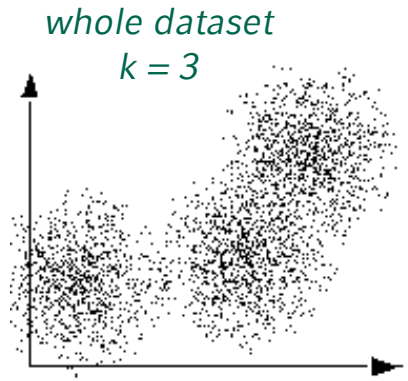
Contents

Initialization of Partitioning Clustering Methods

Just two examples:

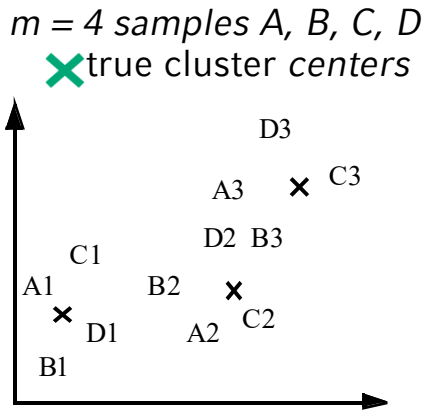
[naïve]

- Choose sample A of the dataset
- Cluster the sample and use centers as initialization



[Fayyad, Reina, and Bradley 1998]

- Choose m different (small) samples $A, \ldots, M$ of the dataset
- Cluster each sample to get m estimates for k representatives $A = (A_1, A_2, \ldots, A_k)$, $B = (B_1, \ldots, B_k)$, ..., $M = (M_1, \ldots, M_k)$
- Then, cluster the set $DS = A \cup B \cup \ldots \cup M$ m times. Each time use the centers of A, B, ..., M as respective initial partitioning
- Use the centers of the best clustering as initialization for the partitioning clustering of the whole dataset



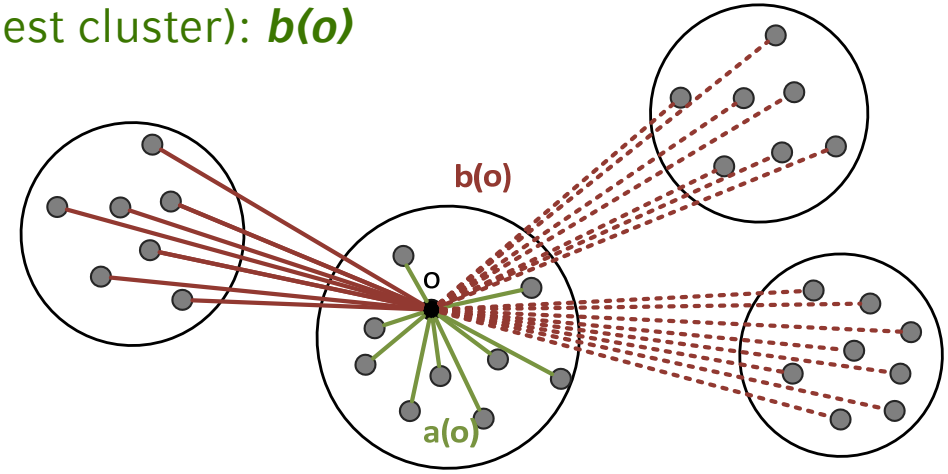
Fayyad U., Reina C., Bradley P. S., „Initialization of Iterative Refinement Clustering Algorithms", *In KDD 1998)*, pp. 194—198.

Choice of the Parameter *k*

- Idea for a method:
  - Determine a clustering for each $k = 2, \ldots, K_{max} \leq n\text{-}1$
  - Choose the "best" clustering
- But how to measure the quality of a clustering?
  - A measure should not be monotonic over *k* because the measures for the compactness of a clustering SSE and TD are monotonously decreasing with increasing value of *k*.
- Silhouette-Coefficient [Kaufman & Rousseeuw 1990]
  - Measure for the quality of a *k*-means or a *k*-medoid clustering that is not monotonic over *k*.

The Silhouette coefficient (1)

- Basic idea:
  - How good is the clustering = how appropriate is the mapping of objects to clusters
  - Elements in cluster should be „similar" to their representative
    → measure the average distance of objects to their representative: ***a(o)***
  - Elements in different clusters should be „dissimilar"
    → measure the average distance of objects to alternative clusters (i.e. second closest cluster): ***b(o)***



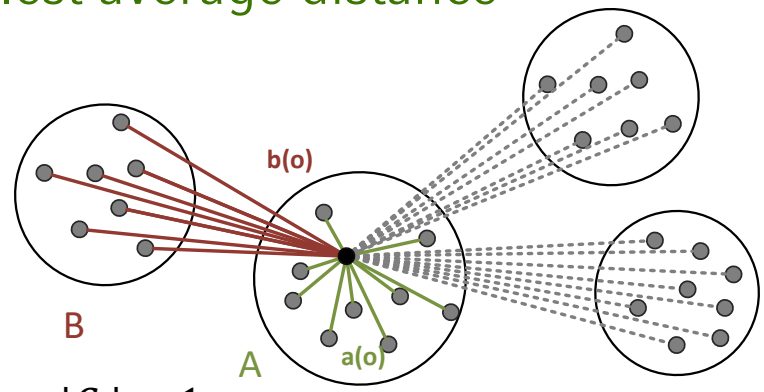
The Silhouette coefficient (2)

- $a(o)$: average distance between object o and the objects in its cluster A

$$a(o) = \frac{1}{|C(o)|} \sum_{p \in C(o)} dist(o,p)$$

- $b(o)$: for each other cluster C_i compute the average distance between o and the objects in C_i. Then take the smallest average distance

$$b(o) = \min_{C_i \neq C(o)} \left(\frac{1}{|C_i|} \sum_{p \in C_i} dist(o,p) \right)$$



- The silhouette of o is then defined as

$$s(o) = \begin{cases} 0 & if\ a(o) = 0, e.g. |C_i| = 1 \\ \dfrac{b(o) - a(o)}{\max\{a(o), b(o)\}} & else \end{cases}$$

- The values of the silhouette coefficient range from −1 to +1

The Silhouette coefficient (3)

- The silhouette of a cluster C_i is defined as:

$$silh(C_i) = \frac{1}{|C_i|} \sum_{o \in C_i} s(o)$$

- The silhouette of a clustering $\mathcal{C} = (C_1, \dots, C_k)$ is defined as:

$$silh(\mathcal{C}) = \frac{1}{|D|} \sum_{o \in D} s(o),$$

where D denotes the whole dataset.

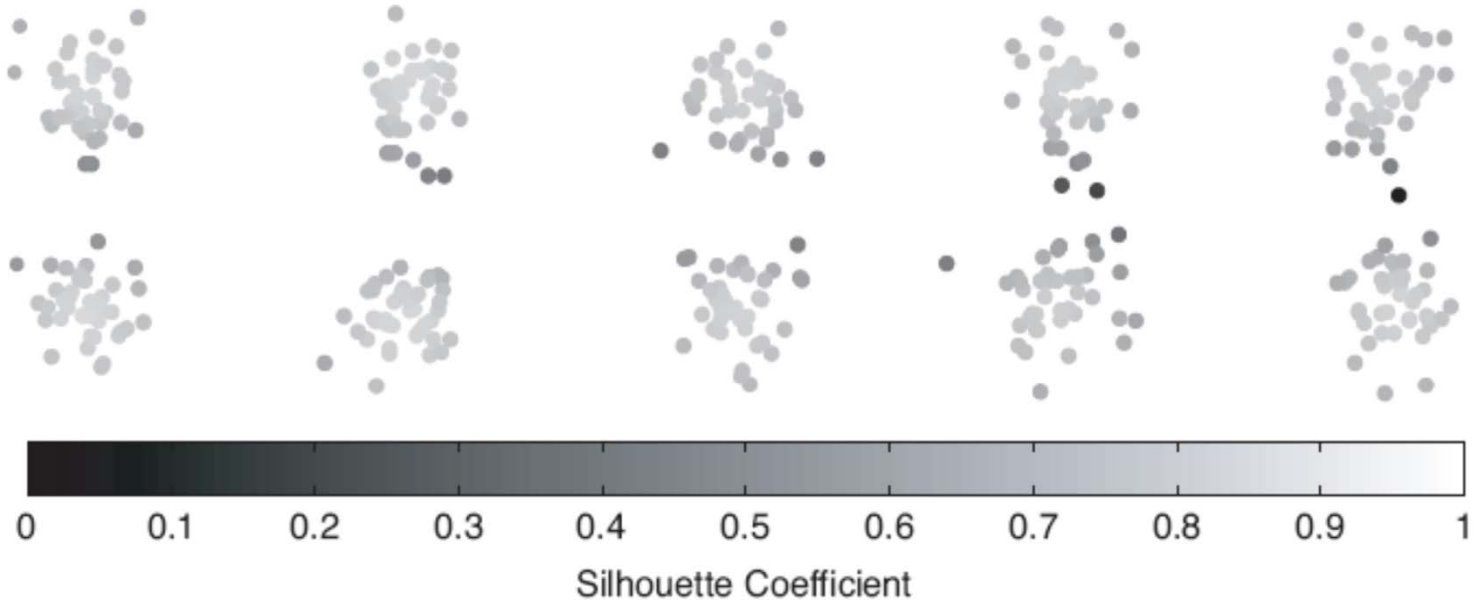
The Silhouette coefficient (4)

- „Reading" the silhouette coefficient:
  Let $a(o) \neq 0$.
  - $b(o) \gg a(o) \Rightarrow s(o) \approx 1$: good assignment of *o* to its cluster *A*
  - $b(o) \approx a(o) \Rightarrow s(o) \approx 0$: *o* is in-between *A* and *B*
  - $b(o) \ll a(o) \Rightarrow s(o) \approx -1$: bad, on average *o* is closer to members of *B*

- Silhouette Coefficient $s_{\mathcal{C}}$ of a clustering: average silhouette of all objects
  - $0.7 < s_C \leq 1.0$ strong structure, $0.5 < s_C \leq 0.7$ medium structure
  - $0.25 < s_C \leq 0.5$ weak structure, $s_C \leq 0.25$ no structure

The Silhouette coefficient (5)

Silhouette Coefficient for points in ten clusters



in: Tan, Steinbach, Kumar: Introduction to Data Mining (Pearson, 2006)

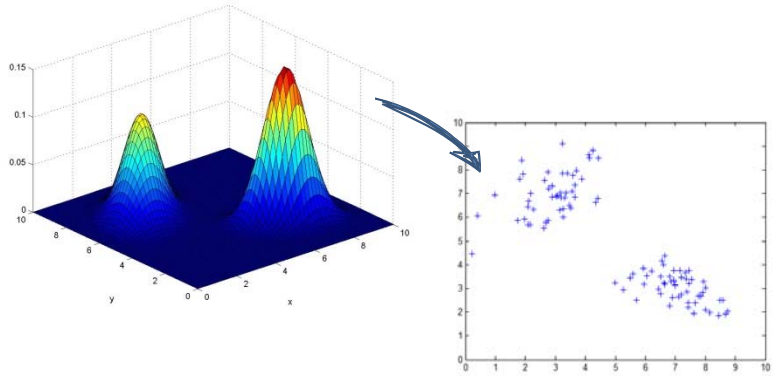
Contents

Expectation Maximization (EM)

Statistical approach for finding maximum likelihood estimates of parameters in probabilistic models

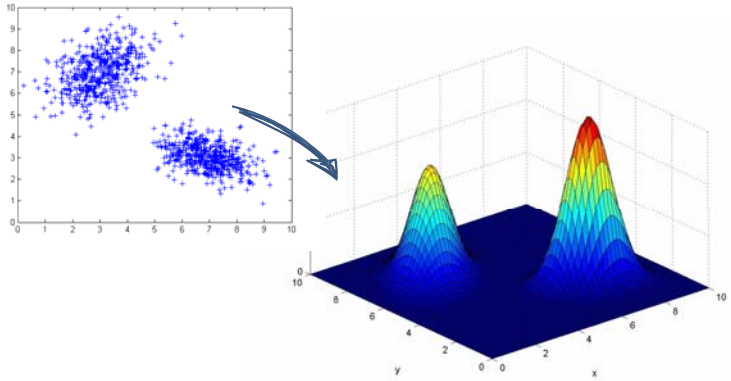
Here: using EM as clustering algorithm

Approach:
Observations are drawn from one of several components of a mixture distribution.



Main idea:

- Define clusters as probability distributions
  → each object has a certain probability of belonging to each cluster
- Iteratively improve the parameters of each distribution (e.g. center, "width" and "height" of a Gaussian distribution) until some quality threshold is reached



Additional Literature: C. M. Bishop „Pattern Recognition and Machine Learning", Springer, 2009

Excursus: Gaussian Mixture Distributions

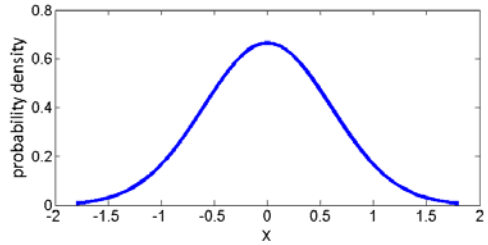
Note: EM is not restricted to Gaussian distributions, but they will serve as example in this lecture.

Gaussian distribution:

- Univariate: single variable x ∈ ℝ:

$$p(x|\mu,\sigma^2) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{1}{2\sigma^2}\cdot(x-\mu)^2}$$

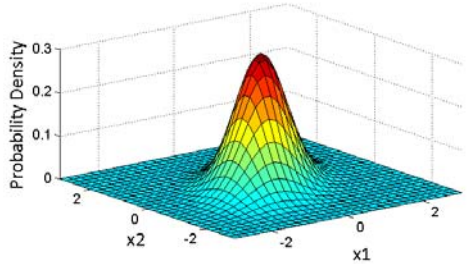
mean $\in \mathbb{R}$ *variance* $\in \mathbb{R}$



- Multivariate: d-dimensional vector $\boldsymbol{x} \in \mathbb{R}^d$:

$$p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}} \cdot e^{-\frac{1}{2}\cdot(\boldsymbol{x}-\boldsymbol{\mu})^T\cdot(\boldsymbol{\Sigma})^{-1}\cdot(\boldsymbol{x}-\boldsymbol{\mu})}$$

mean vector $\in \mathbb{R}^d$ *covariance matrix* $\in \mathbb{R}^{d\times d}$

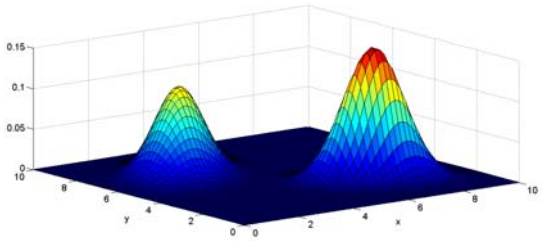


Gaussian mixture distribution with K components:

- d-dimensional vector $\boldsymbol{x} \in \mathbb{R}^d$:

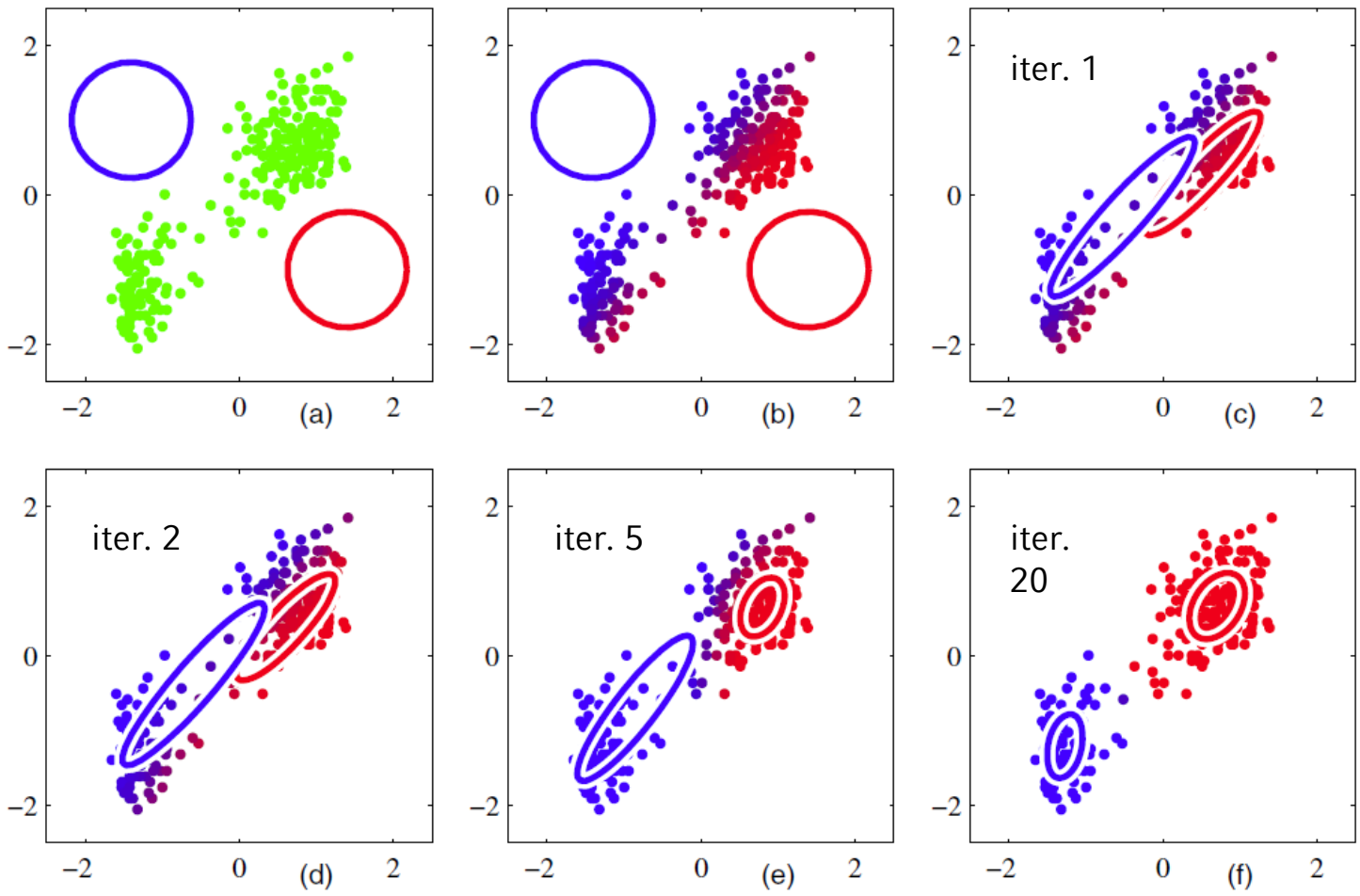
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$$

mixing coefficients $\in \mathbb{R}$ *:* $\sum_k \pi_k = 1$ *and* $0 \le \pi_k \le 1$



Expectation Maximization (EM): Exemplary Application

Example taken from: C. M. Bischop „Pattern Recognition and Machine Learning“, 2009

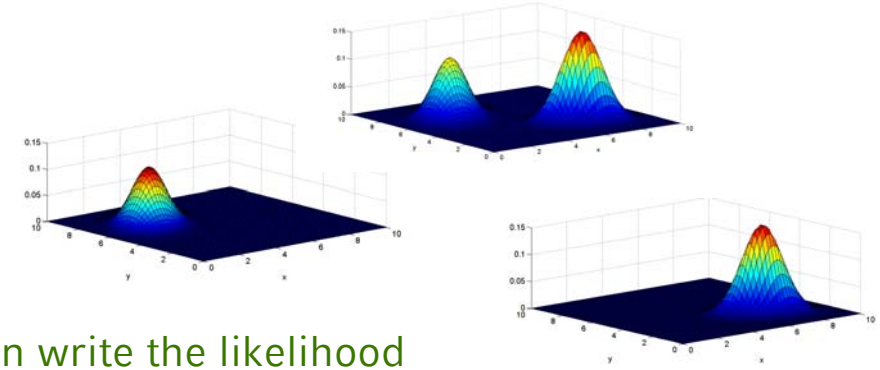


Expectation Maximization (EM)

Note: EM is not restricted to Gaussian distributions, but they will serve as example in this lecture.

A *clustering* $\mathcal{M} = \{C_1, \dots, C_K\}$ is represented by a mixture distribution with parameters $\Theta = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \dots, \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\}$:

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

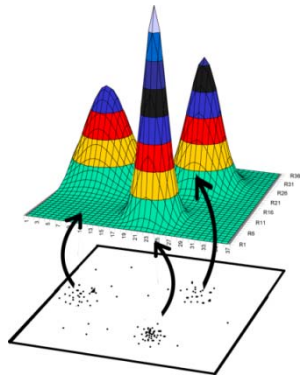


Each *cluster* is represented by one component of the mixture distribution:

$$p(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Given a dataset $\mathbf{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \subseteq \mathbb{R}^d$, we can write the likelihood that all data points $\mathbf{x}_n \in \mathbf{X}$ are generated (independently) by the mixture model with parameters Θ as:

$$\log p(\mathbf{X}|\Theta) = \log \prod_{n=1}^{N} p(x_n|\Theta)$$



Goal: Find the parameters Θ_{ML} with **maximal (log-)likelihood estimation** (MLE)

$$\Theta_{ML} = \arg\max_{\Theta} \{\log p(\mathbf{X}|\Theta)\}$$

Expectation Maximization (EM)

- Goal: Find the parameters Θ_{ML} with the **maximal (log-)likelihood estimation**!

$$\Theta_{ML} = \arg\max_{\Theta}\{\log p(\mathbf{X}|\Theta)\}$$

$$\log p(\mathbf{X}|\Theta) = \log \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \cdot \mathrm{p}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \cdot \mathrm{p}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Maximization with respect to the means:

$$\frac{\partial}{\partial \boldsymbol{\mu}_j} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \boldsymbol{\Sigma}_\mathrm{j}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_j)\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

$$\frac{\partial \log p(\mathbf{X}|\Theta)}{\partial\, \boldsymbol{\mu}_j} = \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n|\Theta)}{\partial\, \boldsymbol{\mu}_j} = \sum_{n=1}^{N} \frac{\frac{\partial p(\boldsymbol{x}_n|\Theta)}{\partial\, \boldsymbol{\mu}_j}}{p(\boldsymbol{x}_n|\Theta)} = \sum_{n=1}^{N} \frac{\frac{\partial\, \pi_j \cdot p(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\partial\, \boldsymbol{\mu}_j}}{\sum_{k=1}^{K} p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} = \sum_{n=1}^{N} \frac{\pi_j \cdot \boldsymbol{\Sigma}_\mathrm{j}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_j)\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

$$\frac{\partial \log p(\mathbf{X}|\Theta)}{\partial\, \boldsymbol{\mu}_j} = \boldsymbol{\Sigma}_\mathrm{j}^{-1} \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu}_j) \frac{\pi_j \cdot \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \stackrel{\text{def}}{=} \mathbf{0}$$

- Define

$$\gamma_j(x_n) := \pi_j \cdot \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j).$$

$\gamma_j(x_n)$ is the probability that component j generated the object $\boldsymbol{x}_n$.

Expectation Maximization (EM)

Maximization w.r.t. the means yields:

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\boldsymbol{x}_n)\, \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_j(\boldsymbol{x}_n)} \qquad \text{(weighted mean)}$$

Maximization w.r.t. the covariance yields:

$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma_j(\boldsymbol{x}_n)(\boldsymbol{x}_n - \boldsymbol{\mu}_j)(\boldsymbol{x}_n - \boldsymbol{\mu}_j)^T}{\sum_{n=1}^{N} \gamma_j(\boldsymbol{x}_n)}$$

Maximization w.r.t. the mixing coefficients yields:

$$\pi_j = \frac{\sum_{n=1}^{N} \gamma_j(\boldsymbol{x}_n)}{\sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_k(\boldsymbol{x}_n)}$$

Expectation Maximization (EM)

Problem with finding the optimal parameters Θ_{ML}:

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\boldsymbol{x}_n)\, \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_j(\boldsymbol{x}_n)} \quad \text{and} \quad \gamma_j(\boldsymbol{x}_n) = \frac{\pi_j \cdot \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}$$

- Non-linear mutual dependencies.
- Optimizing the Gaussian of cluster j depends on all other Gaussians.

→ There is no closed-form solution!

→ Approximation through iterative optimization procedures

→ Break the mutual dependencies by optimizing $\boldsymbol{\mu}_j$ and $\gamma_j(\boldsymbol{x}_n)$ independently

Expectation Maximization (EM)

EM-approach: iterative optimization

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

$$\gamma_j^{new}(\boldsymbol{x}_n) = \frac{\pi_j \cdot \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

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

$$\boldsymbol{\mu}_j^{new} = \frac{\sum_{n=1}^{N} \gamma_j^{new}(\boldsymbol{x}_n)\, \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_j^{new}(\boldsymbol{x}_n)}$$

$$\boldsymbol{\Sigma}_j^{new} = \frac{\sum_{n=1}^{N} \gamma_j^{new}(\boldsymbol{x}_n)(\boldsymbol{x}_n - \boldsymbol{\mu}_j^{new})(\boldsymbol{x}_n - \boldsymbol{\mu}_j^{new})^T}{\sum_{n=1}^{N} \gamma_j^{new}(\boldsymbol{x}_n)}$$

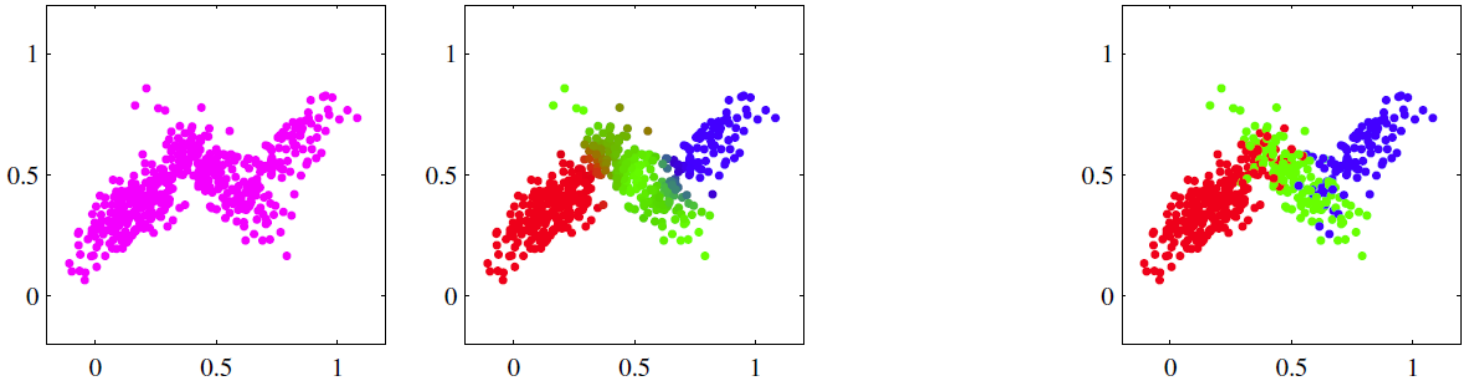
$$\pi_j^{new} = \frac{\sum_{n=1}^{N} \gamma_j^{new}(\boldsymbol{x}_n)}{\sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_k^{new}(\boldsymbol{x}_n)}$$

4. Evaluate the new log likelihood $\log p(\mathbf{X}|\Theta^{\text{new}})$ and check for convergence of parameters or log likelihood ($|\log p(\mathbf{X}|\Theta^{\text{new}}) - \log p(\mathbf{X}|\Theta)| \leq \epsilon$).
   If the convergence criterion is not satisfied, set $\Theta = \Theta^{\text{new}}$ and go to step 2.

EM: Turning the Soft Clustering into a Partitioning

EM obtains a *soft* clustering (each object belongs to each cluster with a certain probability) reflecting the uncertainty of the most appropriate assignment.

Example taken from: C. M. Bishop „Pattern Recognition and Machine Learning", 2009



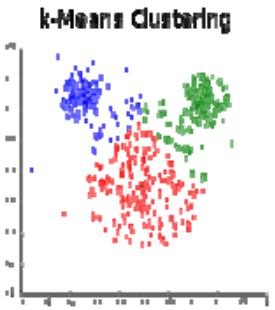
a) input for EM *b) soft clustering result of EM* *c) original data*

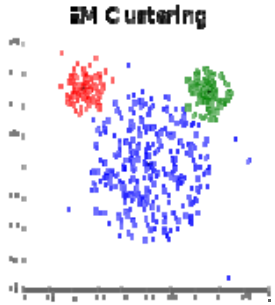
Modification to obtain a *partitioning* variant

- Assign each object to the cluster to which it belongs with the highest probability

$$\text{Cluster}(\text{object}_n) = argmax_{k \in \{1,\ldots,K\}}\{\gamma(z_{nk})\}$$

Discussion





Superior to k-Means for clusters of varying size
or clusters having differing variances
→ more accurate data representation

Convergence to (possibly local) maximum

Computational effort for *N* objects, *K* derived clusters, and *t* iterations:

- $O(t \cdot N \cdot K)$
- #iterations is quite high in many cases

Both - result and runtime - strongly depend on

- the initial assignment
  - → do multiple random starts and choose the final estimate with highest likelihood
  - → Initialize with clustering algorithms (e.g., K-Means usually converges much faster)
  - → Local maxima and initialization issues have been addressed in various extensions of EM
- a proper choice of parameter *K* (= desired number of clusters)
  - → Apply principles of model selection (see next slide)

EM: Model Selection for Determining Parameter K

Classical trade-off problem for selecting the proper number of components K

- If K is too high, the mixture may overfit the data
- If K is too low, the mixture may not be flexible enough to approximate the data

Idea: determine candidate models Θ_K for a range of values of K (from K_{min} to K_{max}) and select the model $\Theta_{K^*} = \max\{\text{qual}(\Theta_K) | K \in \{K_{min}, \dots, K_{max}\}\}$

- Silhouette Coefficient (as for k-Means) only works for partitioning approaches.
- The MLE (Maximum Likelihood Estimation) criterion is nondecreasing in K

Solution: deterministic or stochastic *model selection* methods[MP'00] which try to balance the goodness of fit with simplicity.

- Deterministic: $qual(\Theta_K) = \log p(\mathbf{X}|\Theta_K) + \mathcal{P}(K)$
  where $\mathcal{P}(K)$ is an increasing function penalizing higher values of K
- Stochastic: based on Markov Chain Monte Carlo (MCMC)

[MP'00] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000.