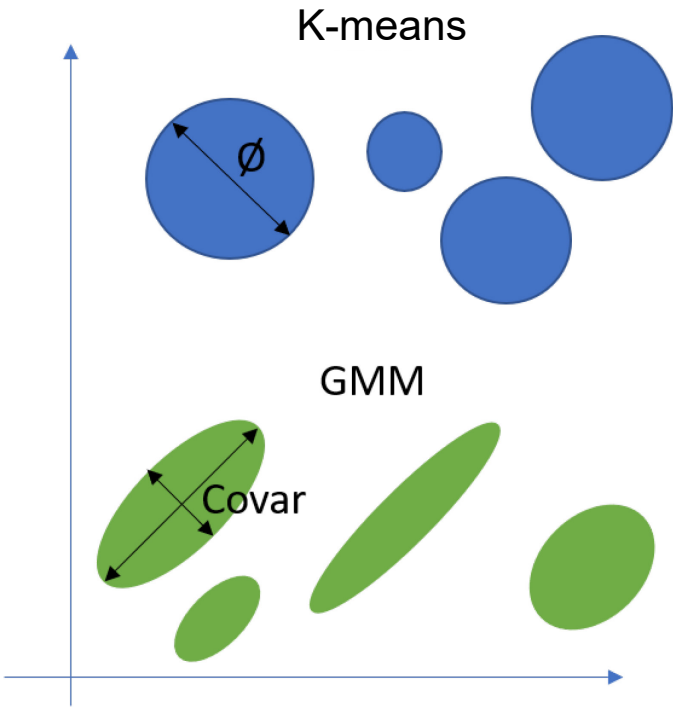
NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz



MSE MachLe
GMM & EM

Christoph Würsch
Institute for Computational Enineering ICE
Interstaatliche Hochschule für Technik Buchs, FHO

Gaussian Mixture Models and the EM algorithm

Agenda

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- **K-means** revisited
- **Gaussian Mixture Models (GMM)**
- **Expectation Maximization (EM)**
- Summary

© NTB, christoph.wuersch@ntb.ch

K-means revisited

- best-known clustering algorithm
- Assume that K is known. The **optimization** that leads to the clusters is the following:

$$\min_{\mathbf{z},\boldsymbol{\mu}} \mathcal{L}(\mathbf{z},\boldsymbol{\mu}) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \left\|\mathbf{x}_n - \boldsymbol{\mu}_k\right\|^2$$

K-means cost function

$$\boldsymbol{\mu}_k \in \mathbb{R}^D, \quad z_{nk} \in \{0,1\}, \quad \sum_{k=1}^{K} z_{nk} = 1$$

- Where:

$$\mathbf{z}_n = \left[z_{n1}, z_{n2}, \ldots, z_{nK}\right]^T$$

$$\mathbf{z} = \left[\mathbf{z}_1, \mathbf{z}_1, \ldots, \mathbf{z}_N\right]^T$$

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N\right]^T$$

← indicator variable:
Assignment of datapoint n to a certain cluster:
membership, (latent variable)

e.g. point n belongs to cluster 3.

$$\mathbf{z}_n = \left[0, 0, 1, 0, \ldots, 0\right]^T$$

K-means algorithm

- For fixed centers $\boldsymbol{\mu}_k$, the cost is minimized if we map each sample to its nearest center, where we measure distance in terms of **Euclidean distance**.
- In words, each sample is assigned **exactly to one center** (cluster) and this is indicated by setting the corresponding indicator variable z_{nk} to 1 and all the other ones $z_{nk'}$ to 0.
- This leads directly to a very intuitive algorithm: initialize $\boldsymbol{\mu}_k \forall\, k$

1. For all n, compute $\boldsymbol{z}_n$ given $\boldsymbol{\mu}$
2. For all k, compute $\boldsymbol{\mu}_k$ given $\boldsymbol{z}$

- taking the derivatives of the cost function w.r.t. μ_k and solving for the cluster centers, we get:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = 0 \Rightarrow \boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} z_{nk} \mathbf{x}_n}{\sum_{n=1}^{N} z_{nk}}$$

K-means as coordinate descent algorithm

- **Convergence** to a local optimum is assured since each step decreases the cost (see Bishop, Exercise 9.1). But note that we are **not guaranteed to reach the globally optimal solution** with this iterative algorithm.
- K-means is a **coordinate descent** algorithm:

$$\min_{\mathbf{z},\boldsymbol{\mu}} \mathcal{L}(\mathbf{z}, \boldsymbol{\mu}) \quad \begin{cases} \mathbf{z}^{(t+1)} = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{z}, \boldsymbol{\mu}^{(t)}) \\ \boldsymbol{\mu}^{(t+1)} = \arg\min_{\boldsymbol{\mu}} \mathcal{L}(\mathbf{z}^{(t+1)}, \boldsymbol{\mu}) \end{cases}$$

© NTB, christoph.wuersch@ntb.ch

Probabilistic K-means

- Assume that, conditioned that a point is associated to cluster k, we consider it a **sample from the D-dimensional Gaussian** with mean $\boldsymbol{\mu}_k$ and covariance matrix $\mathbb{I}$. i.e., the **likelihood** of a sample x given the cluster assignment $\boldsymbol{z}$ and the centers $\boldsymbol{\mu}$ is**:**

$$p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\mu}) = \prod_{k=1}^{K} \left[\mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \mathbb{1}\right)\right]^{z_k}$$

- Then, the likelihood associated for the whole data set is:

$$p(\mathbf{X} \mid \mathbf{z}, \boldsymbol{\mu}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \mathbb{1}\right)\right]^{z_{nk}}$$

- Taking the negative logarithm in order to minmize the negative loglikelihood, we get again the **K-means cost function**:

$$-\log p(\mathbf{X} \mid \mathbf{z}, \boldsymbol{\mu}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\|\mathbf{x}_n - \boldsymbol{\mu}_k\right\|^2$$

Hard and Soft Clustering

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- given: a simple data set consisting of class heights $\boldsymbol{X} = \{x_i\}$ with groups $Z = \{z_1, z_2\}$ separated by gender
- Imaging that we did not have the convenient gender labels (male, female) associated with each data point. How could we estimate the two group means?
- We assume that height data (observed values $\boldsymbol{X}$) are drawn from two independent Gaussian distributions with mean μ_k and variance σ_k^2 $(k = 1,2)$
- Understanding the range the z_j values can take is important: In **K-means**, the two z_j can only take the values of 0 or 1. **This is called hard clustering.**
- In **Gaussian Mixture Models (GMM),** the z_j can take on any value between 0 and 1 . **This is called soft or fuzzy clustering.**

Clustering with Gaussians

- K-means is equivalent to assuming that the data came from K *spherically symmetric* Gaussians. Instead of isotropic covariances $\mathbb{1}$, we now use full covariance matrices $\mathbf{\Sigma}_k$ to model **elliptical clusters.**

$$p(\mathbf{X} \mid \mathbf{z}, \boldsymbol{\mu}, \mathbf{\Sigma}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \mathbf{\Sigma}_k\right)\right]^{z_{nk}}$$

- In K-means, each sample belongs to exacltly one cluster. This is not always a good choice, esp. for points near the boundary.
- By interpreting z_n *as random variable* taking the values $\{1,2,\ldots,K\}$ with a *prior distribution* that follows a multinomial distribution, we can define a fractional assignment (**soft clustering**).

$$p(z_n = k) = \pi_k \qquad \pi_k \geq 0 \,\forall k \qquad \sum_{k=1}^{K} \pi_k = 1$$

$$p(\mathbf{z}) = \prod_{k=1}^{K} (\pi_k)^{z_k}$$

Gaussian Mixture Models (GMM)

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

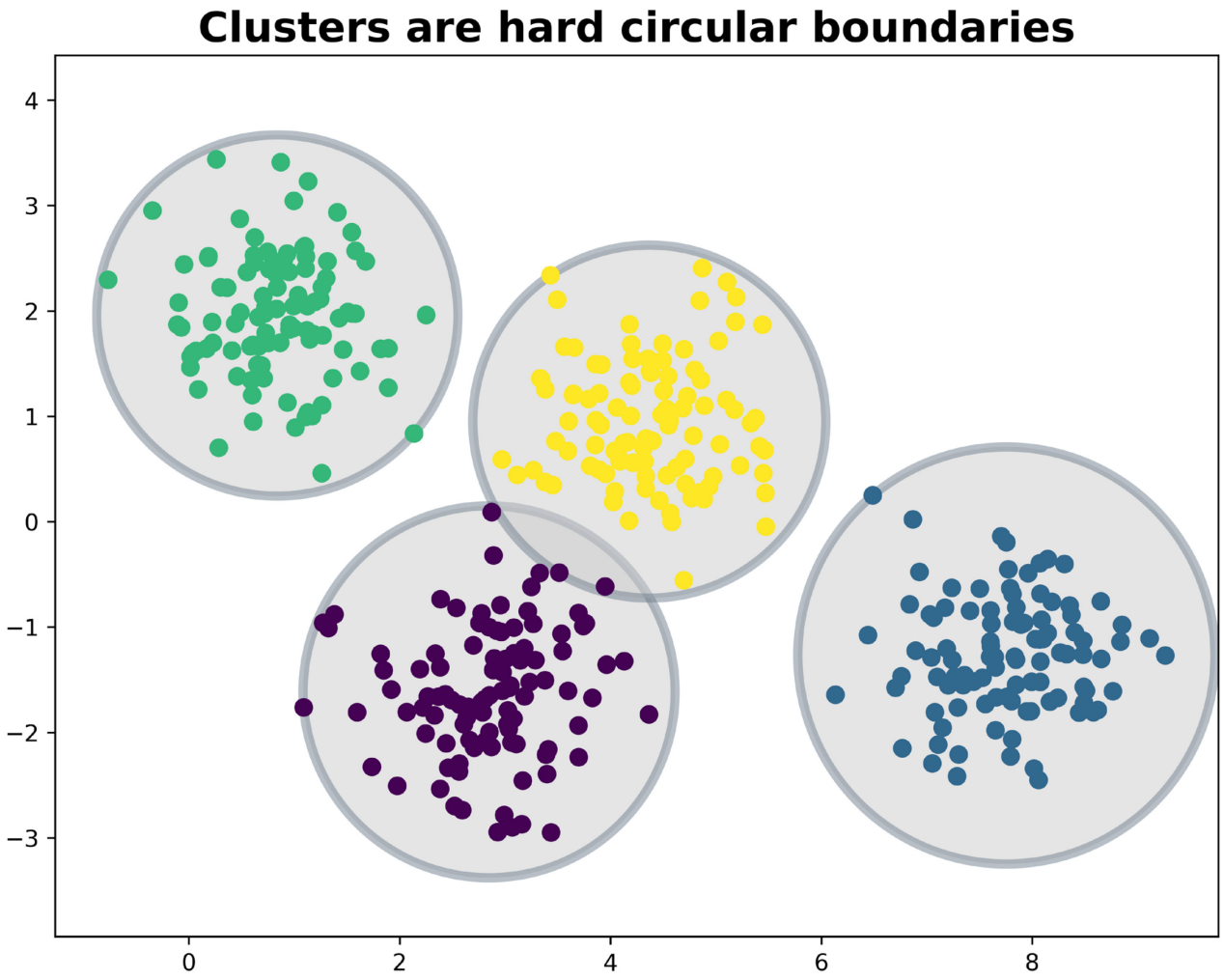
- A **Gaussian mixture model** is a probabilistic model that assumes all the data points are generated from a mixture of a finite number K of Gaussian distributions with unknown parameters (μ_k, Σ_k).
- One can think of mixture models as **generalizing K-means clustering** to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.

Speed:	It is the **fastest algorithm** for learning mixture models
Agnostic:	As this algorithm maximizes only the likelihood, it will not bias the means towards zero, or bias the cluster sizes to have specific structures that might or might not apply.
Singularities:	When one has insufficiently many points per mixture, estimating the covariance matrices becomes difficult, and the algorithm is known to diverge and find solutions with infinite likelihood unless one regularizes the covariances artificially.

See also: https://scikit-learn.org/stable/modules/mixture.html#gmm

K-means for circular clusters

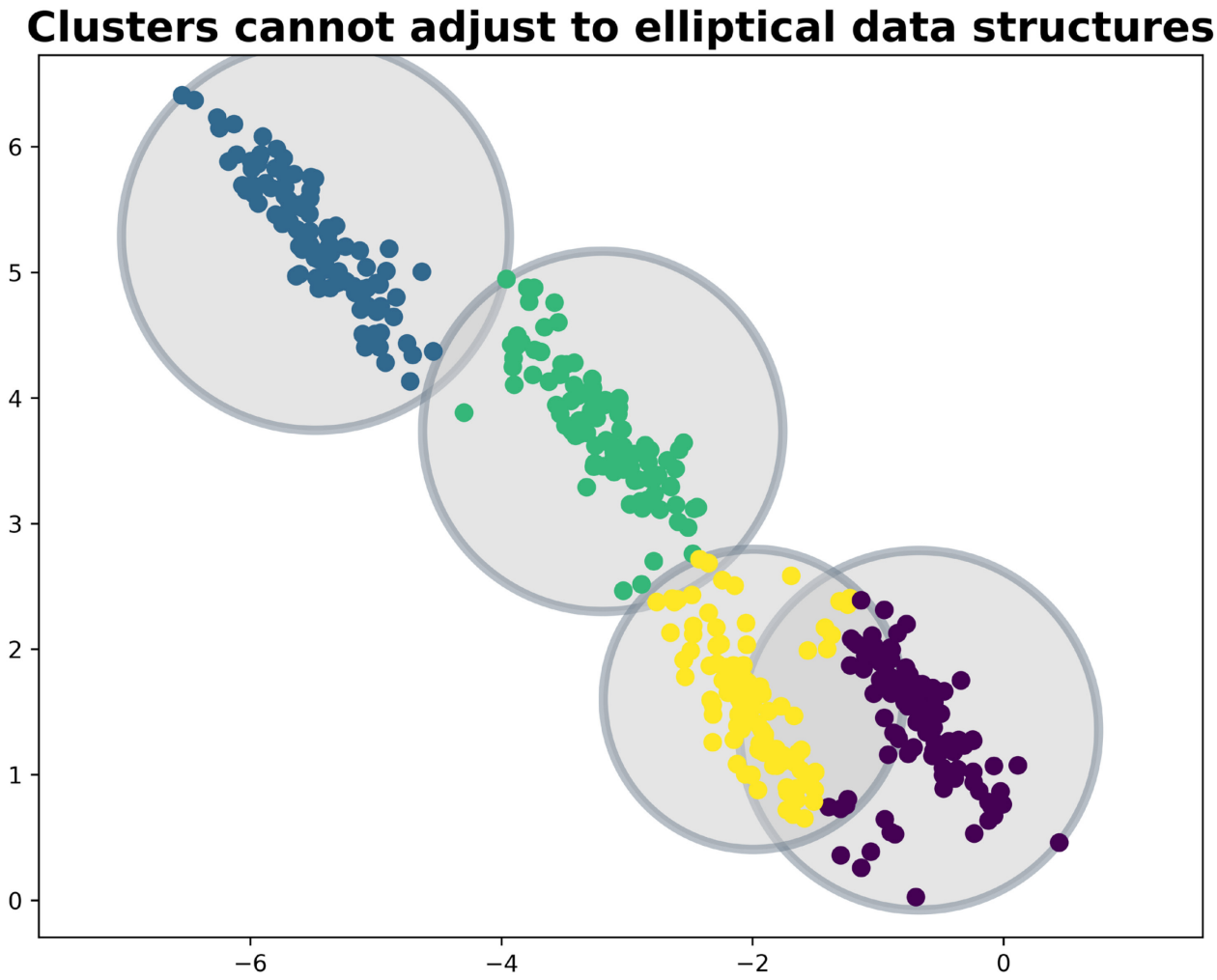
NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz



© NTB, christoph.wuersch@ntb.ch

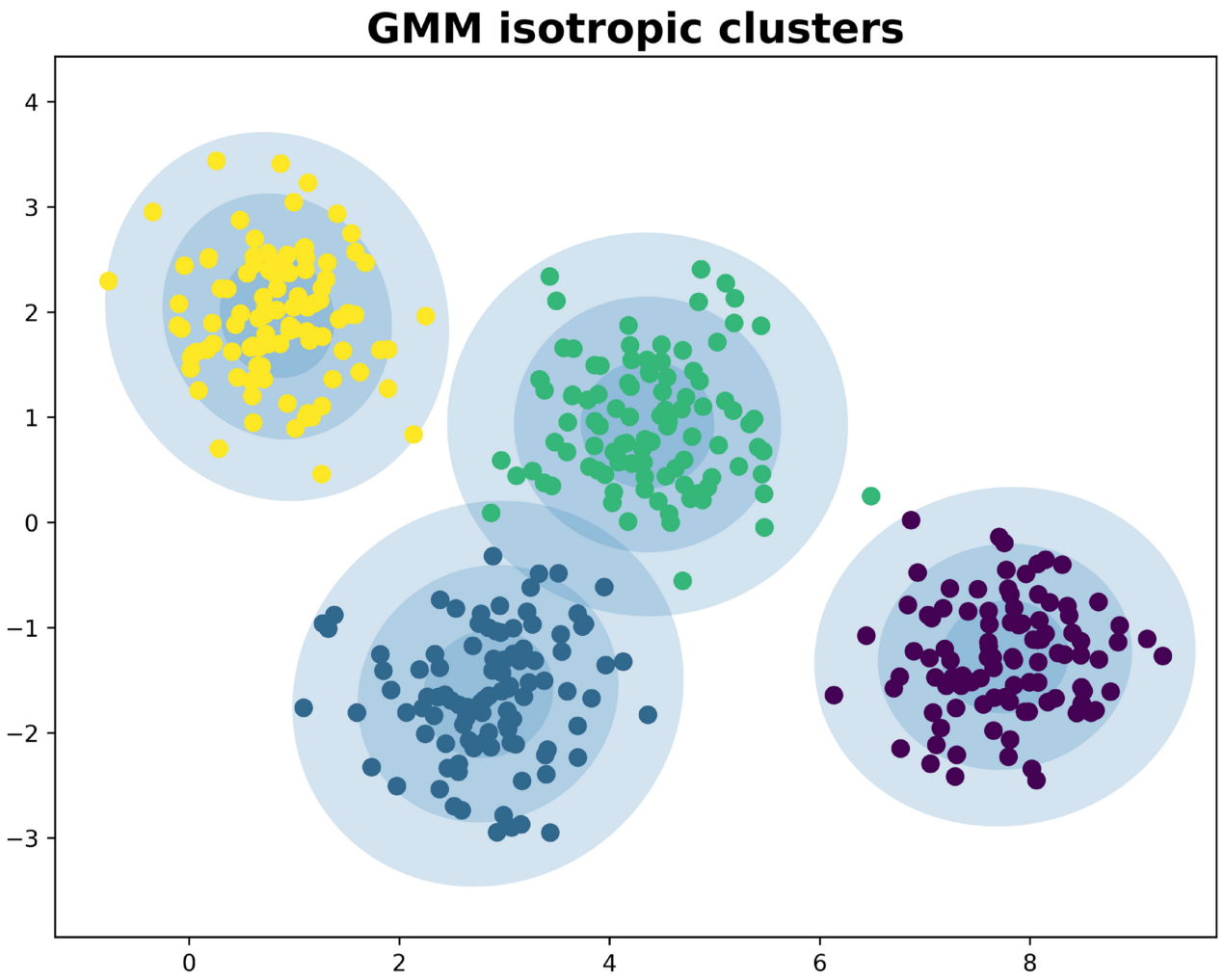
K-means for elliptical clusters

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz



© NTB, christoph.wuersch@ntb.ch

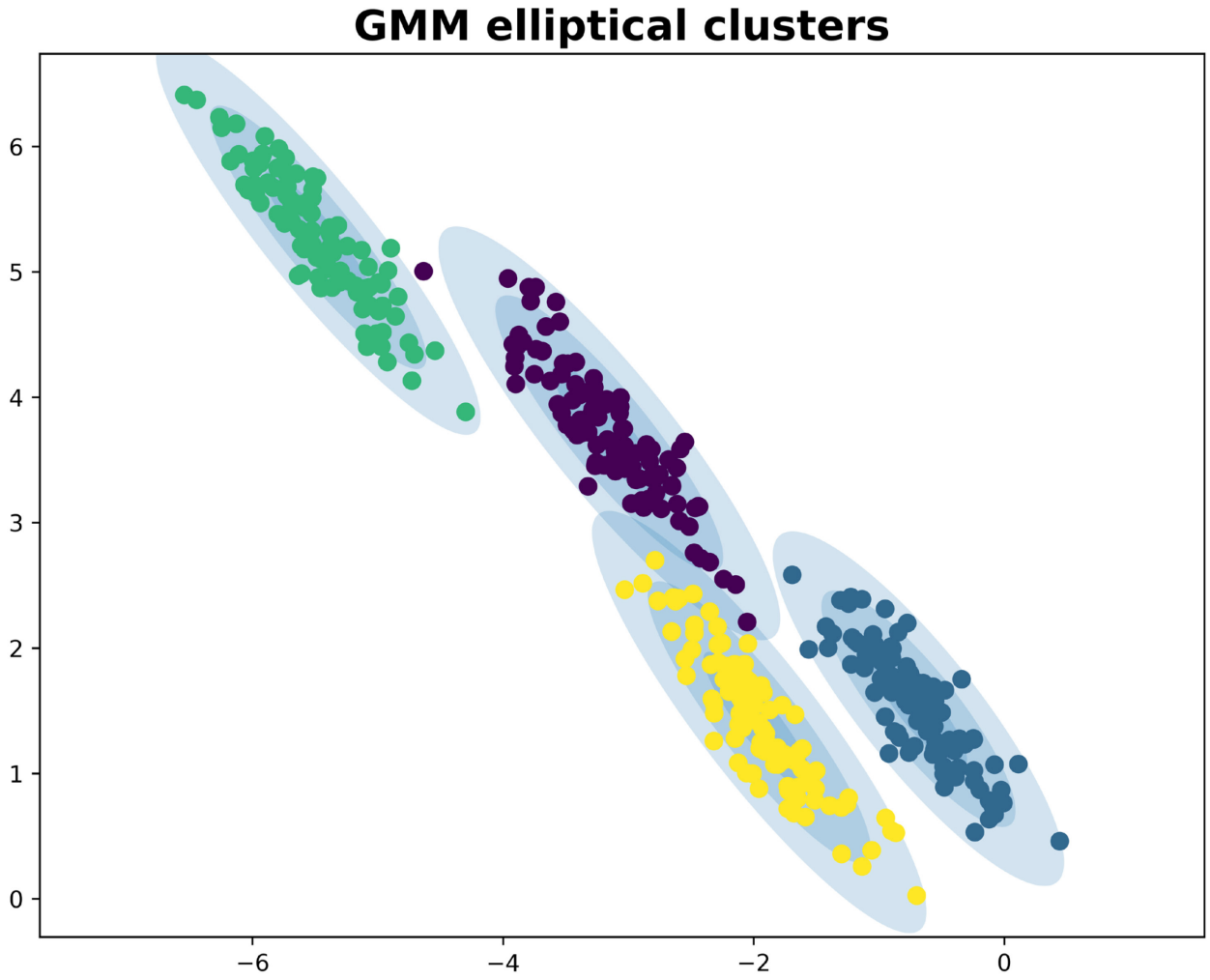
GMM for circular clusters



© NTB, christoph.wuersch@ntb.ch

GMM for elliptical clusters

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz



© NTB, christoph.wuersch@ntb.ch

Gaussian Mixture Model (II)

- We assume that the samples $\{\boldsymbol{x}_n\}$ are iid samples from a weighted sum of K D-dimensional Gaussians. The (probability) density is characterized by the following parameters $\boldsymbol{\theta}$:

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

- The (joint) probability density is given by:

$$p(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^{N} p(\mathbf{x}_n \mid z_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \cdot p(z_n \mid \boldsymbol{\pi})$$

$$= \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]^{z_{nk}} \prod_{k=1}^{K} \left[\pi_k\right]^{z_{nk}}$$

- **Marginal likelihood**: the z_n are **latent variables** that can be marginalized out to get a cost function that does not depend on z_n.

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

Gaussian Mixture Model (III)

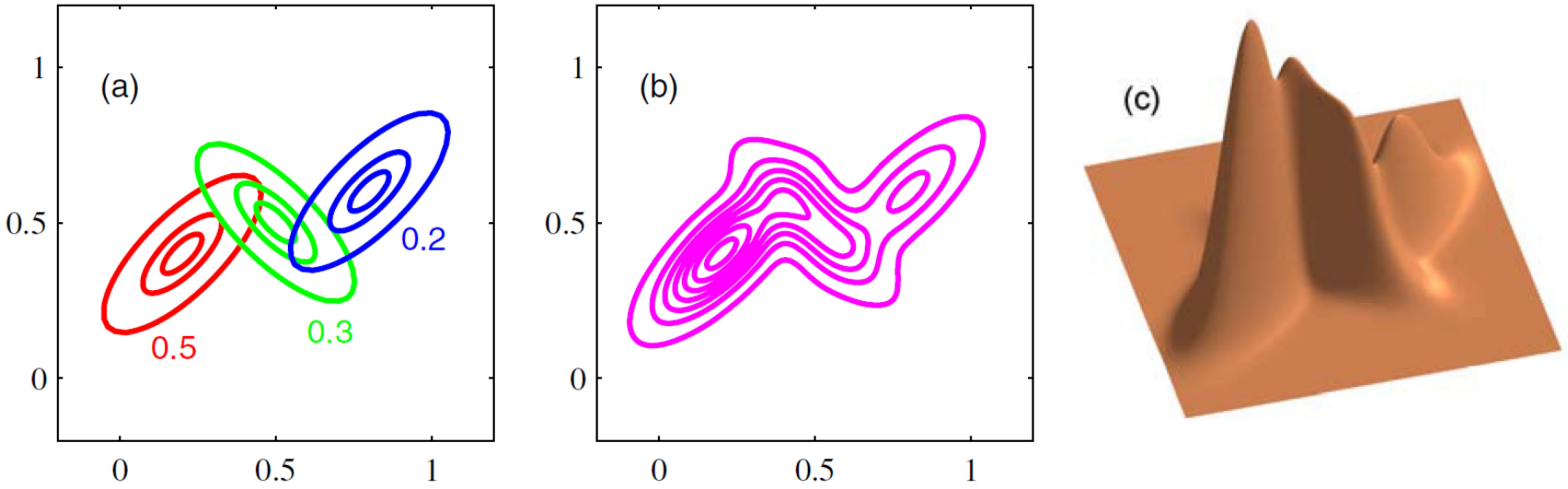


Figure 2.23 Illustration of a mixture of 3 Gaussians in a two-dimensional space. (a) Contours of constant density for each of the mixture components, in which the 3 components are denoted red, blue and green, and the values of the mixing coefficients are shown below each component. (b) Contours of the marginal probability density $p(\mathbf{x})$ of the mixture distribution. (c) A surface plot of the distribution $p(\mathbf{x})$.

Source: C.M. Bishop, Pattern Recognition and Machine Learning, Springer (2006)

The EM Algorithm

- To determine the unknown parameters we maximise the log likelihood.

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

- Compute the **cluster assignments** (E-step):

$$\gamma_{nk}^{(t)} = p(z_{nk} = k \mid \boldsymbol{\theta}^{(t)}, \mathbf{x}_n) = \frac{\pi_k^{(t)} \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}\right)}{\sum_{k=1}^{K} \pi_k^{(t)} \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}\right)}$$

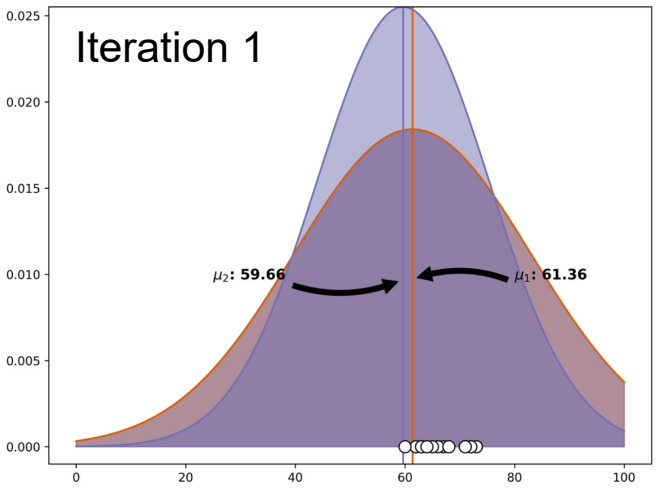
- **Update the cluster centers, covariances and probabilities** (M-Step)

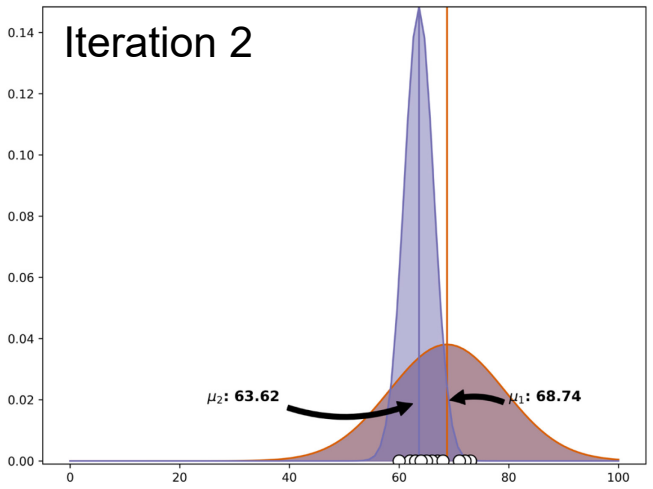
$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_n \gamma_{nk}^{(t)} \mathbf{x}_n}{\sum_n \gamma_{nk}^{(t)}} = \frac{\sum_n \gamma_{nk}^{(t)} \mathbf{x}_n}{Z^{(t)}} \qquad \pi_k^{(t+1)} = \frac{1}{N} \sum_n \gamma_{nk}^{(t)}$$

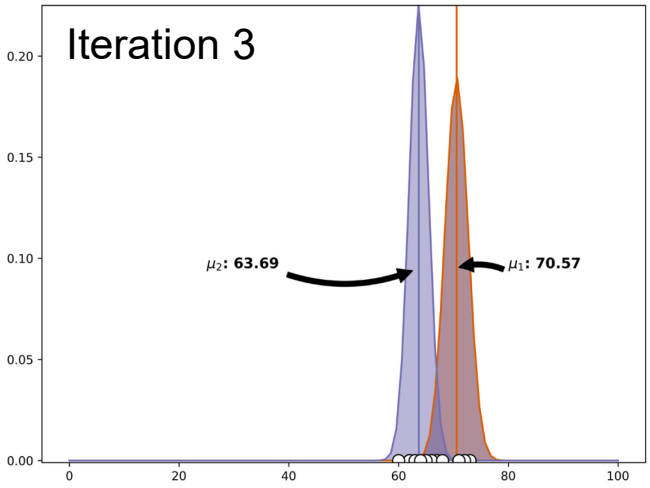
$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{1}{Z^{(t)}} \cdot \sum_n \gamma_{nk}^{(t)} \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t+1)}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t+1)}\right)^T$$

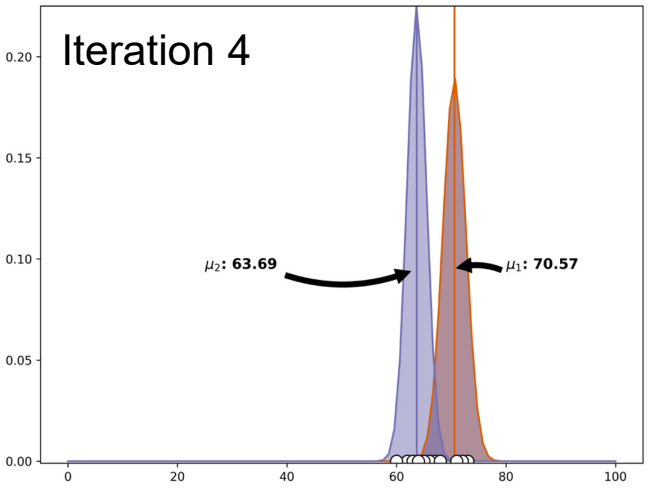
Convergece in 4 iteration steps (see exercise)

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz









© NTB, christoph.wuersch@ntb.ch

EM algorithm (see chapter 9a)

NTB Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- The main difficulty in learning Gaussian mixture models from unlabeled data is that one usually doesn't know which points came from which **latent component** (if one has access to this information, it gets very easy to fit a separate Gaussian distribution to each set of points).
- Expectation-maximization is a well-founded statistical algorithm to get around this problem by an iterative process.
- **E-Step:** First one assumes random components (randomly centered on data points, learned from k-means, or even just normally distributed around the origin) and computes for each point a probability of being generated by each component of the model.
- **M-Step:** Then, one tweaks the parameters to maximize the likelihood of the data given those assignments. Repeating this process is guaranteed to always converge to a local optimum.

Watch Alexander Ihler: https://www.youtube.com/watch?v=qMTuMa86NzU

Summary

Gaussian mixture models

- Flexible class of probability distributions
- Explain variation with hidden groupings or clusters of data
- Latent "membership" z_i
- Feature values x_i are Gaussian given z_i

Expectation-Maximization

- Compute soft membership probabilities, "responsibility" γ_{ik}
- Update mixture component parameters given soft memberships
- Ascent on log-likelihood: convergent, but local optima

Selecting the number of clusters

- Penalized likelihood or validation data likelihood

Practice: Comparison of MIN and EM-Clustering

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- *We assume EM clustering using the Gaussian (normal) distribution.*
- MIN is ***hierarchical***, EM clustering is ***partitional***.
- Both MIN and EM clustering are ***complete***.
- MIN has a **graph-based** (contiguity-based) notion of a cluster, while EM clustering has a prototype (**or model-based)** notion of a cluster.
- MIN will not be able to distinguish poorly separated clusters, but EM can manage this in many situations.
- MIN can find clusters of different shapes and sizes; EM clustering prefers g*lobular clusters* and can have trouble with clusters of different sizes.
- Min has trouble with clusters of different densities, while EM can often handle this.
- Neither MIN nor EM clustering finds subspace clusters.

Practice: Comparison of MIN and EM-Clustering

- *MIN can handle outliers*, but noise can join clusters; EM clustering can tolerate noise, but can be *strongly affected by outliers*.
- EM can only be applied to data for which a centroid is meaningful; MIN only requires a meaningful definition of proximity.
- *EM will have trouble as dimensionality increases* and the number of its parameters (the number of entries in the covariance matrix) increases as the square of the number of dimensions; MIN can work well with a suitable definition of proximity.
- EM is designed for Euclidean data, although versions of EM clustering have been developed for other types of data. MIN is shielded from the data type by the fact that it uses a similarity matrix.
- MIN makes no distribution assumptions; the version of EM we are considering assumes Gaussian distributions.

Practice: Comparison of MIN and EM-Clustering

- EM has an O(n) time complexity; MIN is $O(n^2 \log(n))$.
- Because of random initialization, the clusters found by EM can vary from one run to another; *MIN produces the same clusters unless there are ties in the similarity matrix*.
- Neither MIN nor EM automatically determine the number of clusters.
- MIN does not have any user-specified parameters; EM has the number of clusters and possibly the weights of the clusters.
- EM clustering can be viewed as an optimization problem; MIN uses a graph model of the data.
- Neither EM or MIN are order dependent.

General Literature (excellent reference books)

NTB
Interstaatliche Hochschule für Technik Buchs
FHO Fachhochschule Ostschweiz

- Daume, A Course in Machine Learning
- Barber, Bayesian Reasoning and Machine Learning
- Hastie, Tibshirani, and Friedman, The Elements of Statistical Learning
- MacKay, Information Theory, Inference, and Learning Algorithms

The following print textbooks are good quality, but in some cases more advanced or mathematical than this course:

- Bishop, Pattern Recognition and Machine Learning
- Murphy, Machine Learning: A Probabilistic Perspective
- Duda, Hart, and Stork, Pattern Classification
- Rogers and Girolami, A First Course in Machine Learning
- Mitchell, Machine Learning