Part 7:
Elements of Data Clustering

Battista Biggio
battista.biggio@unica.it

Introduction

- **Supervised learning** assumes that the training samples used to design a classifier are labeled according to their class membership

- We describe here a number of **unsupervised** procedures that use **unlabeled data**

- Two terms are commonly used to indicate the topics we are going to deal with:
  - Unsupervised learning
  - Data clustering

Five Basic Reasons for Unsupervised Learning

1. Collecting and labeling a large set of patterns can be extremely costly
   - e.g., speech recognition, malware detection

2. It might be more convenient to proceed in the reverse direction:
   - using large amounts of unlabeled data to identify "clusters" and then
   - using supervision to label the groupings found

3. The data distribution can change over time
   - An unsupervised model can track these changes and achieve improved performance

4. We can find relevant features that will then be useful for categorization

5. We can gain useful insights into the underlying data structure

Basic Concepts

- The goal of unsupervised learning (**clustering**) is to find groupings in the data (**clusters**) that reflect the ground truth and the "natural properties" of the data

- Even if intuitive, the concept of cluster is hard to define rigorously, both in general and even in very specific cases
  - We can informally say that *samples belonging to the same cluster must present a higher degree of similarity than that shown among 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:

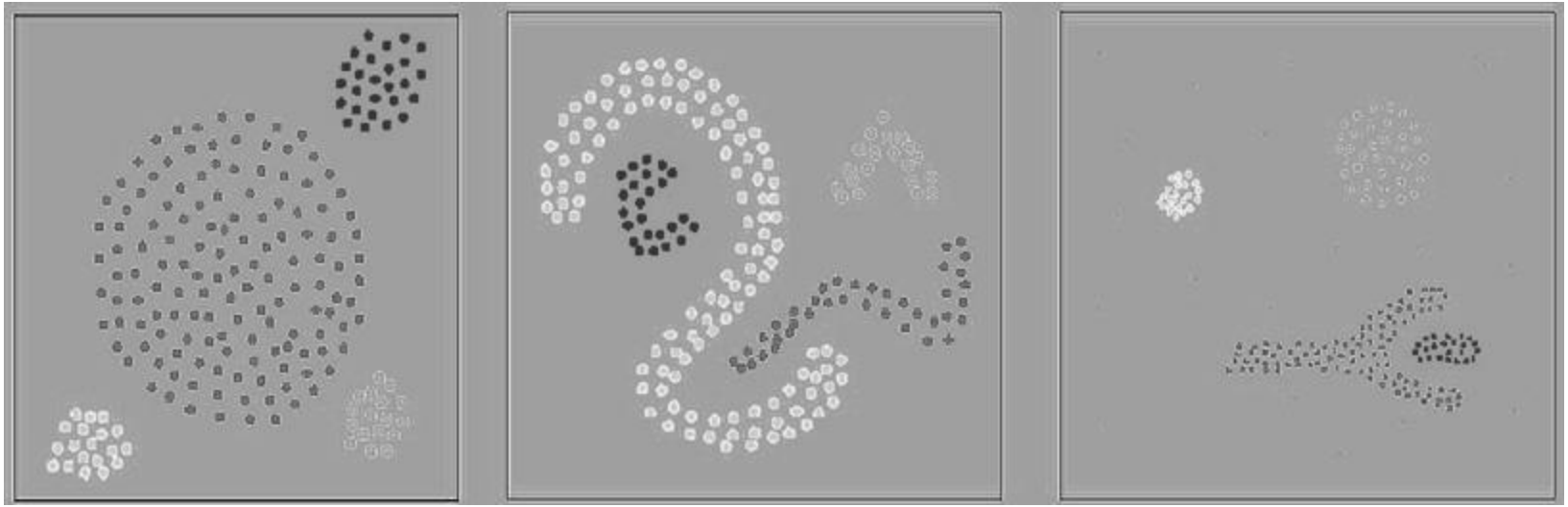
$$D^j \neq \emptyset \quad D$$

$$D^i \cap D^j = \emptyset \quad i \neq j$$

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

Simple Clustering Examples

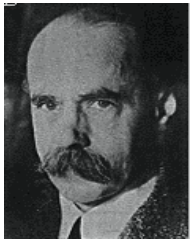
- It is straightforward to see that even if intuitive, the concept of cluster is hard to define!



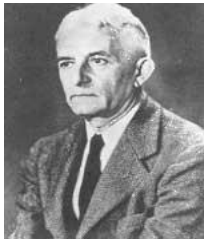
Basic Ideas of Grouping in Humans: The Gestalt School

- **Gestalt qualities (*Gestaltqualitat*).** Elements in a collection of elements can have properties that result from relationships
  - A series of factors (*Gestalt factors*) affect whether elements should be grouped together

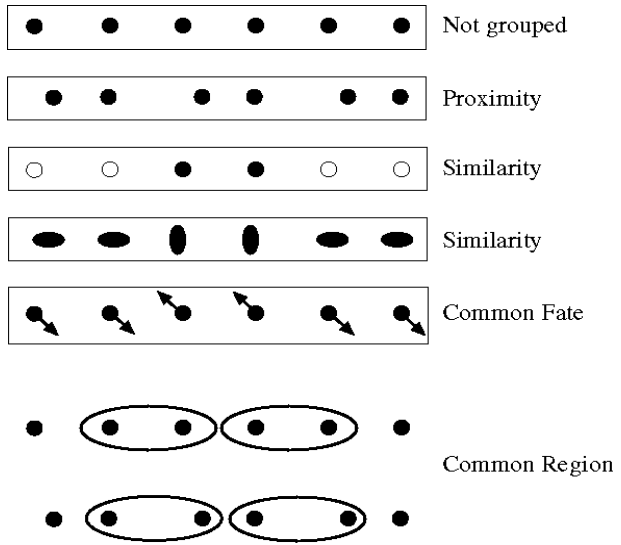
Wertheimer



Koffka



Koehler



Clustering

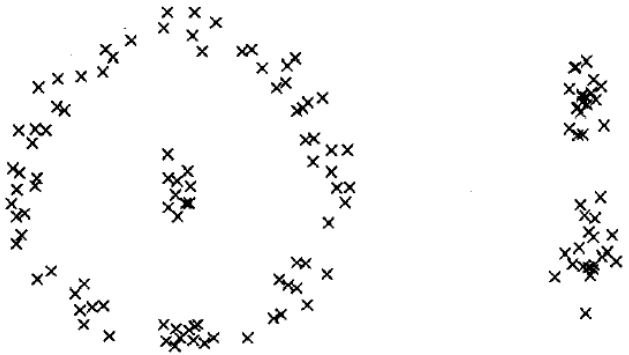
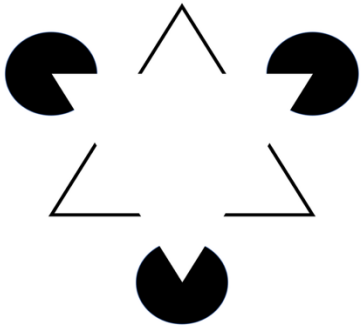


Figure 1: How many groups?

Clustering, Optical Illusions, and Cognitive Biases



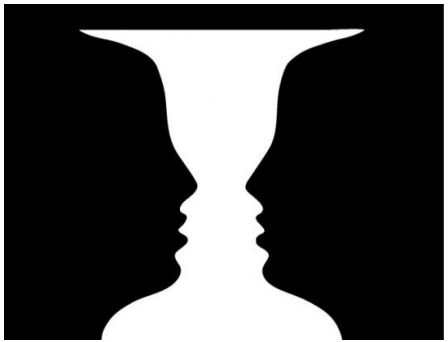
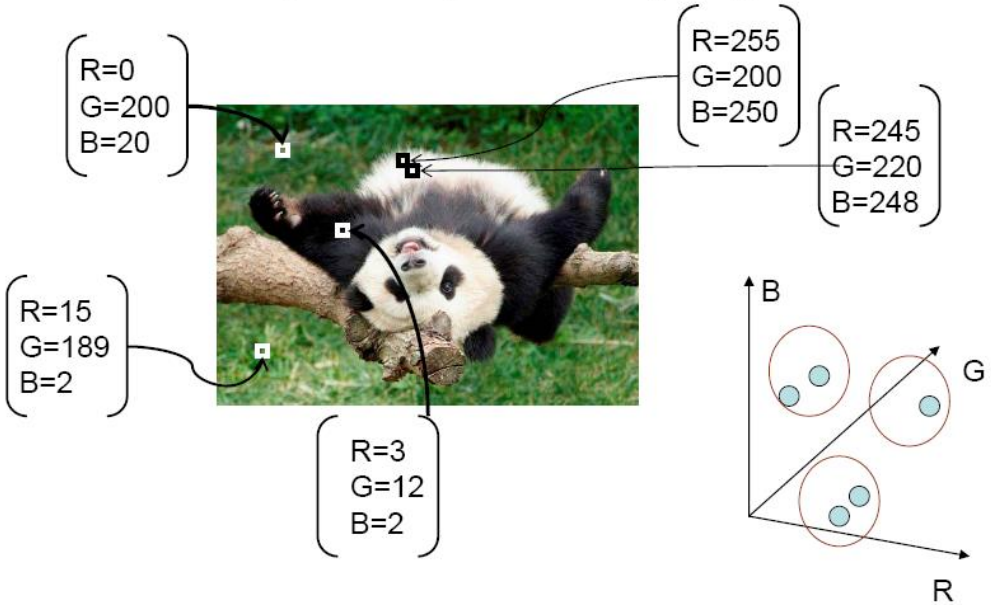


Image Segmentation as Clustering

- Cluster similar pixels (features) together



Source: K. Grauman

Clustering Algorithms: Categorization

Clustering algorithms can be grouped into different categories

- **Connectivity-based (hierarchical) clustering**
  - Linkage clustering is a simple example
- **Centroid-based clustering**
  - The K-means algorithm is a simple example
- **Model-based clustering**
  - Gaussian mixture
- **Others: graph-based (*spectral*), density-based …**

Linkage Clustering

- **Initialization:** Each sample is in a cluster of its own (*singleton*)
- The clusters are then sequentially combined into larger clusters until all elements end up being in the same cluster
- At each step, the two clusters separated by the **shortest distance** are combined
  - The **linkage function** defines the distance between two clusters

- **Single-linkage:** $D(C_1, C_2) = \min_{\boldsymbol{x}_1 \in C_1, \boldsymbol{x}_2 \in C_2} d(\boldsymbol{x}_1, \boldsymbol{x}_2)$
- **Complete/Maximum-linkage:** $D(C_1, C_2) = \max_{\boldsymbol{x}_1 \in C_1, \boldsymbol{x}_2 \in C_2} d(\boldsymbol{x}_1, \boldsymbol{x}_2)$
- **Average-linkage:** $D(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{\boldsymbol{x}_1 \in C_1} \sum_{\boldsymbol{x}_1 \in C_2} d(\boldsymbol{x}_1, \boldsymbol{x}_2)$
- **Centroid-linkage:** $D(C_1, C_2) = \left|\left|\mu_1 - \mu_2\right|\right|^2$, being μ_k the centroid of C_k

- The sample-wise distance $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$ can be: L2, L1, Mahalanobis, etc.

Single-linkage Clustering: Example

- Given the sample-wise distances among 5 samples, find the single-linkage clusters
- **Step 1:** group the closest samples

	a	b	c	d	e
a	0	17	21	31	23
b	17	0	30	34	21
c	21	30	0	28	39
d	31	34	28	0	43
e	23	21	39	43	0

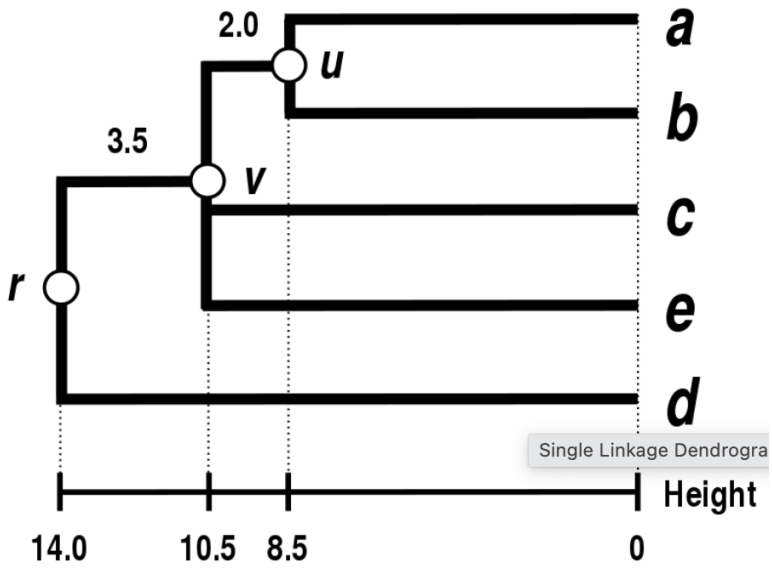
Single-linkage Clustering: Example

- **Steps 2 and 3:** Recompute distances w.r.t. the new cluster, and then aggregate again

	(a,b)	c	d	e
(a,b)	0	**21**	**31**	**21**
c	**21**	0	*28*	*39*
d	**31**	*28*	0	*43*
e	**21**	*39*	*43*	0

	((a,b),c,e)	d
((a,b),c,e)	0	**28**
d	**28**	0

Single-linkage Clustering: Dendrogram



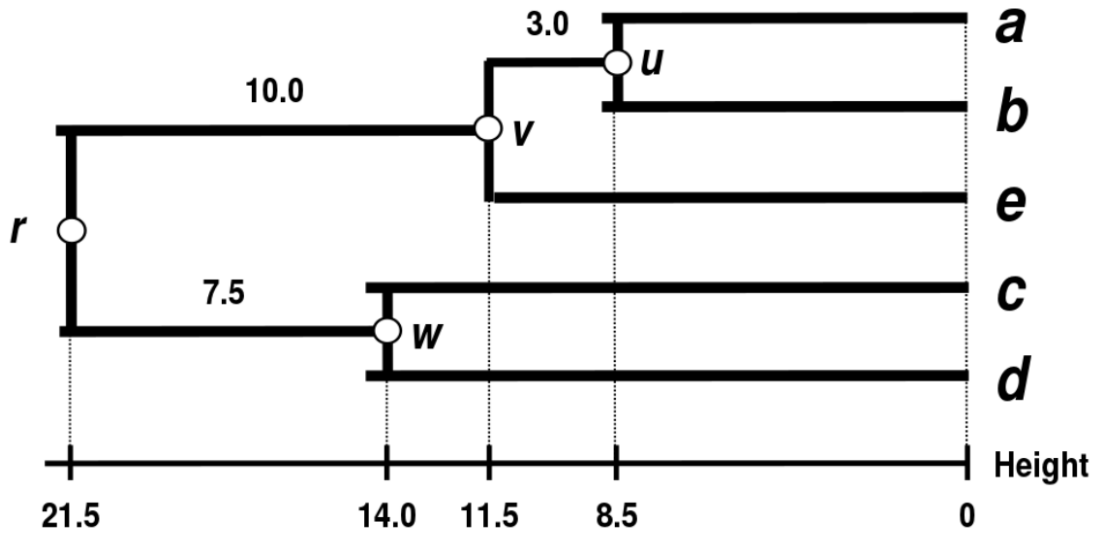
Complete-linkage Clustering: Example

- The first step is the same as for single-linkage, aggregating **(a,b)**
- **The difference is how we compute the distance w.r.t. the corresponding cluster!**

	(a,b)	c	d	e
(a,b)	0	**30**	**34**	**23**
c	**30**	0	*28*	*39*
d	**34**	*28*	0	*43*
e	**23**	*39*	*43*	0

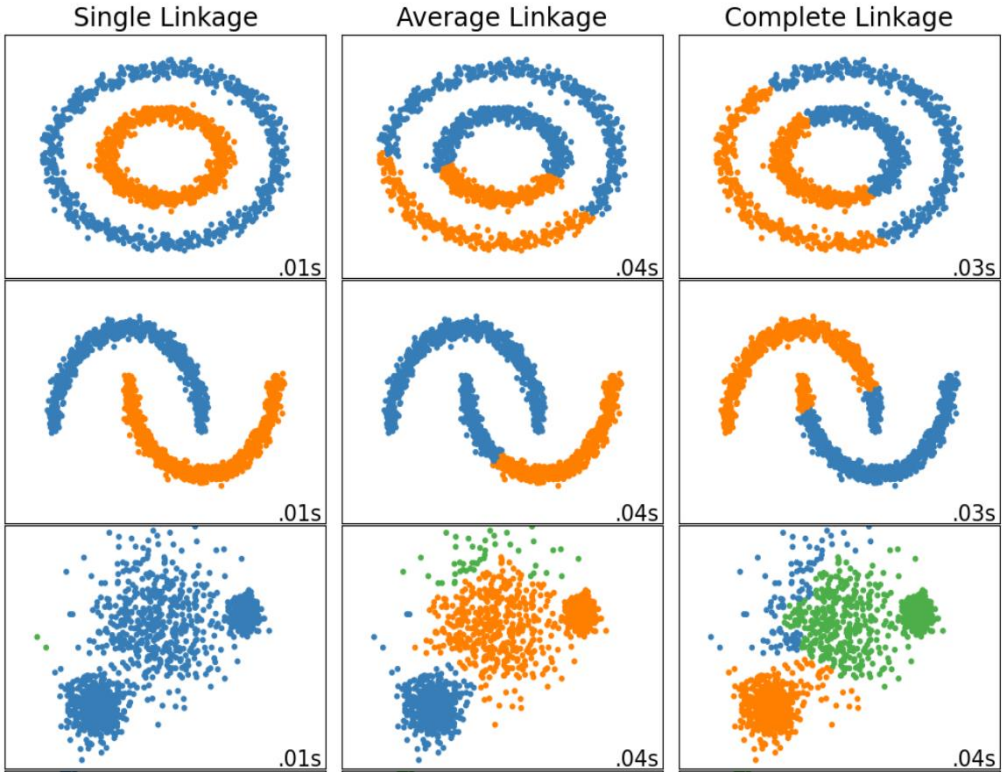
	((a,b),e)	(c,d)
((a,b),e)	0	**43**
(c,d)	**43**	0

Complete-linkage Clustering: Dendrogram



How Do the Clusters Look Like?

- At the end of the process, the **dendrogram** can be cut to retrieve the desired number of clusters
- **Single-linkage** tends to follow "paths" that connect samples
- **Complete-** and **average-linkage** tend to find more spherical clusters



K-means Clustering

Goal: minimize the objective / distortion function J

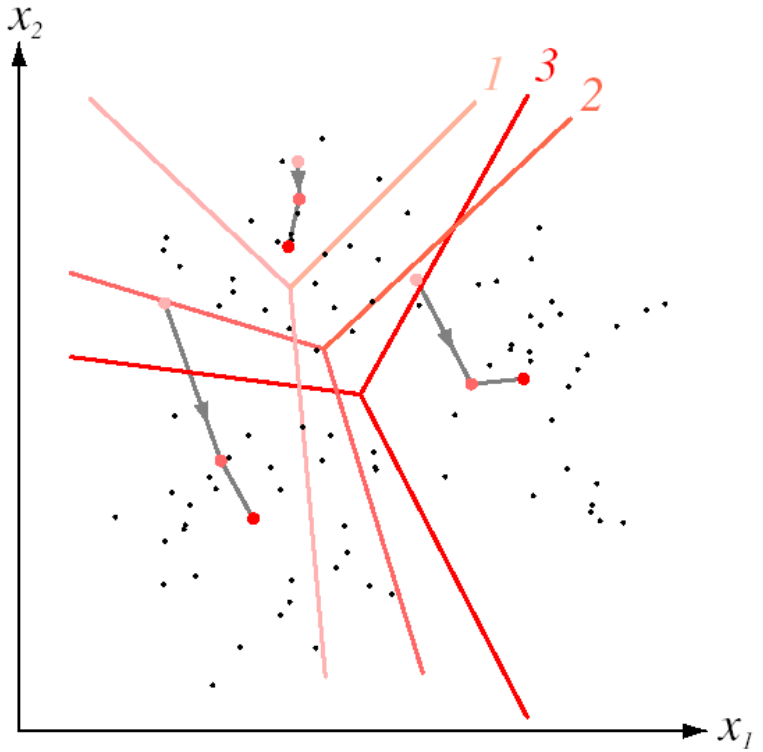
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

$r_{nk} = 1$ **if sample n belongs to cluster k, 0 otherwise**

1. Set the number K of clusters to be found and a distance measure d(a,b) between samples (e.g., L2)
2. Initialize the algorithm by defining K cluster centers
   - K points from D can be randomly selected as centers
3. Assign each point in D to the cluster whose centroid is the closest one (**expectation-step, E-step**)
4. Recompute cluster centers (**maximization step, M-step**)
5. Repeat steps 3 and 4 until cluster centers do not change anymore

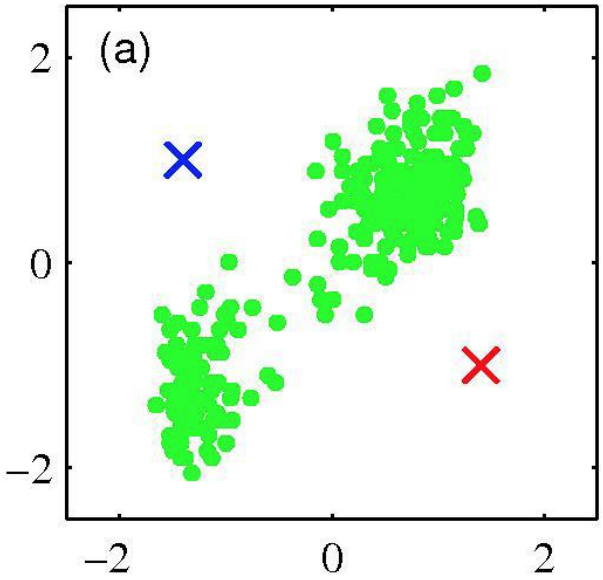
K-means Clustering: Example

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



K-means Clustering: Example

Bishop, PRML, Chapter 9

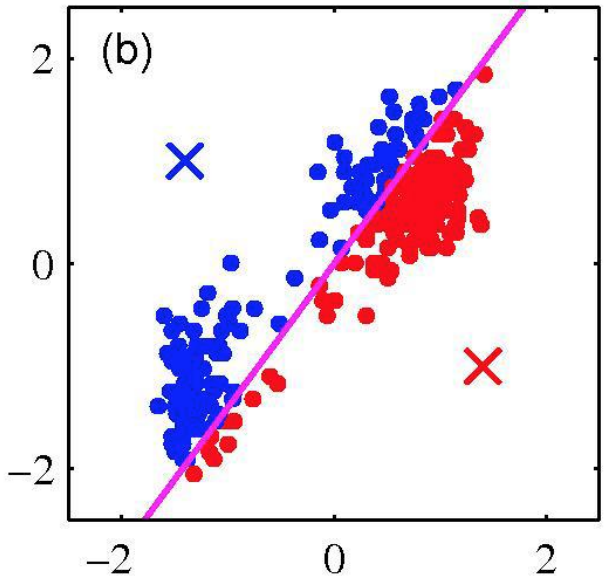


Initialization:
Pick K random points as cluster centers

Shown here for K=2

K-means Clustering: Example

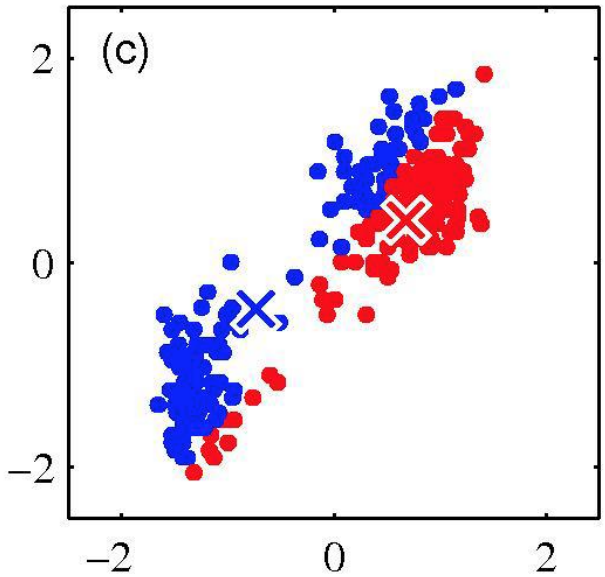
Bishop, PRML, Chapter 9



Iterative Step 1:
Assign data points to closest cluster center

K-means Clustering: Example

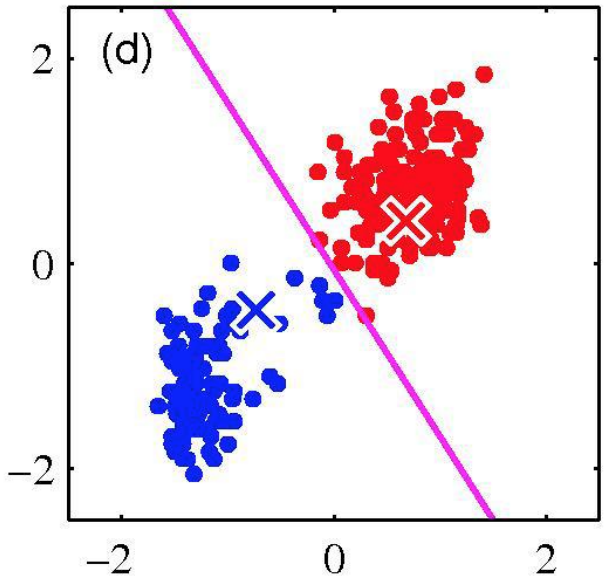
Bishop, PRML, Chapter 9



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

K-means Clustering: Example

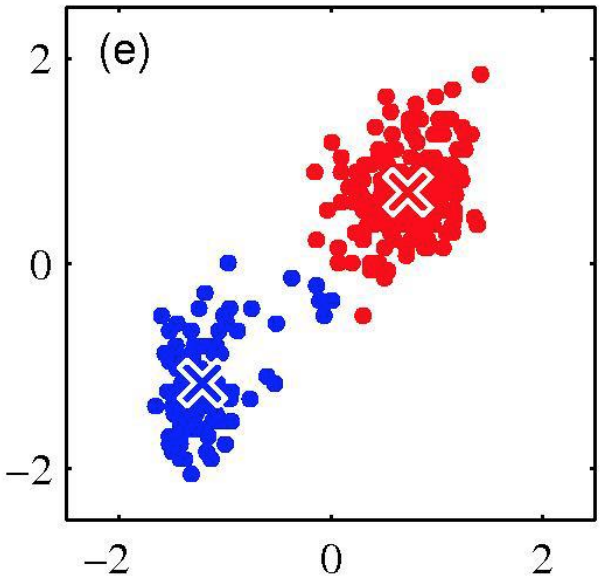
Bishop, PRML, Chapter 9



Repeat until convergence

K-means Clustering: Example

Bishop, PRML, Chapter 9



Final output

K-means Clustering for Quantization and Compression

Bishop, PRML, Chapter 9

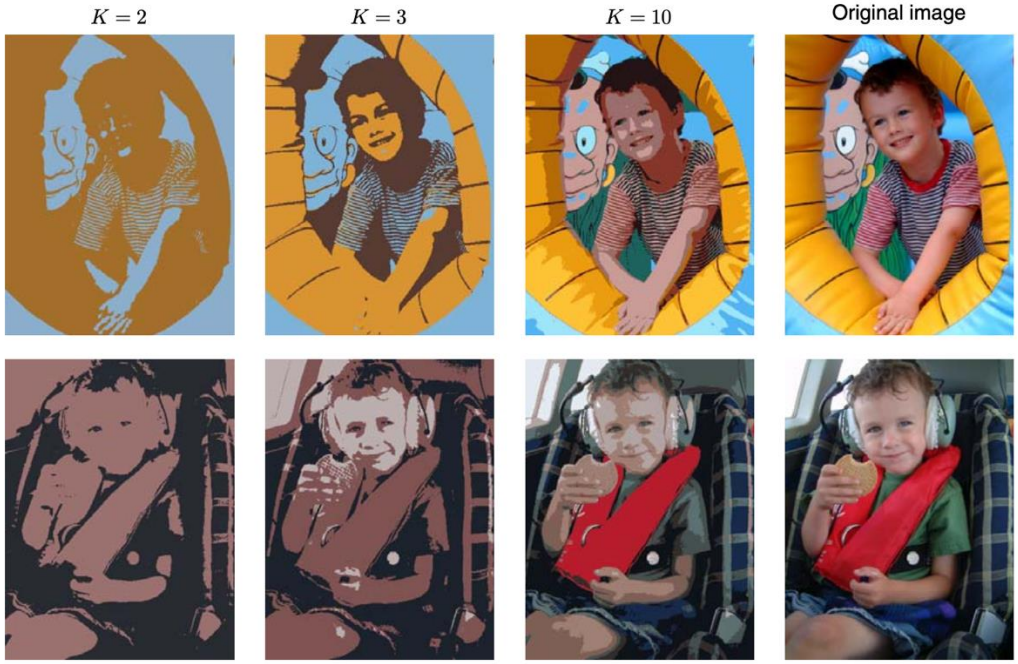


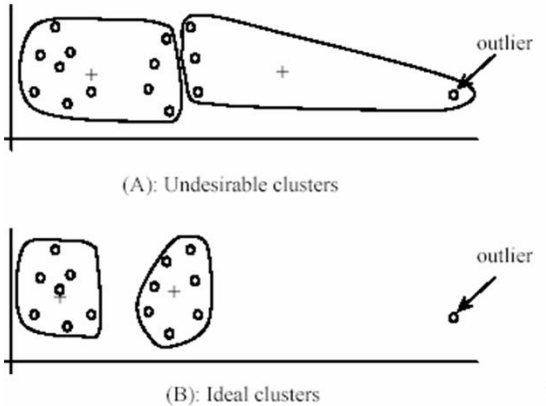
Figure 9.3 Two examples of the application of the K-means clustering algorithm to image segmentation showing the initial images together with their K-means segmentations obtained using various values of K. This also illustrates of the use of vector quantization for data compression, in which smaller values of K give higher compression at the expense of poorer image quality.

Properties of K-means

- Guaranteed to converge in a finite number of steps

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

- **Pros:** Very simple, efficient
- **Cons:**
  - Need to know the number of desired clusters K
  - Sensitive to initialization and outliers
    - Converges to a local minimum of the error function
  - Only finds *spherical* clusters



Gaussian Mixture Models (GMMs)

Bishop, PRML, Chapter 9

- K-means is a special case of **GMM clustering**
- **GMMs** assume that each cluster is Gaussian, and find their parameters (priors, means and covariances) to maximize the likelihood of generating the given data

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Each point belongs to each cluster with a certain probability (**fuzzy clustering**)
- **Expectation-maximization (EM)** is used to maximize the log-likelihood

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

EM for Gaussian Mixture Models (GMMs)

Bishop, PRML, Chapter 9

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.23}$$

EM for Gaussian Mixture Models (GMMs)

Bishop, PRML, Chapter 9

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

where

$$N_k = \sum_{n=1}^{N}\gamma(z_{nk}). \tag{9.27}$$

EM for Gaussian Mixture Models (GMMs)

Bishop, PRML, Chapter 9

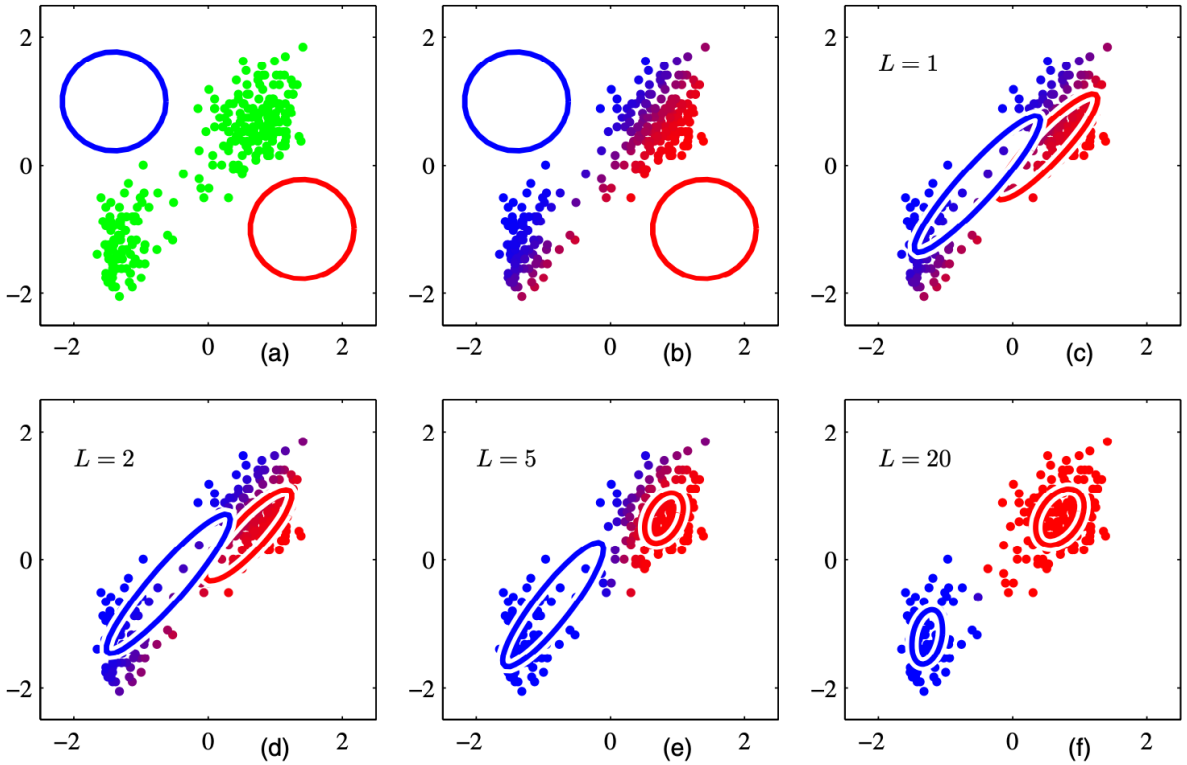
4. Evaluate the log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{9.28}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

GMM Clustering via EM: Example

Bishop, PRML, Chapter 9



Key Issues with Clustering

- Several issues related to clustering should be considered
  - We mention them here for the sake of completeness
- Do the clusters found reflect existing groupings in the problem domain (natural clusters), or have they 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 the cluster centroids and the sum-of-squared-errors respectively as

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

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

- J_e is an intuitive and straightforward way to evaluate clustering validity
  - It measures the total deviation/distortion of the clustering w.r.t. the cluster centroids

Cluster Validation Functions: An Example

- The value of J_e depends on both the number of clusters and how samples are spread around clusters.

- The optimal clustering minimizes J_e.
  - "Clusters" obtained according to this criterion are called **minimum-variance clusters**

- Unfortunately, J_e is not always a good cluster validity measure

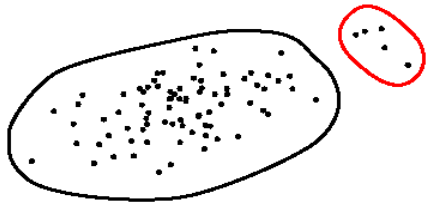
- It works well only if:
  - Clusters are both compact and well separated from each other
  - Clusters have almost the same size (in terms of the number of samples included)

Cluster Validation Functions: An Example

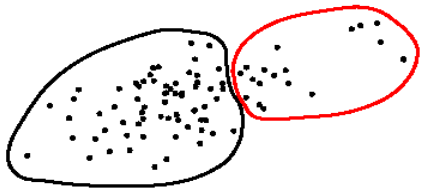
- When two natural groupings are strongly different regarding the number of samples a clustering algorithm based on minimizing J_e can achieve misleading results.

- In the example below, the value of J_e is higher for the clustering above (which is the correct one) than for the clustering below.

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



$J_e = large$



$J_e = small$

Cluster Validation Functions: Other Functions

- Many other functions do exist that can be used to evaluate clustering quality
- Most of them privilege clustering algorithms that 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

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

the lower the better

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

the higher the better

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
- C. Bishop, PRML, Chapter 9.