Part 7

Elements of data clustering

Introduction

➢Until now we have assumed that the training samples used to design a classifier were *labeled* by their category membership. **Procedures** that use **labeled samples** are said to be **supervised**.

➢Now we shall investigate a number of *unsupervised* procedures, which use unlabeled samples.

Two terms are commonly used to indicate the topics we are going to deal with:

- ✓ Unsupervised learning
- ✓ Data clustering

Introduction

There are at least five basic reasons for developing unsupervised procedures:

1) Collecting and labeling a large set of patterns can be surprisingly costly (e.g., for speech recognition)
2) It might result more convenient to proceed in the reverse direction: using large amounts of unlabeled data to identify “clusters”, and only then use supervision to label the groupings found.
3) The characteristics of the patterns can change slowly with time. An unsupervised classifier can track these changes and then achieve improved performance.
4) We can find “significant” features that will then be useful for categorization.
5) We can gain useful insights into the structure of the data

Basic concepts

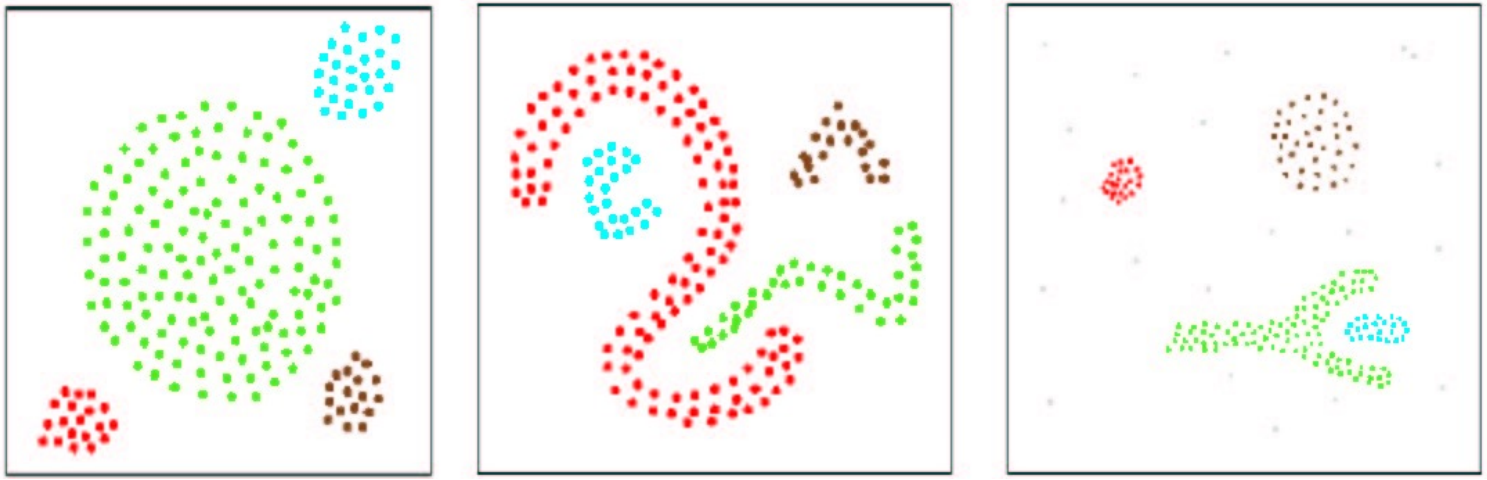
- The goal of **unsupervised learning** ("clustering") is basically to find groupings in the data ("clusters") which actually reflect the ground truth and the "natural properties" of the domain the data comes from.
- It is straightforward to see that even if intuitive, the concept of cluster is hard to be defined rigorously (both in general and even in very specific cases)
- We can informally say that "samples belonging to the same cluster must present a degree of similarity higher than samples belonging to different clusters"
- Starting from a set of unlabeled samples, a "**hard clustering**" algorithm generates a partitioning D=(**D**^{1}, **D**^{2},….,**D**^{c}), where:

$$D^j \subseteq D$$

$$D^i \cap D^j = \varnothing \quad \text{i} \neq \text{j}$$

$$\bigcup_{i=1}^{c} D^i = D$$

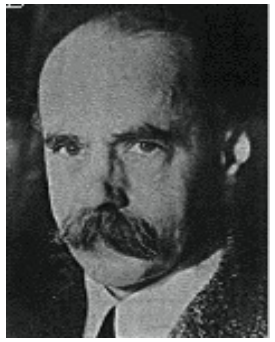
Simple clustering examples



It is straightforward to see that even if intuitive, the concept of cluster is hard to be defined !!

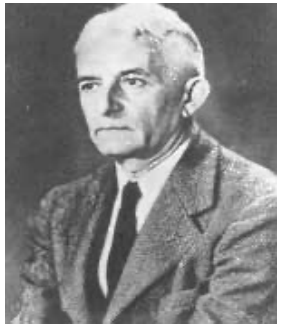
Basic ideas of grouping in humans: The Gestalt school

Wertheimer



Koehler

Koffka



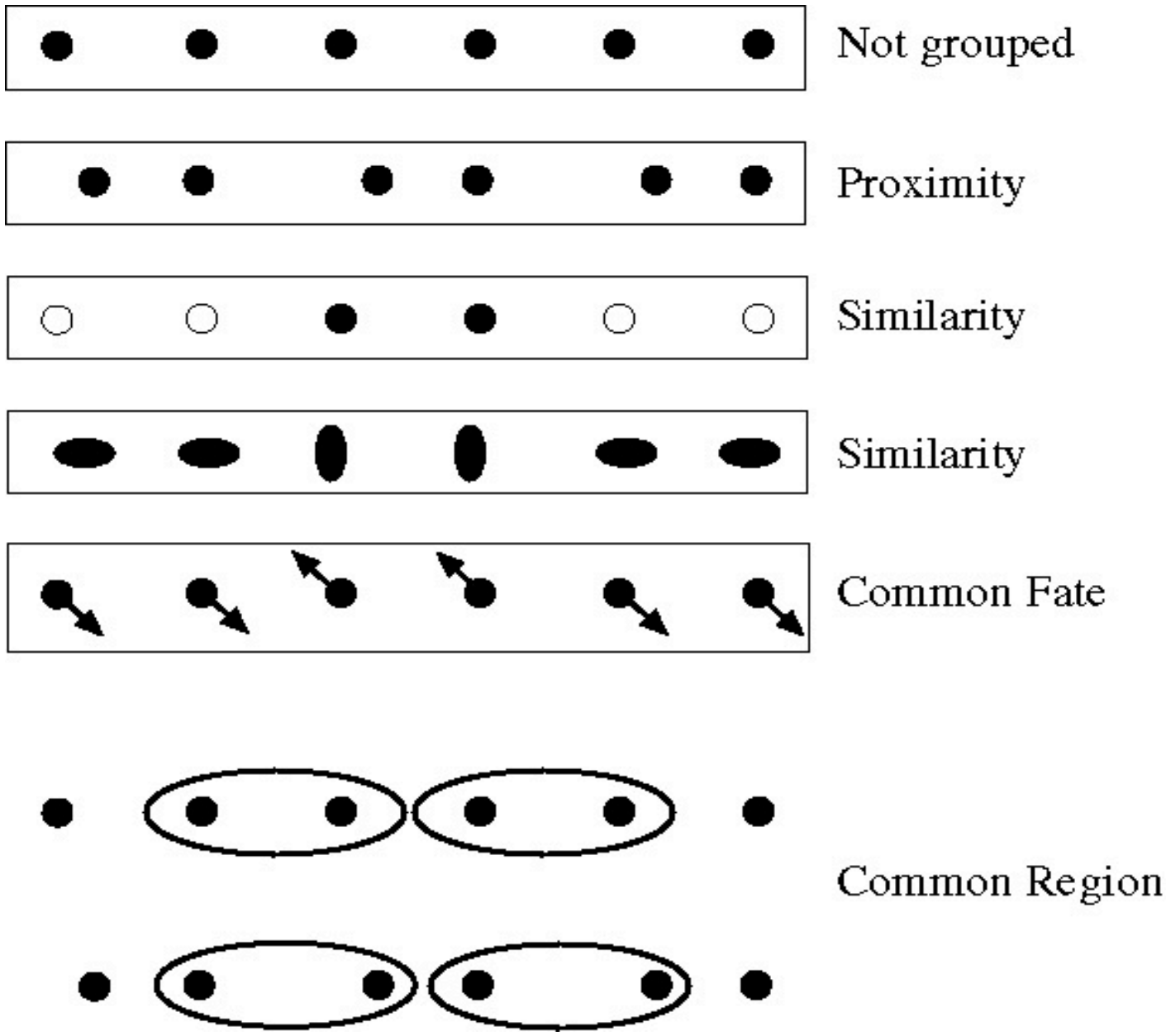
Gestalt properties

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

Clustering

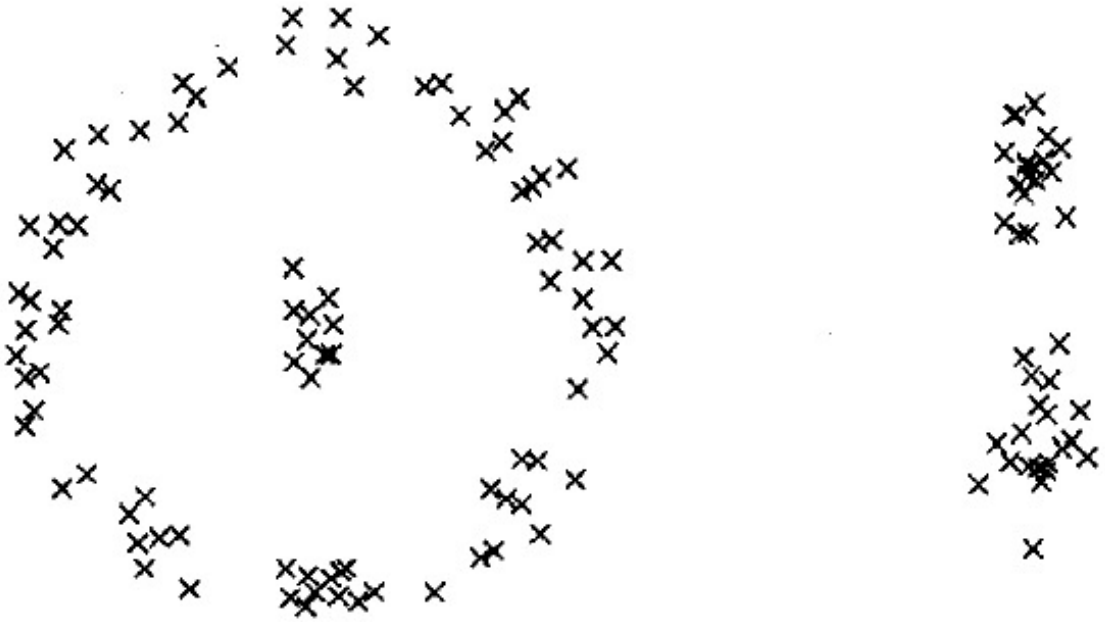
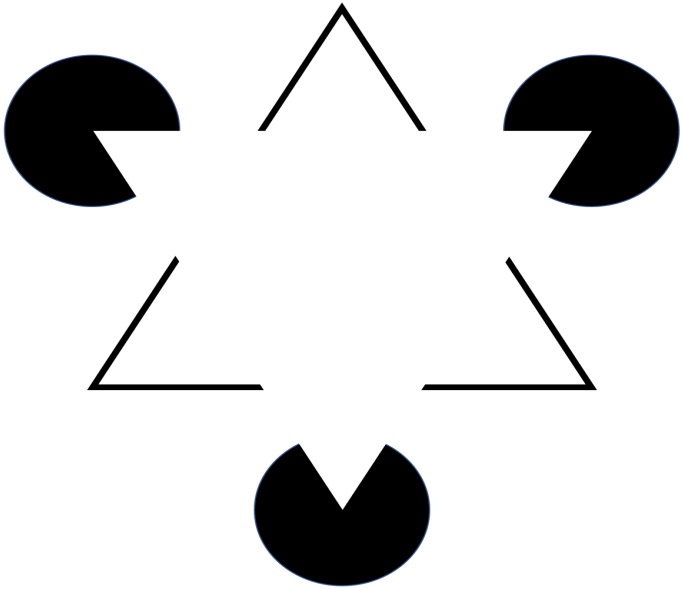


Figure 1: How many groups?

Clustering, optical illusions, and cognitive biases



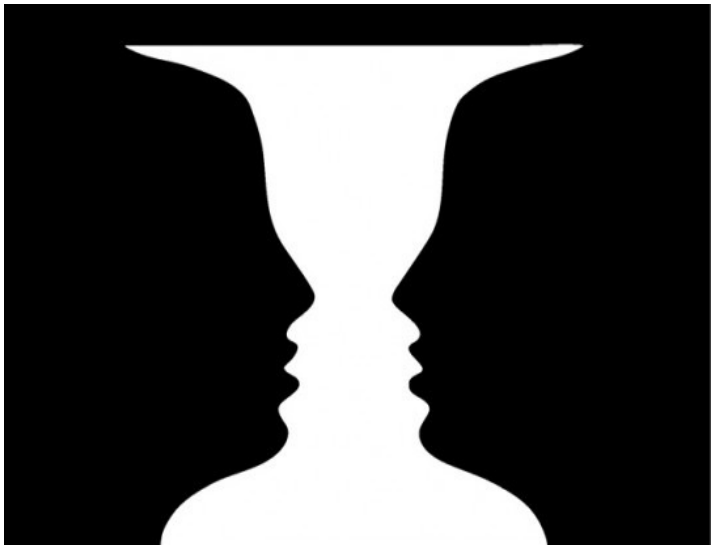
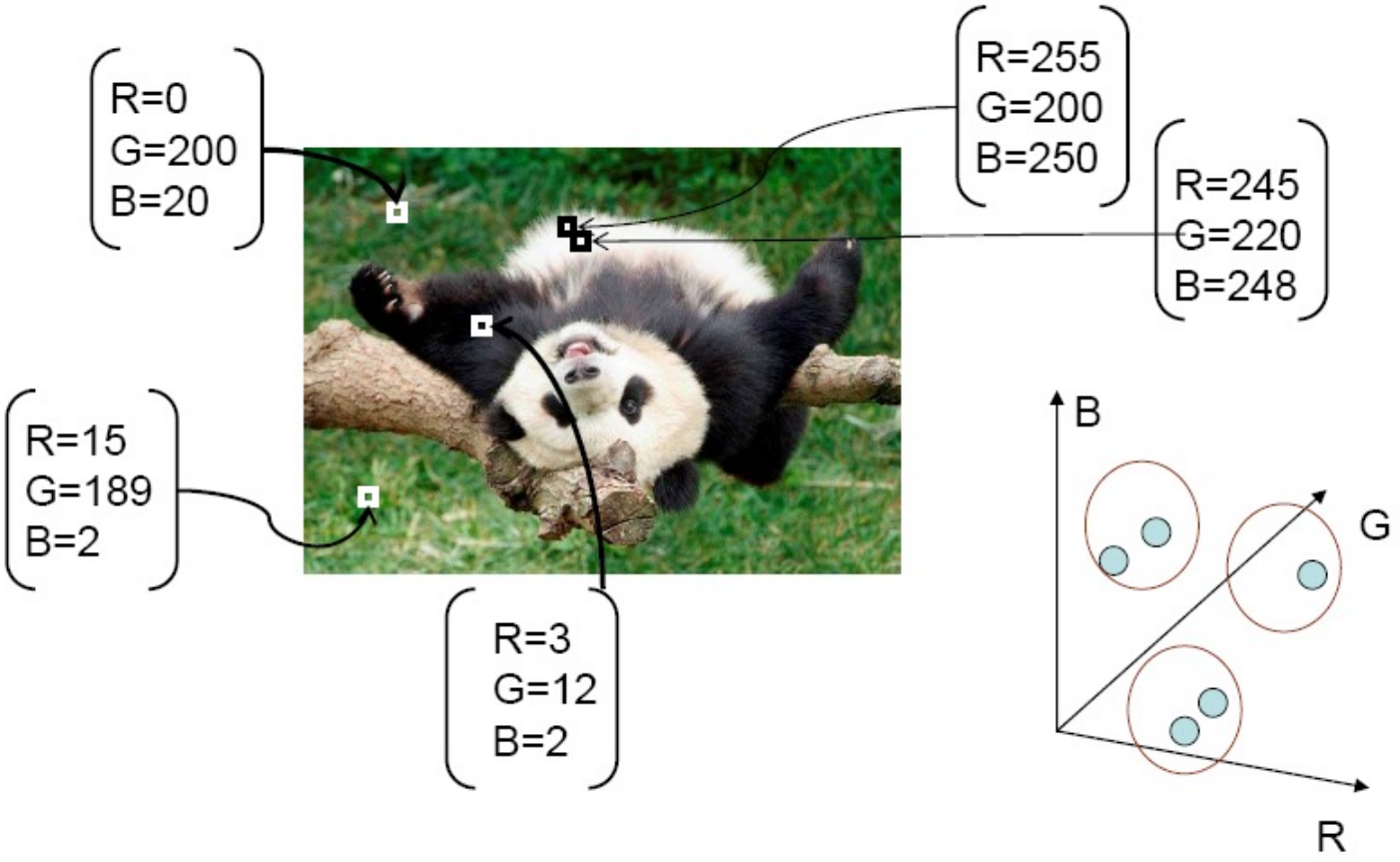


Image Segmentation as clustering

- Cluster similar pixels (features) together



Source: K. Grauman

Clustering algorithms: categorization

- Clustering algorithms can be grouped into two main categories:

  - ✓ Hierarchical algorithms

    The "single linkage clustering" is a simple example

  - ✓ Non-hierarchical algorithms

    The "k-means" algorithm is a simple example

Single Linkage Clustering Algorithm

1. Set the final number of cluster *c* to be found and define a similarity measure S(a,b) among pairs of patterns.

   Initialize the algorithm by assigning each pattern in **D** to a different "cluster"

2. Identify the two most similar clusters (according to the S similarity measure) and merge them in the "same" cluster.

   The **similarity** among two clusters A and B is measured as

$$\min_{a \in A, b \in B} S(a,b)$$

3. Repeat step 2 until the number of clusters obtained is equal to *c*

Similarity Measures

- The Minkowsky metric is a class of metrics in a *d*-dimensional space

$$d(\mathbf{x},\mathbf{x}') = \left[\sum_{k=1}^{d}\left(\mathbf{x}_k - \mathbf{x}'_k\right)^q\right]^{1/q}$$

- Particular cases:
  - q=1: Manhattan metric
  - q=2: Euclidean distance

- Another popular similarity measure is the Mahalanobis distance

$$d(\mathbf{x},\mathbf{x}') = (\mathbf{x}-\mathbf{x}')^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mathbf{x}')$$

Similarity Measures

- Similarity can be also evaluated by using *non-metric similarity functions* s(**x**, **x´**):
  - The value of s(**x**, **x´**) is high when **x** e **x´** are to some extent similar
- **Example**: the normalized inner product can be used as a similarity measure if the angle beetween two vectors is a measure of their similarity

$$s(\mathbf{x},\mathbf{x}') = \frac{\mathbf{x}^t \mathbf{x}'}{\|\mathbf{x}\| \|\mathbf{x}'\|}$$

Similarity Measures

•Let us consider now "features" that can assume only binary values (0 – 1), and let us also assume that the the pattern **x** is provided with the *i-th* feature if x_i=1. Under these assumptions

$$s(\mathbf{x},\mathbf{x}') = \frac{\mathbf{x}^t\ \mathbf{x}'}{\|\mathbf{x}\|\ \|\mathbf{x}'\|}$$

provides a measure of how many features are in common among **x** and **x´**

• ($\mathbf{x}^t$ **x´**) is the number of features provided for both **x** and **x´** and

$$\|\mathbf{x}\|\ \|\mathbf{x}'\| = \left(\mathbf{x}^t\mathbf{x}\ \mathbf{x}'^t\mathbf{x}'\right)^{1/2}$$

is the geometric mean among the **x** and the **x´** features

Similarity Measures

- Simple modifications:

$$s(\mathbf{x},\mathbf{x}') = \frac{\mathbf{x}^t\ \mathbf{x}'}{d}$$

is the fraction of shared attributes

$$s(\mathbf{x},\mathbf{x}') = \frac{\mathbf{x}^t\ \mathbf{x}'}{\mathbf{x}^t\mathbf{x} + \mathbf{x}'^t\mathbf{x}' - \mathbf{x}^t\ \mathbf{x}'} \qquad \text{(Tanimoto distance)}$$

which is the ratio among the number of attributes in common among $\mathbf{x}$ *and* $\mathbf{x}'$ and the number of attributes of $\mathbf{x}$ *or* $\mathbf{x}'$.

Similarity measures of data clusters

$$d_{\min}(D_i, D_j) = \min_{\substack{x \in D_i \\ x' \in Dj}} \|x - x'\|$$

$$d_{\max}(D_i, D_j) = \max_{\substack{x \in D_i \\ x' \in Dj}} \|x - x'\|$$

$$d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{x \in D_i} \sum_{x' \in Dj} \|x - x'\|$$

$$d_{mean}(D_i, D_j) = \|m_i - m_j\|$$

"k-means" (or c-means) algorithm

1. Set the number c of clusters to be found and a similarity measure S(a,b) among pairs of "patterns".

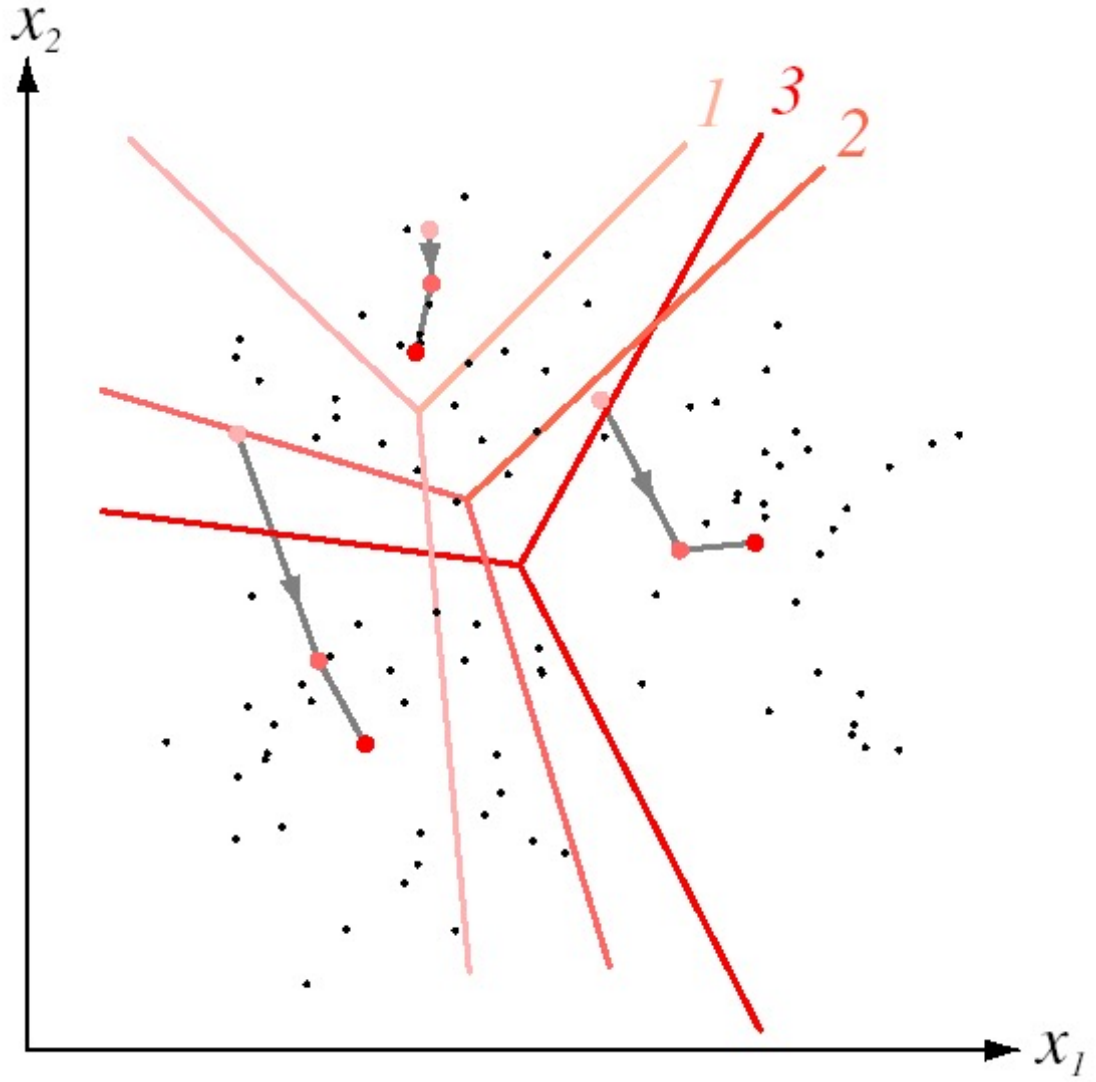
   Initialize the algorithm by defining c cluster centres (c points from **D** can be randomly selected as centres)

2. Assign each point in **D** to one of the c clusters according to the similarity measure

3. Recompute cluster centres

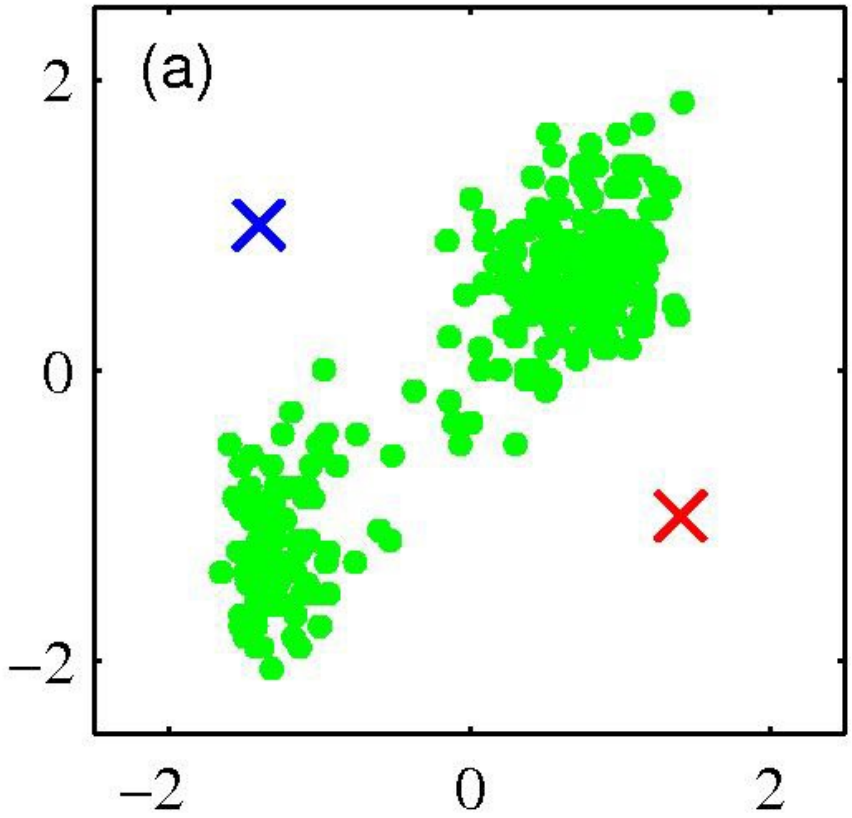
4. Repeat steps 2 and 3 until cluster centres do not change anymore

Example: “k-means clustering”



Trajectories of the means of the c-means clustering procedure applied to two-dimensional data. In this case convergence is obtained in three iterations

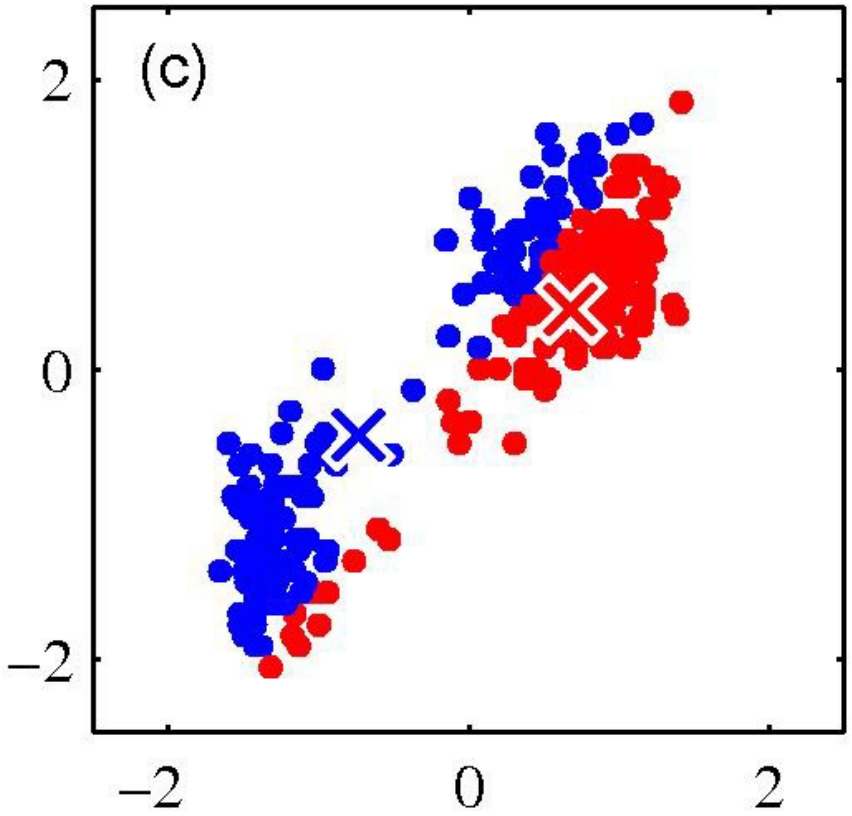
K-means clustering: Example



Initialization:
Pick K random points as cluster centers

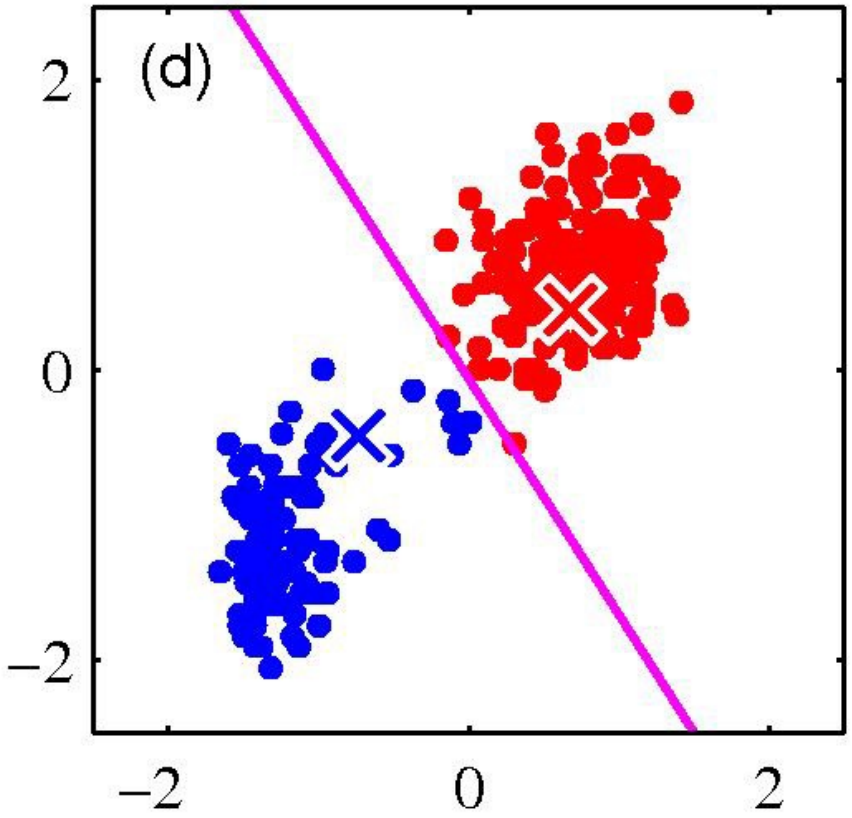
Shown here for K=2

K-means clustering: Example



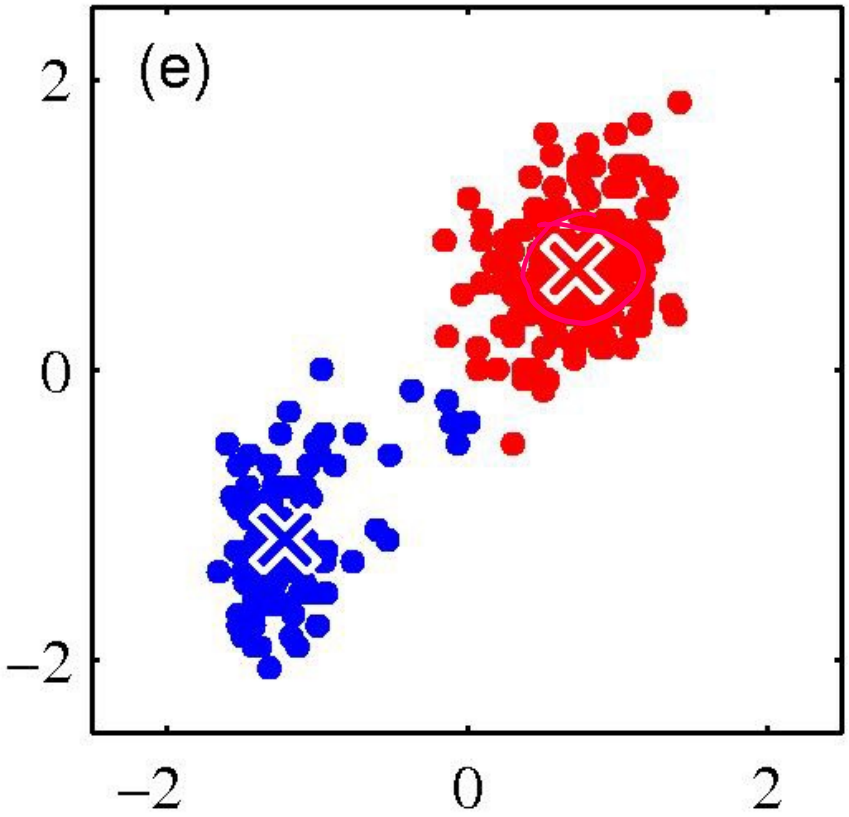
Iterative Step 2:
Change the cluster center to the average of the assigned points

K-means clustering: Example



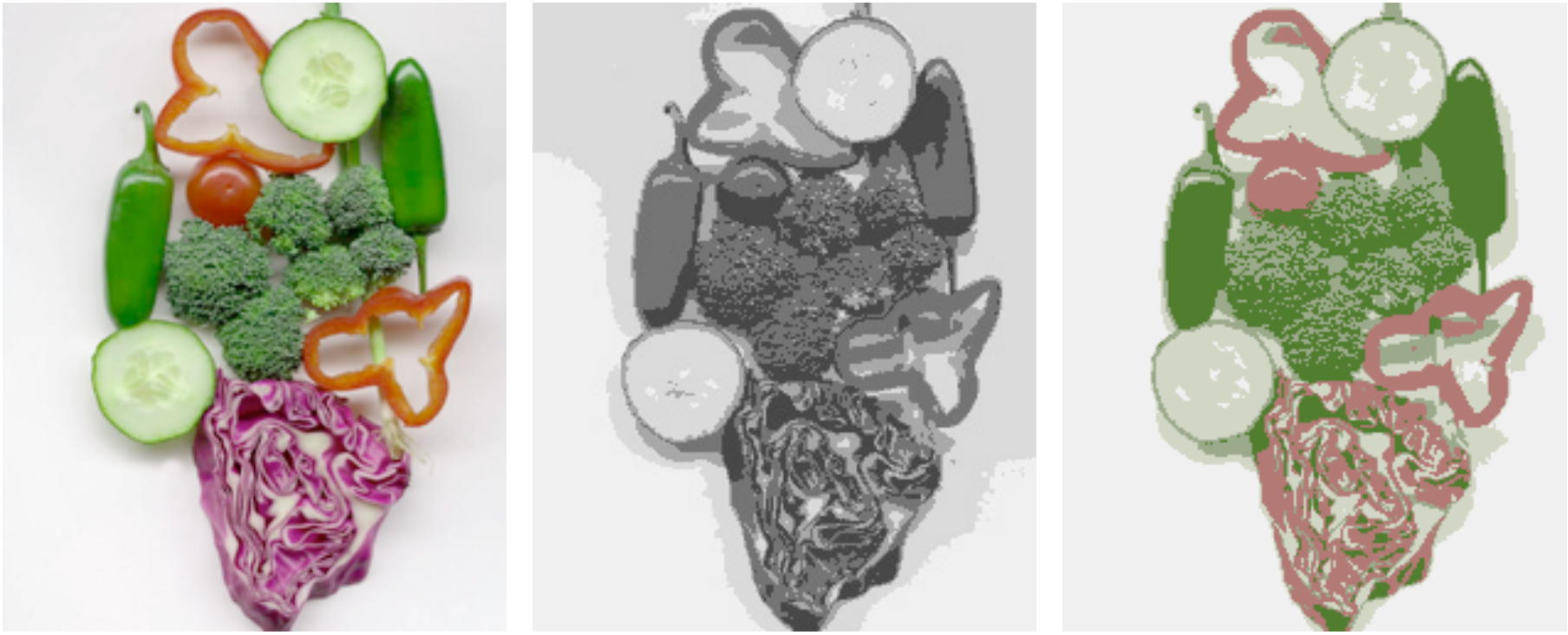
Repeat until convergence

K-means clustering: Example



Final output

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

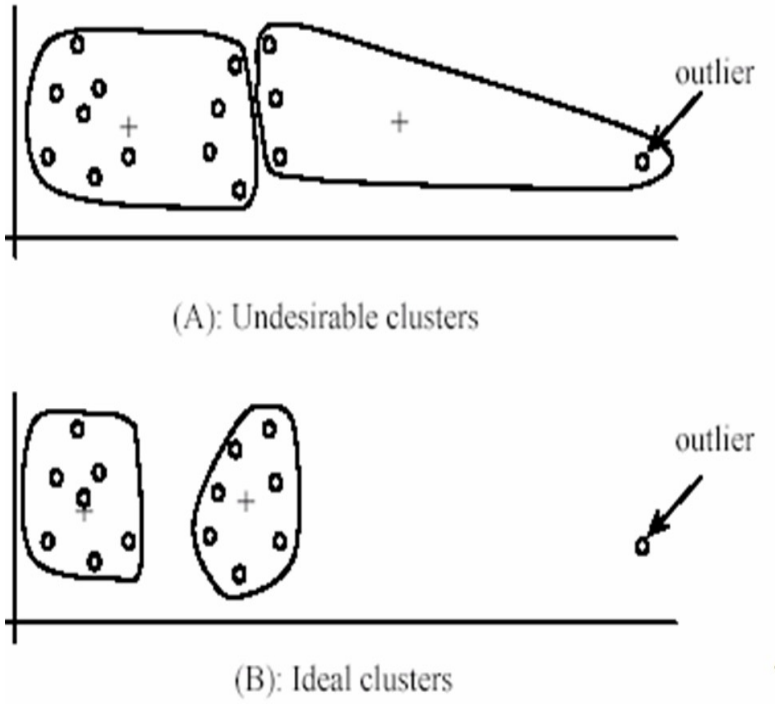
Properties of K-means

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



Key issues with clustering

- Several issues arise that are related with clustering and that are not addressed by this course. Nevertheless, we mention them for the sake of completeness.

- ✓ The clusters found do actually reflect existing groupings in the problem domain (*natural clusters*) or have just been "forced" by the clustering algorithm?
- ✓ How many clusters should we search for?
- ✓ How can cluster validity be measured?
- ✓ How can a good similarity measure be defined ?

Cluster validation functions: an example

- Let us define n_i as the number of samples in "cluster" D_i, and m_i the mean of the samples within D_i.

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}$$

- We can also define the sum-of-squared-errors as:

$$J_e = \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2 = \sum_{i=1}^{c} J_i; \qquad J_i = \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

- For each "cluster" D_i $\mathbf{m}_i$ represents patterns D_i in the sense that it minimizes the sum-of-squared-errors $(\mathbf{x}-\mathbf{m}_i)^2$.

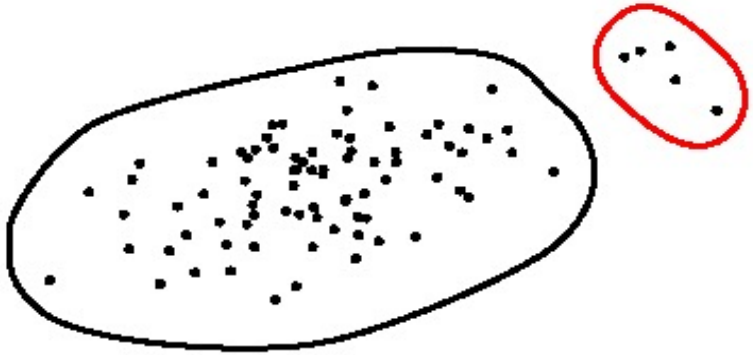
➢ J_e is an intuitive and straightforward way to evaluate clustering validity.

Cluster validation functions: an example

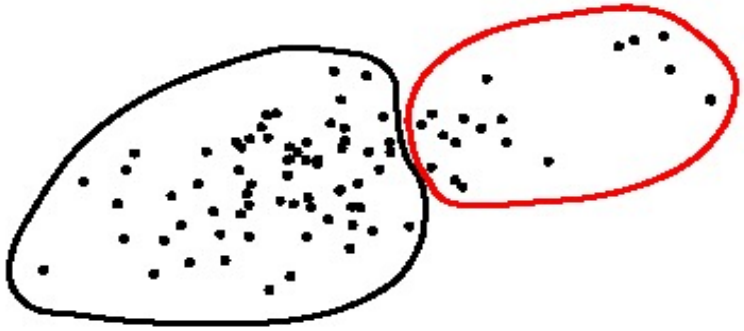
- The value of J_e depends on both the number of clusters and how patterns are spread around clusters.
  - The *optimal clustering* is the one which minimizes J_e.
  - "Clusters" obtained according to this criterion are called minimum variance clusters

- Unfortunately, not always J_e is a good cluster validity measure.
- It works well only if:
  - Clusters are both compact and well separated from each other
  - Clusters have almost the same size (in terms of number of patterns included)

Cluster validation functions: an example



$J_e = large$



$J_e = small$

- When two natural groupings are strongly different in terms of number of samples a clustering algorithm based on minimizing J_e can achieve misleading results.
- In the example here, the value of J_e is higher for the clustering above (which is the correct one) than for the clustering below.

Cluster validation: other functions

- Many other functions do exist that can be used to evaluate clustering quality
- Most of them privilege clustering algorithms that do produce "*compact and well separated*" clusters.

- The concept of "compact and well separated" cluster can be measured concretely by using the "within-cluster scatter matrix" S_W and the "between-cluster scatter matrix" S_B

$$\mathbf{S}_W = \sum_{i=1}^{c} \mathbf{S}_i \qquad \mathbf{S}_i = \sum_{\mathbf{x} \in D_i} \left(\mathbf{x} - \mathbf{m}_i\right)\left(\mathbf{x} - \mathbf{m}_i\right)^t$$

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i \left(\mathbf{m}_i - \mathbf{m}\right)\left(\mathbf{m}_i - \mathbf{m}\right)^t$$

Fuzzy *k*-means clustering

- For every iteration of the classical k-means procedure, each data point is assumed to be in exactly one cluster. We can relax this condition and assume that each sample has some degree of "membership" to a cluster.
- This degree of membership is basically equivalent to

$$P(\omega_i \mid \mathbf{x}_k, \boldsymbol{\theta})$$

- More formally, the fuzzy k-means clustering seeks a minimum of a heuristic global cost function

$$J_{fuz} = \sum_{i=1}^{c} \sum_{j=1}^{n} \left(\left[P(\omega_i \mid \mathbf{x}_j, \boldsymbol{\theta}) \right]^b \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2 \right)$$

Fuzzy *k*-means clustering

- If the parameter b is set to 0, we return to the sum-of-squared-errors criterion
- For $b > 1$, the criterion allows each pattern to belong to multiple clusters.
- At the solution (i.e., the minimum of $J_{fuz)}$ we have:

$$\frac{\partial J_{fuz}}{\partial \boldsymbol{\mu}_i} = 0; \quad \frac{\partial J_{fuz}}{\partial P(\omega_i)} = 0$$

- It can be shown that the solution that meets the above two constraints leads to the following solutions (see next slide):

Fuzzy *k*-means clustering

$$g) \quad \boldsymbol{\mu}_i = \frac{\sum_{j=1}^{n} [\hat{P}(\omega_i \mid \boldsymbol{x}_j)]^b \boldsymbol{x}_j}{\sum_{j=1}^{n} [\hat{P}(\omega_i \mid \boldsymbol{x}_j)]^b}$$

$$h) \quad \hat{P}(\omega_i \mid \boldsymbol{x}_j) = \frac{(1/d_{ij})^{1/(b-1)}}{\sum_{r=1}^{c} (1/d_{rj})^{1/(b-1)}} \; and \quad d_{ij} = \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_j \right\|^2$$

In general, the J_{fuz} criterion is minimized when the cluster centers μ_j are near to those points that have high estimated probability of being in cluster j.

Because equations g) and h) rarely have analytic solutions, the cluster means and point probabilities are once again estimated iteratively.

Fuzzy *k*-means clustering

Fuzzy *k*-Means Clustering Algorithm

begin initialize $n, c, b, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_c, P(\omega_i \mid \mathbf{x}_j), i = 1 \ldots c, j = 1 \ldots n$

normalize $P(\omega_i \mid \mathbf{x}_j)$

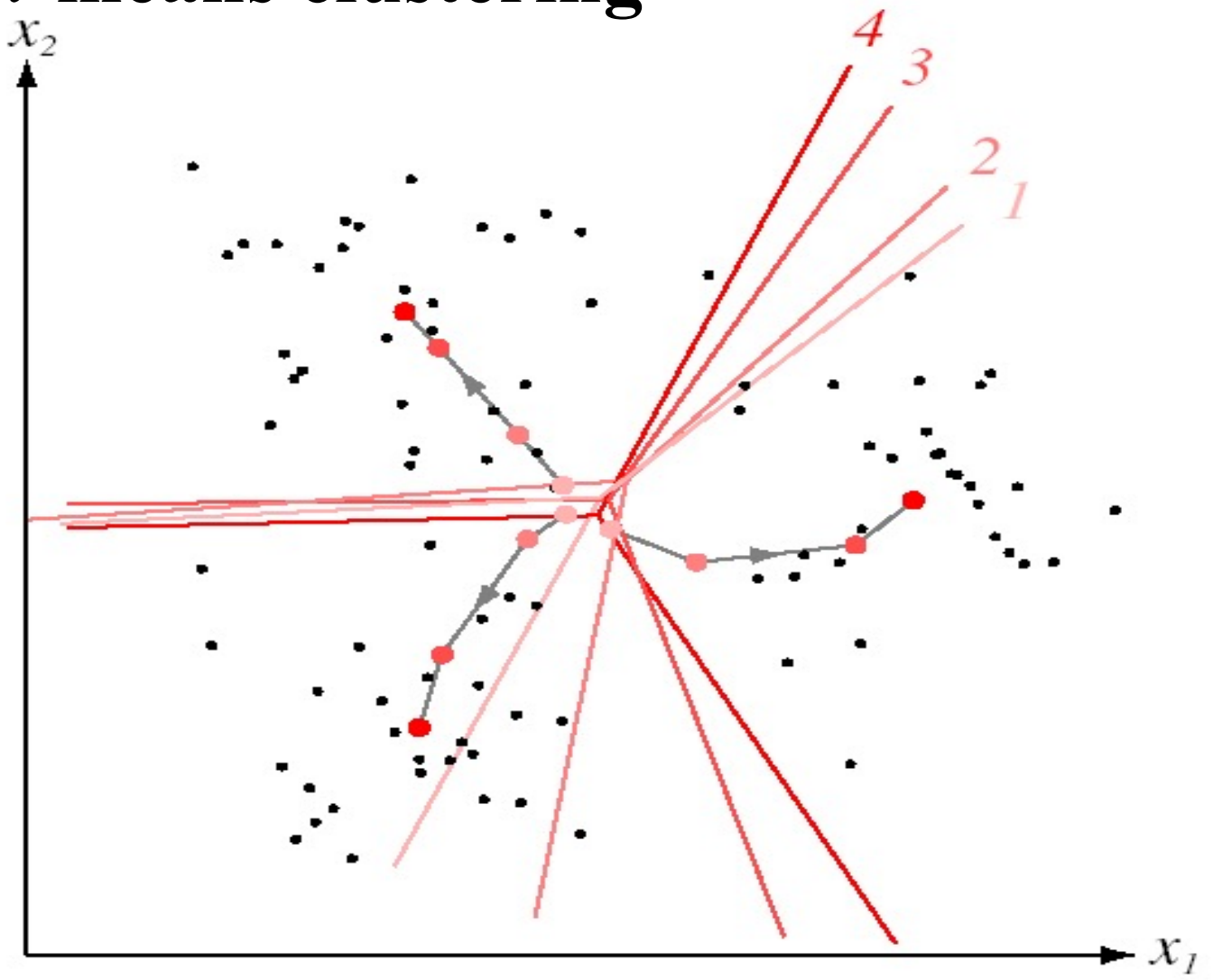
do recompute $\boldsymbol{\mu}_i$ by equation g)

recompute $P(\omega_i \mid \mathbf{x}_j)$ by equation h)

until small change in $\boldsymbol{\mu}_i$ e $P(\omega_i \mid \mathbf{x}_j)$

return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_c$

Fuzzy *k*-means clustering



At each iteration the probability of category memberships for each point are adjusted. After four iterations, the algorithms has converged to the red cluster centers and associated Voronoi tessellation.

Fuzzy *k*-means clustering

- The incorporation of probabilities (as graded memberships) sometimes improves the convergence of the fuzzy k-means over its classical counterpart.

- One drawback of the method is that according to the normalization

$$\sum_{i=1}^{c} \hat{P}(\omega_i \mid \mathbf{x}_j) = 1; \quad j = 1,\ldots,n$$

The probability of membership of a point $\mathbf{x}_j$ in a cluster i depends implicitly on the guessed number of clusters. A wrong estimate in this sense will probably affect the validity of the resulting clustering.

"Clustering" with iterative optimization

•Let us consider now the use of iterative improvement to minimize the sum-of-squared-error J_e written as:

$$J_e = \sum_{i=1}^{c} J_i$$

where the error for each cluster is defined as

$$J_i = \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

and the mean for each cluster is defined us

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}$$

Iterative optimization

- If the pattern $\mathbf{x}'$ assigned to cluster D_i is moved to D_j , $\mathbf{m}_j$ and J_j change consequently:

$$\mathbf{m}_j^* = \mathbf{m}_j + \frac{\mathbf{x}' - \mathbf{m}_j}{n_j + 1}$$

$$J_j^* = \sum_{\mathbf{x} \in D_j} \left\| \mathbf{x} - \mathbf{m}_j^* \right\|^2 + \left\| \mathbf{x}' - \mathbf{m}_j^* \right\|^2 =$$

$$= \sum_{\mathbf{x} \in D_j} \left(\left\| \mathbf{x} - \mathbf{m}_j - \frac{\mathbf{x}' - \mathbf{m}_j}{n_j + 1} \right\|^2 \right) + \left\| \frac{n_j}{n_j + 1} \left(\mathbf{x}' - \mathbf{m}_j \right) \right\|^2 =$$

$$= J_j + \frac{n_j}{n_j + 1} \left\| \mathbf{x}' - \mathbf{m}_j \right\|^2$$

Iterative optimization

•Obviously, also values related to cluster D_i do change. Under the ipothesys ($n_i > 1$) they become:

$$\mathbf{m}_i^* = \mathbf{m}_i + \frac{\mathbf{x}' - \mathbf{m}_i}{n_i - 1}$$

$$J_i^* = J_i - \frac{n_i}{n_i - 1}\left\|\mathbf{x}' - \mathbf{m}_i\right\|^2$$

•The transfer of $\mathbf{x}'$ from D_i to D_j is advantageous if the decrease in J_i is greather than the increase in J_j, so that the value of J_e decreases.

Iterative optimization

- We observe a decrease in J_e when

$$\frac{n_i}{n_i - 1}\left\|\mathbf{x}' - \mathbf{m}_i\right\|^2 > \frac{n_j}{n_j + 1}\left\|\mathbf{x}' - \mathbf{m}_j\right\|^2$$

which typically happens whenever $\mathbf{x}'$ is closer to $\mathbf{m}_j$ than $\mathbf{m}_i$.

- If this reassignment is profitable, the greatest decrease in J_e is obtained by selecting the cluster for which

$$n_j \frac{\left\|\mathbf{x}' - \mathbf{m}_j\right\|^2}{n_j + 1}$$

is minimum.

Iterative optimization

- Previous considerations lead to the following clustering procedure:

Basic Iterative Minimum - Square - Error Clustering

1 **begin** initialize $n, c, \mathbf{m}_1, \ldots, \mathbf{m}_c$

2 **do** randomly select a pattern $\mathbf{x}'$

3 $i \leftarrow \arg\min_{i'} \|\mathbf{m}_{i'} - \mathbf{x}'\|$ (classify $\mathbf{x}'$)

4 **if** $n_i \neq 1$ **then** *compute*

5 $$\rho_j = \begin{cases} \dfrac{n_j}{n_j + 1}\|\mathbf{x}' - \mathbf{m}_j\|^2 & \text{if } j \neq i \\ \dfrac{n_j}{n_j - 1}\|\mathbf{x}' - \mathbf{m}_i\|^2 & \text{if } j = i \end{cases}$$

6 **if** $\rho_k \leq \rho_j \ \forall j$ **then** transfer $\mathbf{x}'$ in D_k

7 recompute J_e, $\mathbf{m}_i$, $\mathbf{m}_k$

8 **until** no change in J_e n attempts

9 **end**

Iterative optimization

- The iterative optimization algorithm is quite similar to the *k-means* algorithm
  - With *k*-means, means are re-computed once all the patterns have been re-classified.
  - With iterative optimization, means are re-computed every time a single pattern is re-classified.

- The final result is affected:
  - By the presence of local minima
  - By the order according to which patterns are selected

References

- Sections 10.1, 10.4, 10.6, 10.7, 10.8, 10.10, Pattern Classification, R. O. Duda, P. E. Hart, and D. G. Stork, John Wiley & Sons, 2000