Master's degree ICT Internet Multimedia Engineering

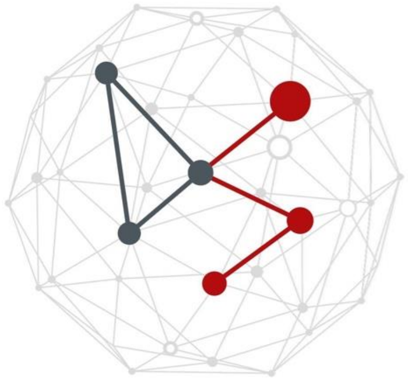
INTRODUCTION TO CLUSTERING

Michele Rossi

michele.rossi@unipd.it

Dept. of Information Engineering

University of Padova, IT

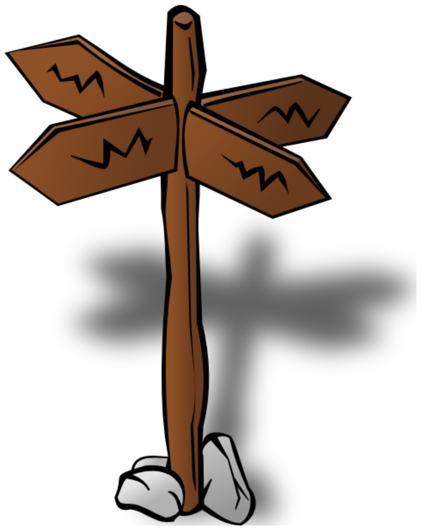


DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO
MATEMATICA

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Overview



- K-means clustering
  - Hard assignment of points to clusters
  - Lloyd algorithm
  - Examples and discussion
- Mixtures of Gaussians r.v.s.
  - K-means with Gaussian mixtures (“soft” K-means)
  - Examples and discussion
- X-means
  - Automatically finding the best K
  - Numerical results

The problem

- Data points in a multidimensional space
- Dataset: $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ in an m dimensional space $\boldsymbol{x}_i \in \mathbb{R}^m$
- Goal (informal)
    - partition the points into K disjoint clusters (K is given)
    - Use Euclidean distance (i.e., norm-2)
- For each cluster i =1,2,…, K
    - We define a prototype (or centroid) $\boldsymbol{\mu}_i \in \mathbb{R}^m$
    - 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”):

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

- 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, for each n, sets r_{nk} to 1 for whichever value of k returns the minimum value of $\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$

- Formally

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

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)

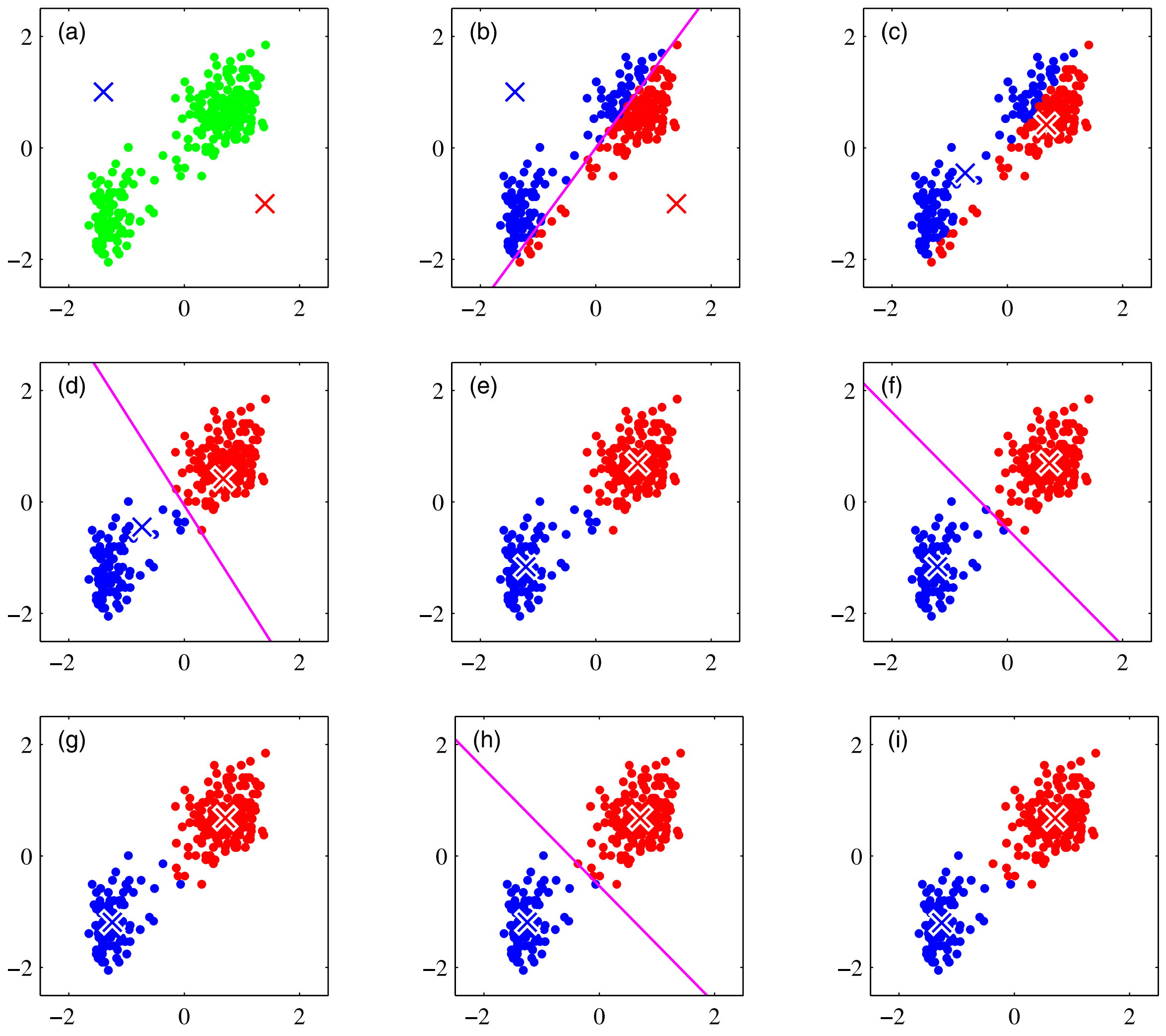
$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = 0 \Rightarrow 2 \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

- This leads to the update (rearranging terms):

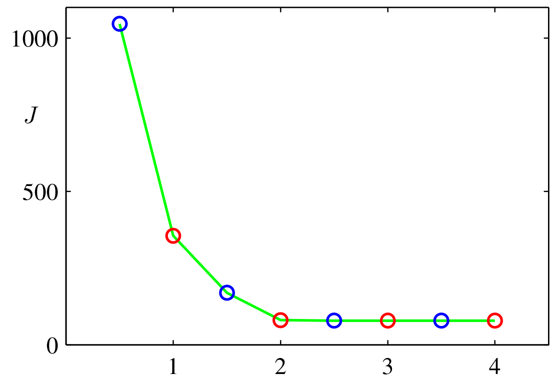
$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n}{\sum_{n=1}^{N} r_{nk}}$$

(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



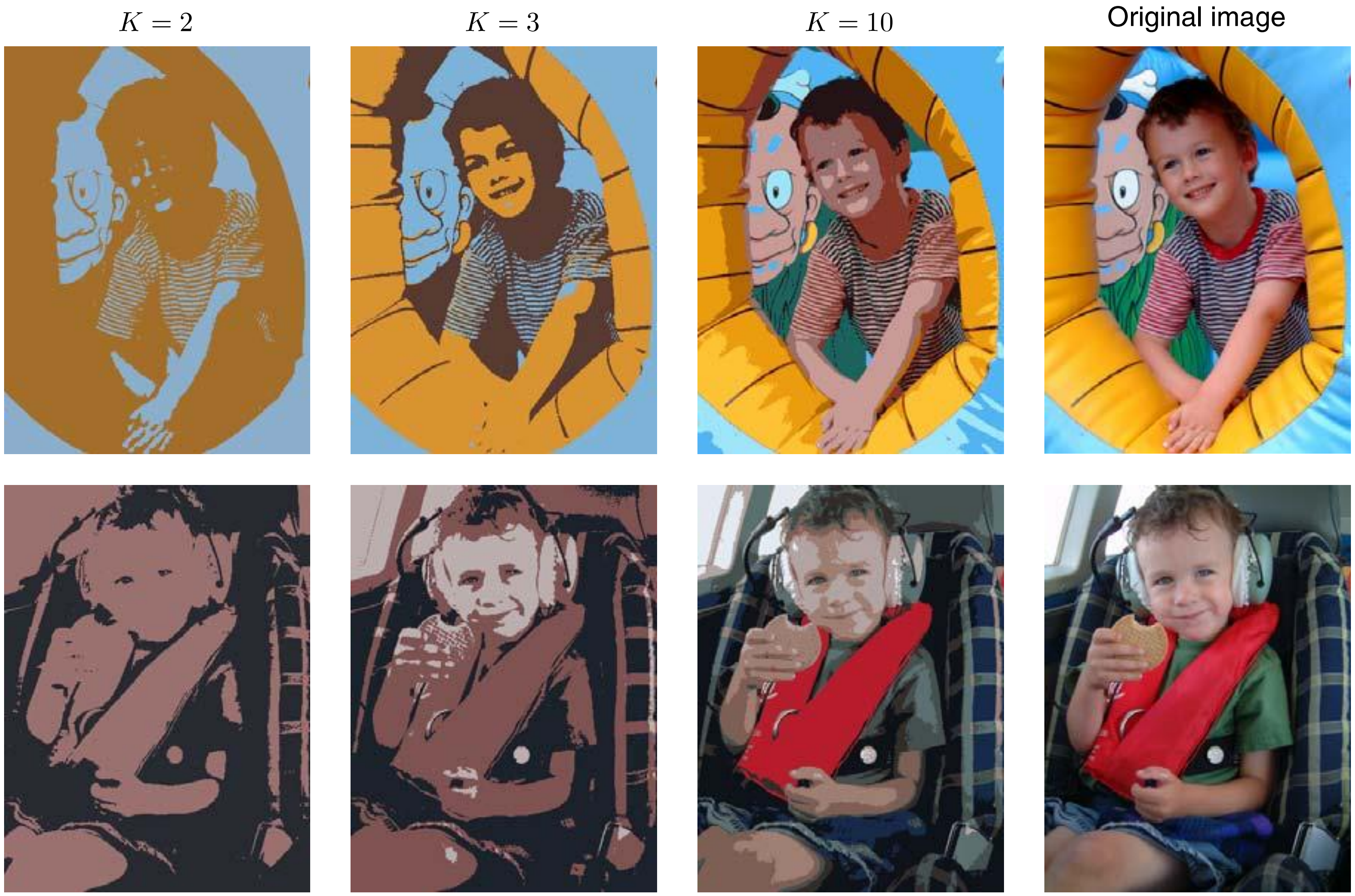
K-medoids

- 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



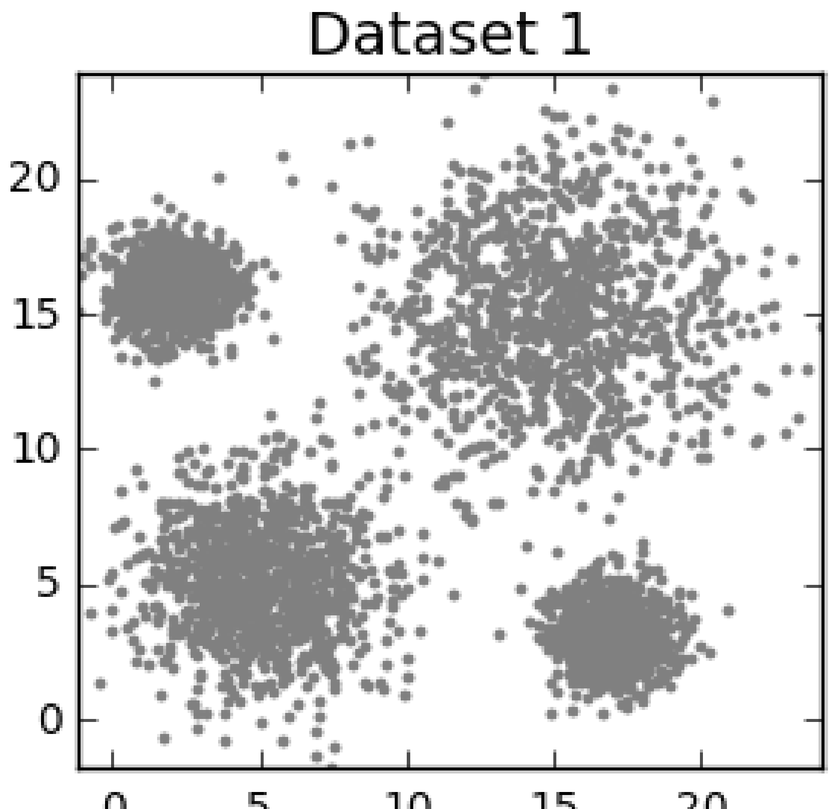
Cluster pixels according to their color (R,G,B) triple

Fewer clusters = more “compression”

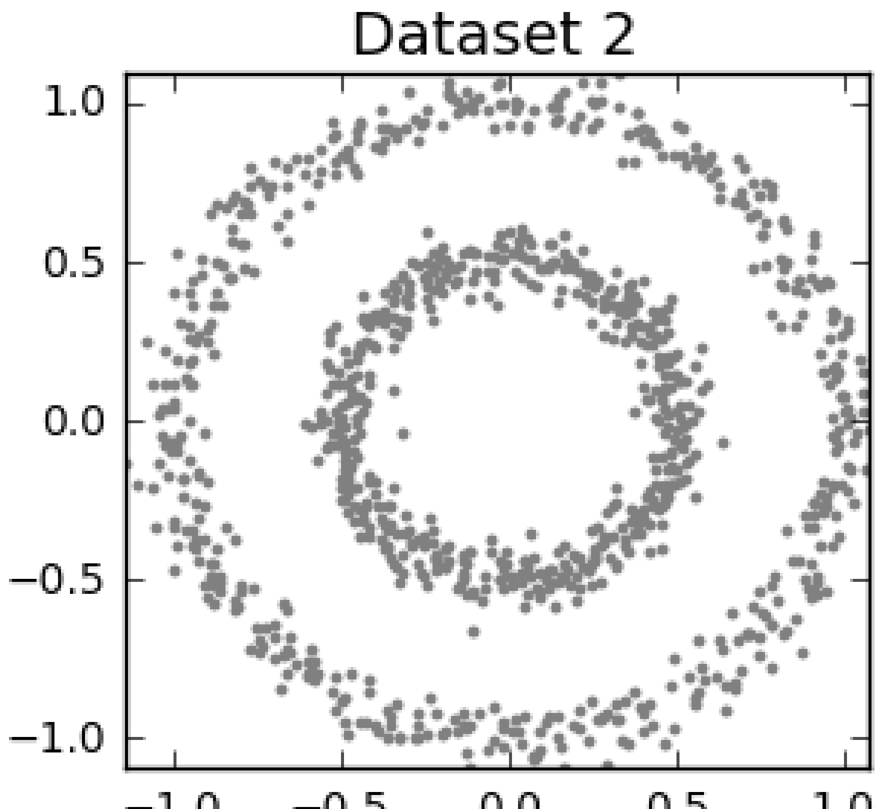
A few questions about K-means

- 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

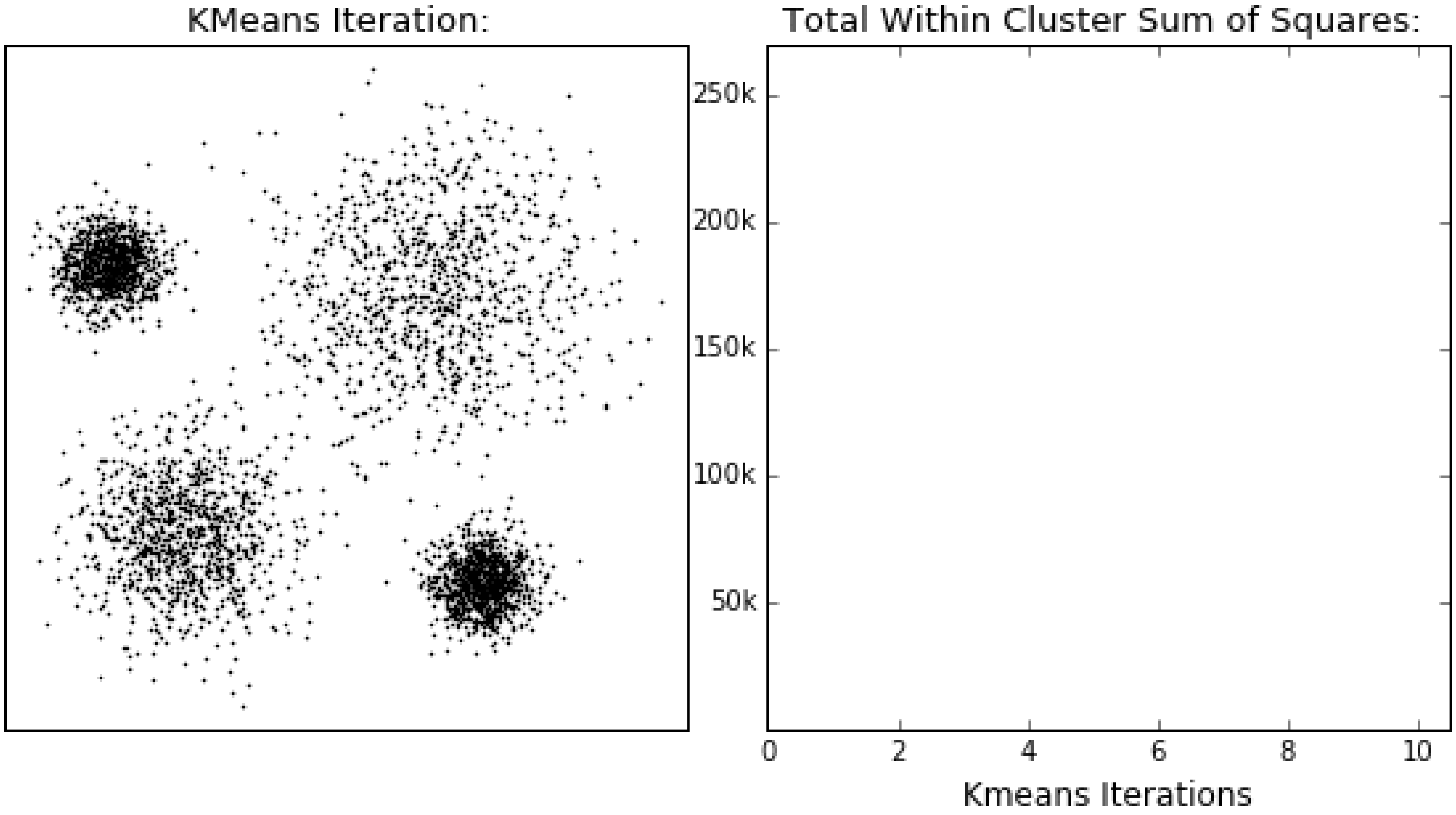


globular clusters



clusters on rings

Dataset 1



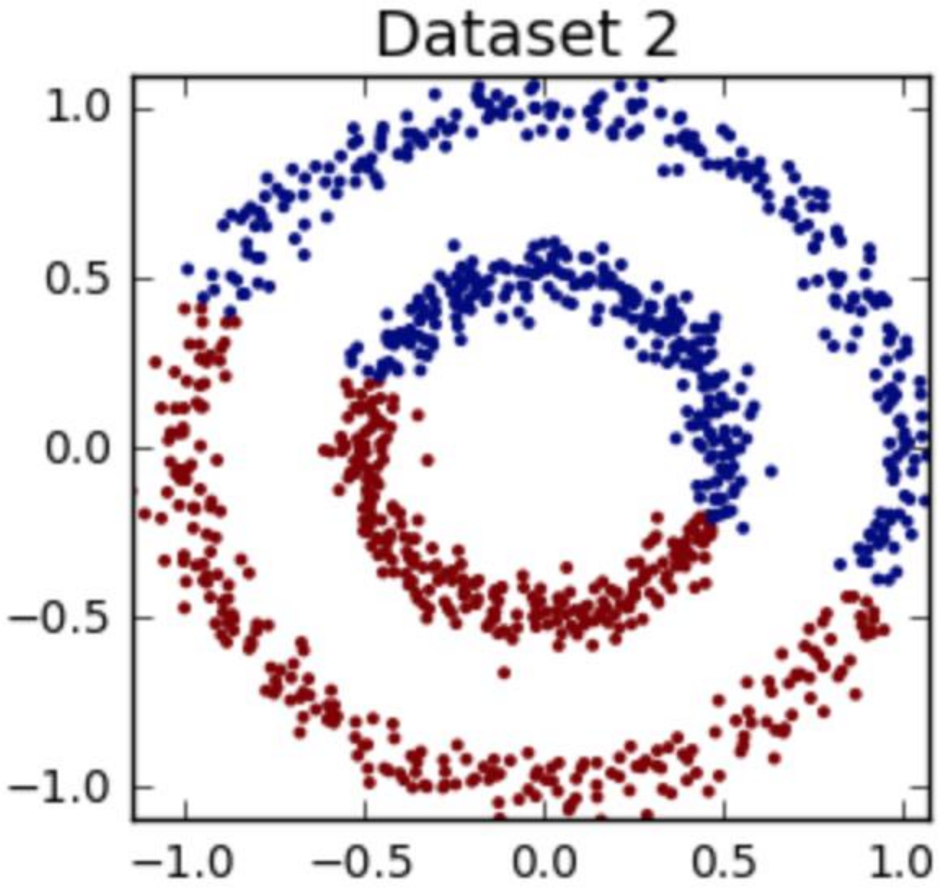
- All OK, clusters are satisfactory!

Dataset 2

K-means identifies globular clusters (essentially spherical clusters)

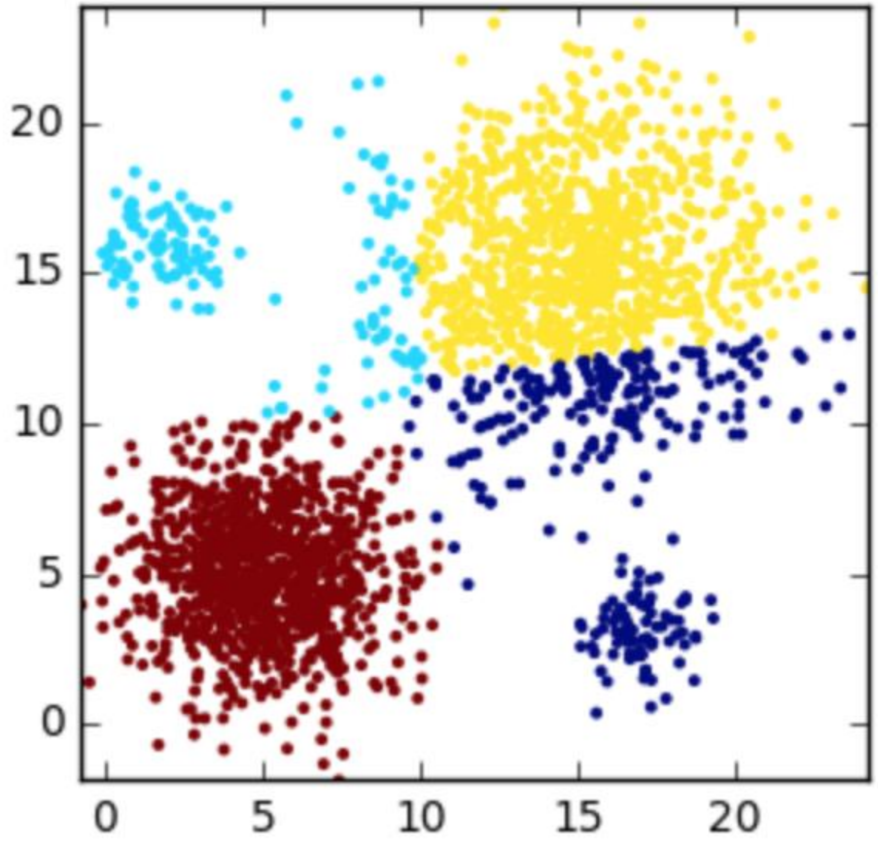
If this assumption does not hold K-means may perform poorly

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



- With this dataset 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 (see DBSCAN)

Bottom line

- Understanding the assumptions behind a method & its limits is essential: it does not just tell 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:

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

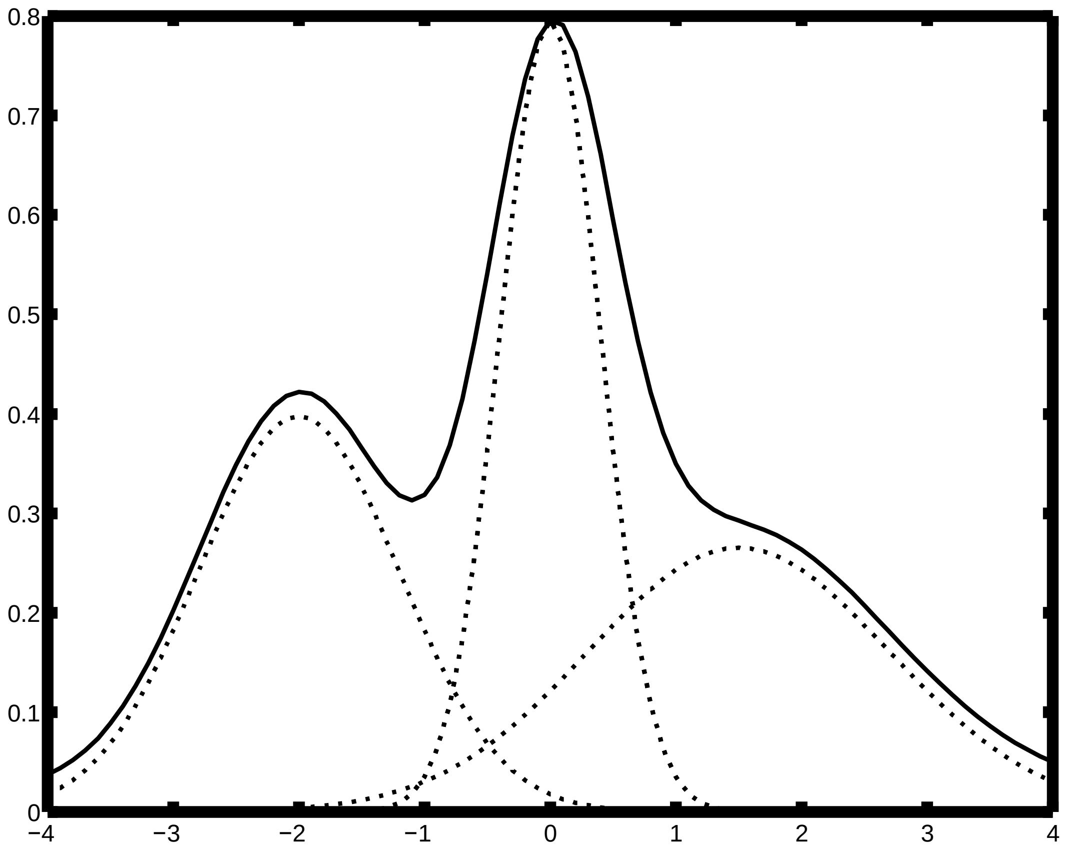
- With:

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

$\boldsymbol{\mu}_k$ mean $\quad$ $\boldsymbol{\Sigma}_k$ m x m 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
  - Allows for powerful optimizations (e.g., Expectation Maximization)
  - Here, **z** is used to model the K clusters the data points belong to

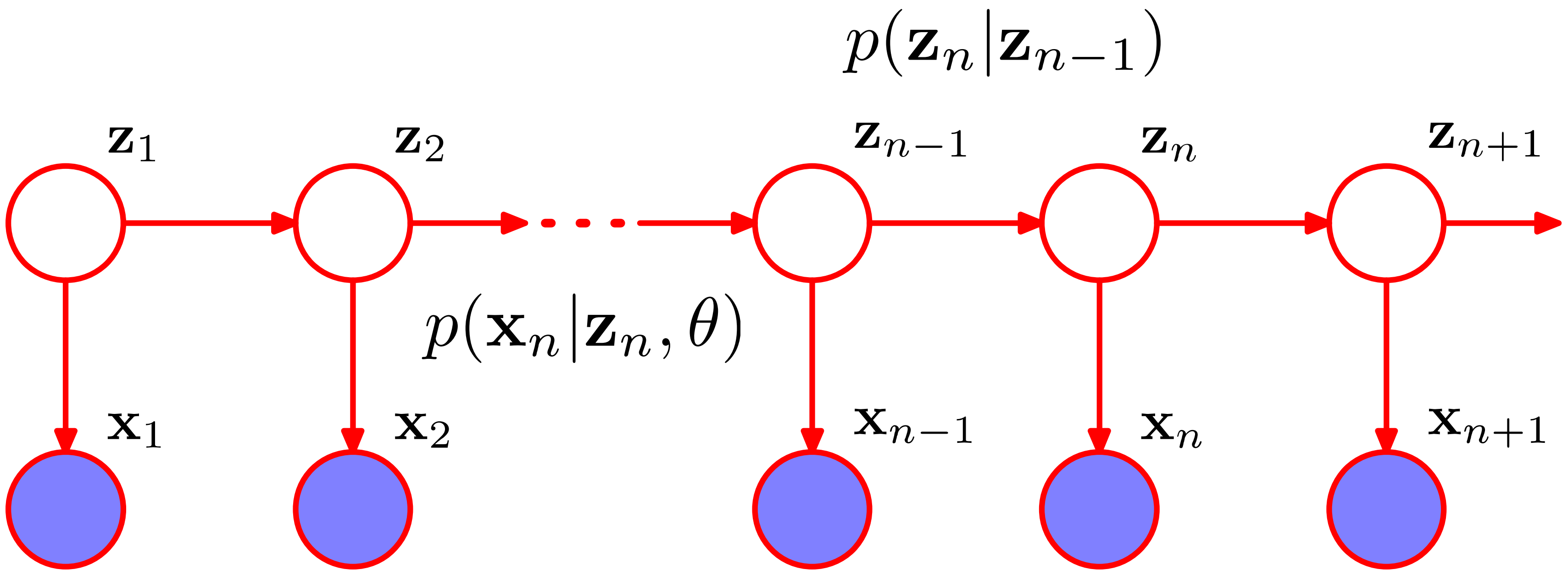
Latent variables z

- 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

Example: Hidden Markov Model

- Graphical model, sequential (tracks memory in time)
- States: white filled circles ($\mathbf{z}_n$, latent variables, not accessible)
- Emissions or observations: blue filled circles (pdf)
- Arrows: dependencies (transition probability matrix, Markov)



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:

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

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

Latent variables **z**

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

marginalizing over $\mathbf{z}_n$

- 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 $\mathbf{z}_n$)
- Although now it may be unclear, this is very important, as it allows to operate using Expectation Maximization (Bayesian Machine Learning) on latent variables
- Powerful tool for automated model optimization

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

- Using Bayes result for conditional probability

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

$$\gamma(z_k) \triangleq p(z_k = 1|\boldsymbol{x}) = \frac{p(z_k = 1, \boldsymbol{x})}{p(\boldsymbol{x})}$$

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

- 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 data Log likelihood

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

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

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

Maximizing the log likelihood

- We want to compute: $\dfrac{\partial \ln p(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} = 0$

- It holds:

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

$$= \sum_{n=1}^{N} \frac{\left(\sum_{k=1}^{K} \pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right)'}{\sum_{k=1}^{K} \pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)} \longleftarrow \boxed{\frac{d \ln f(x)}{dx} = \frac{f(x)'}{f(x)}}$$

$$= \sum_{n=1}^{N} \frac{\pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)} \left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)'$$

Maximizing the log likelihood

- It follows (Eq. (86) of matrix cookbook [cookbook-2012])

$$\frac{\partial(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k))}{\partial\boldsymbol{\mu}_k} = -\frac{1}{2}\left(\frac{\partial((\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k))}{\partial\boldsymbol{\mu}_k}\right)$$

$$= -\frac{1}{2}\left(-2\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right) = \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)$$

- Hence, we write

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

Posterior probability

Maximizing the log likelihood

- Hence, setting: $$\frac{\partial \ln p(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} = 0$$
- Leads to: $$\sum_{n=1}^{N} \gamma(z_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

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

- Which finally leads to

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

Discussion of result

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

- We can interpret N_k as the effective number of points assigned to cluster k
- The mean vector $\boldsymbol{\mu}_k$ 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 $\mathbf{x}_n$

Maximizing the log likelihood

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

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

- 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

Maximizing the likelihood

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

$$= \sum_{n=1}^{N} \frac{\pi_k}{\sum_{k=1}^{K} \pi_k G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)} \boxed{\frac{\partial G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\Sigma}_k}}$$

- We consider

$$G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{m/2}} \cdot \underbrace{\frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}}}_{f(x) = 1/|\boldsymbol{\Sigma}_k|^{1/2}} \underbrace{\exp\left\{ -\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \right\}}_{g(x)}$$

Maximizing the likelihood

$$\frac{\partial G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\Sigma}_k} = ?$$

$$G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{m/2}} \cdot \underbrace{\frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}}}_{f(x)} \underbrace{\exp\left\{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right\}}_{g(x)}$$

- We use

$$\frac{d(f(x)g(x))}{dx} = f(x)'g(x) + f(x)g(x)' \qquad \frac{d\ln f(x)}{dx} = \frac{f(x)'}{f(x)}$$

$$\frac{d(1/f(x))}{dx} = -\frac{f(x)'}{f(x)^2} = -\frac{1}{f(x)} \cdot \frac{\partial \ln f(x)}{dx}$$

Maximizing the likelihood

$$\frac{\partial G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\Sigma}_k} = -\frac{1}{(2\pi)^{m/2}}\cdot\frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}}\frac{\partial \ln|\boldsymbol{\Sigma}_k|^{1/2}}{\partial \boldsymbol{\Sigma}_k}\exp\{\dots\}$$

$$+\frac{1}{(2\pi)^{m/2}}\cdot\frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}}\exp\{\dots\}\frac{\partial(-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu}_k))}{\partial \boldsymbol{\Sigma}_k}$$

$$=\boxed{G(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}\left(-\frac{1}{2}\boldsymbol{\Sigma}_k^{-1}+\frac{1}{2}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu}_k)(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}\right)$$

Using

- derivative rules in previous slide
- Eqs. (57) and (61) of the matrix cookbook [cookbook-2012]
- Covariance is symmetric (does not change swapping X and Y)

$$\sigma(X,Y)=E[(X-E[X])(Y-E[Y])]$$

- It follows that, covariance matrix is also symmetric, and (see Appendix 1)

$$(\boldsymbol{\Sigma}_k^{-1})^T=(\boldsymbol{\Sigma}_k^T)^{-1}=\boldsymbol{\Sigma}_k^{-1}$$

Maximizing the likelihood

From previous results, we obtain

$$\frac{\partial \ln p(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k} =$$

$$\sum_{n=1}^{N} \gamma(z_{nk}) \left(-\frac{1}{2}\boldsymbol{\Sigma}_k^{-1} + \frac{1}{2}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} \right) = 0$$

Left multiplying by $2\boldsymbol{\Sigma}_k$ and rearranging, leads to

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

Maximizing the log likelihood

- 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

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

Maximizing the log likelihood

- 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 replace λ and rearranging gives:

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

Maximizing the log likelihood

- Multiplying both sides by π_k and summing over k:

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

$$\sum_{n=1}^{N} 1 + \lambda = 0 \Rightarrow \lambda = -N$$

Maximizing the log likelihood

- Using this λ to replace N and rearranging:

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

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

multiply both sides by π_k

$$\sum_{n=1}^{N} \gamma(z_{nk}) - \pi_k N = 0 \Rightarrow \pi_k = \frac{\sum_{n=1}^{N}\gamma(z_{nk})}{N} = \frac{N_k}{N}$$

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 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 (mixing coefficients, mean vectors and variances):

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

EM for GM (2/2)

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

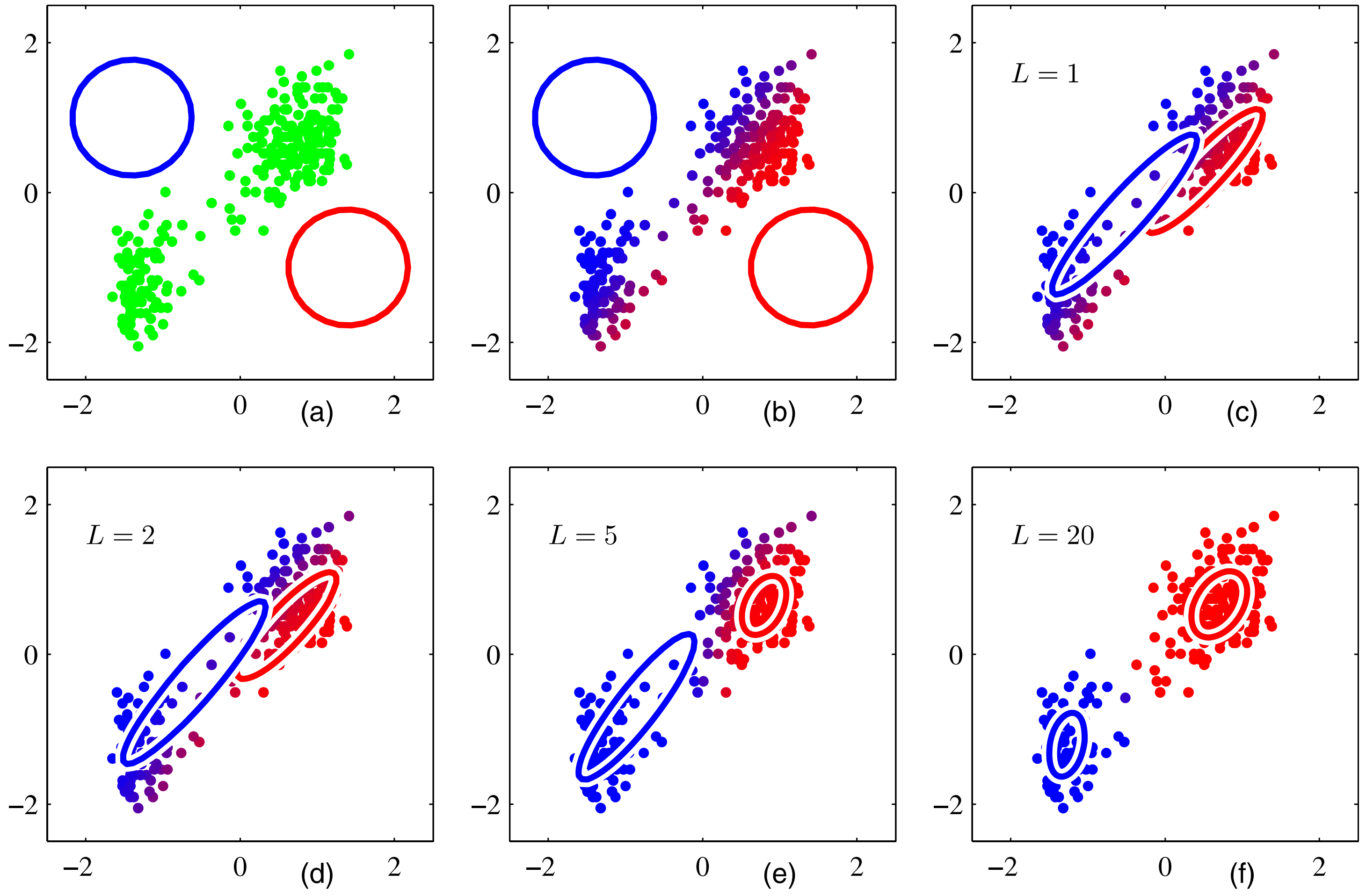
$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\boldsymbol{x}_n$$

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

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

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.

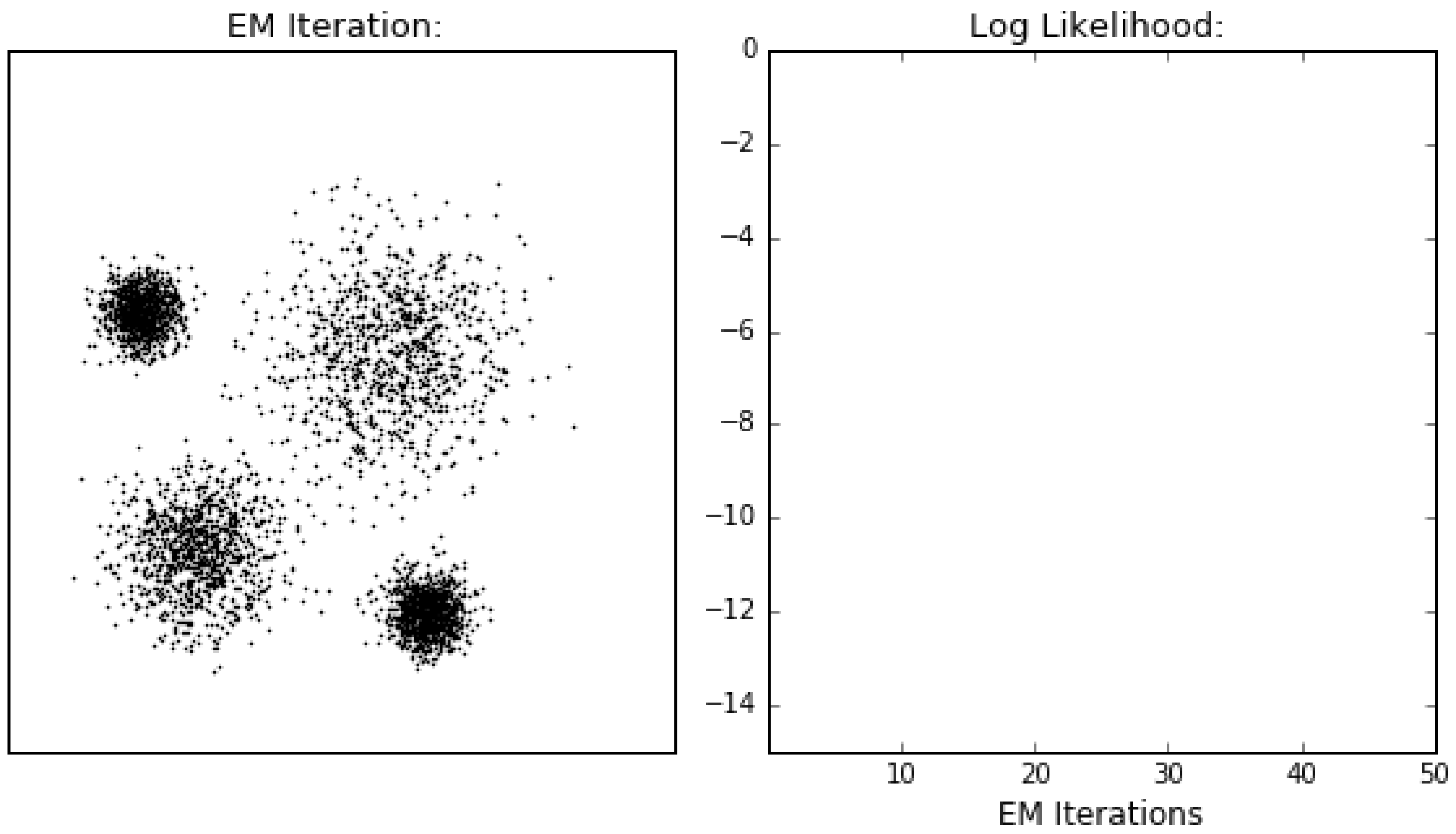
Example



Observations

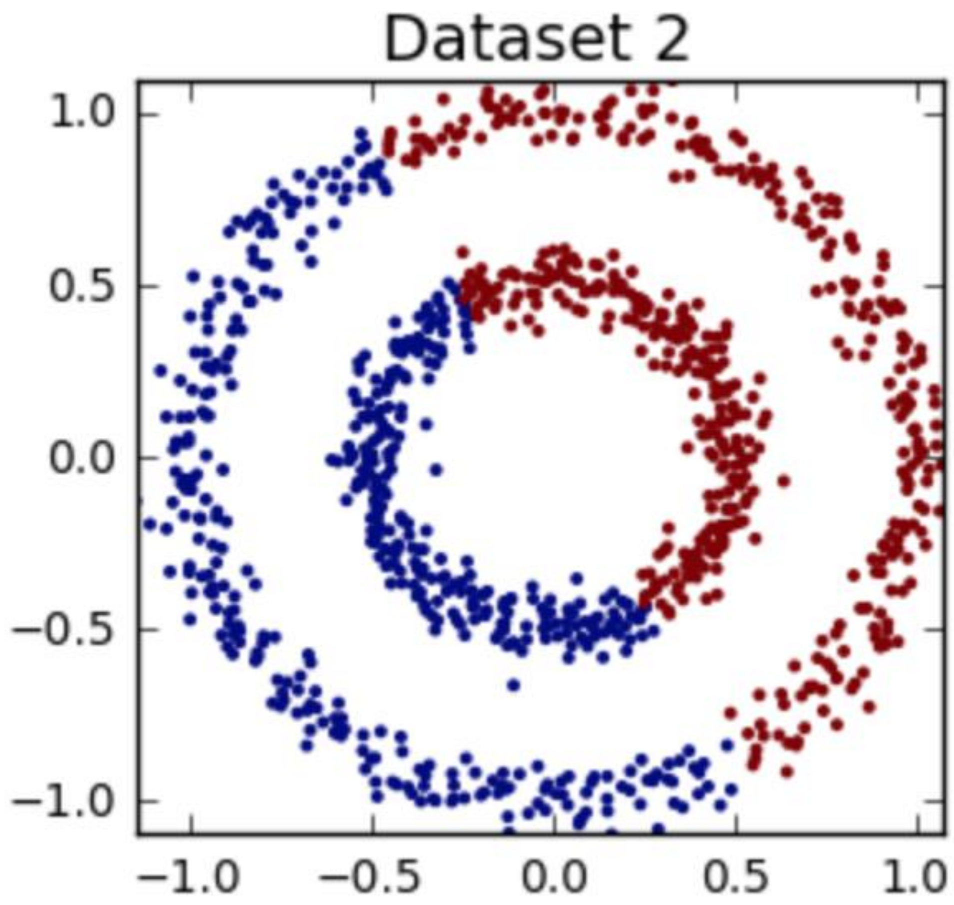
- EM can take many more iterations than K-means to converge
  - Higher computational complexity
- It is common to use K-means to initialize the GM model pars
  - The GM model is then adapted through EM
  - The GM covariance matrices can be conveniently initialized by the sample covariance matrices of 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

Dataset 1



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

A problematic dataset

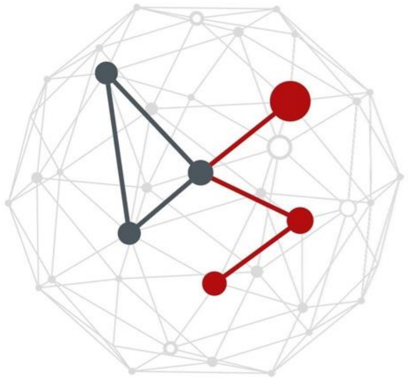


- The data distribution cannot be accurately modeled by a GMM → GM-based soft-clustering fails

MIME
Master's degree ICT Internet Multimedia Engineering

X-MEANS

How to pick the right number of clusters



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO
MATEMATICA

UNIVERSITAS STUDII PADUANI MCCXXII

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Right number of clusters?

- Up to now
  - The number of clusters K was *supplied by the user*

- Would it be possible to learn K from data?
  - A very popular approach is proposed in [Peleg-00]
  - X-means
    - Uses a divisive (hierarchical) approach
    - Intelligently exploiting 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. [3903 citations, Oct 2024]

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 outputted by the algorithm

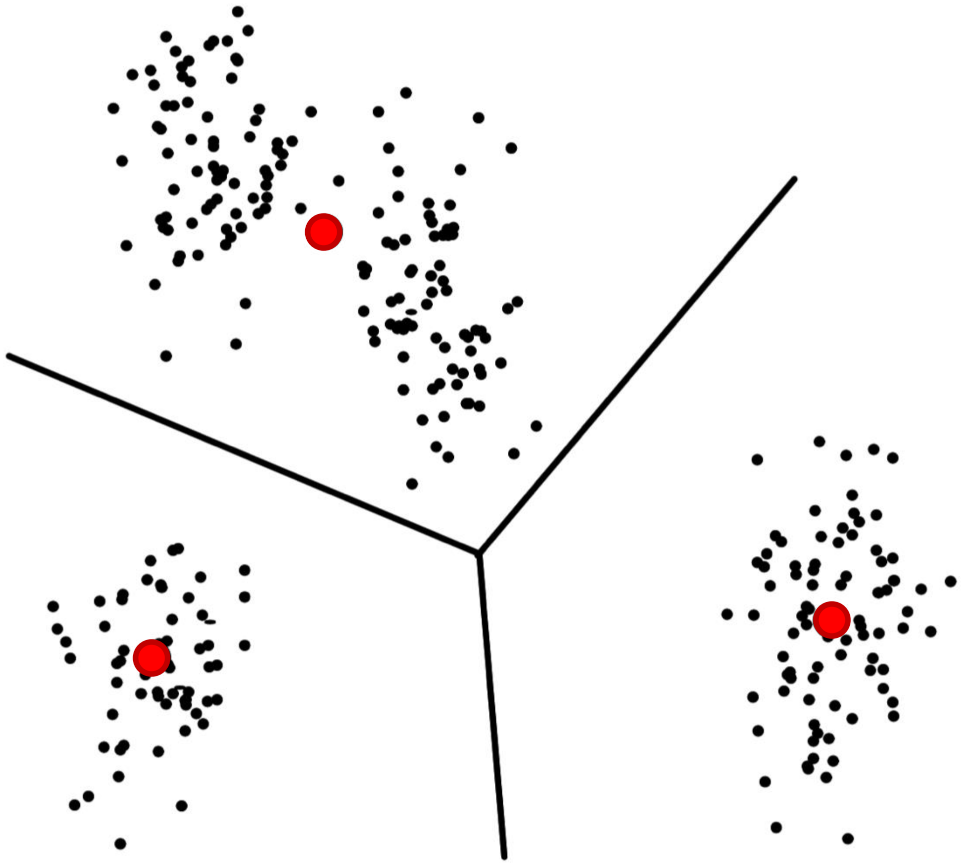
X-means in a nutshell (2/2)

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 into two
3. **If** all configurations are explored, **stop** and report the best model found, **Else**, Goto 1

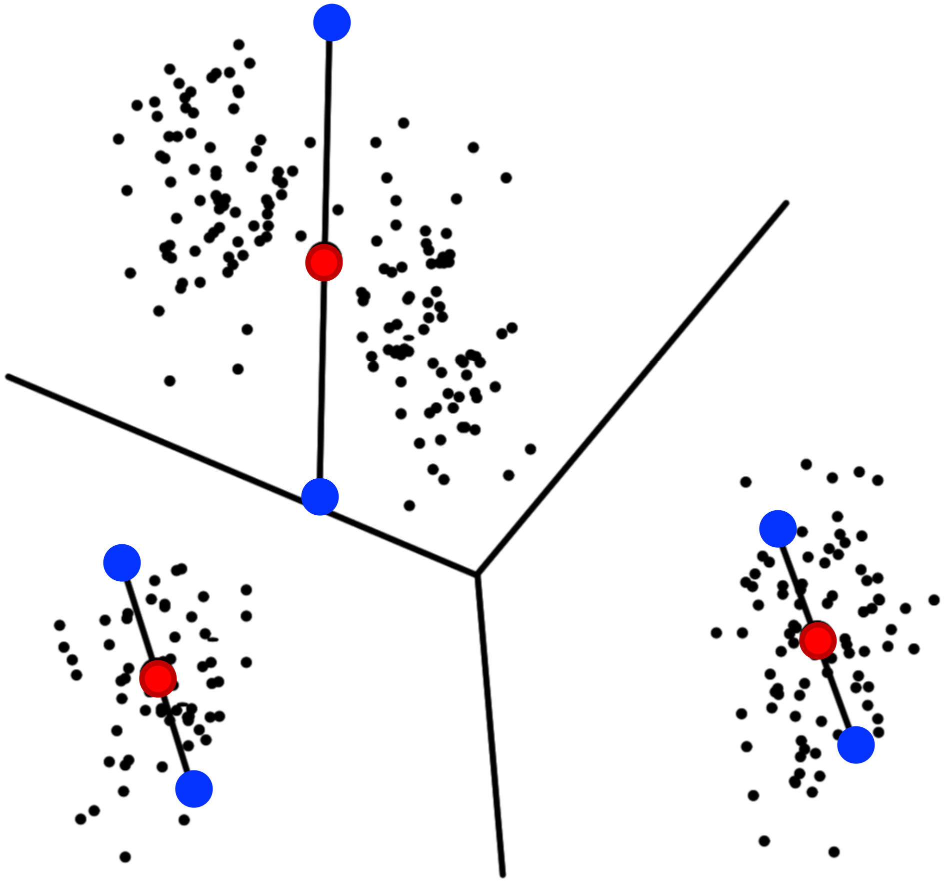
X-means by example

- Current model: stable current solution with 3 centroids



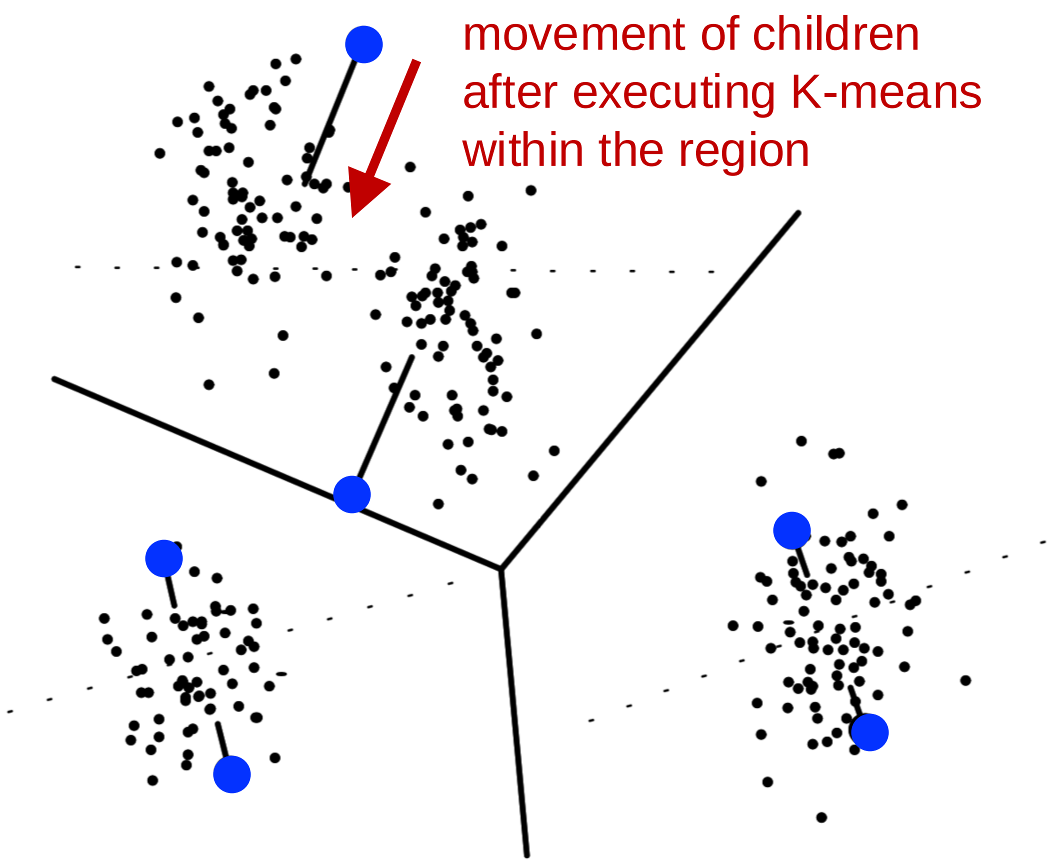
X-means by example

- 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, following a random direction



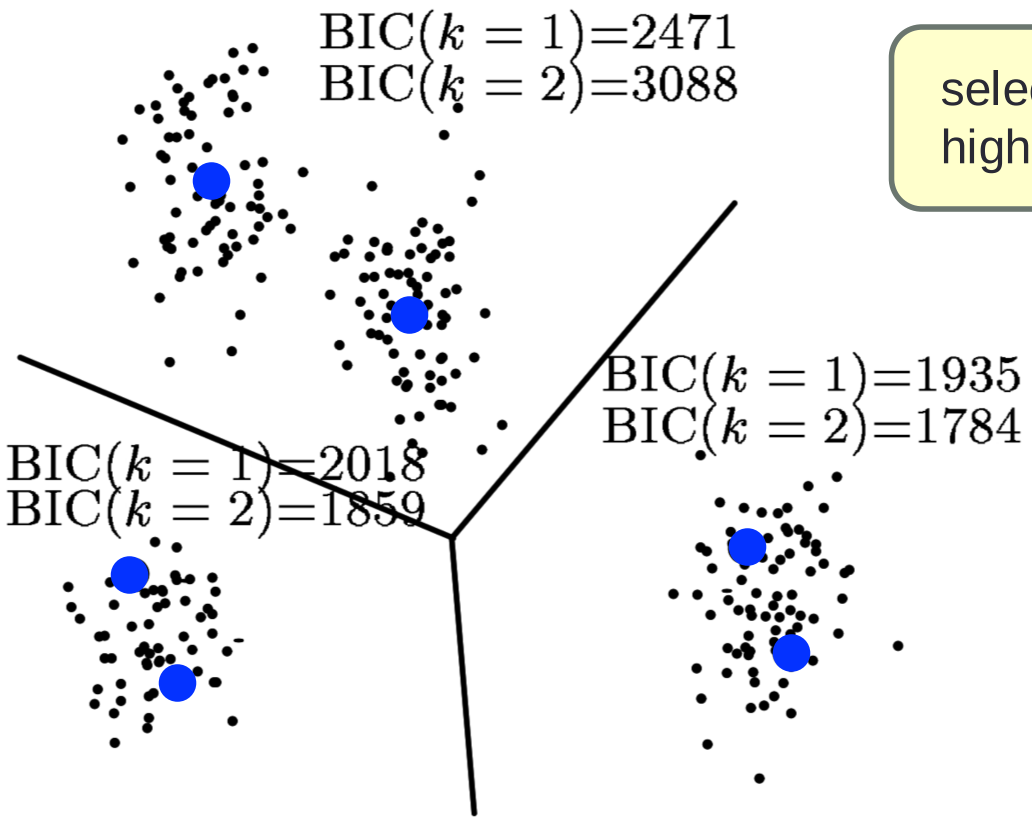
X-means by example

- 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


movement of children after executing K-means within the region

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?*"



select model with highest BIC

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 2 sub-clusters 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:

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

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

For each hypothesis (M_1, M_2)

- The data log-likelihood "$L(M_i)$" for the two models i=1,2 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, 2 Gaussian components, no. of points N_1 and N_2 ($N=N_1+N_2$)
  - 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 in a sub-region

- If $\mathcal{P}$ represents the set of data points in the sub-region
- Models M_j 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. [1673 citations, Oct 2024]

Scoring the models (M_1, M_2)

- If $\mathcal{P}$ represents the set of data points in the sub-region
- Models M_j 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 largely depends on the underlying data structure

Scoring the models (M_1, M_2)

- If $\mathcal{P}$ represents the set of data points in the sub-region
- Models M_j 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

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

number of points in the sub-region

- p_j: number of model parameters

$$p_j = \boxed{K_j - 1} + \boxed{K_j m} + \boxed{C_j}$$

class probabilities

centroid coordinates

number of parameters to estimate in covariance matrices

Shape of covariances

- By whitening the input data (e.g., PCA)
- We can obtain Gaussians with elliptical shape

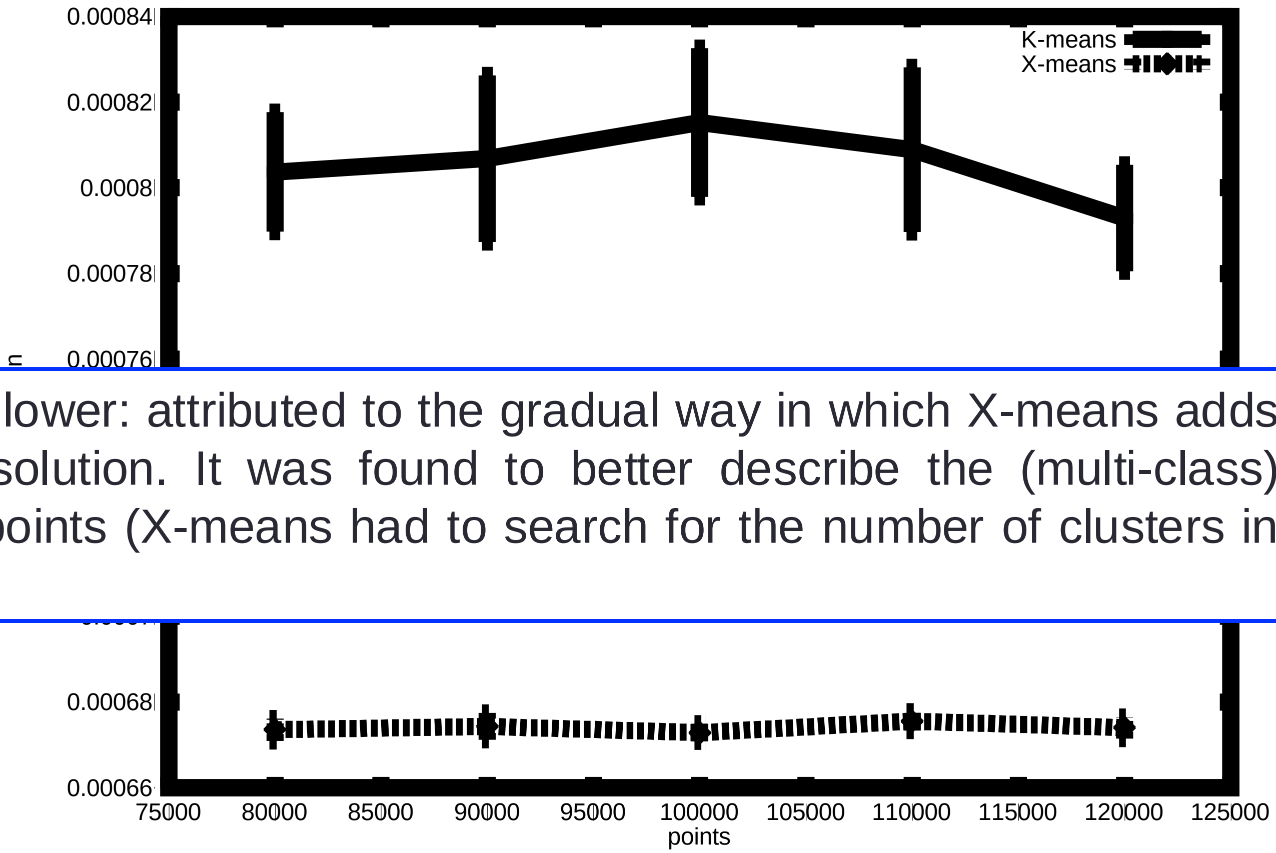
$$\mathbf{\Sigma}_j = \begin{bmatrix} \sigma^2_{j,1} & 0 & \dots & 0 \\ 0 & \sigma^2_{j,2} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \sigma^2_{j,m} \end{bmatrix}$$

- If $\sigma^2_{i,j} = \sigma^2 \ , \ \forall j$: spherical Gaussians
- Number of free parameters C_j:
  - C_j = m (M_1), 2m (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 = m^2 (M_1), $2m^2$ (M_2): full covariance matrices

Example results – distortion J

Average distortion K-means uses the *exact* number of classes K. Results are averaged over 30 runs for 3D data and 250 classes

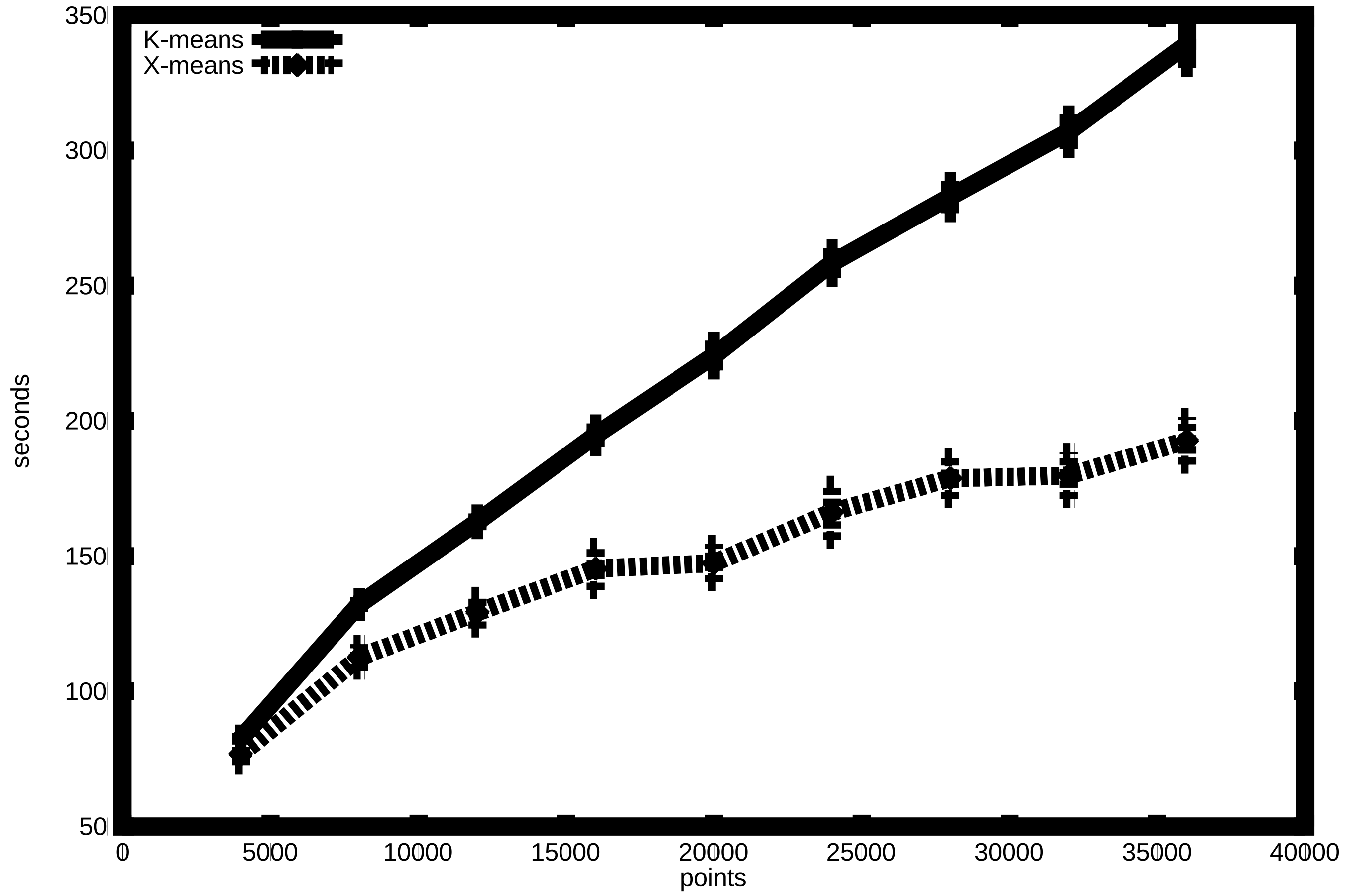
Distortion = average squared distance between points and centroids



Distortion of X-mean is lower: attributed to the gradual way in which X-means adds new centroids to the solution. It was found to better describe the (multi-class) specific distribution of points (X-means had to search for the number of clusters in the range {2, …, K})

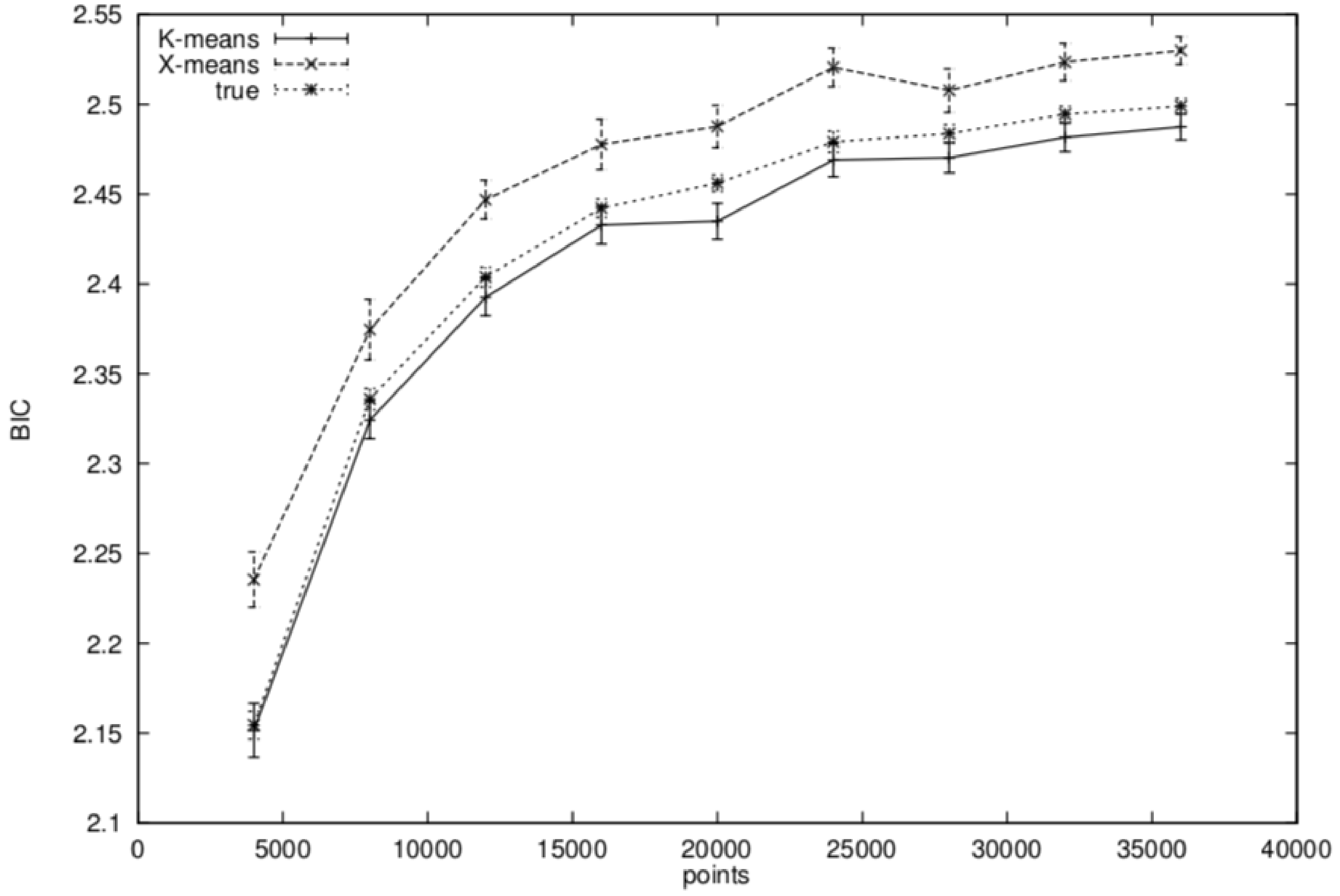
Example results – time complexity

Time complexity X-means scales much better than iterated K-means in terms of execution time. Results are averaged over 30 runs for 3D data and 250 classes



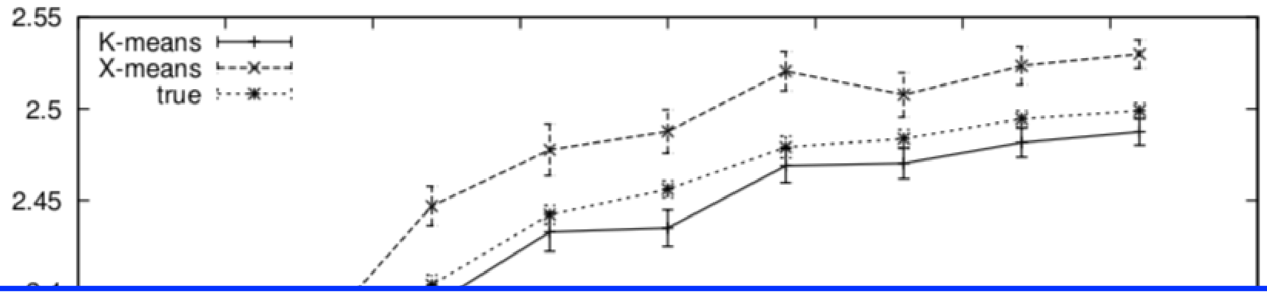
Example results – average BIC

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



Example results – average BIC

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



True – shows the BIC for the actual distribution (true centroids)

K-means variant – searches for the best K using the BIC criterion on the final solution (permissible range for no. of centroids [2, …, 2K])

X-means outperforms both: due to random deviations in the data that cause it to be better modeled by fewer classes than they actually are. For instance, two (or mode) centroids (chosen at random) may fall very close to one another, and, in turn, the corresponding data clouds may be "better modeled" by a single class (that means, fewer parameters and correspondingly higher BIC)

points

References

Matrix algebra:

[cookbook-2012] Kaare Brandt Petersen, Michael Syskind Pedersen, “The Matrix Cookbook,” Tech. Rep, Nov. 15, 2012. [3128 citations, Oct 2023]

Reference book:

[Bishop-2006] Christopher M. Bishop, “Pattern Recognition and Machine Learning,” Springer 2006.

Papers:

[Peleg-2000] 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. [3681 citations, Oct 2023]

[Kass-1995] 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. [1600 citations, Oct 2023]

Appendix 1

- Prove that: $(\boldsymbol{A}^{-1})^T = (\boldsymbol{A}^T)^{-1}$

Proof.

We use the standard identity for matrices: $\boldsymbol{A}^T\boldsymbol{B}^T = (\boldsymbol{B}\boldsymbol{A})^T$

It holds:

$$\boldsymbol{A}^T(\boldsymbol{A}^{-1})^T = (\boldsymbol{A}^{-1}\boldsymbol{A})^T = \boldsymbol{I}^T = \boldsymbol{I}$$
$$(\boldsymbol{A}^{-1})^T\boldsymbol{A}^T = (\boldsymbol{A}\boldsymbol{A}^{-1})^T = \boldsymbol{I}^T = \boldsymbol{I}$$

This proves that the inverse of $\boldsymbol{A}^{-1}$ is $(\boldsymbol{A}^T)^{-1}$

QED

Appendix 2

Setup

- Regression problem
- Dataset $\mathcal{D} = \{\boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$
- Noisy measurements

$$f_{\text{meas}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x}) + \epsilon$$

- Gaussian, zero mean and i.i.d. noise

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Question: find the most probable model (best fit for the data)

$$\boldsymbol{\theta} = \operatorname*{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{D})$$

Appendix 2

The pdf that a point (**x**,y) belongs to the dataset is

$$P((\boldsymbol{x},y)\in\mathcal{D}|\boldsymbol{\theta}) = P(f_{\boldsymbol{\theta}}(\boldsymbol{x})+\epsilon = y|\epsilon\sim\mathcal{N}(0,\sigma^2)) =$$

$$=\mathcal{N}(y-f_{\boldsymbol{\theta}}(\boldsymbol{x}),\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-f_{\boldsymbol{\theta}}(\boldsymbol{x}))^2}{2\sigma^2}\right)$$

Applying the logarithm

$$\log P((\boldsymbol{x},y)\in\mathcal{D}|\boldsymbol{\theta}) = -\frac{(y-f_{\boldsymbol{\theta}}(\boldsymbol{x}))^2}{2\sigma^2} + \log\left(\overset{\text{constant}}{\frac{1}{\sqrt{2\pi\sigma^2}}}\right)$$

The pdf extended to the entire dataset is

$$\log P(\mathcal{D}|\boldsymbol{\theta}) = -\sum_{(\boldsymbol{x}_i,y_i)\in\mathcal{D}}\frac{1}{2\sigma^2}(y_i-f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 + \text{const}$$

Appendix 2

Applying conditional probability Bayes rule,we get

$$P(\boldsymbol{\theta}|\mathcal{D}) \propto P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})$$

And applying the logarithm,

$$\log P(\boldsymbol{\theta}|\mathcal{D}) = \log P(\mathcal{D}|\boldsymbol{\theta}) + \overset{\text{prior prob}}{\log P(\boldsymbol{\theta})} + \text{const}$$

From the previous slide we can extend the first term

$$\log P(\boldsymbol{\theta}|\mathcal{D}) = -\frac{1}{2\sigma^2} \sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 + P(\boldsymbol{\theta}) + \text{const}$$

Appendix 2

Final result

$$\log P(\boldsymbol{\theta}|\mathcal{D}) = -\frac{1}{2\sigma^2} \overbrace{\sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2}^{\text{MSE}} + \underset{\text{prior. on the pars}}{P(\boldsymbol{\theta})} + \text{const}$$

- Hence, maximizing the likelihood amounts to minimizing the Mean Square Error (MSE, i.e., norm-2)
- In other words, minimizing the MSE maximizes the likelihood of the parameters, we have found the maximum likelihood estimator
- NOTE: setting a prior acts as a regularizer on the pars

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log P(\boldsymbol{\theta}|\mathcal{D})$$

Appendix 2

What if the noise has a Laplace distribution? (still zero mean)

- In this case,

$$P((\boldsymbol{x}, y) \in \mathcal{D} | \boldsymbol{\theta}) = P(f_{\boldsymbol{\theta}}(\boldsymbol{x}) + \epsilon = y | \epsilon \sim \mathcal{L}(0, b)) =$$

$$= \mathcal{L}(y - f_{\boldsymbol{\theta}}(\boldsymbol{x}), b) = \frac{1}{\sqrt{2b}} \exp\left(-\frac{|y - f_{\boldsymbol{\theta}}(\boldsymbol{x})|}{b}\right)$$

- Going through the same steps, we obtain

$$\log P(\boldsymbol{\theta} | \mathcal{D}) = -\frac{1}{b} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}} |y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)| + P(\boldsymbol{\theta}) + \text{const}$$

- In this case, the L1 norm would be optimal

Master's degree ICT Internet Multimedia Engineering

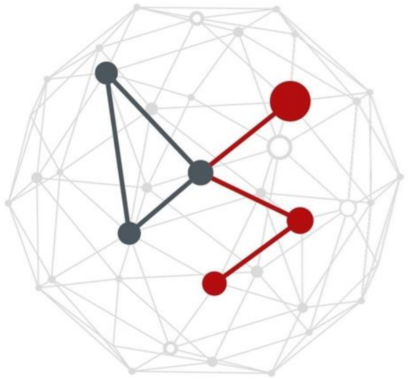
INTRODUCTION TO CLUSTERING

Michele Rossi

michele.rossi@unipd.it

Dept. of Information Engineering

University of Padova, IT



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

UNIVERSITÀ DEGLI STUDI DI PADOVA