K-MEANS & EXPECTATION MAXIMIZATION

Michele Rossi

rossi@dei.unipd.it

Dept. of Information Engineering

University of Padova, IT

Overview

- K-means clustering
  - Hard assignment of points to clusters
  - Lloyd algorithm
- Mixtures of Gaussians r.v.s.
  - K-means with Gaussian mixtures (“soft” K-means)
- Expectation Maximization (EM)
  - Rationale
  - General procedure
  - EM for soft K-means
- X-means
  - Choosing K

The problem

- Data points in a multidimensional space
- Dataset: $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ in a D dimensional space $\boldsymbol{x}_i \in \mathbb{R}^D$
- Goal (informal)
  - partition the points into K disjoint clusters (K is given)
  - Use Euclidean distance
- For each cluster i =1,2,…, K
  - We define a prototype (or centroid) $\boldsymbol{\mu}_i \in \mathbb{R}^D$
  - Can be thought of as the “center” of the cluster
- Goal (restated)
  - Assign the N data points to the K clusters (*each point assigned to a single cluster*) and find a set of centroids such that:
  - the sum of the squares of the distance of each data point to its closest centroid is a minimum

The model

- For each data point, we introduce a binary indicator variable $r_{nk} \in \{0, 1\}$
- Where $r_{nk} = 1$ if point $\boldsymbol{x}_n$ is associated with cluster k, and it is equal to zero otherwise
- This is known as the 1-of-K coding scheme

- We define a cost function (aka "distortion measure"):

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

- Goal: minimize J

The procedure (Lloyd algorithm)

- Iterative procedure involving
  - Two subsequent steps (“E” and “M”)

0) Choose some initial value for $\boldsymbol{\mu}_k$

1) Minimize J with respect to r_{nk} (expectation step, E)

2) Minimize J with respect to $\boldsymbol{\mu}_k$ (maximization step, M)

3) Stop when max. no. of iterations is reached
   or improvement in cost function J smaller than ε

E step

- Minimize J with respect to r_{nk} keeping $\boldsymbol{\mu}_k$ fixed
- Given the shape of J

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

- The E step is achieved by setting r_{nk} to be 1 for whichever value of k gives the minimum value of $\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$

- Formally:

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

M step (1/2)

- Minimize J with respect to $\boldsymbol{\mu}_k$ keeping r_{nk} fixed
- Given the shape of J

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

- The M step is achieved by taking the derivative of J with respect to μ_k and setting it equal to zero:

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = 0 \Rightarrow 2 \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

M step (2/2)

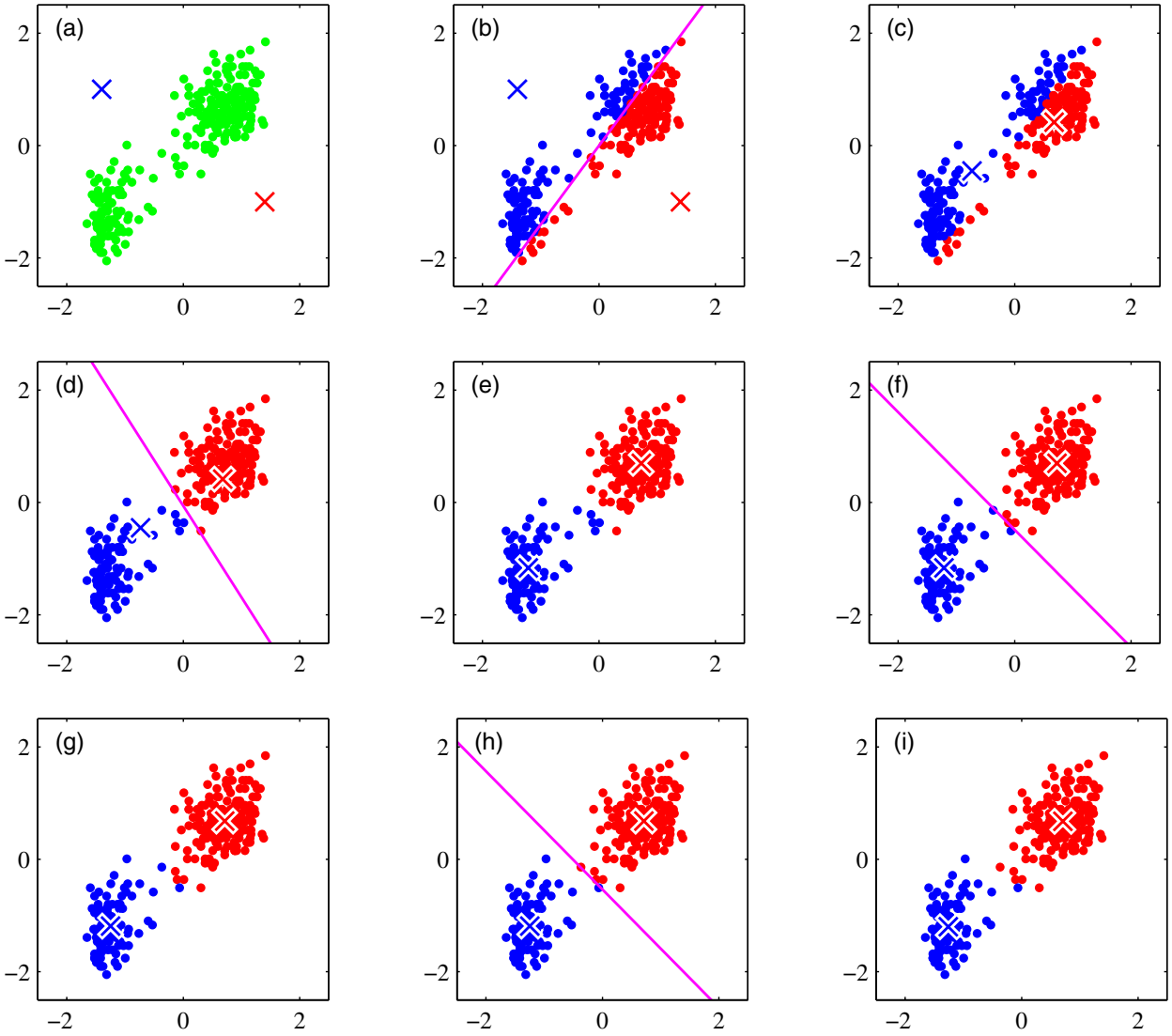
$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = 0 \Rightarrow 2\sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

- This leads to the update:

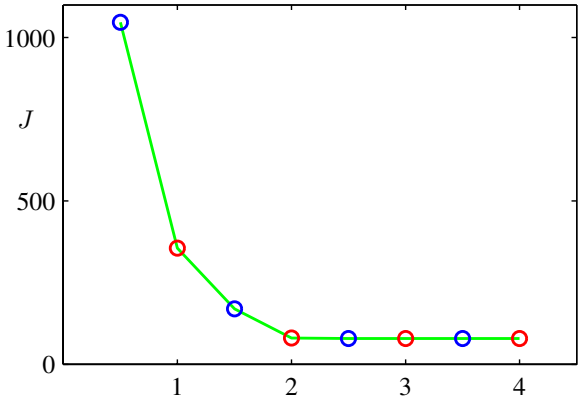
$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n}{\sum_{n=1}^{N} r_{nk}}$$

(the denominator equals the number of points assigned to cluster k. Hence, μ_k is equal to the mean of all of the data points that are assigned to cluster k. This explains the name: “k-means”)

K-means example



- With poor assignment of initial centroids
- 4 iterations to converge



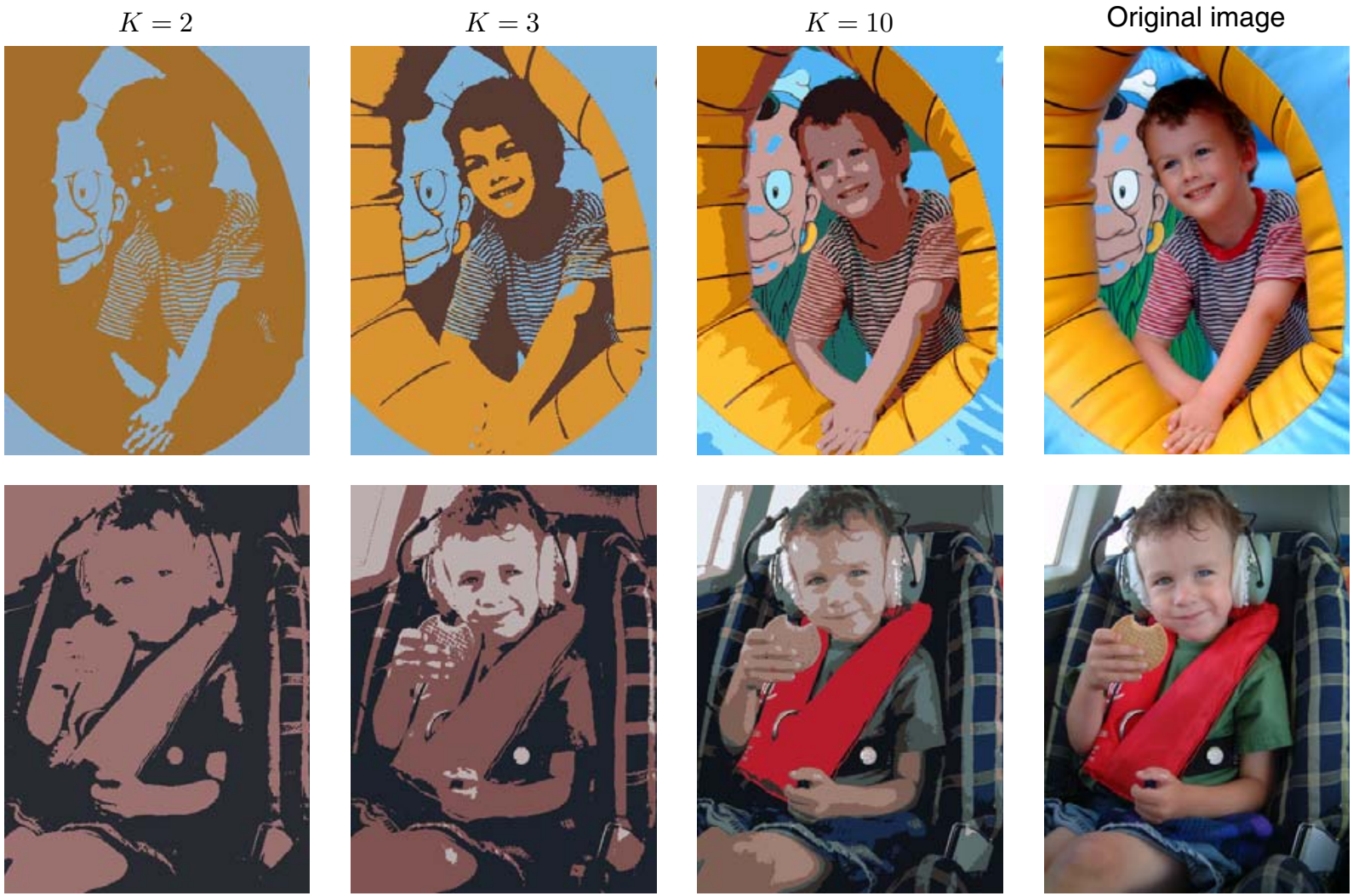
K-medoids

- Euclidean distance may limit the usability
  - Inappropriate for categorical labels, for instance
- A more general distance metric can be used

$$\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\boldsymbol{x}_n, \boldsymbol{\mu}_k)$$

- E step: consists of assigning each point to the centroid whose distance is minimized (cost O(KN), as for k-means)
- M step: potentially more complex than k-means, for this reason is implemented by assigning each centroid to one of the data points in the cluster → the M step involves, for each cluster k, a discrete search over the N_k points assigned to that cluster → requires $O(N_k^2)$ evaluations of $\mathcal{V}$

One more example



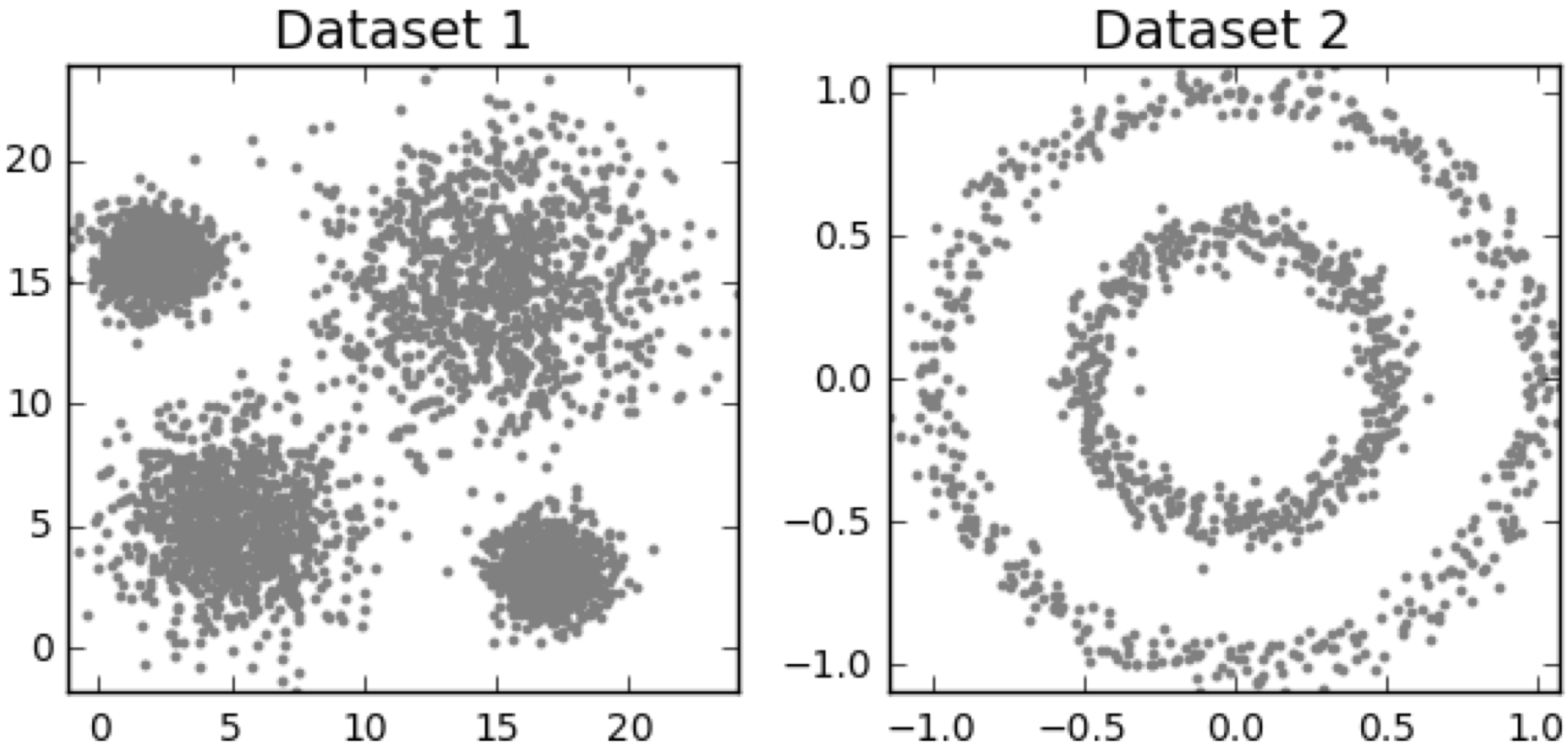
Cluster pixels according to their color (R,G,B) triple

Fewer clusters = more “compression”

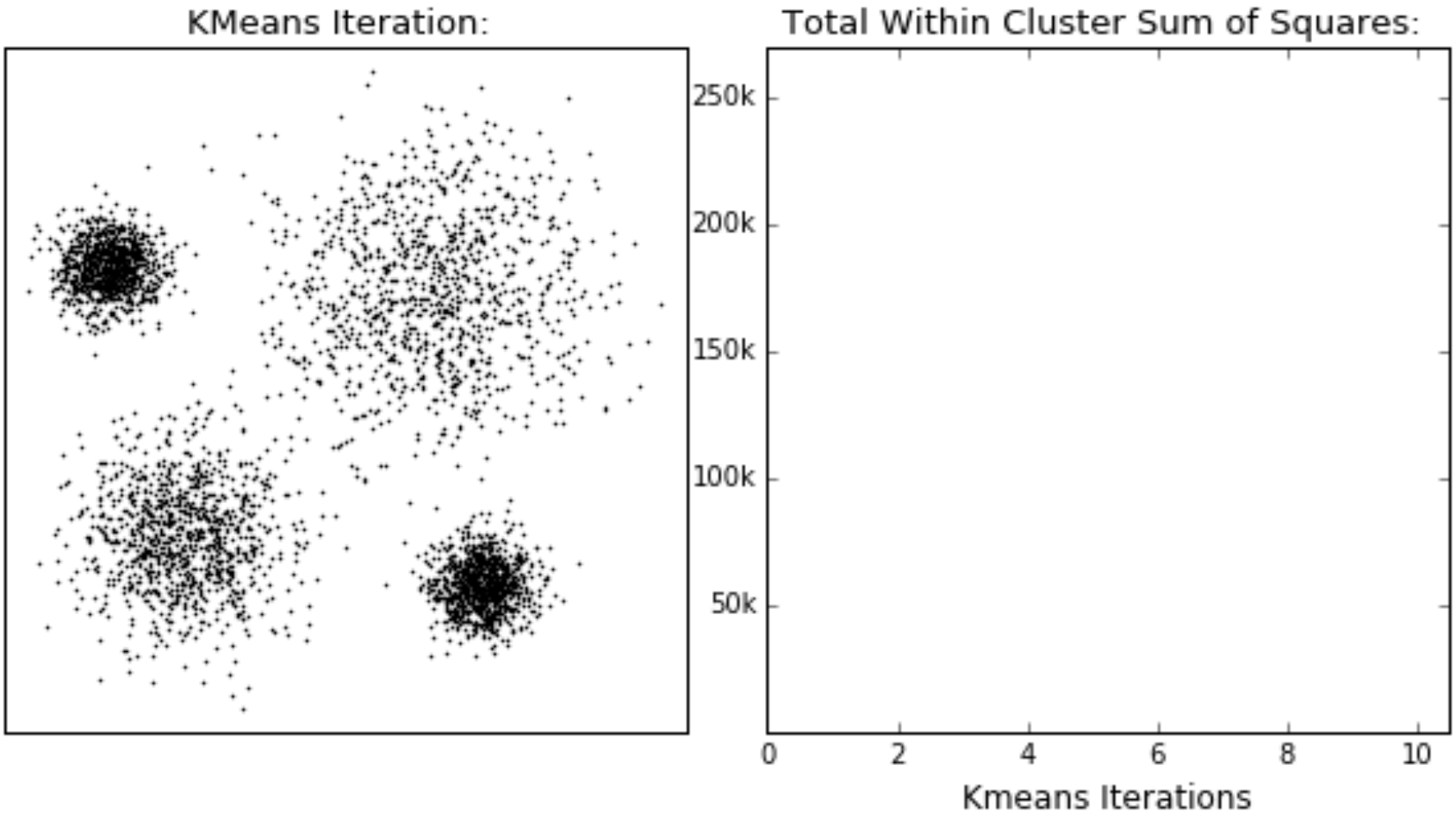
A few questions about K-means

- Where did the distance function come from (Euclidean)?
  - What if we used a different distance function?
  - How could we choose the “best” distance?
- How do we choose the number of clusters K?
- Does K-means always do a good/decent job?

Two datasets



Dataset1



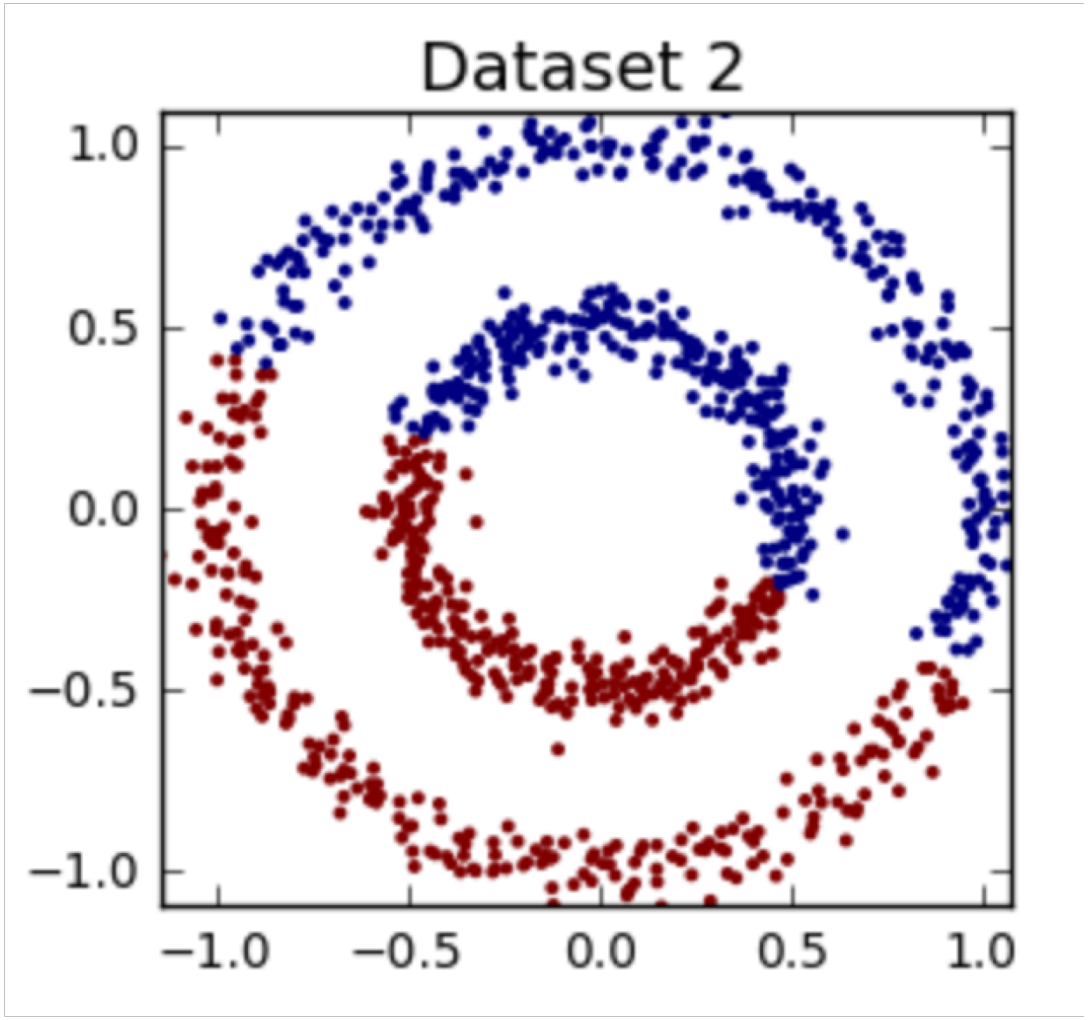
- All OK, clusters are satisfactory!

Dataset2

K-means identifies globular clusters (essentially spherical clusters)

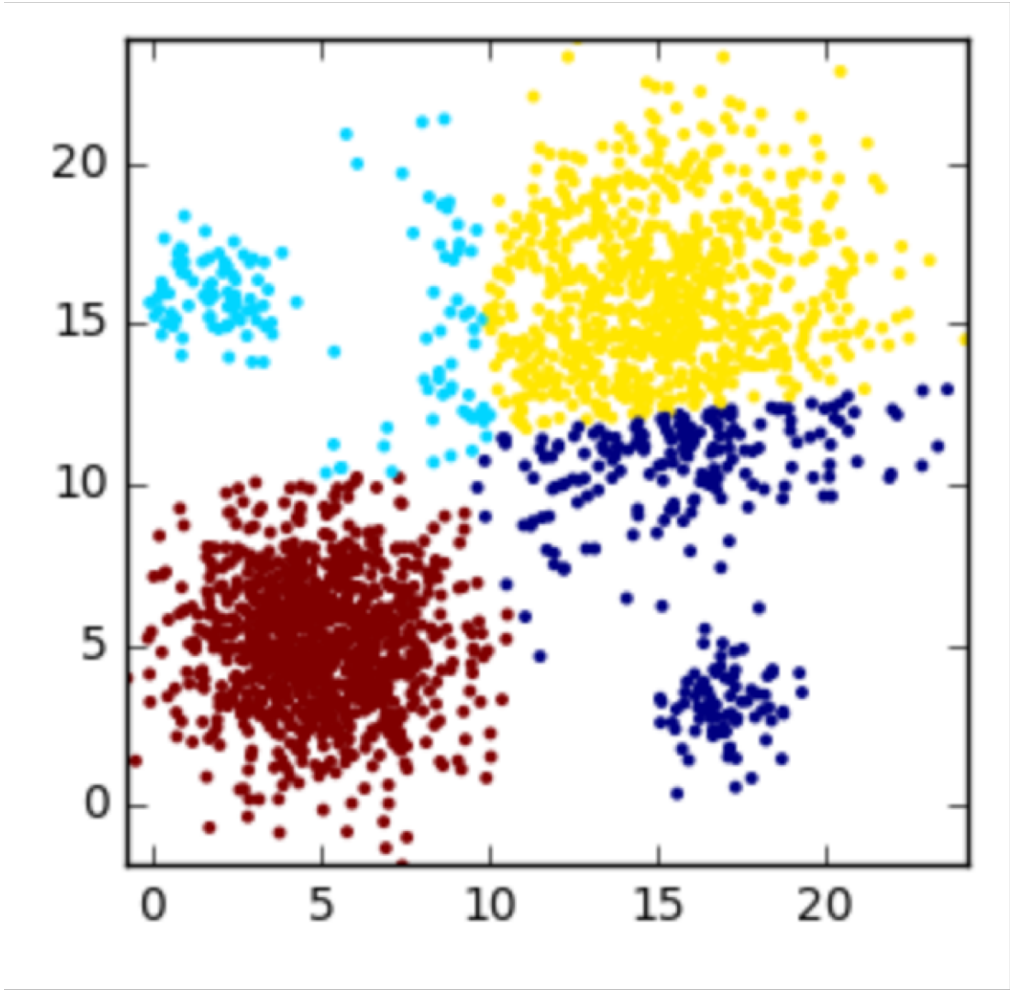
If this assumption does not hold K-means may perform poorly

Note that: if we change the coordinate system, K-means may still perform OK



- K-means fails (miserably)
- The problem is the distance: Euclidean assumes clusters are balls, with mean in the center, and points around it

Yet another dataset



- K-means may also underperform
  - With clusters with different size and density (check DBSCAN)

Bottom line

- Understanding the assumptions behind a method & its limits is essential: it does not just tells you whether a method has drawbacks, it may tell you how to fix them!
- Blindly using a clustering algorithm may be dangerous
  - The clusters that we get may not make much of a sense
- Nevertheless
  - K-means is one of the most popular clustering algorithm out there
  - Its complexity is just O(KNT)
    - K number of clusters
    - N number of data samples
    - T number of iterations
  - It is considered one of the fastest existing clustering algorithms

GAUSSIAN MIXTURES

Gaussian Mixtures (GM)

- Linear superposition of Gaussian pdfs
- Used to approximate distributions with general shape
- Provide a rich model for data fitting
- The Gaussian Mixture (GM) is defined as:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k G(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad \boldsymbol{x} \in \mathbb{R}^D$$

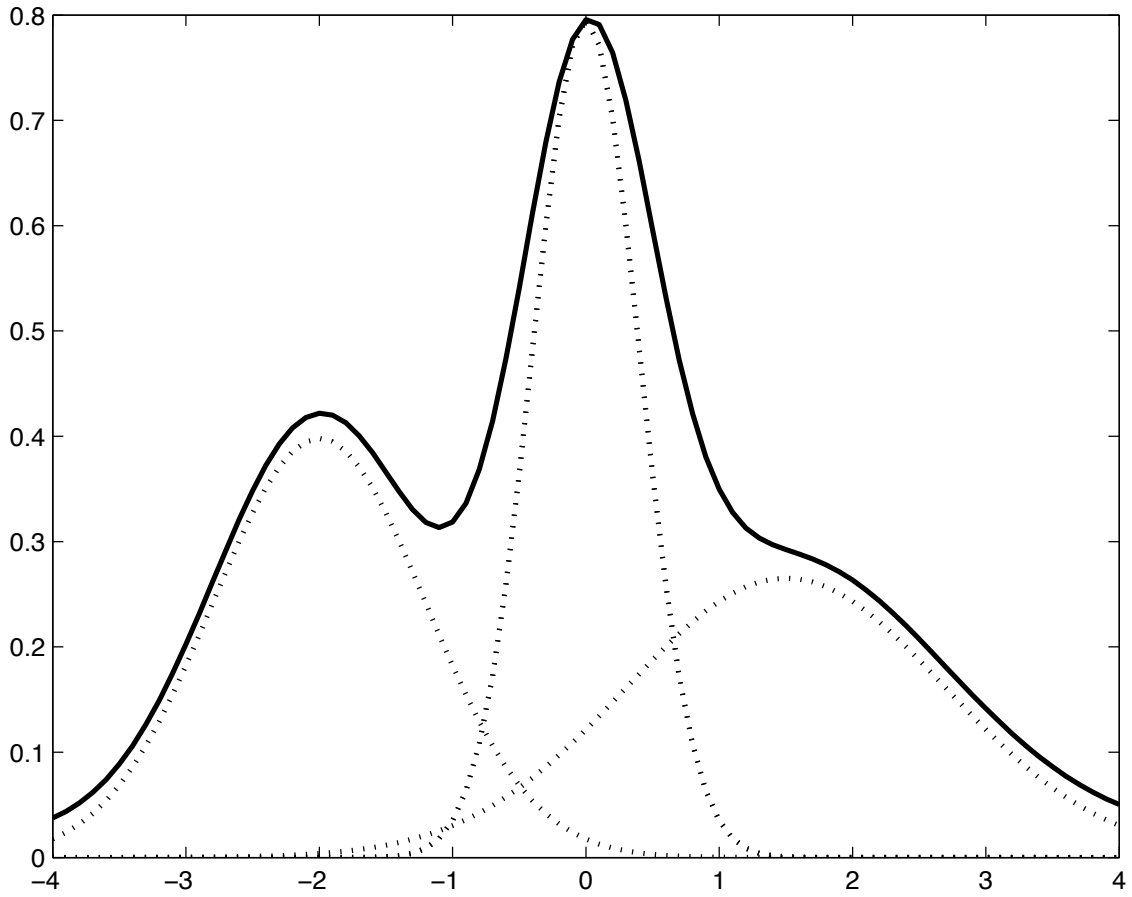
- With:

$$G(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}-\boldsymbol{\mu}_k)\right\}$$

$\boldsymbol{\mu}_k$ mean $\quad$ $\boldsymbol{\Sigma}_k$ D x D covariance matrix $\quad$ $|\boldsymbol{\Sigma}_k|$ determinant

GM: visual insight

- Mixture of 3 Gaussians in 1D



Cluster association probabilities

- We introduce a K-dimensional r.v.

$$\boldsymbol{z} = (z_1, \ldots, z_k, \ldots, z_K)^T$$

- **z** has a 1-of-K representation $z_k \in \{0, 1\}$
  - A single element equal to one
  - All other elements are zero
  - Vector **z** has K possible states, and: $\sum_{k=1}^{K} z_k = 1$
- **z** are called latent variables
  - Very often it is convenient to extend the model to include them
  - Leads to a richer structure, which simplifies analysis
  - And allows for powerful optimizations (e.g., EM, see later)
  - Here, **z** is used to model the K clusters the data points belong to

Latent variables z

- In statistics Latent variables are
  - *Latin: present participle of "lateo" – "lie hidden"*
  - Variables that cannot be directly measured
  - Are inferred from other variables that are observed (measured)
  - Inference occurs through a suitable mathematical model

- They link
  - Data in the real world to symbolic data (in the model)
  - Can be used to aggregate data leading to
    - Represent an underlying concept
    - Making it easier to understand the data
    - This is intimately related to "dimensionality reduction" tasks

Cluster association probabilities

- We define the pdfs $p(\boldsymbol{x}, \boldsymbol{z}),\ p(\boldsymbol{z})\ , p(\boldsymbol{x}|\boldsymbol{z})$
- Mixing coefficients: $\sum_{k=1}^{K} \pi_k = 1$
- We can write:

$$p(\boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{z_k} \qquad p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{k=1}^{K} G(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$$

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

Latent variables **z**

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

marginalizing over **z**

- Given this equation, for any observed data point $\mathbf{x}_n$ there is a corresponding latent variable $\mathbf{z}_n$
- We have found an equivalent formulation of Gaussian mixtures involving latent variables (vector **z**)
- Although now unclear, this is very important, as it allows to operate using Expectation Maximization (see later) on latent variables
- Powerful tool for automated model optimization

Posterior probability p(**z**|**x**)

- Using Bayes theorem:

joint prob. $p(z_k = 1, \boldsymbol{x})$

$$\gamma(z_k) \triangleq p(z_k = 1|\boldsymbol{x}) = \frac{p(z_k = 1)p(\boldsymbol{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x}|z_j = 1)}$$

$$= \frac{\pi_k G(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j G(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- π_k is the prior probability of $z_k = 1$
- $\gamma(z_k)$: posterior probability, once we have observed $\boldsymbol{x}$
  - can be viewed as the responsibility that component (cluster) k takes for explaining the observation **x**

The Log likelihood

- Suppose we have a dataset of observations $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$
- We can represent it as a D x N data matrix **X**
  - n-th column given by $\boldsymbol{x}_n$
- Similarly K x N matrix **Z** (latent variables)
  - n-th column given by $\boldsymbol{z}_n$
- Data points are drawn i.i.d. from $p(\boldsymbol{x})$

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

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

Maximize the log likelihood (1/5)

- Setting: $\dfrac{\partial \ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} = 0$
- Leads to:

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j G(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\boldsymbol{x}_n - \boldsymbol{\mu}_k)$$

- Multiplying this by $\boldsymbol{\Sigma}_k^{-1}$ (assumed to be non-singular):

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \quad \text{where: } N_k \triangleq \sum_{n=1}^{N} \gamma(z_{nk})$$

Maximize the log likelihood (2/5)

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \quad \text{where:} \quad N_k \triangleq \sum_{n=1}^{N} \gamma(z_{nk})$$

- We can interpret N_k as the effective number of points assigned to cluster k
- The mean vector μ_k for the k-th Gaussian component is obtained by taking a weighted mean of all the points in the data set
- The weighting factor for data point $\mathbf{x}_n$ is given by the posterior probability $\gamma(z_{nk})$ that component (cluster) k was responsible for generating $\mathbf{x}_n$

Maximize the log likelihood (3/5)

- Setting: $\dfrac{\partial \ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k} = 0$
- Leads to (the derivation is rather involved, see [1]):

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

- Has the same form of the corresponding result for a single Gaussian fitted to the data set, but with each point weighted by the corresponding posterior probability and with the denominator given by the effective number of points associated with the corresponding component k

[1] Magnus, J. R. and H. Neudecker, Matrix Differential Calculus with Applications in Statistics and Econometrics, Wiley 1999.

Maximize the log likelihood (4/5)

- Finally, maximize with respect to the mixing coefficients:

$$\max_{\pi_k} \left[\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \right]$$

$$\text{subject to: } \sum_{k=1}^{K} \pi_k = 1$$

- To do this, we define the Lagrangian function

$$J \triangleq \ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

Maximize the log likelihood (5/5)

- Maximize the Lagrangian

$$J \triangleq \ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

$$\frac{\partial J}{\partial \pi_k} = 0 \Rightarrow 0 = \sum_{n=1}^{N} \frac{G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j G(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda$$

- If we multiply both sides by π_k and sum over k we get $\lambda = -N$
- Using this to eliminate λ and rearranging gives:

$$\pi_k = \frac{N_k}{N}$$

Discussion

- This result is not obtained in closed-form
- But it suggests a simple iterative procedure to find a solution to the maximum likelihood problem
- This procedure is an instance of the Expectation Maximization algorithm (see later) for the particular case of the GM

EM for Gaussian Mixtures (GM) (1/2)

1. Initialize the means, the variances and the mixing coefficients and evaluate the initial value of the log likelihood

2. **E step.** Evaluate (estimate) the (current) responsibilities (posterior probabilities) using the current parameter values:

$$\gamma(z_{nk}) = \frac{\pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j G(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

EM for GM (2/2)

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

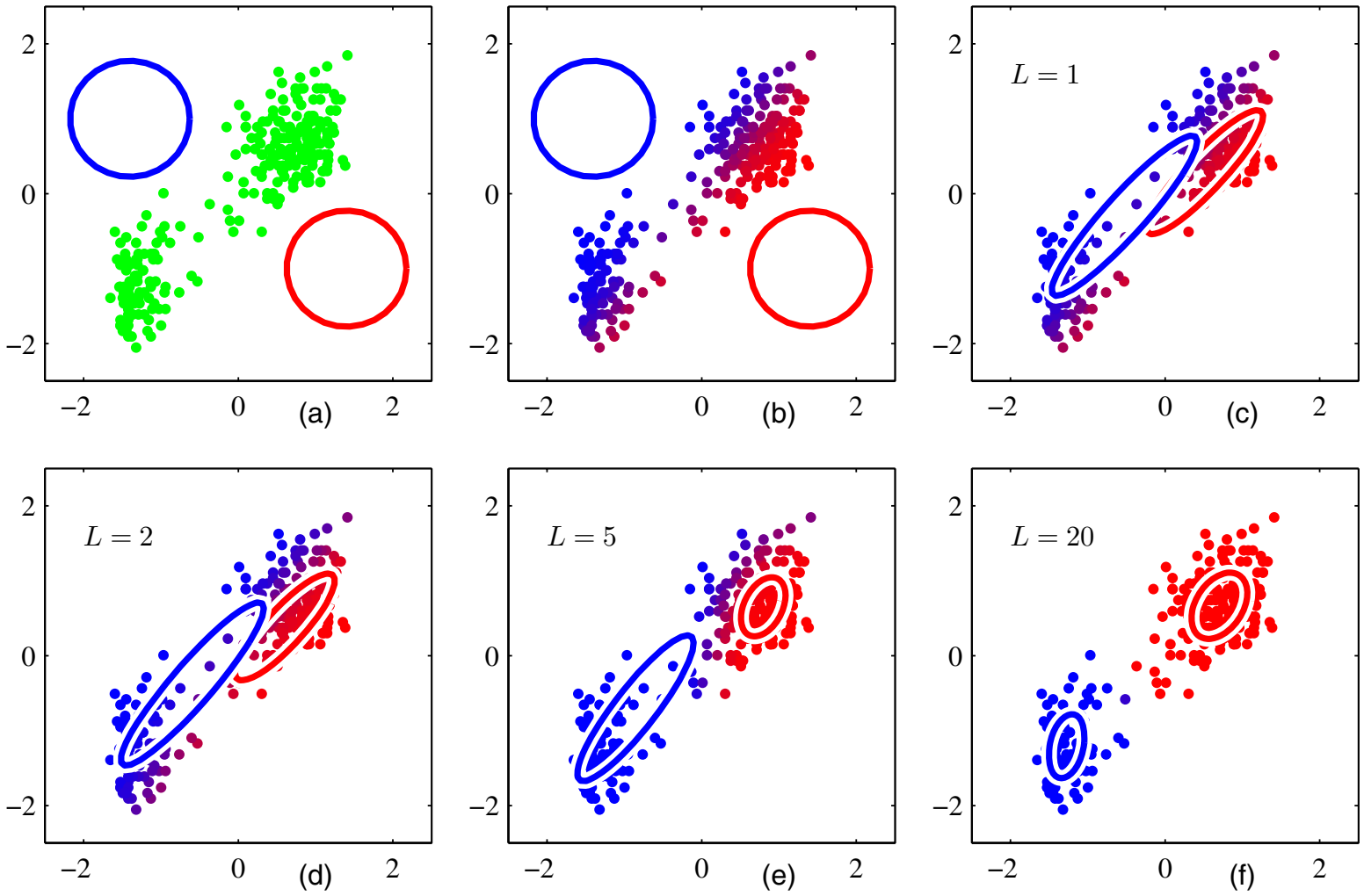
$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n$$

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

$$\pi_k^{\text{new}} = \frac{N_k}{N} \qquad \text{where } N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

4. Evaluate the log likelihood $\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and check for convergence. If NOT, go back to step 2.

Example



Observations

- EM can take many more iterations than K-means to converge
  - Higher computational complexity
- It is common to use K-means to initialize the GM model
  - The GM model is then adapted through EM
  - The GM covariance matrices can be conveniently initialized by the sample covariance matrices of the K clusters found by the K-means algorithm
  - The mixing coefficients can be set to the fraction of data points that are assigned to the clusters that are found by K-means
- Measures should be taken to avoid singularities, i.e., when a Gaussian component (cluster) collapses into a single data point

EXPECTATION MAXIMIZATION

EM in deeper detail

- EM is a general optimization procedure
- Widely used in machine learning
- Used to
  - find maximum likelihood solutions
  - for *models having latent parameters*
- If **X** is the data matrix, **Z** is the matrix of latent variables and vector $\boldsymbol{\theta}$ contains all the *system parameters*
- The log likelihood can be written as (marginalizing over **Z**)

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \right\}$$

(for continous r.v. just replace sum with integral)

General treatment of EM

- Guaranteed to converge (lead to improvement) at each step
- Our goal is to maximize the likelihood function

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

- Very often direct optimization of $p(\boldsymbol{X}|\boldsymbol{\theta})$ is difficult
- Working in the latent space **Z** instead allows for much simpler forms, which are often easier to deal with
- We now introduce a distribution $q(\boldsymbol{Z})$
- We use such distribution to decompose the log likelihood

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta})$$

Likelihood decomposition

- The likelihood:

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \right\}$$

- Can be decomposed as:

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q\|p)$$

- where:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \ln \left\{ \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{q(\boldsymbol{Z})} \right\}$$

$$\mathrm{KL}(q\|p) = -\sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \ln \left\{ \frac{p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})}{q(\boldsymbol{Z})} \right\}$$

(Kullback-Leibler divergence: quantifies the dissimilarity between two distributions)

Verifying the decomposition

- We first use the product rule (aka "chain rule") of probability:

$$p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) = p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{X}|\boldsymbol{\theta})$$

- Using this into $\mathcal{L}(q, \boldsymbol{\theta})$ leads to:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \left[\ln \frac{p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})}{q(\boldsymbol{Z})} + \ln p(\boldsymbol{X}|\boldsymbol{\theta}) \right]$$

- The first term cancels out: $\mathrm{KL}(q||p)$
- The second term is: $\sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \ln p(\boldsymbol{X}|\boldsymbol{\theta})$

QED

Properties of the decomposition

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \mathcal{L}(q,\boldsymbol{\theta}) + \mathrm{KL}(q\|p)$$

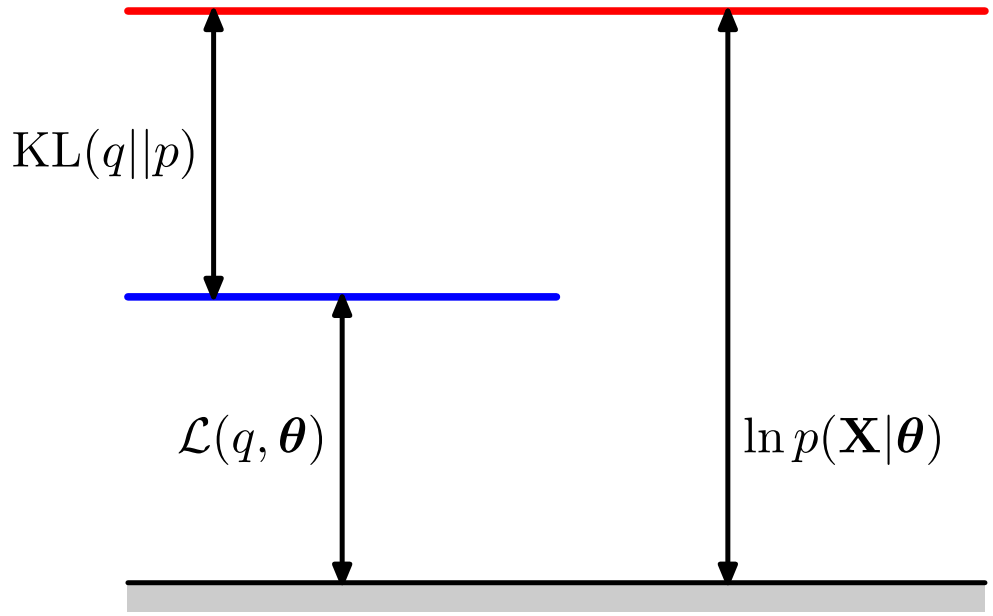
- One key property of KL is that:

$$\mathrm{KL}(q\|p) \geq 0 \text{ and } \mathrm{KL}(q\|p) = 0 \text{ iff } q = p$$

- Hence:

$$\mathcal{L}(q,\boldsymbol{\theta}) \leq \ln p(\boldsymbol{X}|\boldsymbol{\theta})$$

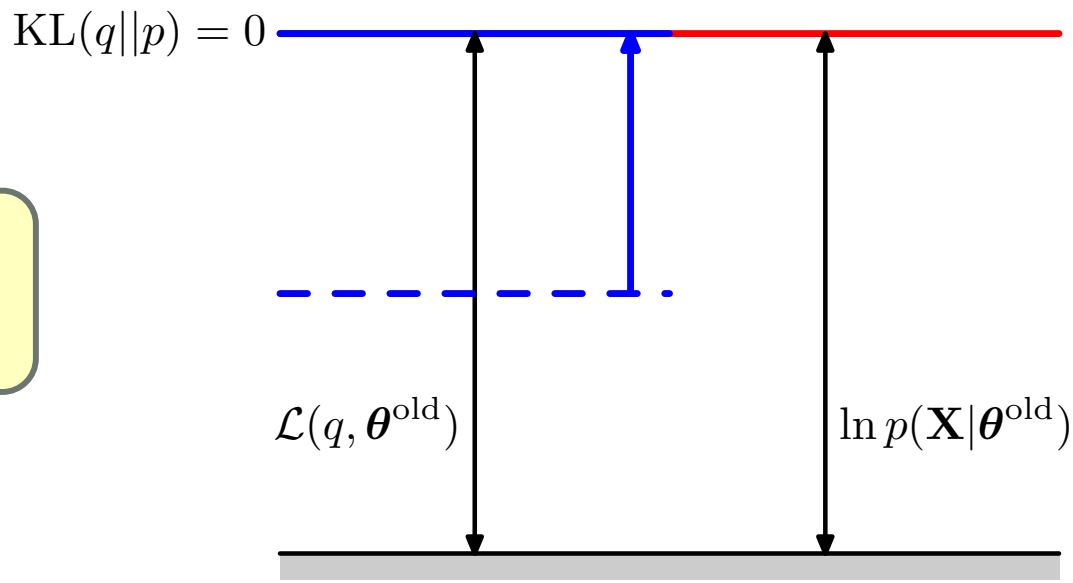
$\mathcal{L}(q,\boldsymbol{\theta})$ is a *lower bound* on the log likelihood function $\ln p(\boldsymbol{X}|\boldsymbol{\theta})$



EM – E step

- Suppose the current value of the parameter vector is $\boldsymbol{\theta}^{\text{old}}$
- **E step:** lower bound $\mathcal{L}(q, \boldsymbol{\theta}^{\text{old}})$ is maximized wrt $q(\boldsymbol{Z})$
  - Keeping $\boldsymbol{\theta}^{\text{old}}$ fixed
- Solution to this maximization problem:
  - $\ln p(\boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})$ does not depend on $q(\boldsymbol{Z})$
  - $\mathcal{L}(q, \boldsymbol{\theta}^{\text{old}})$ is max. when $\text{KL}(q||p) = 0 \Rightarrow q = p$

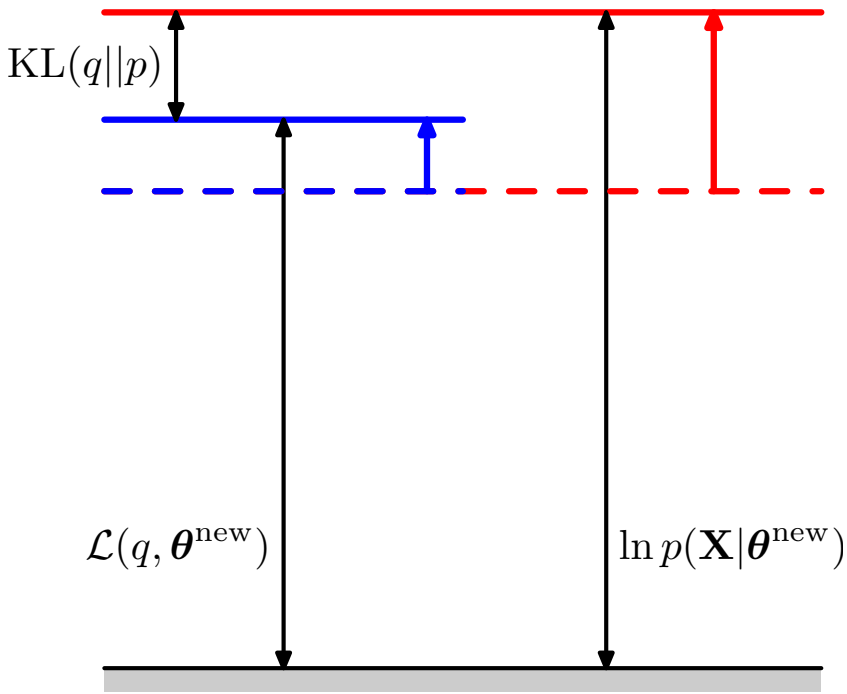
$$q(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}})$$



EM – M step

- **M-step:** now, the distribution $q(\boldsymbol{Z})$ is held fixed and the lower bound $\mathcal{L}(q, \boldsymbol{\theta})$ is maximized wrt $\boldsymbol{\theta}$
  - This returns a new value $\boldsymbol{\theta}^{\text{new}}$
- The lower bound increases (unless it is already max.)
- Since the distribution $q(\boldsymbol{Z})$ is determined using $\boldsymbol{\theta}^{\text{old}}$
- We have $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{new}}) \neq q(\boldsymbol{Z})$
- The KL divergence will be >0
- Increase in the log likelihood is:

KL + increase in lower bound



M step – "bottom line"

- We maximize $\mathcal{L}(q, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$

- If $\boldsymbol{\theta}^{\text{new}} \neq \boldsymbol{\theta}^{\text{old}}$
  - The log likelihood must improve
  - This is granted by the fact that the KL divergence is >0

- This means that the log likelihood can only increase
  - as we keep going through E and M steps
  - until we reach a maximum
  - the maximum can either be global or local

M step in greater detail

- From the E-step we have: $q(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}})$
- Using this equality into the lower bound, we get:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \ln \left\{ \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{q(\boldsymbol{Z})} \right\} =$$

$$= \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) - \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}})$$

$$\triangleq Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \text{const}$$

- The second term is constant (does not depend on $\boldsymbol{\theta}$)
- First term: is the expectation of the log likelihood of p(**X**,**Z**|θ) using the old parameter estimates for this calculation

The general EM algorithm

Given: joint distribution $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$

Goal: maximize likelihood $p(\boldsymbol{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$

1. Chose an initial parameter vector $\boldsymbol{\theta}^{\text{old}}$
2. **E step.** Evaluate $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}})$ (posterior pdf)
3. **M step.** Evaluate $\boldsymbol{\theta}^{\text{new}} = \text{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$

where:

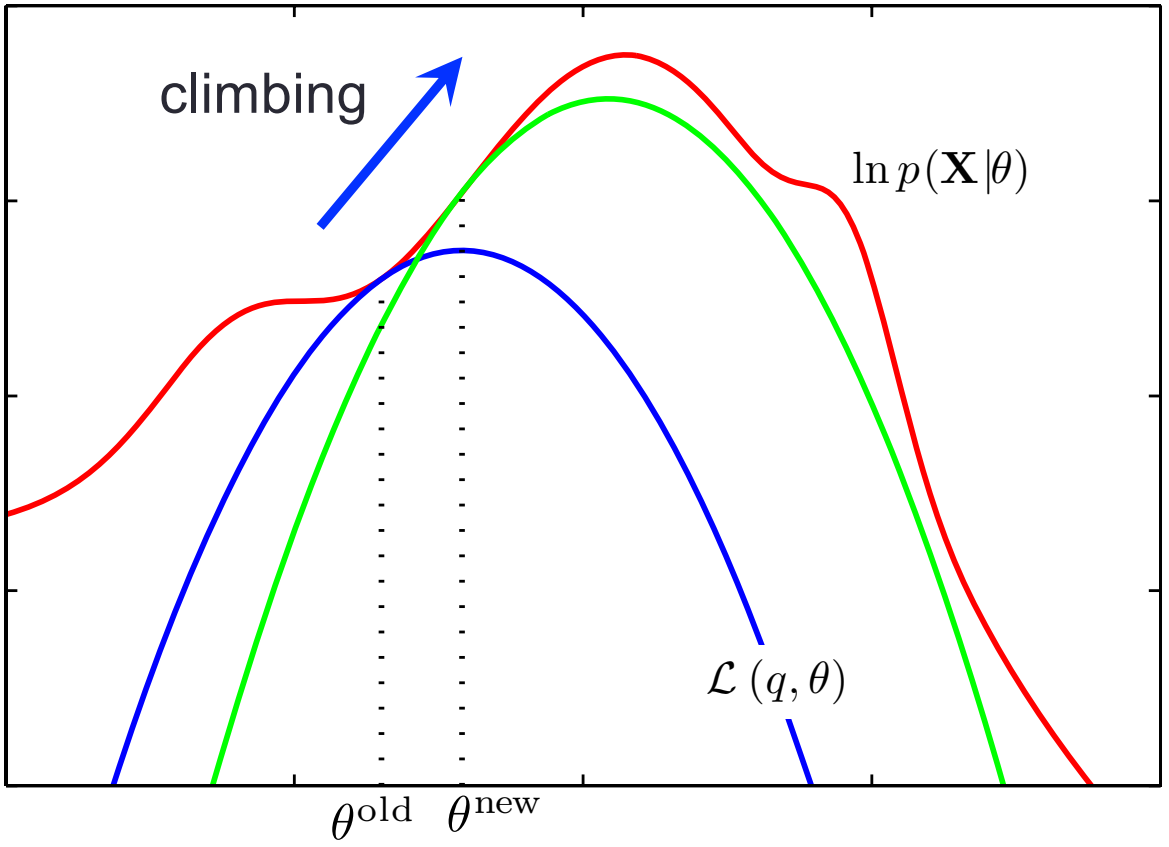
$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) =$$

$$= E_{\boldsymbol{Z}} \left[\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) | \boldsymbol{X}, \boldsymbol{\theta} = \boldsymbol{\theta}^{\text{old}} \right]$$

4. Check for convergence: for either the log likelihood or the parameter values. If convergence is not achieved then set $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$ and go back to step 2

Geometric interpretation

1. **E step:** evaluate posterior pdf over latent variables
2. This leads to a lower bound that makes a tangential contact with the log likelihood in θ^{old}
3. This means that both curves have the same gradient
4. The lower bound is maximized (leading to θ^{new}), leading to a larger value of the likelihood than in θ^{old}



The EM algorithm alternately (**E step**) computes a lower bound on the log likelihood for the current parameter values and then **M step** to maximize this lower bound to obtain the new parameter values. The lower bound is often convex (e.g., for pdfs of the exponential family) → single maximum

K-means revisited – joint pdf

- For a single vector: $p(\boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{k=1}^{K} G(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$$

- For N measurements:

$$p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

$$\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}\}$$

- Taking the logarithm:

$$\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \{\ln \pi_k + \ln G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

K-means revisited – posterior pdf (1/2)

- Posterior pdf (from Bayes):

$$p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}) = \frac{p(\boldsymbol{Z},\boldsymbol{X}|\boldsymbol{\theta})}{p(\boldsymbol{X}|\boldsymbol{\theta})}$$

$$\text{(from previous slide)} = \prod_{n=1}^{N} \left[\frac{\prod_{k=1}^{K} \left[\pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right]^{z_{nk}}}{p(\boldsymbol{X}|\boldsymbol{\theta})} \right]$$

$$\text{(the joint pdf can be factorized)} = \prod_{n=1}^{N} p(\boldsymbol{z}_n|\boldsymbol{X},\boldsymbol{\theta})$$

We have: $\boldsymbol{z}_n = (z_{n1}, z_{n2}, \ldots, z_{nK})^T$

- 1-of-K (only one element =1, all others=0)

K-means revisited – posterior pdf (2/2)

$$p(z_{nk}=1|\boldsymbol{X},\boldsymbol{\theta}) =? \quad \text{with: } \boldsymbol{z}_n = (z_{n1}, z_{n2}, \ldots, z_{nK})^T$$

We have obtained:

$$p(\boldsymbol{z}_n|\boldsymbol{X},\boldsymbol{\theta}) = \frac{\prod_{k=1}^{K}\left[\pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu_k},\boldsymbol{\Sigma}_k)\right]^{z_{nk}}}{\sum_{\boldsymbol{z}_n}\prod_{j=1}^{K}\left[\pi_j G(\boldsymbol{x}_n|\boldsymbol{\mu_j},\boldsymbol{\Sigma}_j)\right]^{z_{nj}}}$$

normalizing constant such that: $\sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{X},\boldsymbol{\theta}) = 1$

IF $z_{nk}=1$ we have ($q(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})$):

$$p(z_{nk}=1|\boldsymbol{X},\boldsymbol{\theta}) = \frac{\pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K}\pi_j G(\boldsymbol{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)} \triangleq \gamma(z_{nk})$$

K-means revisited – expectation of z_{nk}

- The expected value or the indicator variable z_{nk}
  - Under the posterior distribution
  - Is obtained as:

$$E_{\boldsymbol{Z}}[z_{nk}|\boldsymbol{X},\boldsymbol{\theta}] = p(z_{nk}=1|\boldsymbol{X},\boldsymbol{\theta}) = \gamma(z_{nk})$$

(key property of indicator functions: their expected value equals the probability that the event they indicate is verified, in this case that z_{nk}=1)

Expectation of log of joint pdf

- Recall that:

$$\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \{\ln \pi_k + \ln G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

- Applying the expectation over **Z**

$$E_{\boldsymbol{Z}}[\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})] = \sum_{n=1}^{N} \sum_{k=1}^{K} E_{\boldsymbol{Z}}[z_{nk}] \{\ln \pi_k + \ln G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \{\ln \pi_k + \ln G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

from previous slide
note that this is $q(z_{nk})$

Remark

- Note that:

$$E_{\boldsymbol{Z}}\left[\ln p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})|\boldsymbol{X},\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{old}}\right]=Q(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}})$$

- which is the Q function in the EM procedure
- The old parameters correspond to:

$$\boldsymbol{\theta}^{\text{old}}=\{\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k,\boldsymbol{\pi}_k\}$$

- The next step is to maximize this expectation
- This will lead to the new parameters
  - expressed as a function of the old ones

M step

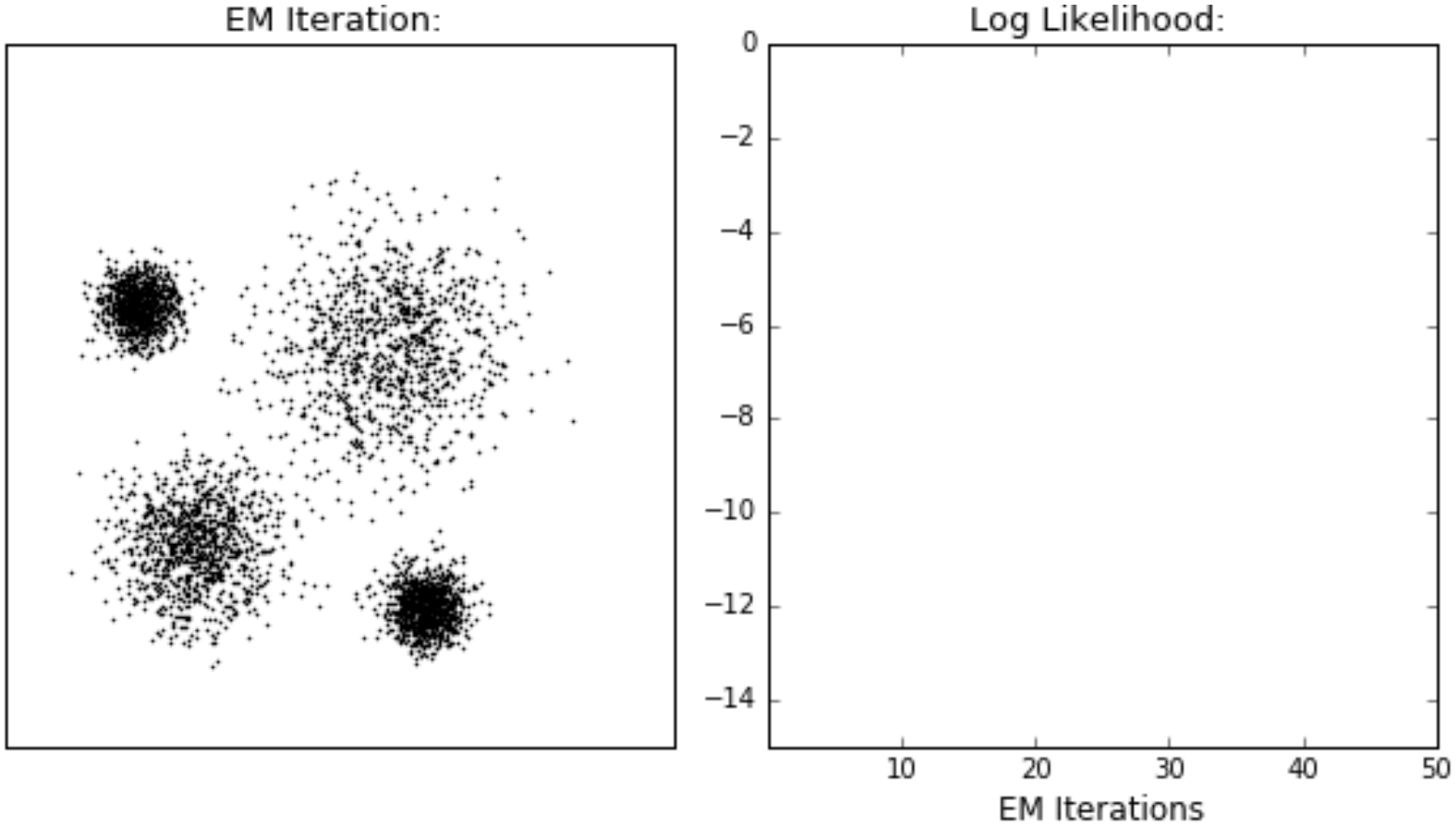
- Maximize the expectation of the (logarithm of) joint pdf
  - Obtain new values of parameters
  - As a function of old parameters
  - The expressions coincide with those that we have previously obtained
  - This involves a simpler calculation
    - the log is applied to each single Gaussian component, rather than to the sum of all components; compare with log-likelihood in slide 26

$$\frac{\partial E_{\boldsymbol{Z}}[\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})]}{\partial \boldsymbol{\pi}_k} = 0 \rightarrow \boldsymbol{\pi}_k^{\text{new}}$$

$$\frac{\partial E_{\boldsymbol{Z}}[\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})]}{\partial \boldsymbol{\mu}_k} = 0 \rightarrow \boldsymbol{\mu}_k^{\text{new}}$$

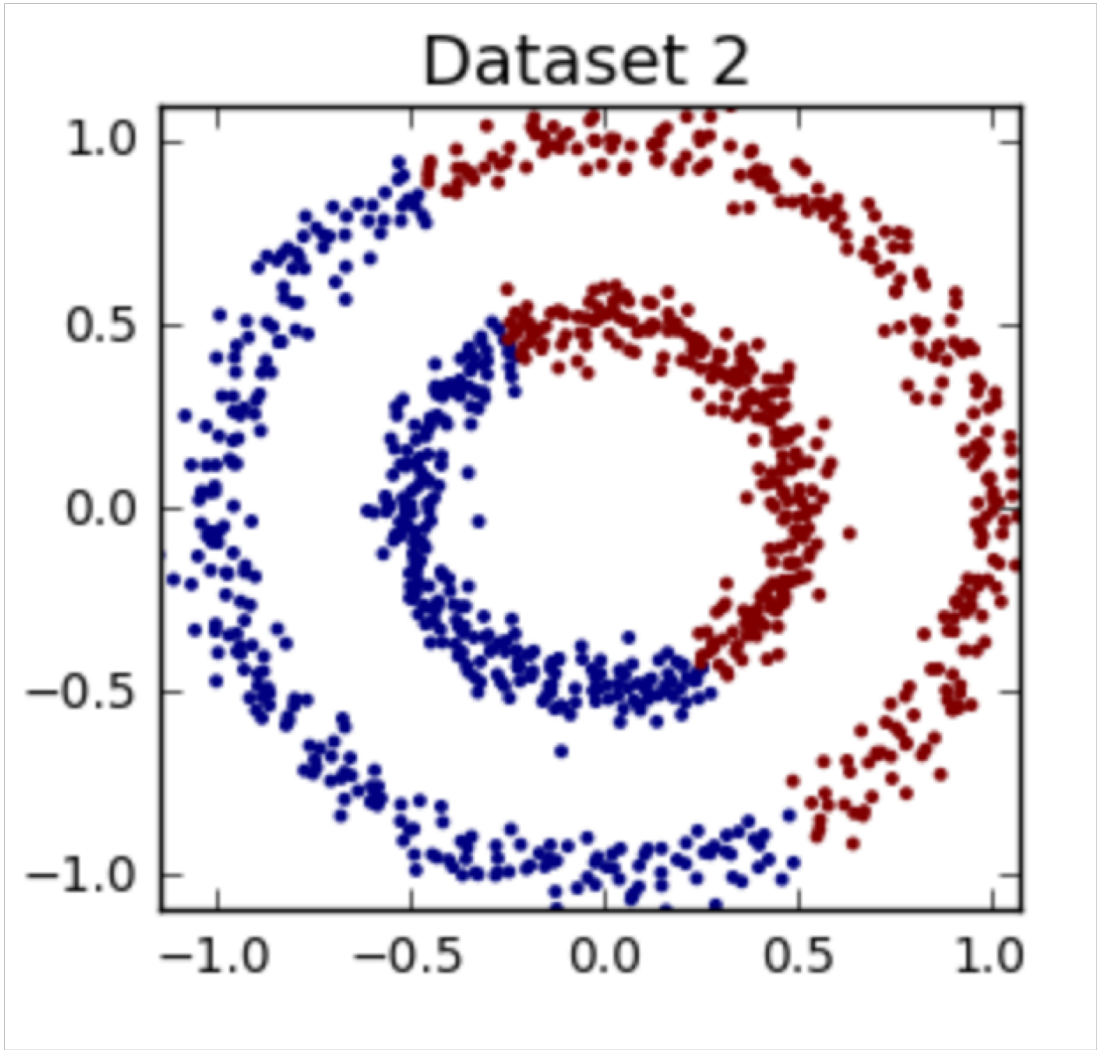
$$\frac{\partial E_{\boldsymbol{Z}}[\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})]}{\partial \boldsymbol{\Sigma}_k} = 0 \rightarrow \boldsymbol{\Sigma}_k^{\text{new}}$$

Dataset1



- Everything… in its right place (Kid A, 2000)
  - Expected: points are Gaussian distributed

A problematic dataset



- The data distribution cannot be accurately modeled by a GMM: also GM-based soft-clustering fails

Right number of clusters?

- Up to now
  - The number of clusters has to be supplied by the user
- Would it be possible to learn K from data?
  - A very popular approach is proposed in [Peleg-00]
  - X-means
    - Intelligently uses the Bayesian Information Criterion (**BIC**)

[Peleg-00] Dan Peleg, Andrew Moore, "**X-means**: Extending K-means with Efficient Estimation of the Number of Clusters," *International Conf. on Machine Learning (ICML)*, Stanford, CA, USA, 2000.

X-means in a nutshell (1/2)

- X-means starts with K equal to the lower bound (user defined)
  - Continues to add centroids
    - Where they are needed
    - Until the upper bound is reached
  - The configuration (number of centroids and their location) achieving the best score is recorded during the search
    - Such configuration is finally output by the algorithm

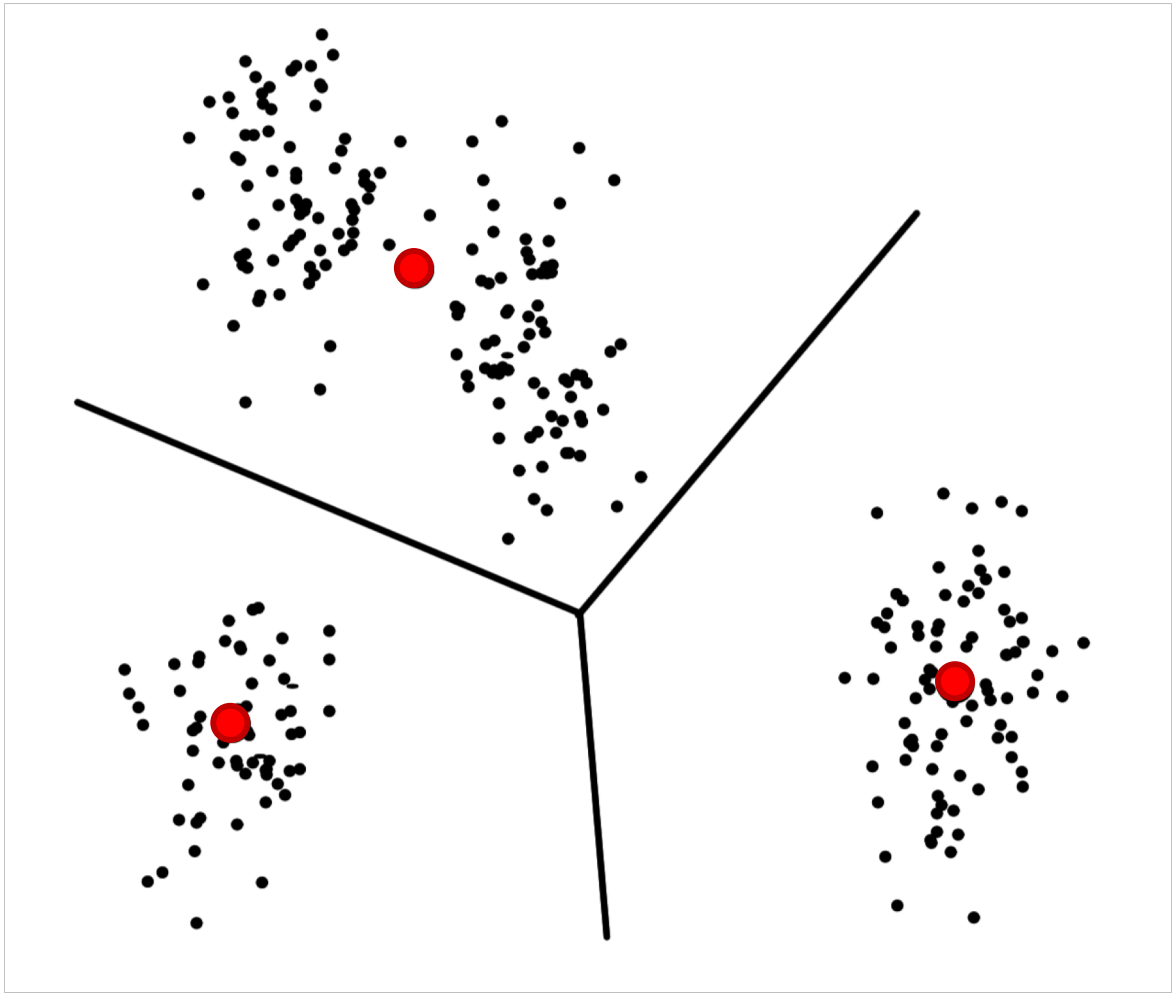
X-means in a nutshell (2/2)

X-means algorithm:

1. **Improve-Params:** consists of running simple K-means to converge to a new configuration
2. **Improve-Structure:** finds out if and where new centroids should appear. This is achieved by letting *some* centroids split in two.
3. **If** all configurations are explored **stop** and report the best model found, **Else**, Goto 1.

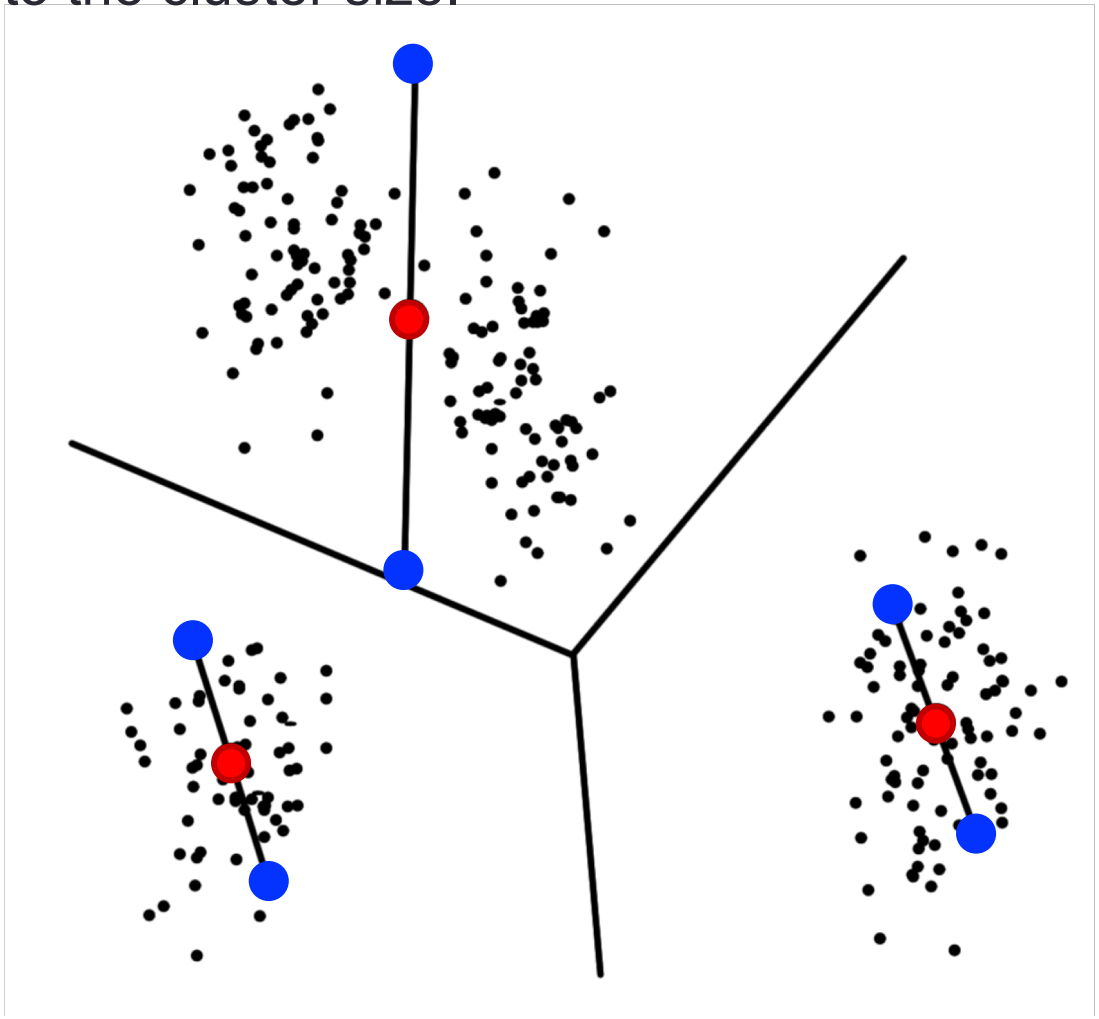
X-means by example

- Current model: stable current solution with 3 centroids



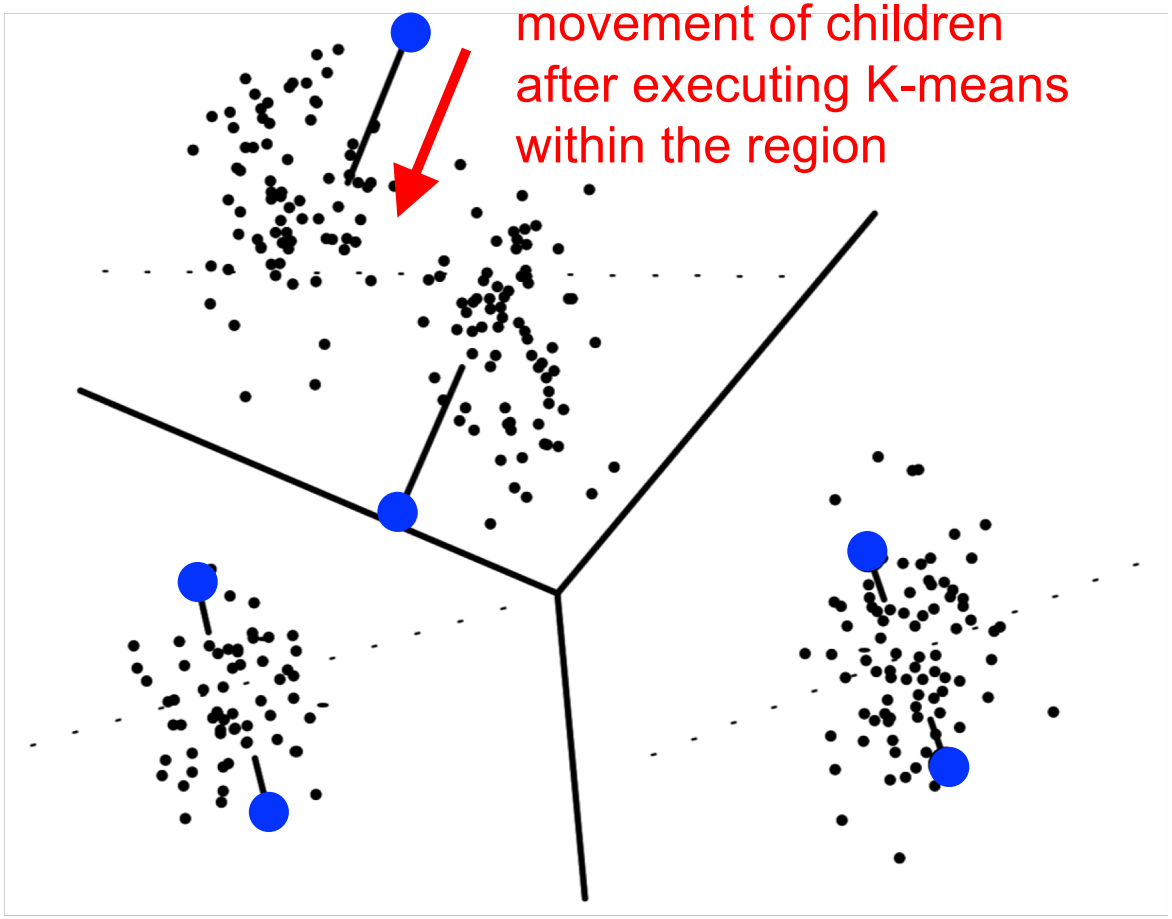
X-means by example

- Structure improvement operation: starts by *splitting each centroid into two children* (blue dots). They are moved at a distance that is proportional to the cluster size.



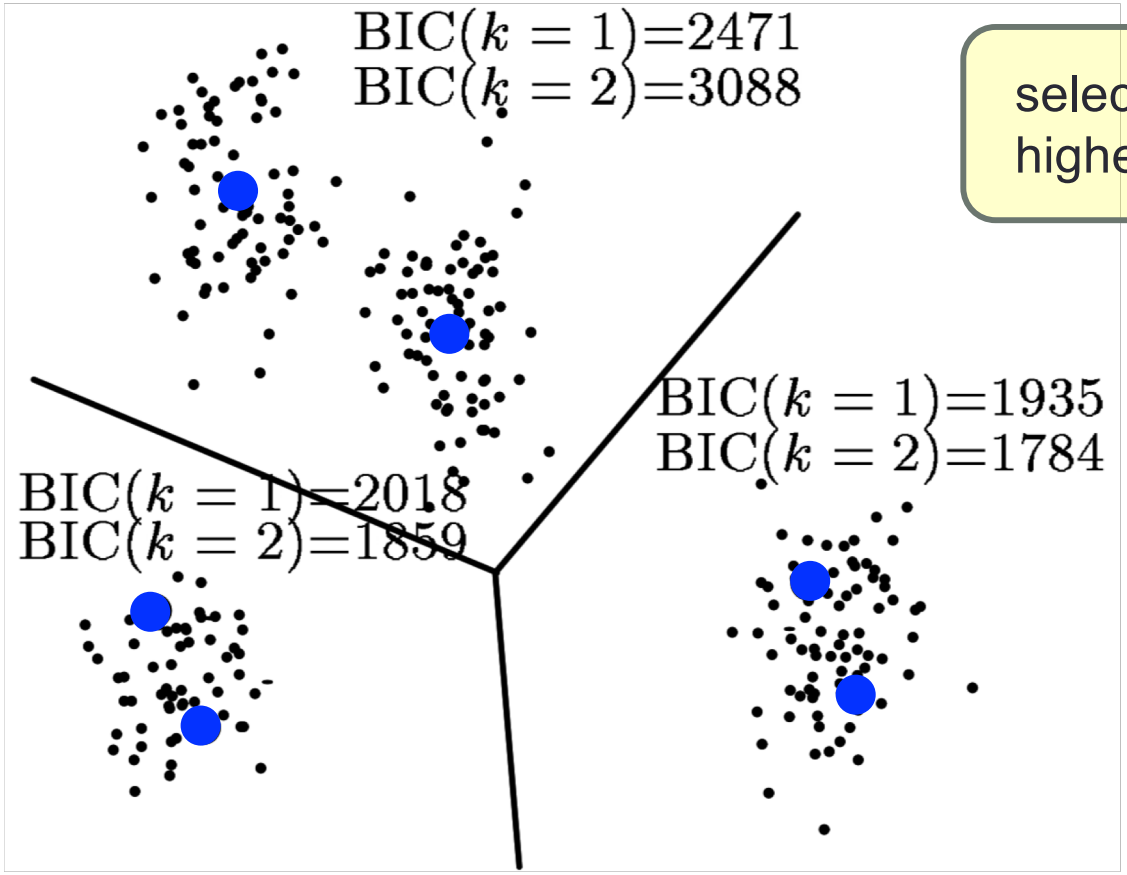
X-means by example

- Improve params: use standard K-means (with K=2) inside each region for each pair of children. It is a *local procedure* as the children fight with each other for the points in the parent's region: no other.



X-means by example

- Improve structure: a model selection test is executed on all pairs of children. In each region, the test asks: "*is there evidence that the two children are modeling real structure? Would the original parent model the distribution equally well?*"



Choosing the best model for each region

- For each region, we need to weigh the following alternatives
  - K=1 – Model M_1 (all the N nodes in one region)
    - The parent node better represents the region
    - This is the configuration with a single parent node (prior to applying K-means)
    - All the N points in the region are generated by a single Gaussian component
  - K=2 – Model M_2 ($N = N_1 + N_2$, nodes split into two sub-regions)
    - The two children nodes better represent the region
    - Two Gaussian components generate the points of the two-subclusters in the region
    - N_1 and N_2 are the number of points in the two sub-regions

Data log-likelihood

- The data log-likelihood L is:

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

- The centroids $\boldsymbol{\mu}_k$ have already been estimated by K-means
- The covariance matrices maximizing L are to be estimated
  - For each Gaussian component (cluster): $\pi_k = N_k/N$
  - We know that, the co-variances that maximize L are:

$$\boldsymbol{\Sigma}_k^* = \frac{1}{N_k'} \sum_{n=1}^{N} \gamma(z_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T$$

$$\text{where: } N_k' \triangleq \sum_{n=1}^{N} \gamma(z_{nk}) \qquad \gamma(z_{nk}) = \frac{\pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j G(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

For each hypothesis (M_1, M_2)

- The data log-likelihood L for the two models is:

- Model M_1
  - Single cluster, single Gaussian component, number of points N
  - Data log-likelihood is (K=1):

$$L(M_1) = L(\boldsymbol{\mu}, \boldsymbol{\Sigma}^*)$$

- Model M_2
  - Two clusters, two Gaussian components, no. of points N_1 and N_2
  - Data log-likelihood is (K=2):

$$L(M_2) = L(\pi_1, \pi_2, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1^*, \boldsymbol{\Sigma}_2^*)$$

Scoring the models (M_1, M_2)

- If $\mathcal{P}$ represents the set of data points in the sub-region
- Models are scored based on posterior probabilities

$$\Pr[M_j|\mathcal{P}]\,,\; j=1,2$$

- A common way to *approximate* them is the BIC formula [Kass-95]

$$\mathrm{BIC}(M_j) = L(M_j) - \frac{p_j}{2}\ln N$$

[Kass-95] Robert E. Kass and Larry Wasserman, "A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion," *Journal of the American Statistical Association*, Vol. 90, No. 431, pp. 928-934, 1995.

Scoring the models (M_1, M_2)

- If $\mathcal{P}$ represents the set of data points in the sub-region
- Models are scored based on posterior probabilities

$$\Pr[M_j|\mathcal{P}]\,,\; j = 1, 2$$

- A common way to *approximate* them is the BIC formula [Kass-95]

$$\mathrm{BIC}(M_j) = L(M_j) - \frac{p_j}{2}\ln N$$

NOTE: BIC is a very popular and effective approximation technique, but other metrics are as well possible, for example, AIC or MDL, this very much depends on the underlying structure of the data

Scoring the models (M_1, M_2)

- If $\mathcal{P}$ represents the set of data points in the sub-region
- Models are scored based on posterior probabilities

$$\Pr[M_j|\mathcal{P}]\,,\; j = 1, 2$$

- A common way to approximate these posteriors is the BIC formula

$$\mathrm{BIC}(M_j) = L(M_j) - \frac{p_j}{2}\ln N$$

- Number of model parameters

$$p_j = K_j - 1 + K_j D + C_j$$

$K_j - 1$	$K_j D$	C_j
class probabilities	centroid coordinates	number of parameters to estimate in covariance matrices

Shape of covariances

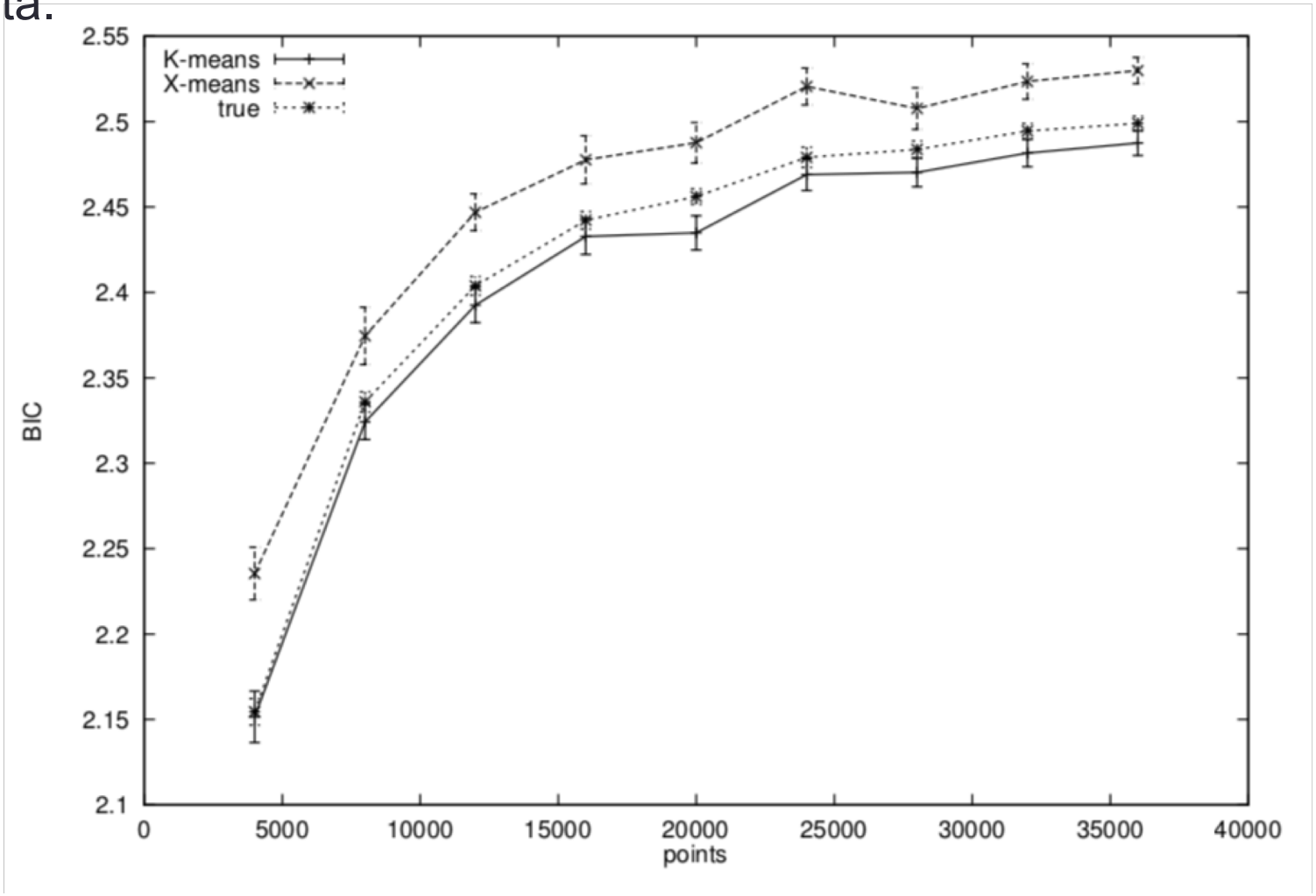
- By whitening the input data
- We can obtain Gaussians with elliptical shape

$$\boldsymbol{\Sigma}_j = \begin{bmatrix} \sigma_{j,1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{j,2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \sigma_{j,D}^2 \end{bmatrix}$$

- If $\sigma_{i,j}^2 = \sigma^2 \ , \ \forall j$: spherical Gaussians
- Number of free parameters C_j:
  - C_j = D (M_1) 2D (M_2): elliptical
  - C_j = 1 (M_1), 2 (M_2): spherical
  - C_j = 1 (M_1 and M_2): spherical with same variance for all clusters
  - C_j = D^2 (M_1), $2D^2$ (M_2): full covariance matrix

Example results

Average BIC per point (BIC / number of data points), 2D data with 100 real classes. The label “true” stands for the BIC score of the centroids used to generate the data.



References

Reference book:

[1] Christopher M. Bishop, “Pattern Recognition and Machine Learning,” Springer 2006. Chapter 9 “Mixture Models and EM”

Papers:

[2] Dan Peleg, Andrew Moore, “X-means: Extending K-means with Efficient Estimation of the Number of Clusters,” *International Conference on Machine Learning (ICML)*, Stanford, CA, USA, 2000. (+2k citations)

[3] Robert E. Kass and Larry Wasserman, "A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion," *Journal of the American Statistical Association*, Vol. 90, No. 431, pp. 928-934, 1995. (+1k citations)

K-MEANS & EXPECTATION MAXIMIZATION

Michele Rossi

rossi@dei.unipd.it

Dept. of Information Engineering

University of Padova, IT