cut

- Weight of cut is directly proportional to the number of edges in the cut.



Normalized cuts

Write graph as V , one cluster as A and the other as B



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

$cut(A, B)$ is sum of weights with one end in A and one end in B

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

with $A \cap B = \emptyset$

$assoc(A, V)$ is sum of all edges with one end in A .

$$assoc(A, B) = \sum_{u \in A, v \in B} W(u, v)$$

A and B not necessarily disjoint

Normalized cut

- Let W be the adjacency matrix of the graph
- Let D be the diagonal matrix with diagonal entries
$$D(i, i) = \sum_j W(i, j)$$
- Then the normalized cut cost can be written as
$$\frac{y^T (D - W)y}{y^T D y}$$

where y is an indicator vector whose value should be 1 in the i -th position if the i -th feature point belongs to A and a negative constant otherwise

Normalized cut

- Finding the exact minimum of the normalized cut cost is NP-complete, but if we *relax* y to take on arbitrary values, then we can minimize the relaxed cost by solving the **generalized eigenvalue problem**

$$(D - W)y = \lambda D y$$

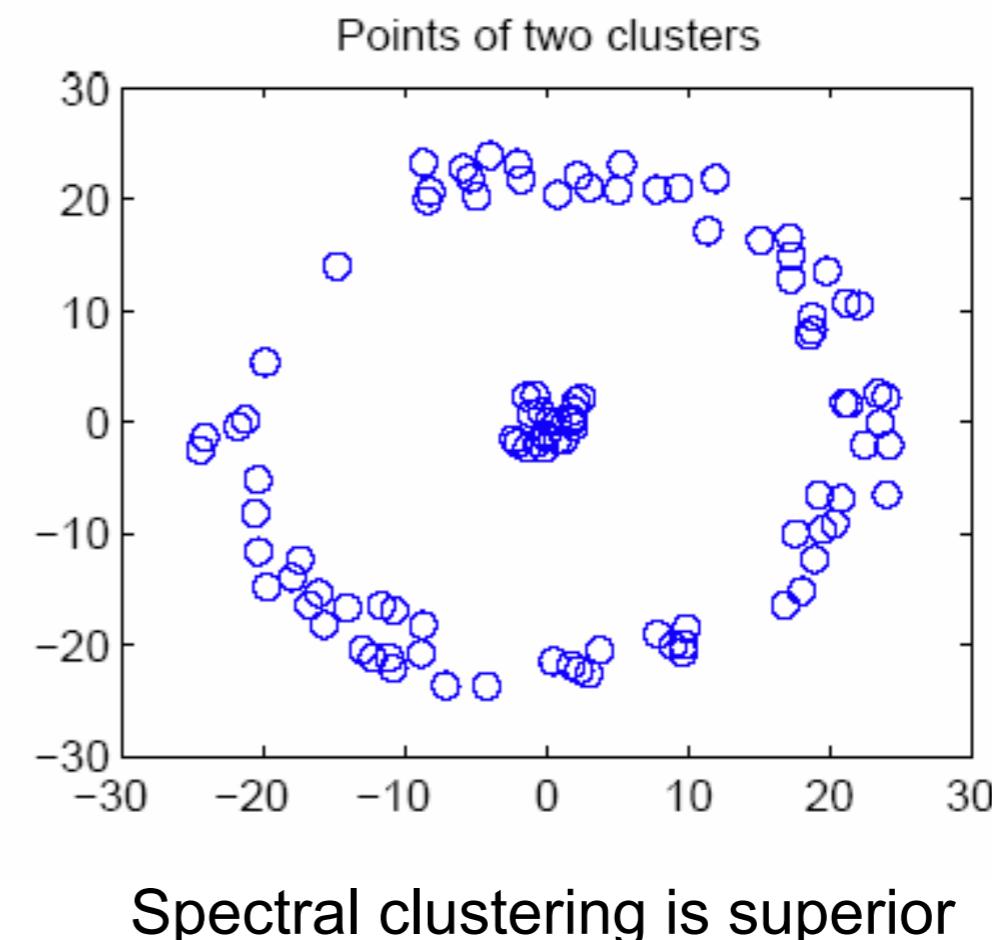
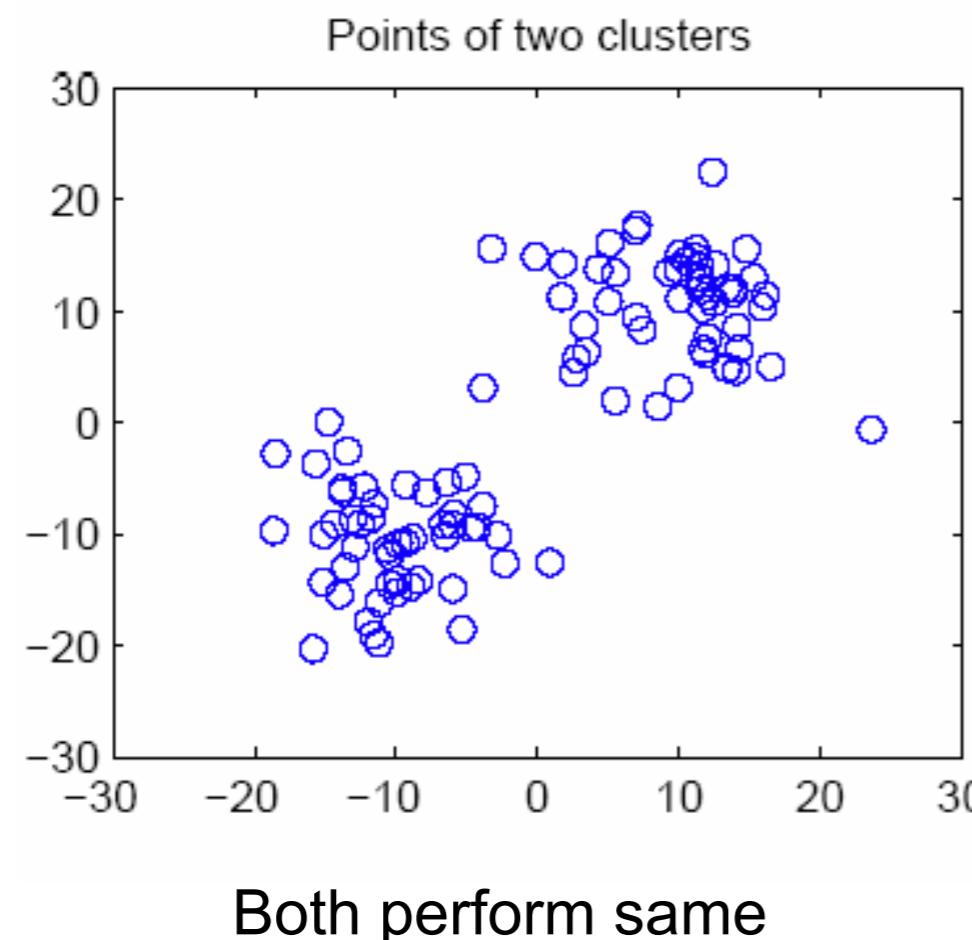
- The solution y is given by the generalized eigenvector corresponding to the second smallest eigenvalue
- Intuitively, the i -th entry of y can be viewed as a “soft” indication of the component membership of the i -th feature
 - Can use 0 or median value of the entries as the splitting point (threshold), or find threshold that minimizes the Ncut cost

Normalized cut 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 being a measure of the similarity between the two nodes.
2. Solve $(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda \mathbf{D}\mathbf{x}$ for eigenvectors with the smallest eigenvalues.
3. Use the eigenvector with second smallest eigenvalue to bipartition the graph.
4. Decide if the current partition should be sub-divided, and recursively repartition the segmented parts if necessary.

K-Means vs. Spectral Clustering

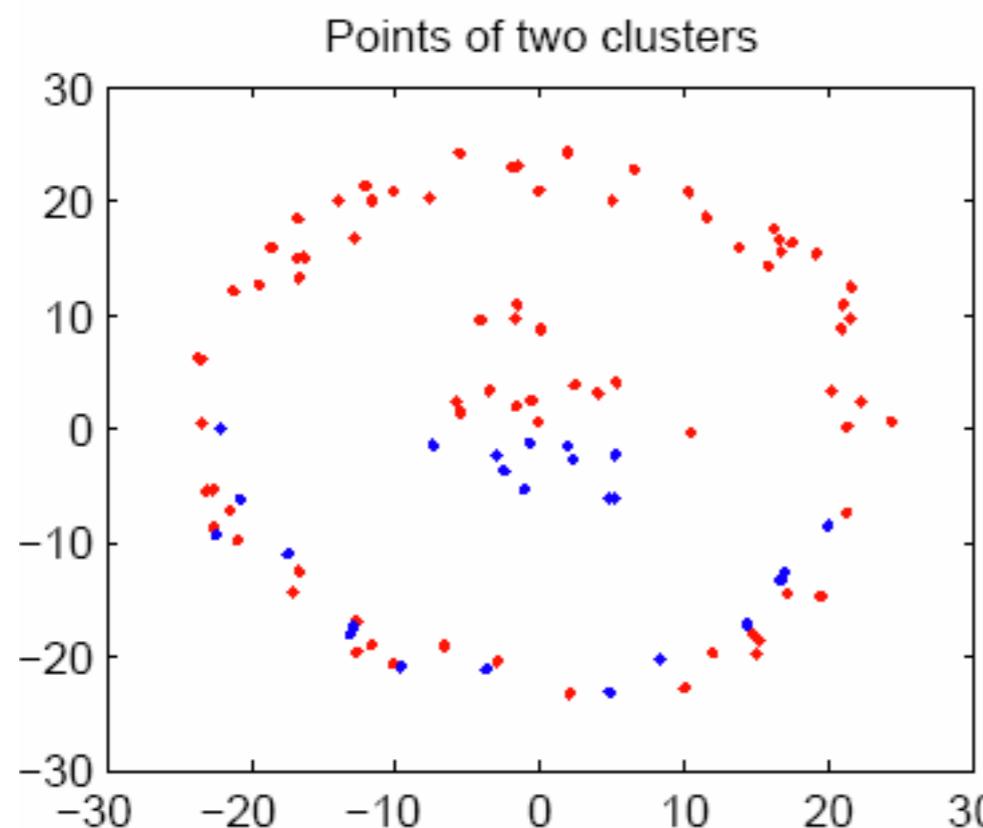
- Applying k-means to Laplacian eigenvectors allows us to **find cluster with non-convex boundaries.**



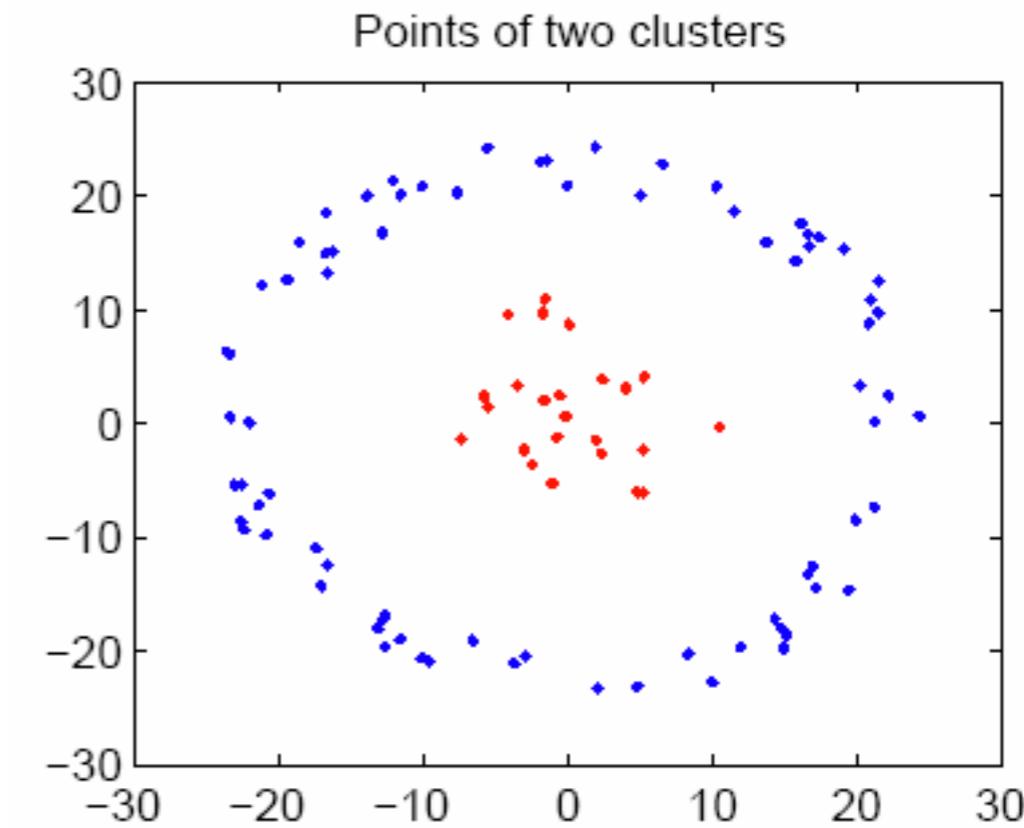
Spectral clustering is superior

K-Means vs. Spectral Clustering

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



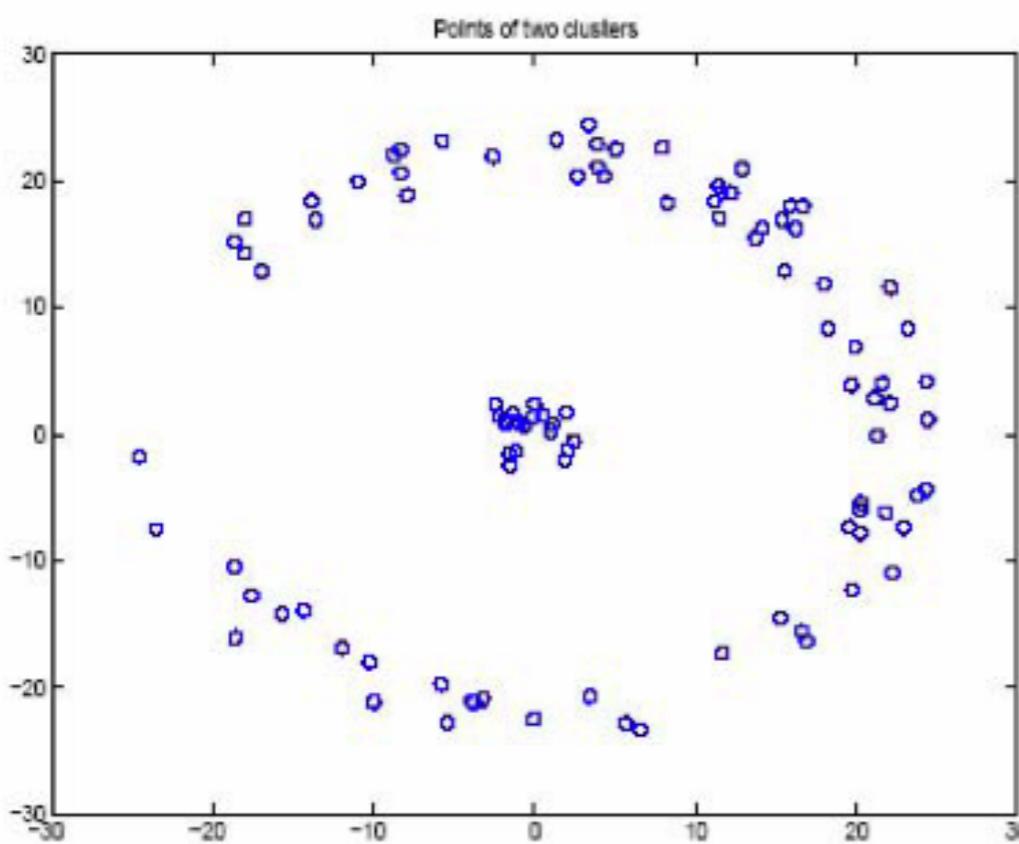
k-means output



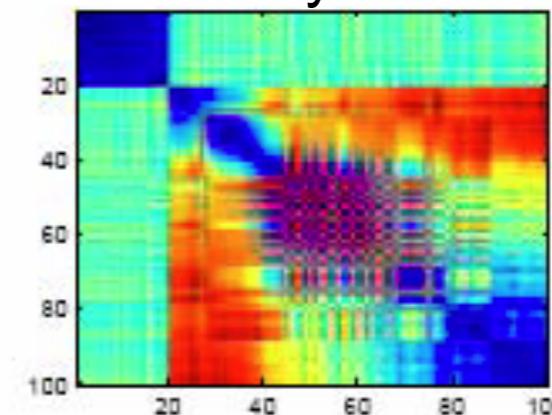
Spectral clustering output

K-Means vs. Spectral Clustering

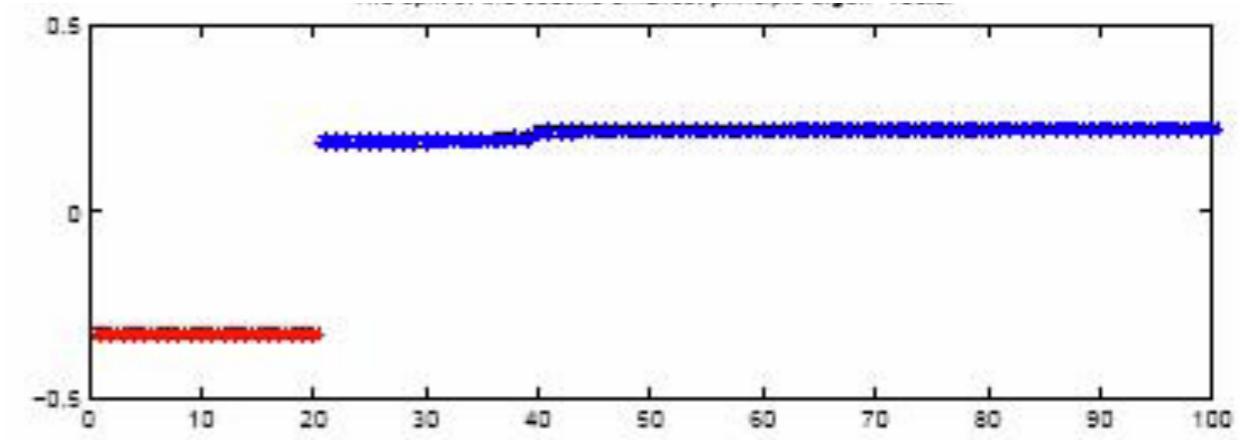
- Applying k-means to Laplacian eigenvectors allows us to **find cluster with non-convex boundaries.**



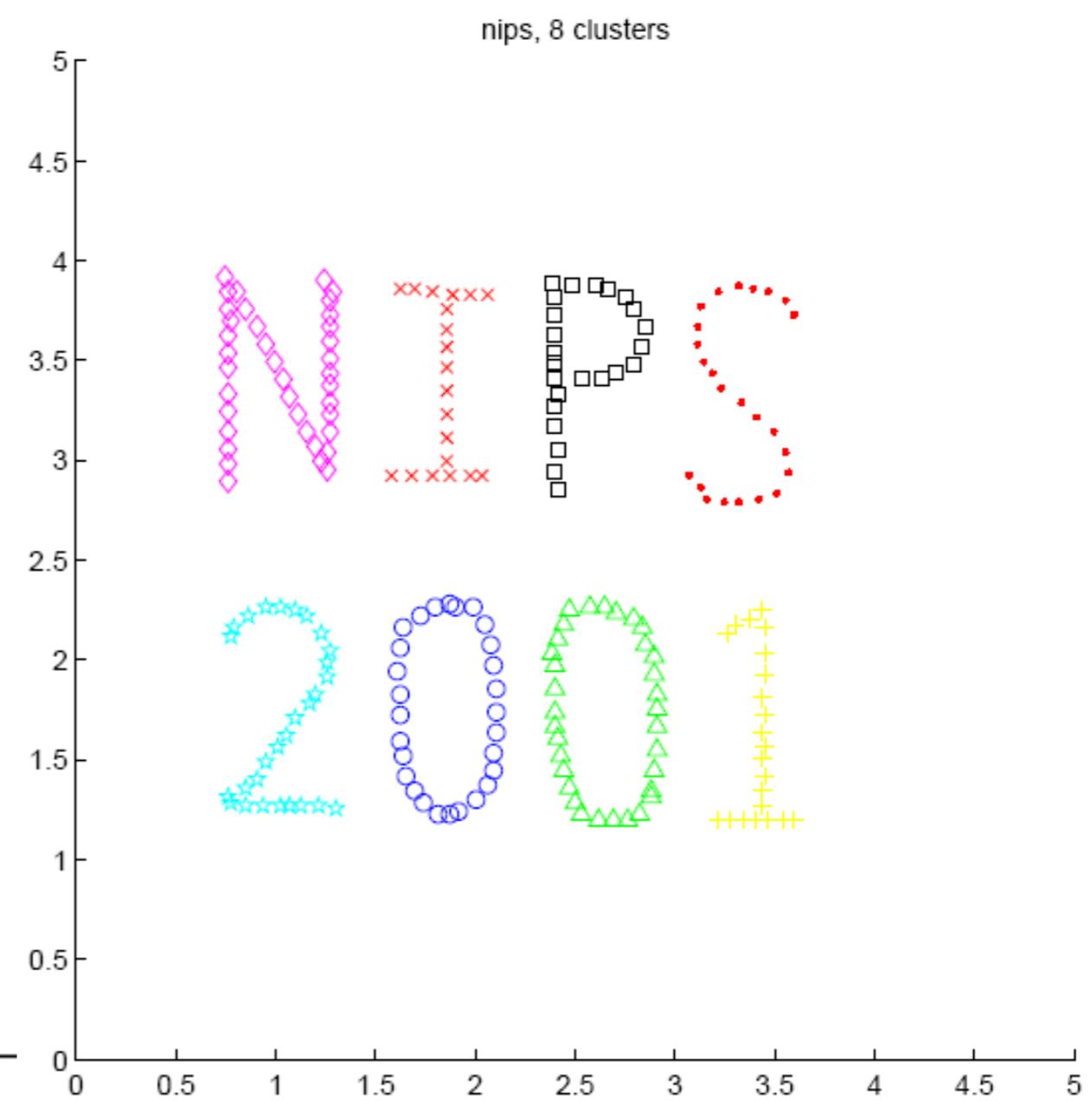
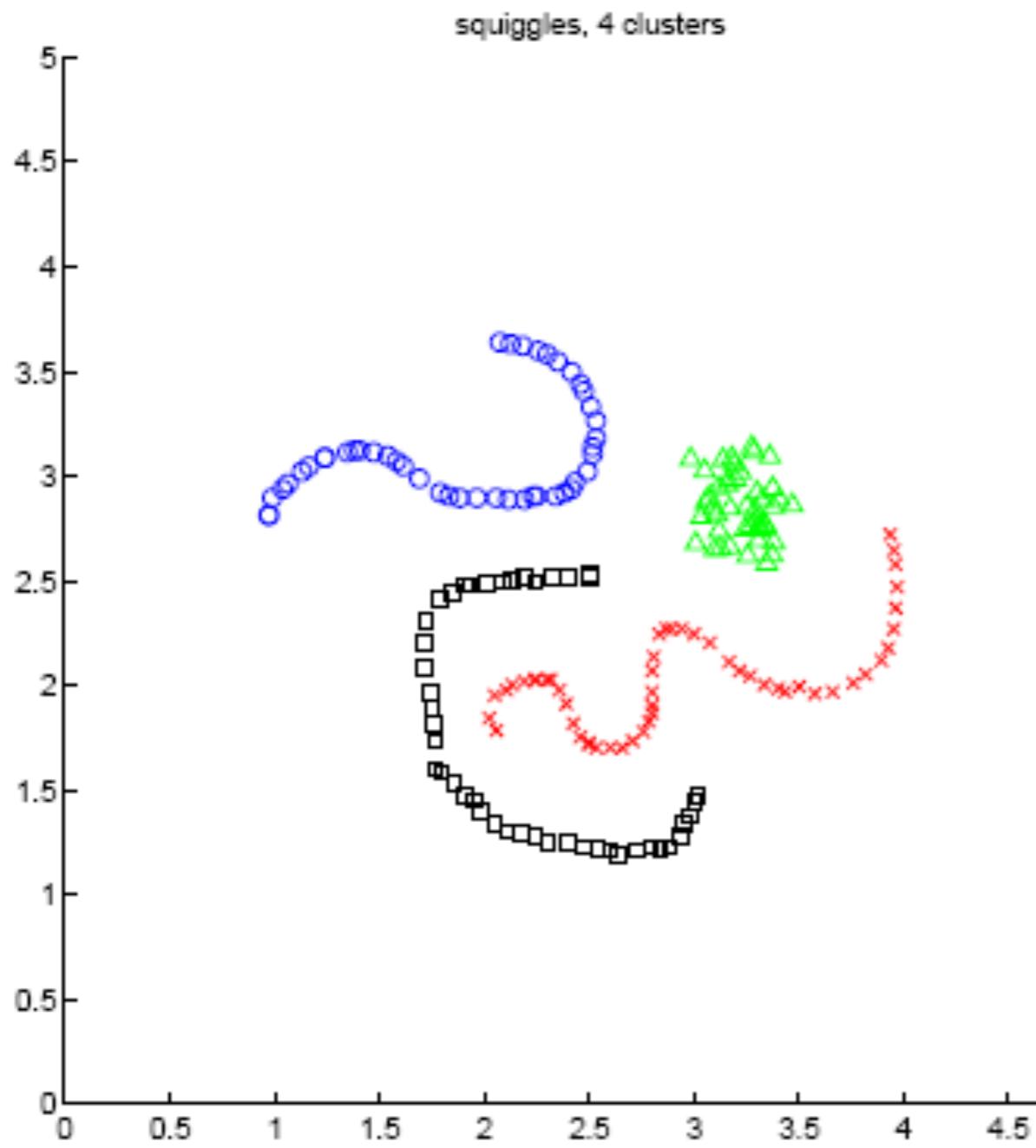
Similarity matrix



Second eigenvector of graph Laplacian



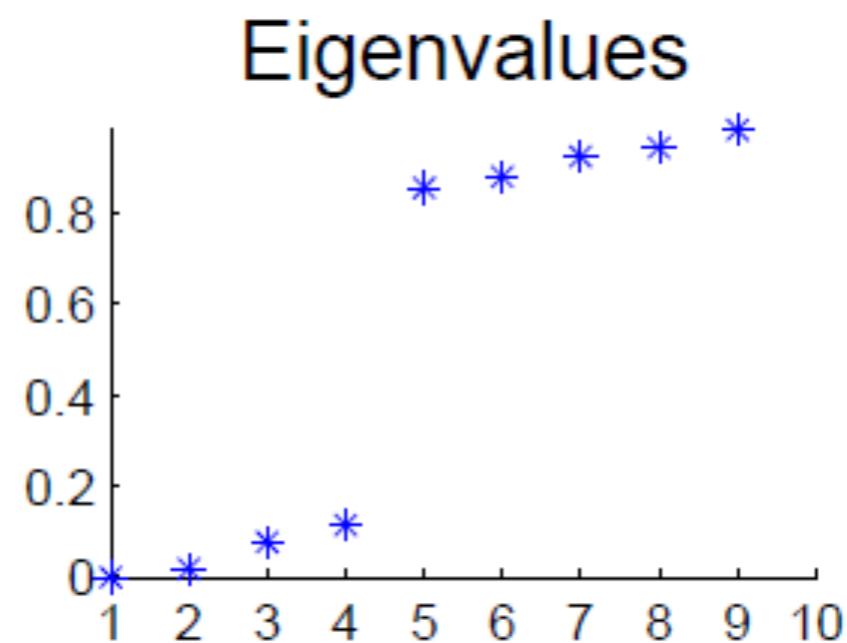
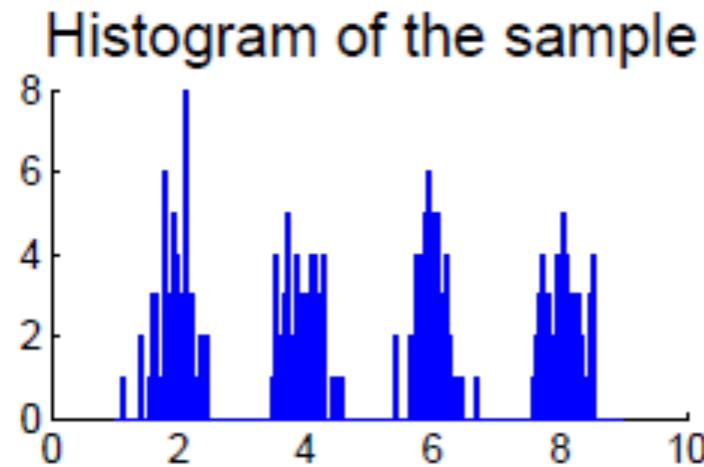
Examples



Some Issues

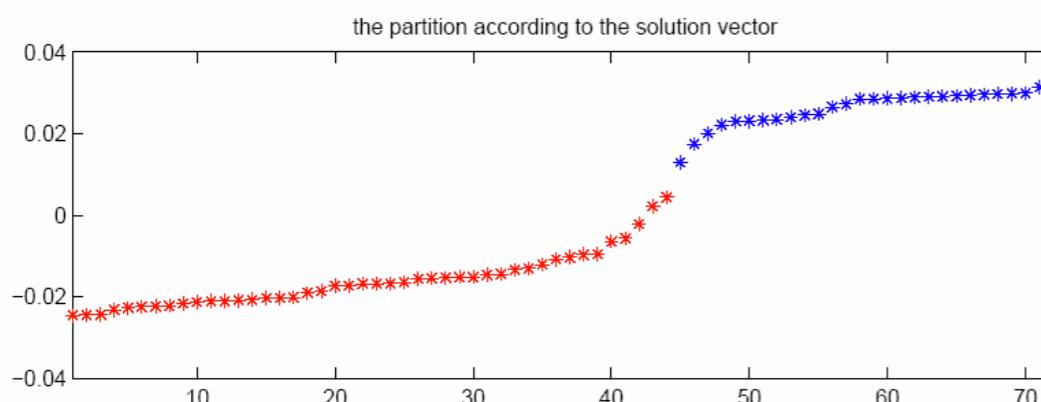
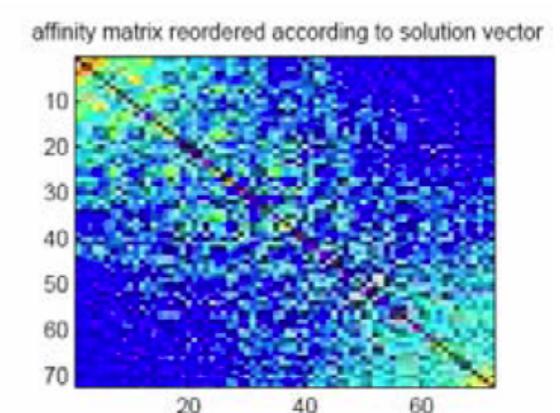
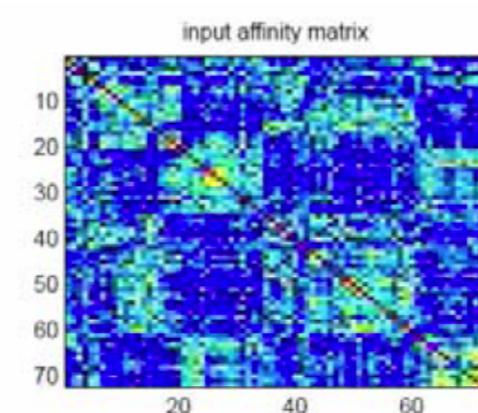
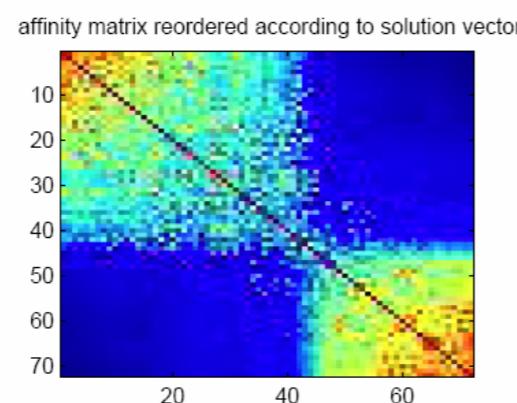
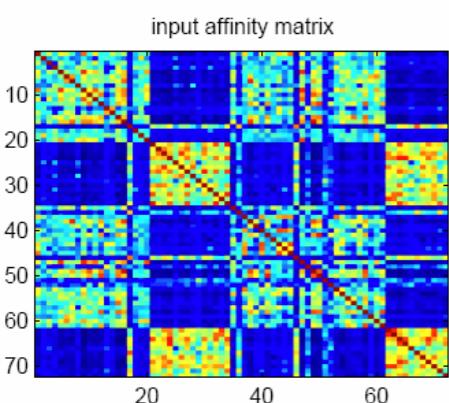
- Choice of number of clusters k
 - Most stable clustering is usually given by the value of k that maximizes the eigengap (difference between consecutive eigenvalues)

$$\Delta_k = |\lambda_k - \lambda_{k-1}|$$

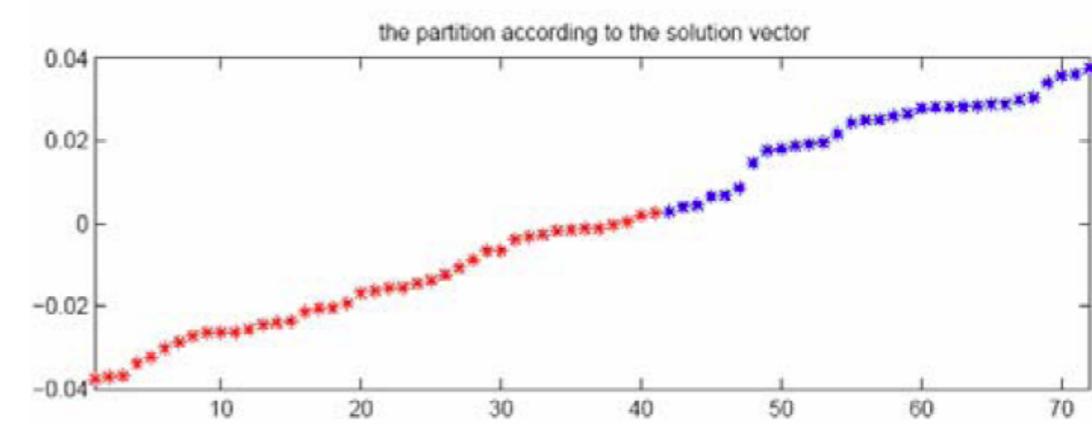


Some Issues

- Choice of number of clusters k
- Choice of similarity
 - Choice of kernel
for Gaussian kernels, choice of σ



Good similarity measure



Poor similarity measure

Some Issues

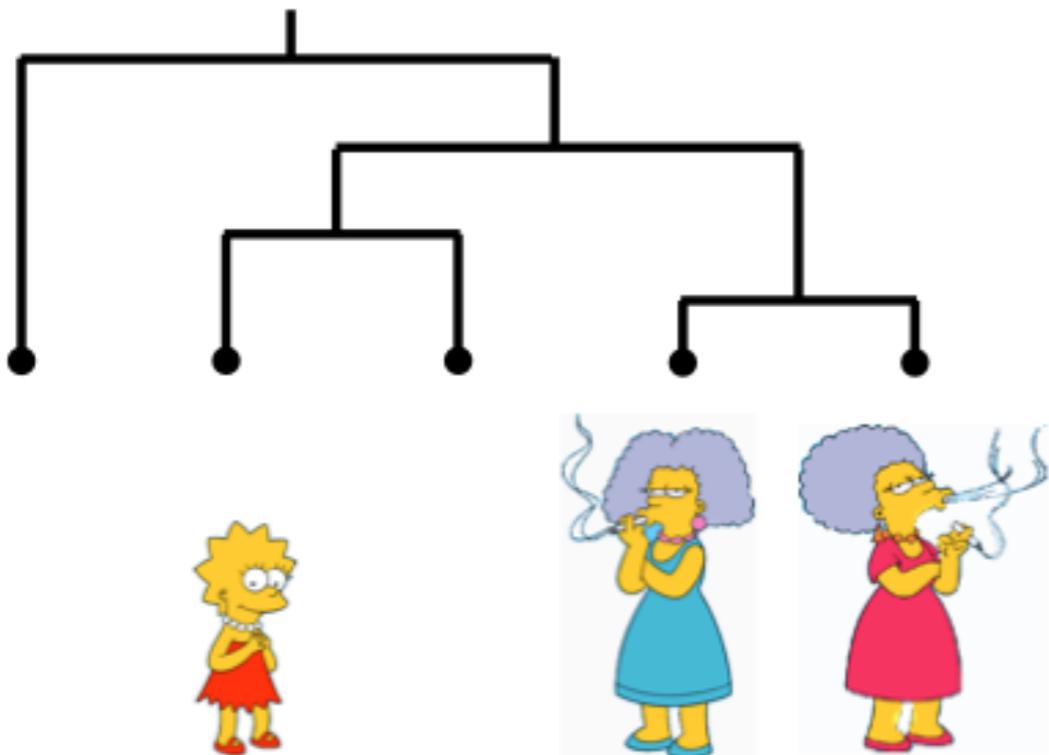
- Choice of number of clusters k
- Choice of similarity
 - Choice of kernel
for Gaussian kernels, choice of σ
- Choice of clustering method
 - k-way vs. recursive 2-way

Hierarchical clustering

Hierarchical Clustering

- Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

- The number of dendograms with
 n leafs = $(2n - 3)!/[(2(n - 2)) (n - 2)!]$



Number of leafs	Number of possible Dendograms
2	1
3	3
4	15
5	105
...	...
10	34,459,425

We begin with a distance matrix which contains the distances between every pair of objects in our dataset

$$D(\text{Lis}, \text{Lis}) = 8$$

$$D(\text{Marge}, \text{Edna}) = 1$$

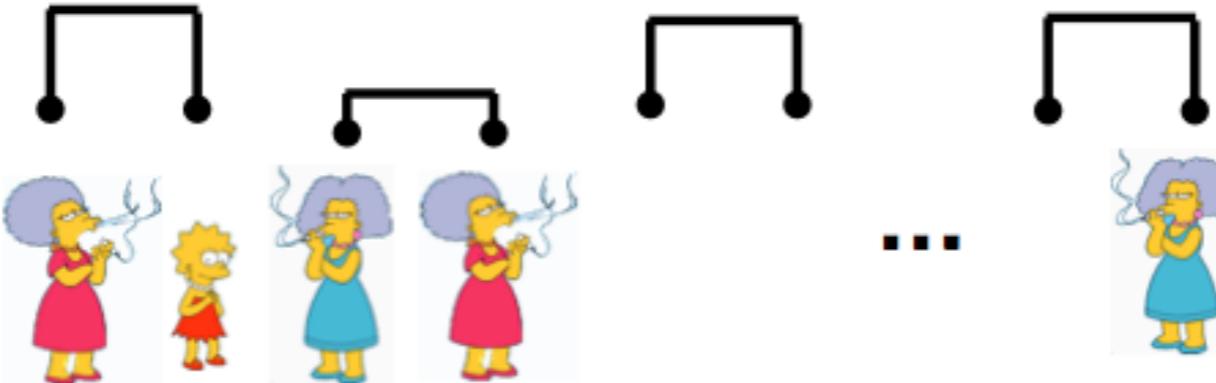


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

Bottom-Up (agglomerative):

Start with each item in its own cluster,
find the best pair to merge into a new
cluster. Repeat until all clusters are
fused together.

Consider all
possible
merges...



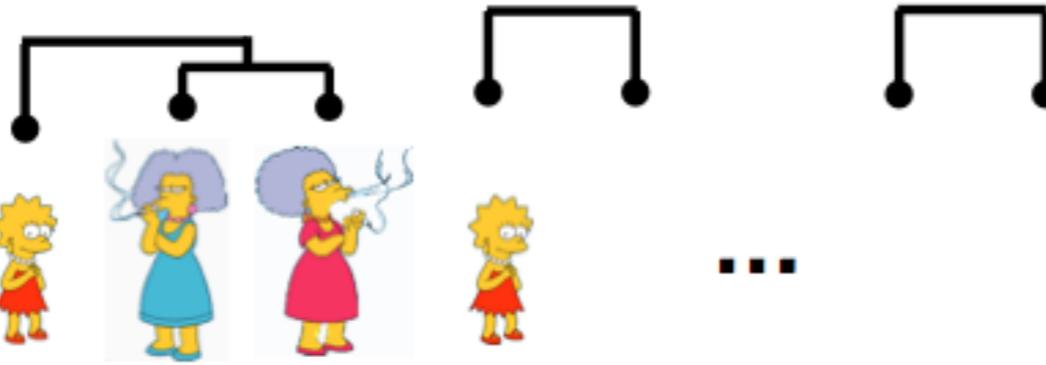
Choose
the best



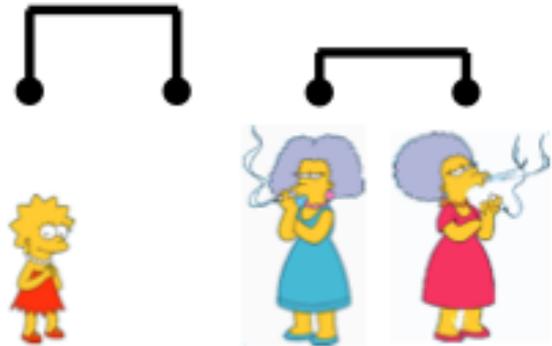
Bottom-Up (agglomerative):

Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

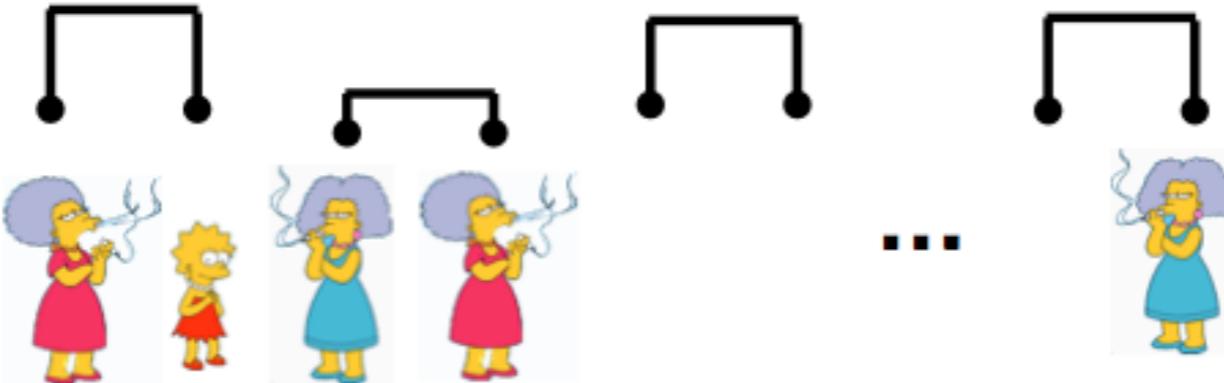
Consider all possible merges...



Choose the best



Consider all possible merges...



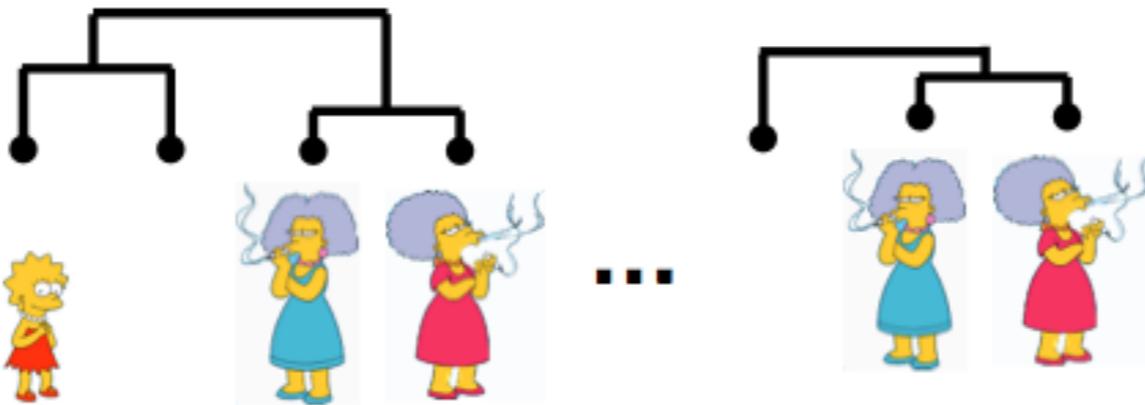
Choose the best



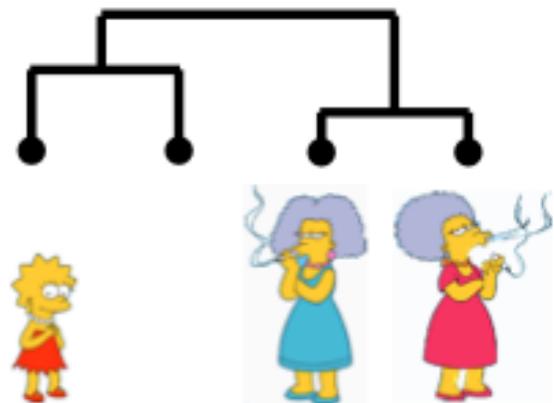
Bottom-Up (agglomerative):

Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

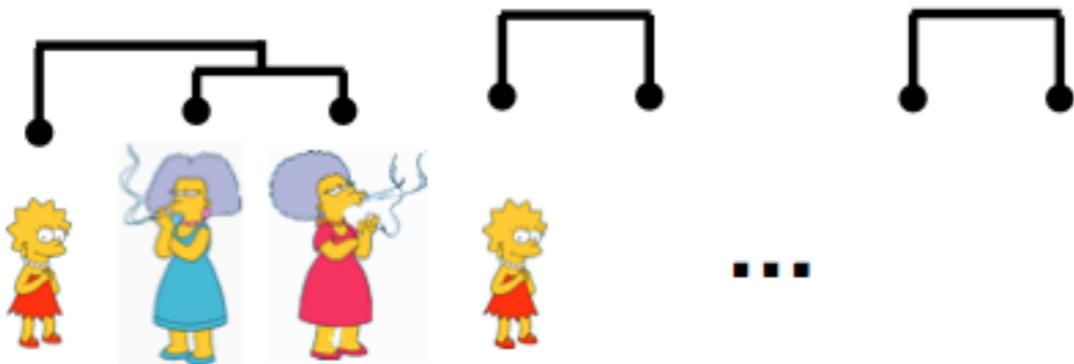
Consider all possible merges...



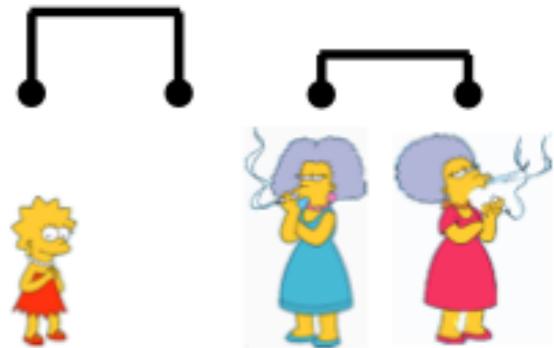
Choose the best



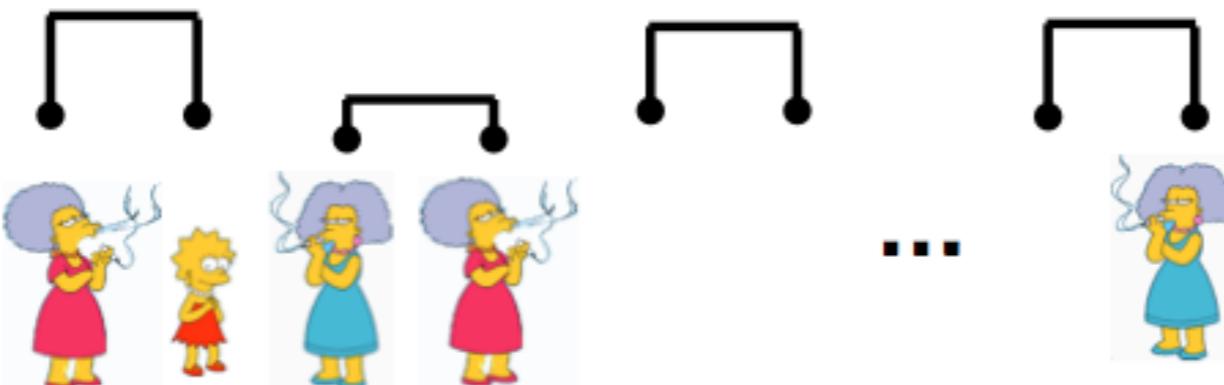
Consider all possible merges...



Choose the best



Consider all possible merges...

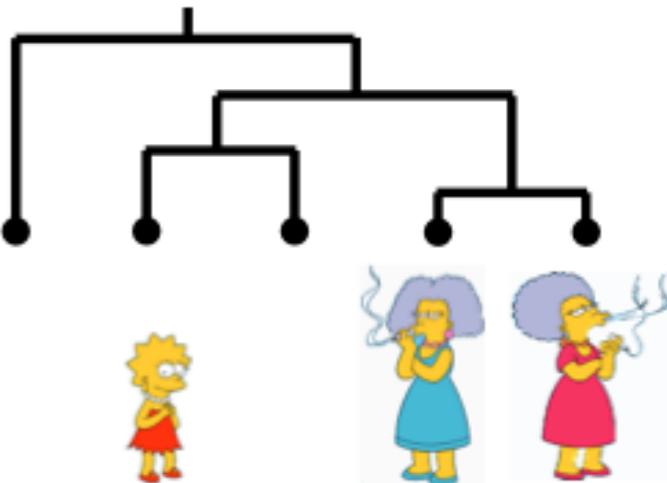


Choose the best

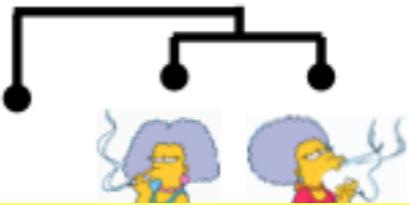
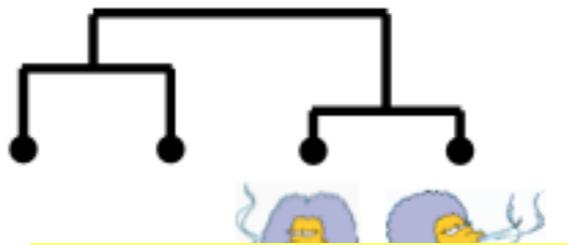


Bottom-Up (agglomerative):

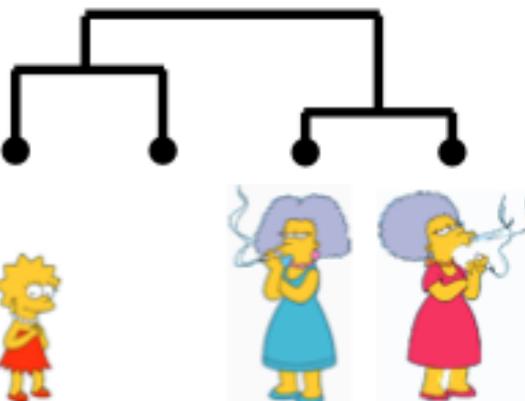
Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



Consider all possible merges...



Choose the best

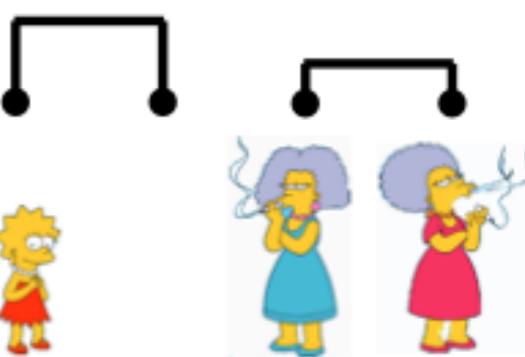


Consider all possible merges...

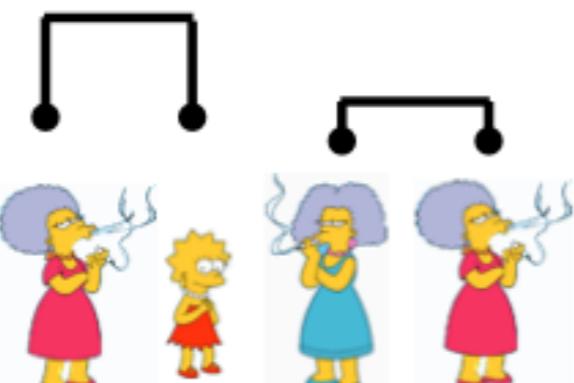


...

the best



Consider all possible merges...



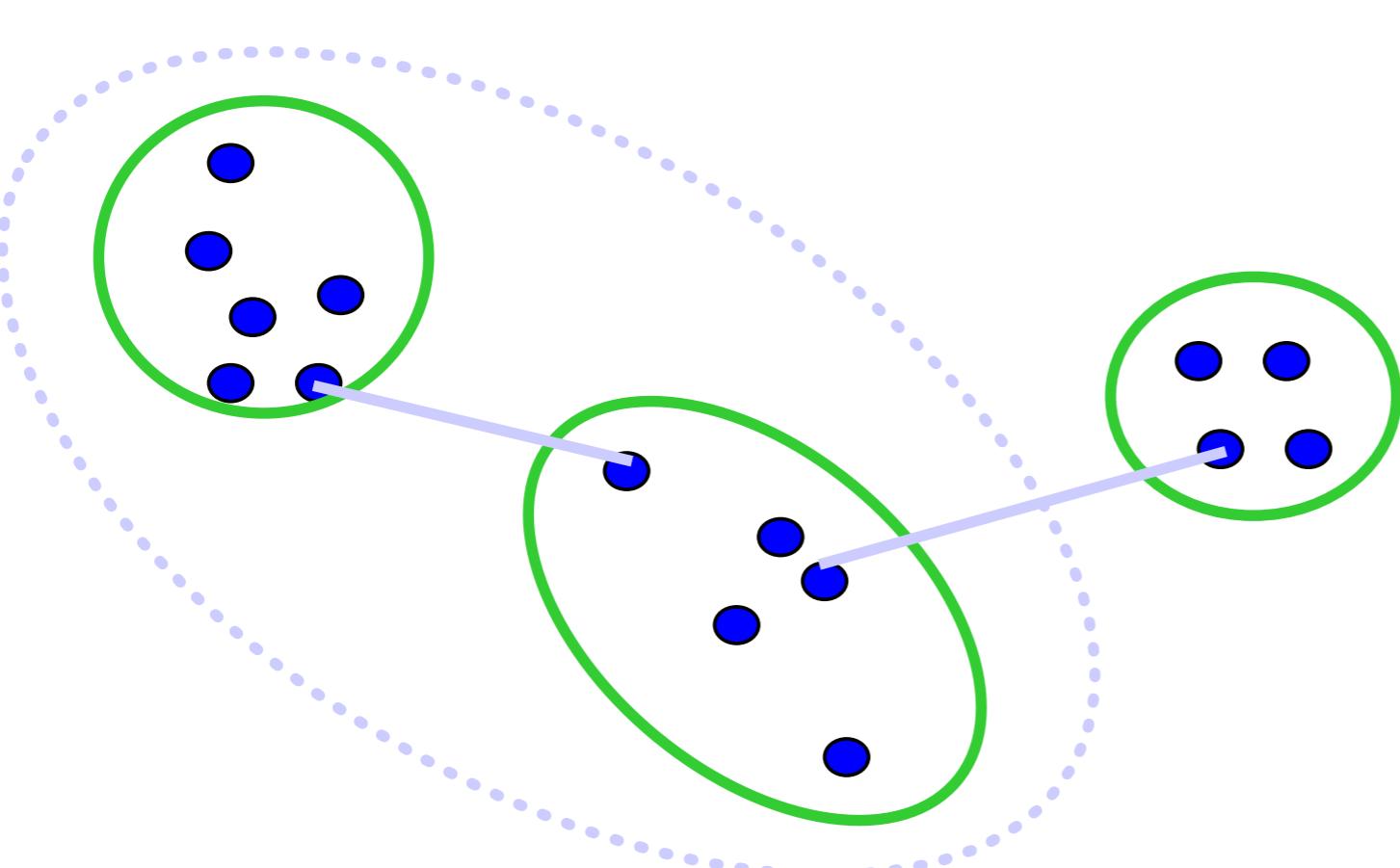
...

Choose the best



Computing distance between clusters: Single Link

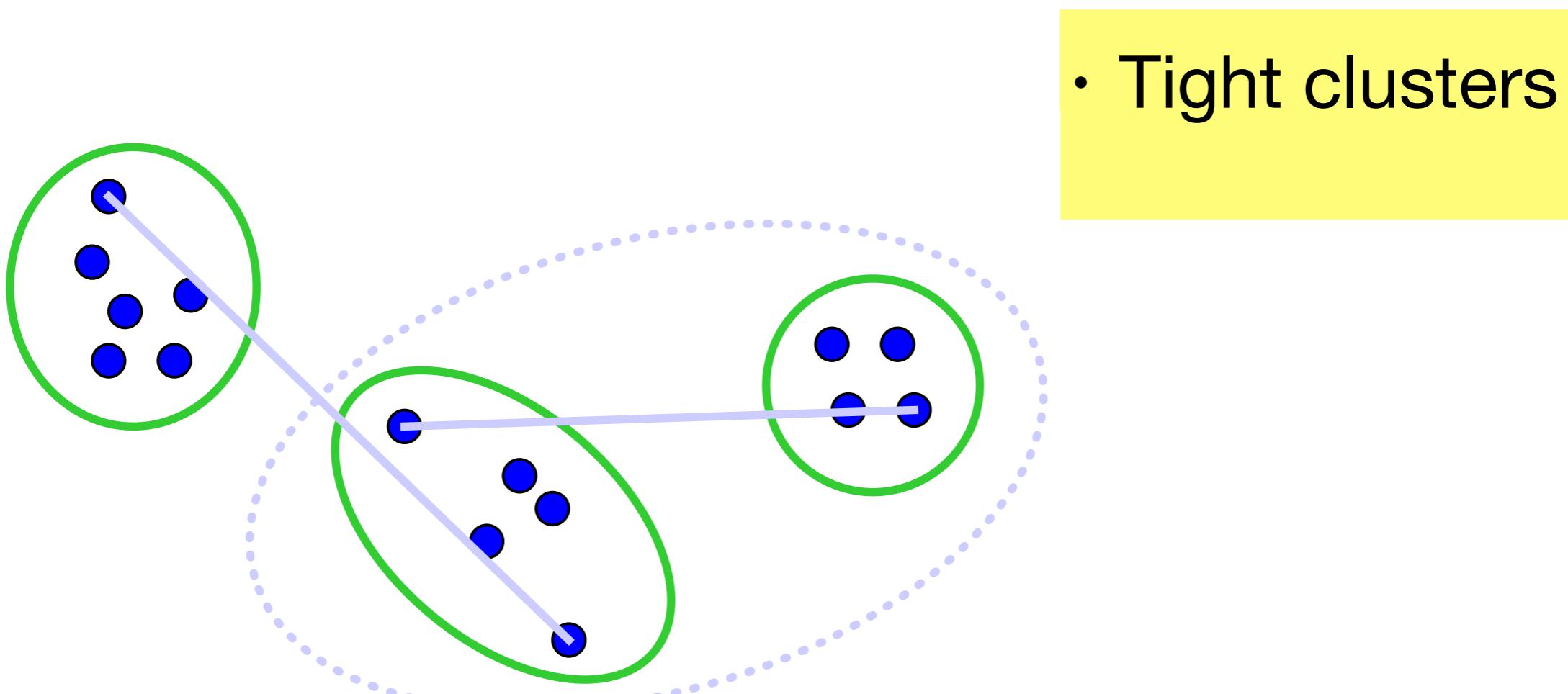
- Cluster distance = distance of two **closest** members in each class



- Potentially long
and skinny
clusters

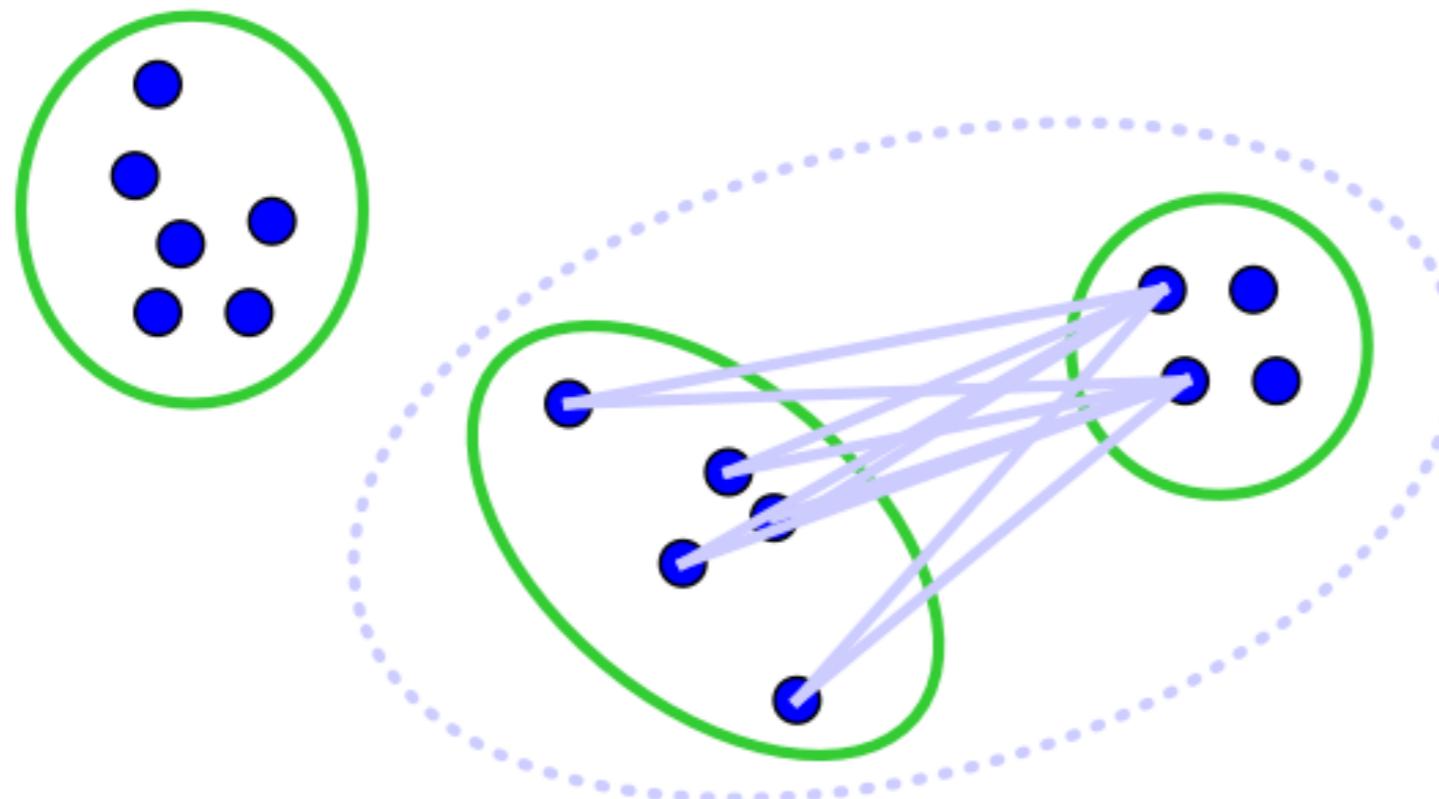
Computing distance between clusters: Complete Link

- Cluster distance = distance of two **farthest** members in each class



Computing distance between clusters: Average Link

- Cluster distance = average distance of all pairs



- The most widely used measure
- Robust against noise

Agglomerative Clustering

Good

- Simple to implement, widespread application
- Clusters have adaptive shapes
- Provides a hierarchy of clusters

Bad

- May have imbalanced clusters
- Still have to choose number of clusters or threshold
 - **silhouette coefficient**
- Need to use an “ultrametric” to get a meaningful hierarchy

What is a good clustering?

What is a good clustering?

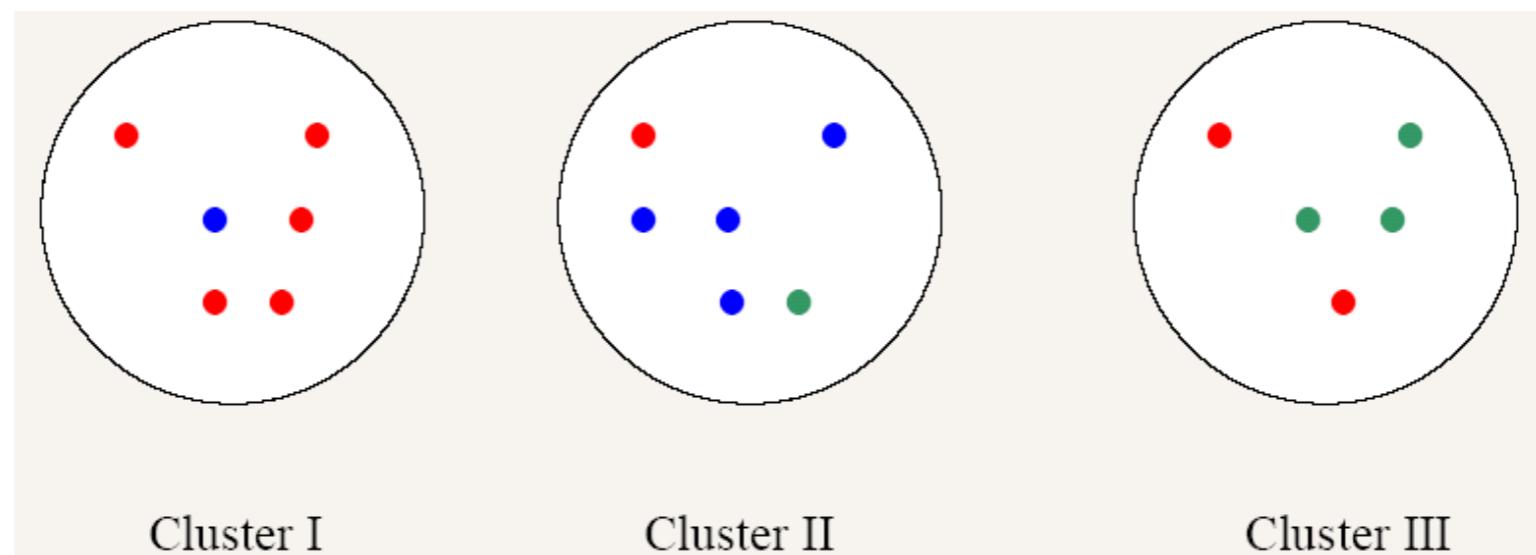
- **Internal criterion:** A good clustering will produce high quality clusters in which:
 - the intra-class (that is, intra-cluster) similarity is high
 - the inter-class similarity is low
 - The measured quality of a clustering depends on both the obj. representation and the similarity measure used
- **External criteria** for clustering quality
 - Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
 - Assesses a clustering with respect to ground truth
 - Example:
 - Purity
 - Entropy of classes in clusters (or Mutual Information between classes and clusters)

External Evaluation of Cluster Quality

- Simple measure: purity, the ratio between the dominant class in the cluster and the size of cluster
 - Assume documents with C gold standard classes, while our clustering algorithms produce K clusters, $\omega_1, \omega_2, \dots, \omega_K$ with n_i members.

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

- Example:



$$\begin{aligned}\text{purity} &= 1/17 * (\max(5, 1, 0) + \max(1, 4, 1) + \max(2, 0, 3)) \\ &= 1/17 * (5+4+3) \approx 0.71\end{aligned}$$

External Evaluation of Cluster Quality

- Let:

$$TC = TC_1 \cup TC_2 \cup \dots \cup TC_n$$

$$CC = CC_1 \cup CC_2 \cup \dots \cup CC_m$$

be the target and computed clusterings, respectively.

- $TC = CC = \text{original set of data}$

- Define the following:

- a : number of pairs of items that belong to the same cluster in both CC and TC
- b : number of pairs of items that belong to different clusters in both CC and TC
- c : number of pairs of items that belong to the same cluster in CC but different clusters in TC
- d : number of pairs of items that belong to the same cluster in TC but different clusters in CC

External Evaluation of Cluster Quality

$$P = \frac{a}{a + c}$$

$$R = \frac{a}{a + d}$$

$$F = \frac{2 \times P \times R}{P + R}$$

$$\frac{a+b}{a+b+c+d}$$

Rand Index

Measure of clustering agreement: how similar are these two ways of partitioning the data?

F-measure

External Evaluation of Cluster Quality

$$\frac{a+b}{a+b+c+d}$$

Rand Index

$$\frac{2(ab - cd)}{(a + c)(c + b) + (a + d)(d + b)}$$

Adjusted Rand Index

Extension of the Rand index that attempts to account for items that may have been clustered by chance

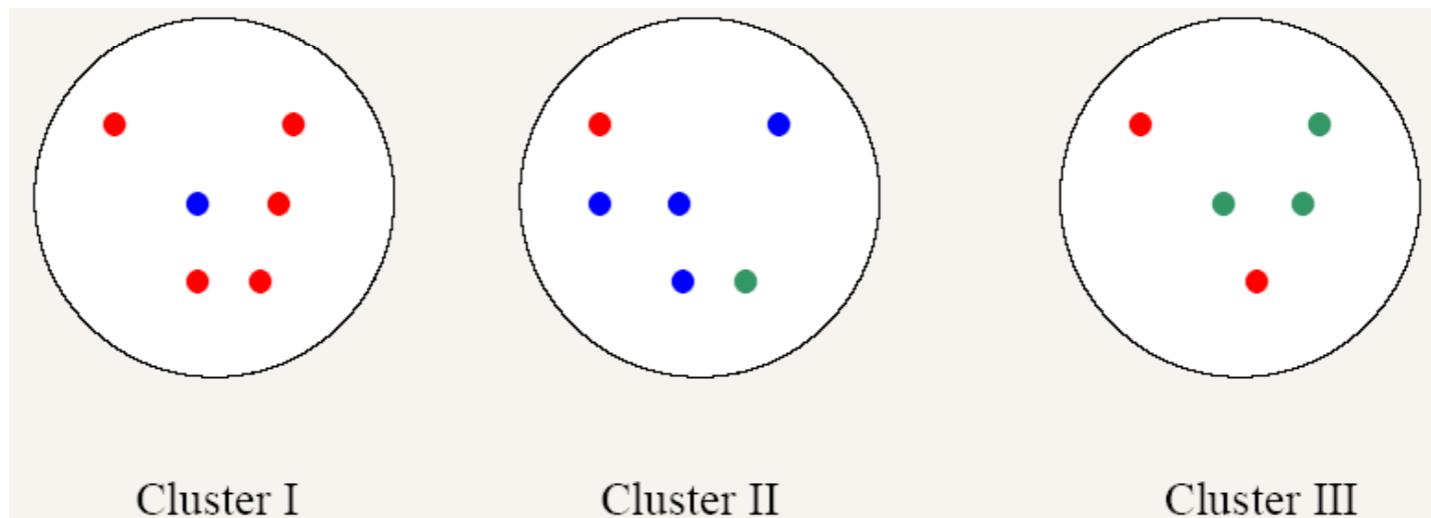
External Evaluation of Cluster Quality

$$Entropy(CC_i) = \sum_{TC_j \in TC} -p(TC_j | CC_i) \log p(TC_j | CC_i)$$

$$AvgEntropy(CC) = \sum_{i=1}^m \frac{|CC_i|}{|CC|} Entropy(CC_i)$$

Measure of purity wrt
the target clustering

- Example:



$$Entropy(CC_1) = (5/6)\log(5/6) + (1/6)\log(1/6) + (0/6)\log(0/6) = -.650$$

$$Entropy(CC_2) = (1/6)\log(1/6) + (4/6)\log(4/6) + (1/6)\log(1/6) = -1.252$$

$$Entropy(CC_3) = (2/5)\log(2/5) + (0/5)\log(0/5) + (3/5)\log(3/5) = -.971$$

$$AvgEntropy(CC) = (-.650 * 6/17) + (-1.252 * 6/17) + (-.971 * 5/17)$$

$$AvgEntropy(CC) = -.956$$

Next Lecture:

Dimensionality Reduction