Mixture Models and Unsupervised Learning

TTT4185 Machine Learning for Signal Processing

Giampiero Salvi

Department of Electronic Systems
NTNU

HT2021

Supervised vs Unsupervised Learning

Supervised

$$\begin{aligned}\mathbf{X} &= \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \\ \mathbf{t} &= \{t_1, \dots, t_N\}\end{aligned}$$

$$\mathbf{x} \xrightarrow{f} t$$

Unsupervised

$$\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

- clustering
- distribution estimation
- dimensionality reduction

Mixture Models

Weighted sum of K simple probability distribution functions:

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\theta_k),$$

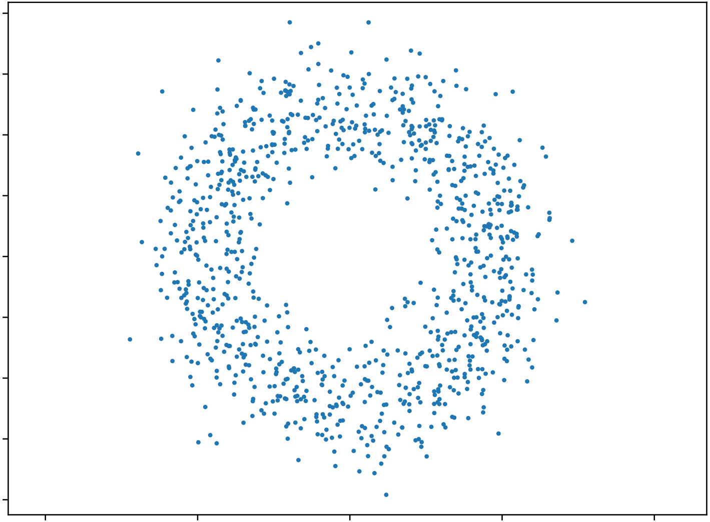
with $\theta = \{\pi_1, \dots, \pi_k, \theta_1, \dots, \theta_K\}$

Two main uses:

- Model complex distributions with simpler pdfs
- Clustering (unsupervised classification)

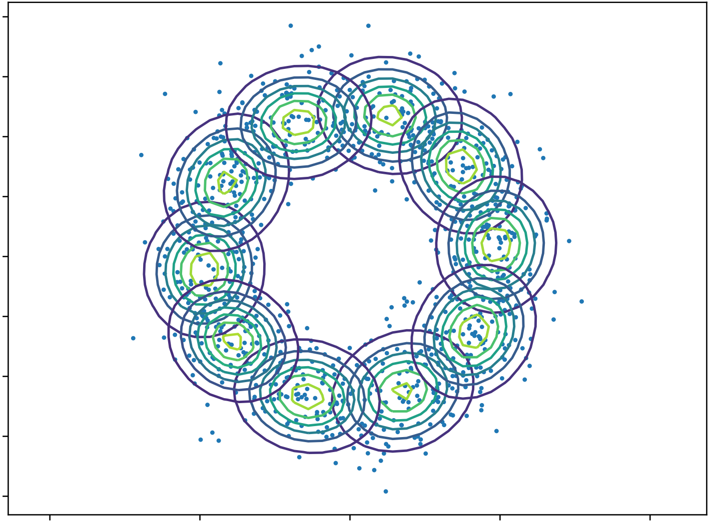
Example: approximating distribution (doughnut data)

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$



Example: approximating distribution (doughnut data)

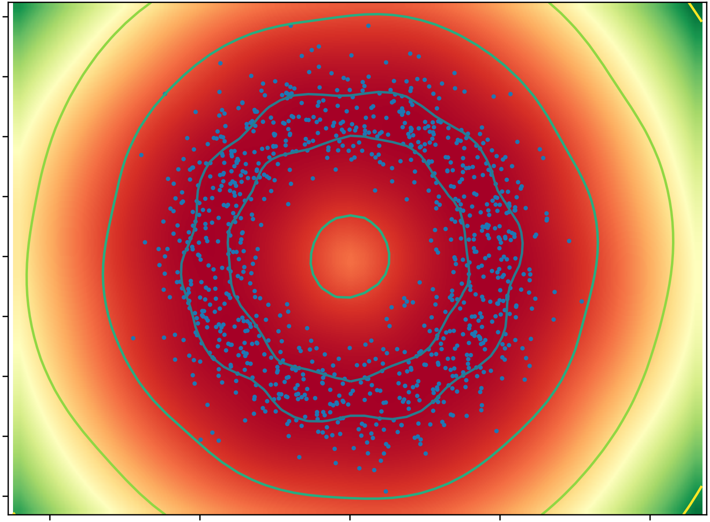
$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$
$$p(\mathbf{x}|\theta_k), \forall k \in [1, K]$$



Example: approximating distribution (doughnut data)

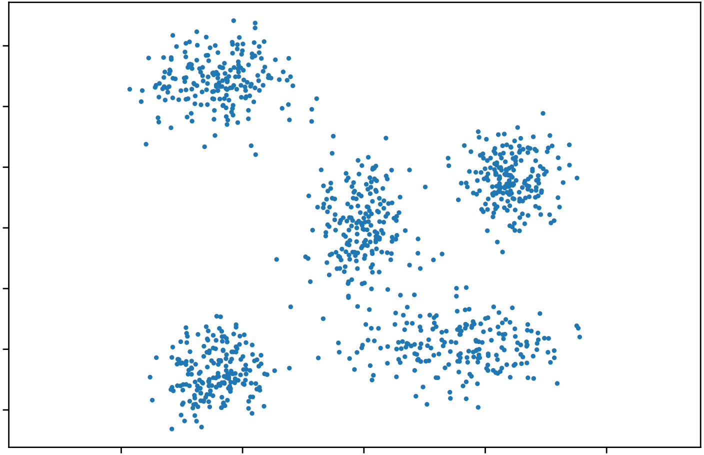
$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k p(x|\theta_k)$$



Example: Clustering

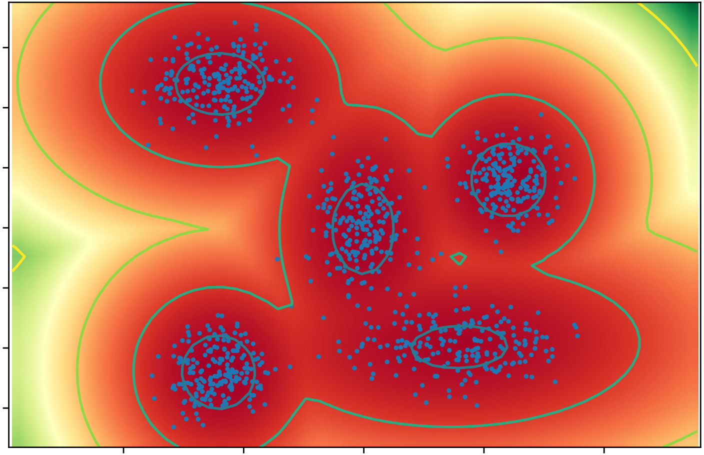
$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$



Example: Clustering

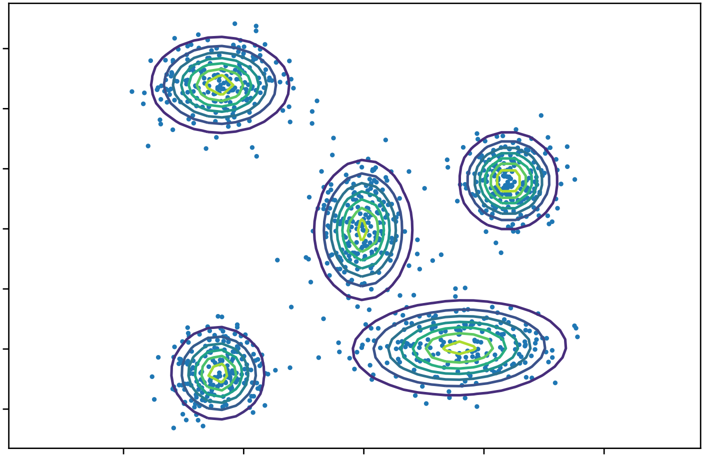
$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k p(x|\theta_k)$$

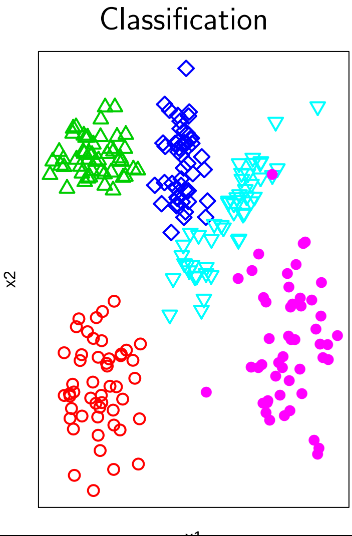


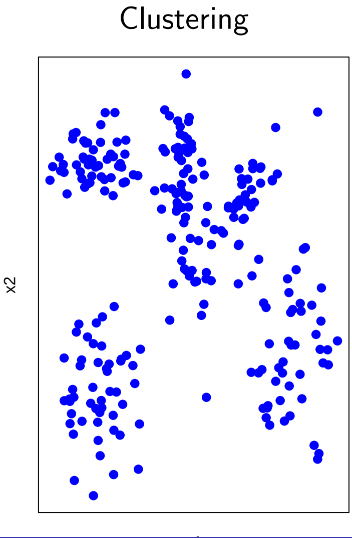
Example: Clustering

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$
$$p(\mathbf{x}|\theta_k), \forall k \in [1, K]$$



Clustering vs Classification





Fitting Mixture of distributions

We would like to find the maximum likelihood solution:

$$\begin{aligned}
\arg\max_\theta \ln p(\mathbf{X}|\theta) &= \arg\max \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\theta) \\
&= \arg\max_\theta \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n|\theta_k),
\end{aligned}$$

with $\theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$

Fitting Mixture of distributions

We would like to find the maximum likelihood solution:

$$\begin{aligned}\arg\max_{\theta} \ln p(\mathbf{X}|\theta) &= \arg\max \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\theta) \\ &= \arg\max_{\theta} \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n|\theta_k),\end{aligned}$$

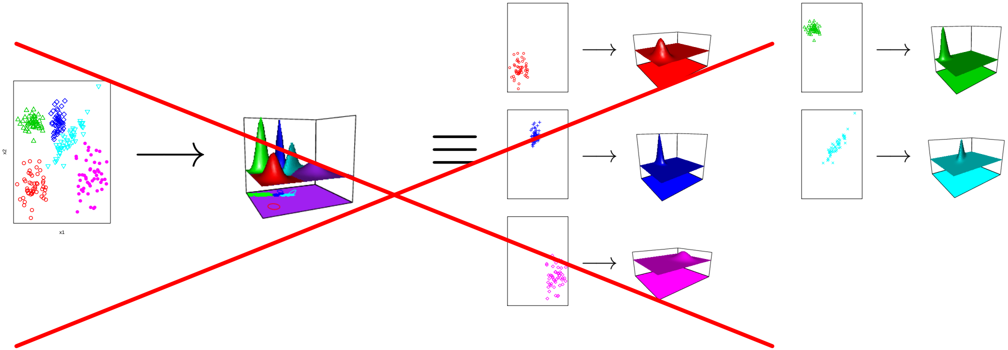
with $\theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$

Problem:

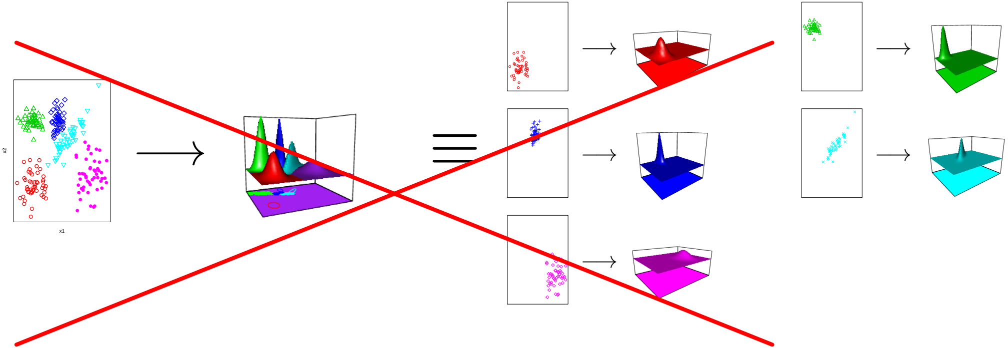
We do not know which point has been generated by which component of the mixture

We cannot optimize $p(\mathbf{X}|\theta)$ directly

No Class Independence Assumption



No Class Independence Assumption



Solution: Expectation Maximization

Expectation Maximization

- introduce latent variables to solve the problem
- very general idea (applies to many different probabilistic models)
- easier to explain in terms of clustering than modelling complex distributions
- We will use K-means as introduction

K-means

- given a set of points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- describes each class with a centroid $\boldsymbol{\mu}_k, \quad k = 1, \dots, K$
- a point belongs to a class if the corresponding centroid is closest (Euclidean distance)
- define binary indicator variable $r_{nk} \in \{0, 1\}$ (1-of-K coding)
- find $\boldsymbol{\mu}_k$ and r_{nk} by optimizing the distortion measure:

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

K-means

- iterative procedure (both steps have closed solution):
  1. optimize J w.r.t. r_{nk} keeping $\boldsymbol{\mu}_k$ fixed

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j ||\mathbf{x}_n - \boldsymbol{\mu}_j||^2 \\ 0 & \text{otherwise.} \end{cases}$$

  2. optimize J w.r.t. $\boldsymbol{\mu}_k$ keeping r_{nk} fixed

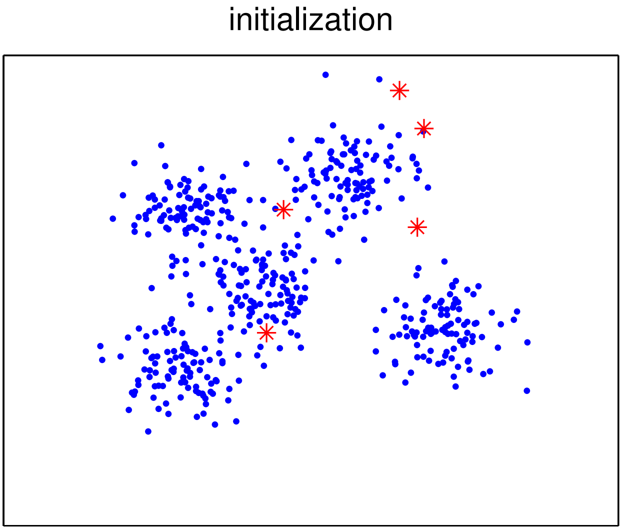
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

- guaranteed to converge
- not guaranteed to find the optimal solution
- used in vector quantization (since the 1950's)

K-means: algorithm

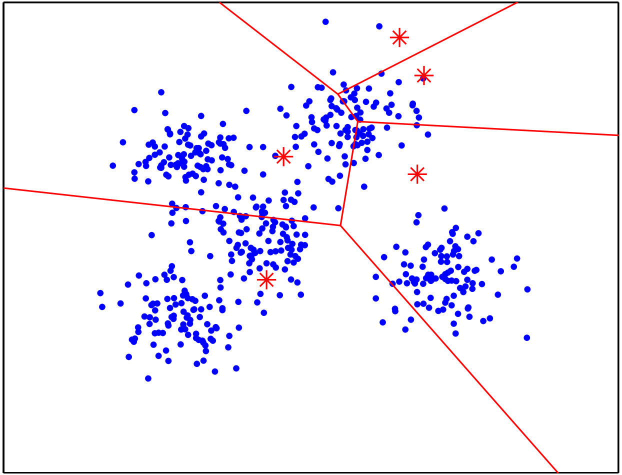
Data: K (number of desired clusters), N data points $\mathbf{x}_n$
Result: K clusters
initialization: assign initial value to K centroids $\boldsymbol{\mu}_k$;
repeat
    assign each point $\mathbf{x}_n$ to closest centroid $\boldsymbol{\mu}_k$;
    compute new centroids as mean of each group of points;
until *centroids do not change*;
return K clusters;

K-means: example



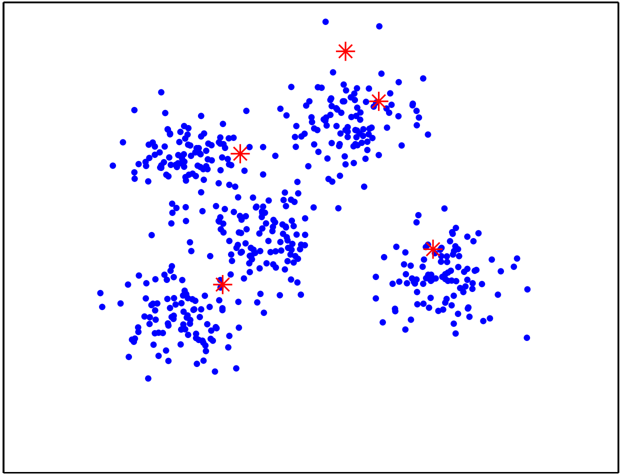
K-means: example

iteration 1, update clusters



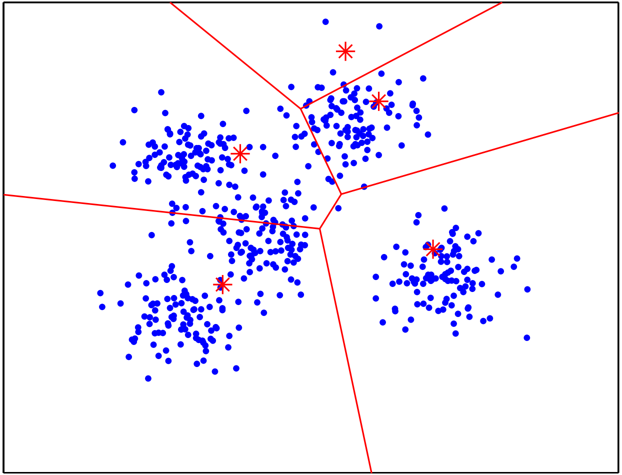
K-means: example

iteration 2, update centroids



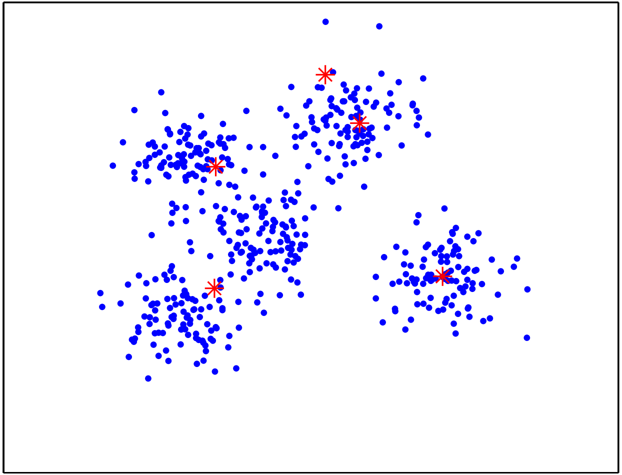
K-means: example

iteration 2, update clusters



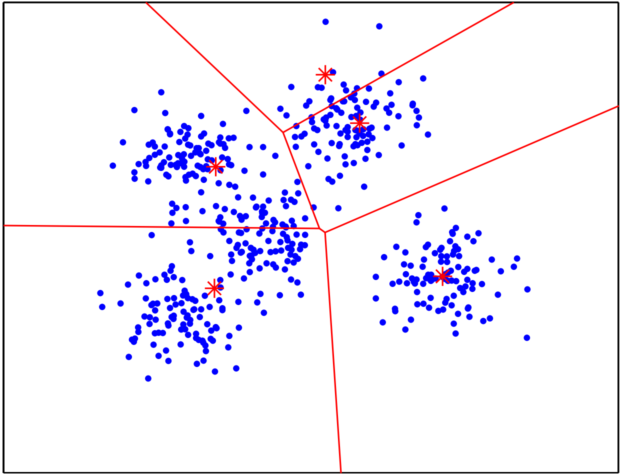
K-means: example

iteration 3, update centroids



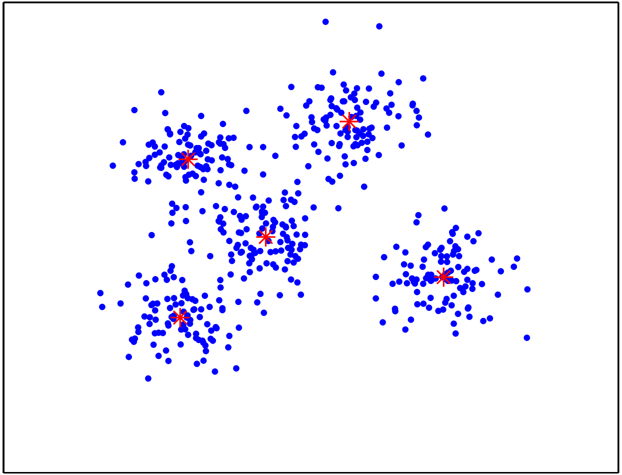
K-means: example

iteration 3, update clusters



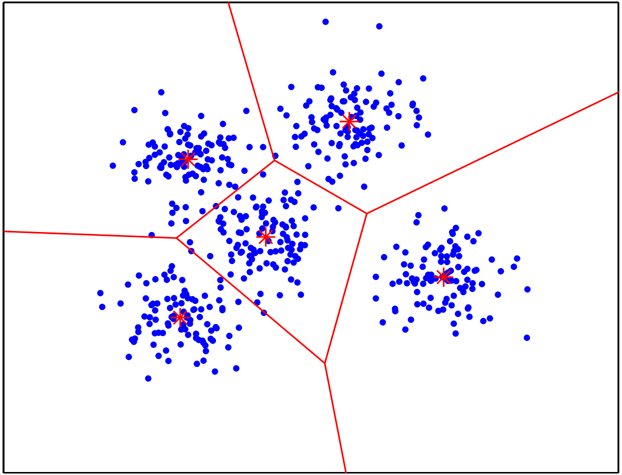
K-means: example

iteration 20, update centroids



K-means: example

iteration 20, update clusters



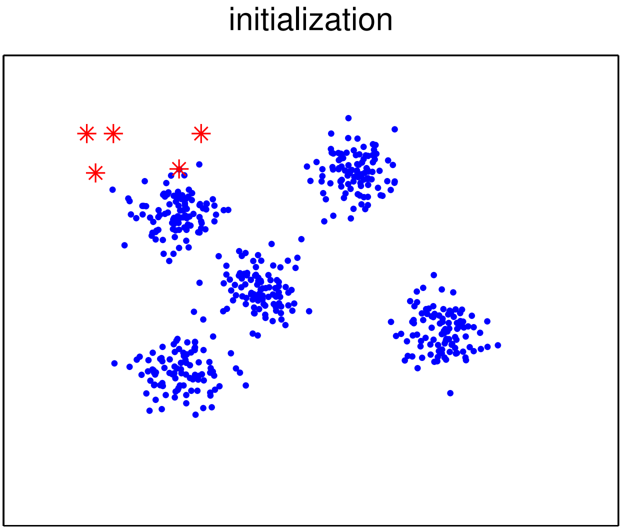
Example: data compression

N pixels, 8 bits RGB
$\rightarrow 24N$ bits

K-means $\rightarrow 24K + N \log_2 K$

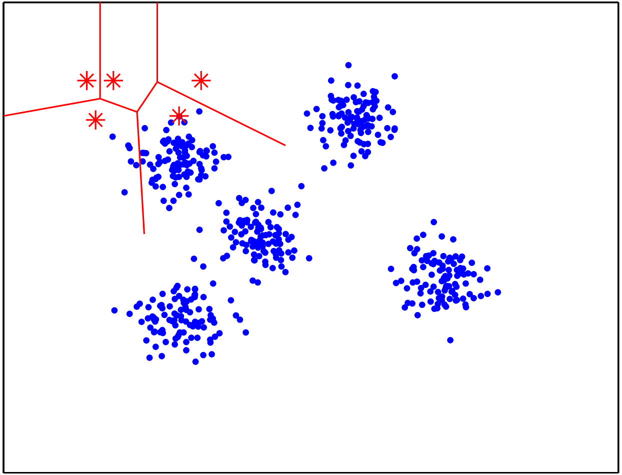
If 1 Megapixel,
$24N = 24$ millions,
K-means, $K = 10 \rightarrow$
~ 3 millions

K-means: sensitivity to initial conditions



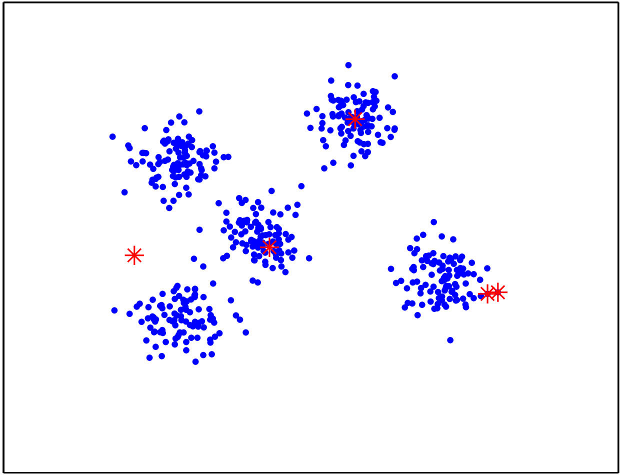
K-means: sensitivity to initial conditions

iteration 1, update clusters



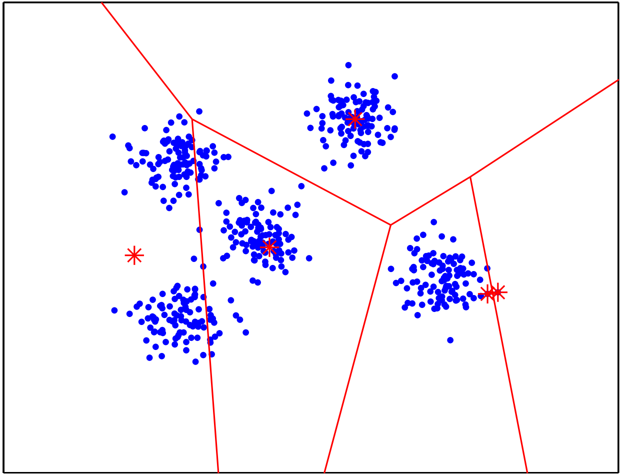
K-means: sensitivity to initial conditions

iteration 2, update centroids

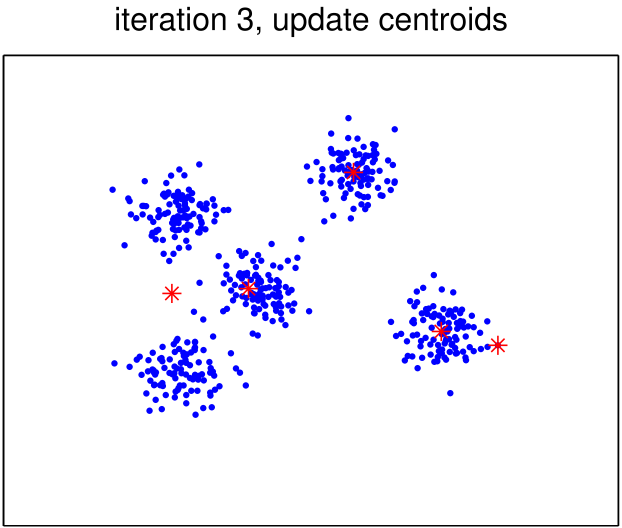


K-means: sensitivity to initial conditions

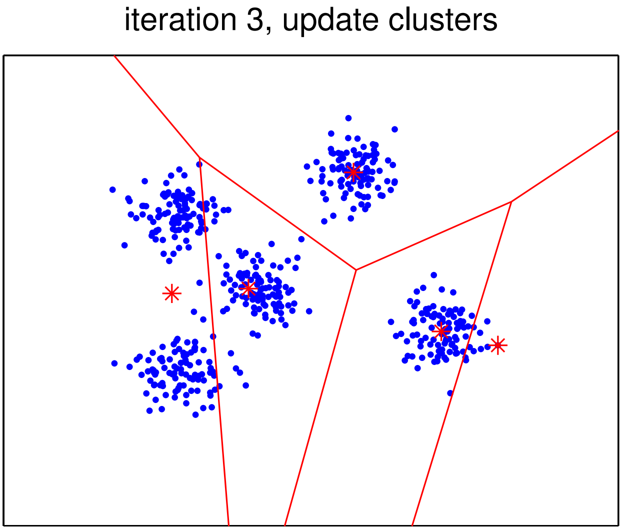
iteration 2, update clusters



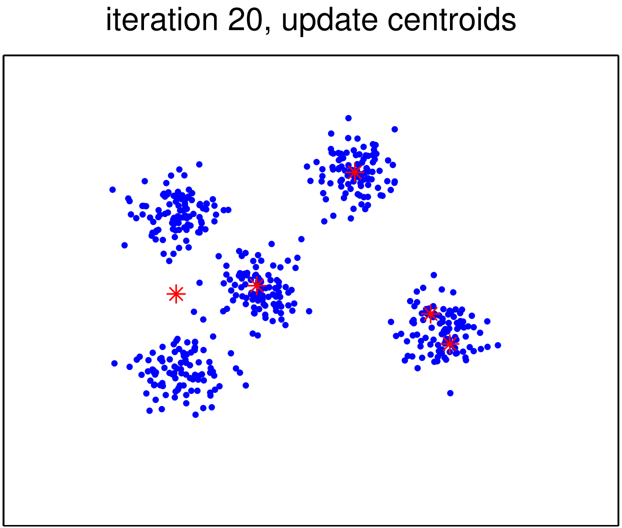
K-means: sensitivity to initial conditions



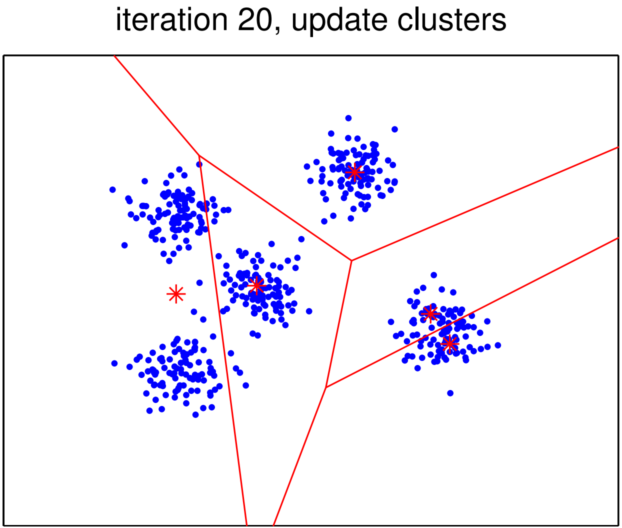
K-means: sensitivity to initial conditions



K-means: sensitivity to initial conditions



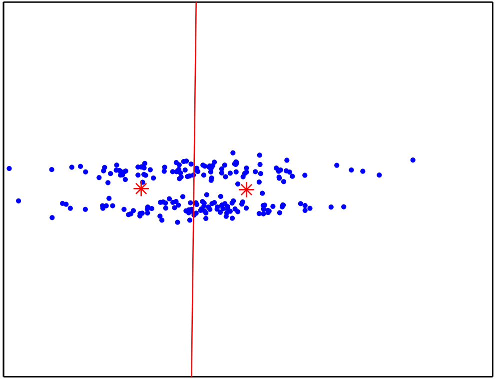
K-means: sensitivity to initial conditions



K-means: limits of Euclidean distance

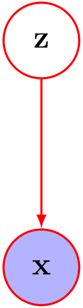
- the Euclidean distance is isotropic (same in all directions in $\mathbb{R}^p$)
- this favours spherical clusters
- the size of the clusters is controlled by their distance

two non–spherical classes



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

We introduce a *latent* variable $\mathbf{z}$ with a 1-of-K representation:
$\mathbf{z} \in \{0,1\}^K$, $\sum_{k=1}^{K} z_k = 1$.



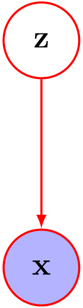
Alternative formulation: $h \in \{1, \dots, K\}$.

Latent Variable: responsibilities

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

Marginal (w.r.t. $\mathbf{z}$):

$$\begin{aligned} p(z_k = 1) &= \pi_k, \text{ or} \\ p(\mathbf{z}) &= \prod_{k=1}^{K} \pi_k^{z_k}, \end{aligned}$$

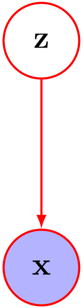


Conditional:

$$\begin{aligned} p(\mathbf{x}|z_k = 1) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \text{ or} \\ p(\mathbf{x}|\mathbf{z}) &= \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}, \end{aligned}$$

Latent Variable: important probabilities

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



Posterior:

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

This is also called the responsibility of the term k in the mixture.

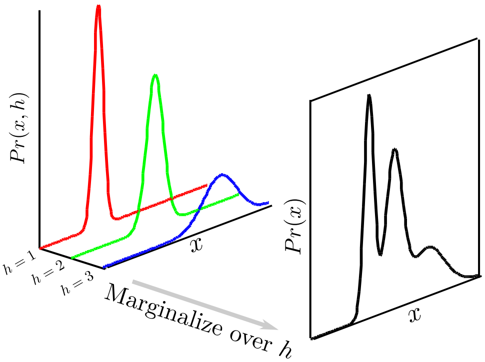
Mixture of Gaussians as a marginalization

Marginal (w.r.t. $\mathbf{x}$):

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{k=1}^{K} p(\mathbf{x}, z_k = 1)$$

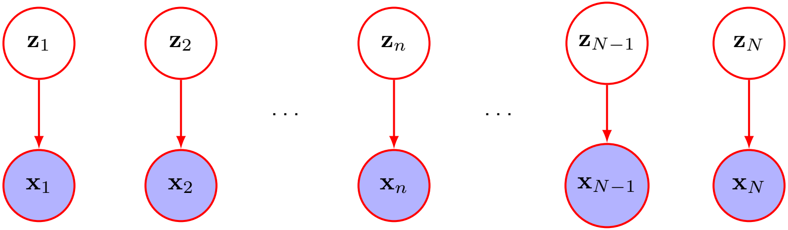
$$= \sum_{k=1}^{K} p(z_k = 1) p(\mathbf{x} \mid z_k = 1)$$

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

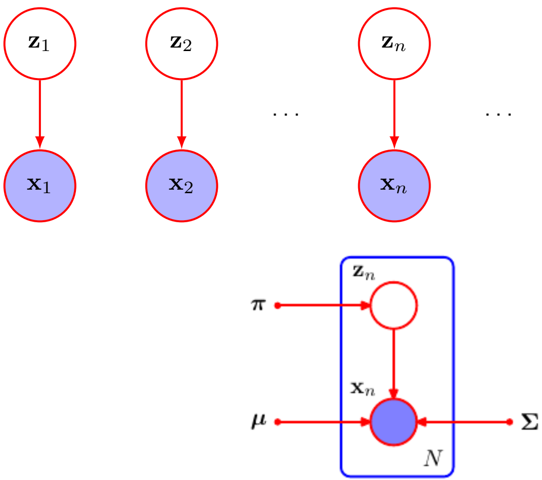


Figures taken from **Computer Vision: models, learning and inference** by Simon Prince.

Set of N i.i.d. points



Set of N i.i.d. points

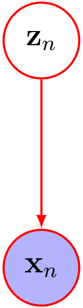


Ancestral Sampling

We assume the data is generated as follows:

For $n \in [1, N]$, do:

1. sample the value of $\mathbf{z}_n$ from the distribution $\{\pi_1, \dots, \pi_K\}$
2. given that $z_{nk} = 1$, sample $\mathbf{x}_n$ from $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$



Maximum Likelihood for Mixture of distributions

We would like to find the maximum likelihood solution:

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

Problems:

- log of sum hard to optimize $\rightarrow$ Expectation Maximization
- singularities
- identifiability

Singularities

- set a lower threshold for variance, or
- detect collapsing Gaussian and reinitialize

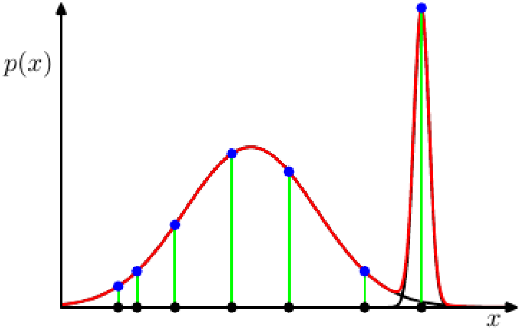


Figure from Bishop

Identifiability

Any permutation of the K distributions is an equivalent solution

Irrelevant if we are interested in $p(\mathbf{x})$ only, but

Important if we are interested in clustering.

Maximum Likelihood

Setting

$$\frac{d \ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{d\boldsymbol{\mu}} = 0$$

we obtain:

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

with

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

Maximum Likelihood

Similarly for $\boldsymbol{\Sigma}$ and $\boldsymbol{\pi}$:

$$\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n, \\
\boldsymbol{\Sigma}_k &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(\mathbf{x}_n-\boldsymbol{\mu}_k)(\mathbf{x}_n-\boldsymbol{\mu}_k)^{\mathsf{T}}, \\
\pi_k &= \frac{N_k}{N}, \text{ with} \\
N_k &= \sum_{n=1}^{N}\gamma(z_{nk}).
\end{aligned}$$

Maximum Likelihood

Similarly for $\boldsymbol{\Sigma}$ and $\boldsymbol{\pi}$:

$$\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n, \\
\boldsymbol{\Sigma}_k &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(\mathbf{x}_n-\boldsymbol{\mu}_k)(\mathbf{x}_n-\boldsymbol{\mu}_k)^{\mathsf{T}}, \\
\pi_k &= \frac{N_k}{N}, \text{ with} \\
N_k &= \sum_{n=1}^{N}\gamma(z_{nk}).
\end{aligned}$$

Not a closed form solution!

We do not know $\gamma(z_{nk})$ until we know $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$.

Expectation Maximization

Solve problem with iterative procedure:

1. Expectation: given $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ estimate responsibilities:

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

2. Maximization: given $\gamma(z_{nk})$, maximise likelihood (formulae from previous slide)

- very general idea (applies to many different probabilistic models)
- optimize the Likelihood of the complete data over N data points

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{z}_1, \dots, \mathbf{z}_N | \theta) = p(\mathbf{X}, \mathbf{Z} | \theta)$$

EM and K-means

EM is very similar to K-means, but:

- $\gamma(z_{nk})$ can be non-zero for all $n \Rightarrow$ all points contribute to all distributions
- probability distribution functions are used instead of Euclidean distance: more flexible cluster shapers

Illustration: EM for two univariate Gaussians

For each sample x_n introduce a *hidden variable* $\mathbf{z}_n$

$$\mathbf{z}_n = \begin{cases} [1,0] & \text{if sample } x_n \text{ was drawn from } \mathcal{N}(x|\mu_1,\sigma_1^2) \\ [0,1] & \text{if sample } x_n \text{ was drawn from } \mathcal{N}(x|\mu_2,\sigma_2^2) \end{cases}$$

Initialize the model parameters (random, or using K-means):

$$\Theta^{(0)} = (\pi_1^{(0)}, \mu_1^{(0)}, \sigma_1^{(0)}, \mu_2^{(0)}, \sigma_2^{(0)})$$

Update the parameters iteratively with Expectation and Maximization steps...

EM for two Gaussians: E-step

Estimate the responsibility $\gamma(z_{nk})$ of k-th Gaussian for each sample x_n (indicated by the size of the projected data point)

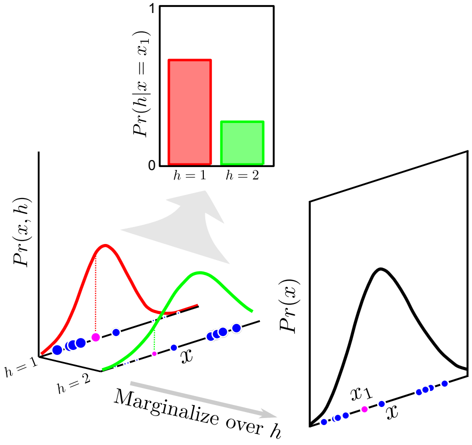


Figure from Prince.

EM for two Gaussians: E-step (cont.)

E-step: Compute the *posterior probability* that x_n was generated by component k given the current estimate of the parameters $\Theta^{(t)}$. (responsibilities)

for $n = 1, \dots N$

for $k = 1, 2$

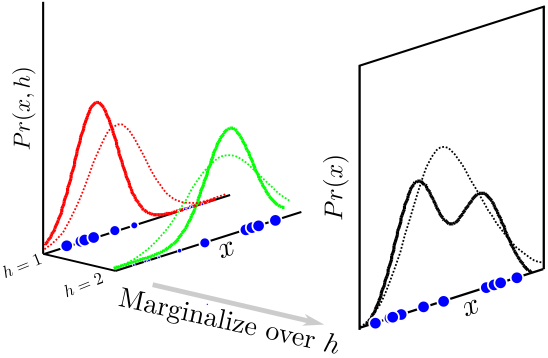
$$\gamma_{nk}^{(t)} = P(z_{nk} = 1 \,|\, x_n, \Theta^{(t)})$$

$$= \frac{\pi_k^{(t)} \mathcal{N}(x_n | \mu_k^{(t)}, \sigma_k^{(t)})}{\pi_1^{(t)} \mathcal{N}(x_n | \mu_1^{(t)}, \sigma_1^{(t)}) + \pi_2^{(t)} \mathcal{N}(x_n | \mu_2^{(t)}, \sigma_2^{(t)})}$$

Note: $\gamma_{n1}^{(t)} + \gamma_{n2}^{(t)} = 1$ and $\pi_1 + \pi_2 = 1$

EM for two Gaussians: M-step

Fitting the Gaussian model for each of k-th constituent.
Sample x_n contributes according to the responsibility γ_{nk}.



(dashed and solid lines for fit before and after update)

Figure from Prince.

EM for two Gaussians: M-step (cont.)

M-step: Compute the *Maximum Likelihood* of the parameters of the mixture model given out data's membership distribution, the $\gamma_i^{(t)}$'s:

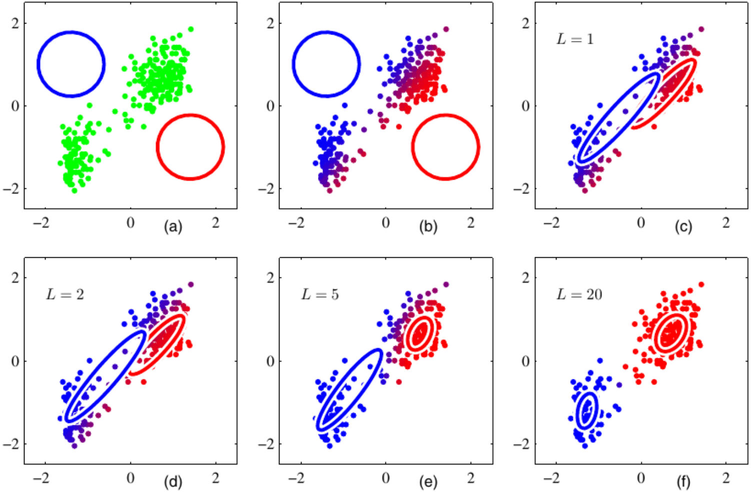
for $k = 1, 2$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t)} x_i}{\sum_{i=1}^n \gamma_{ik}^{(t)}},$$

$$\sigma_k^{(t+1)} = \sqrt{\frac{\sum_{i=1}^n \gamma_{ik}^{(t)} (x_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^n \gamma_{ik}^{(t)}}},$$

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t)}}{N}.$$

EM in practice



EM properties

Similar to K-means

- guaranteed to find a local maximum of the complete data likelihood
- somewhat sensitive to initial conditions

Better than K-means

- Gaussian distributions can model clusters with different shapes
- all data points are smoothly used to update all parameters