Gaussian Mixture Models (GMMs)

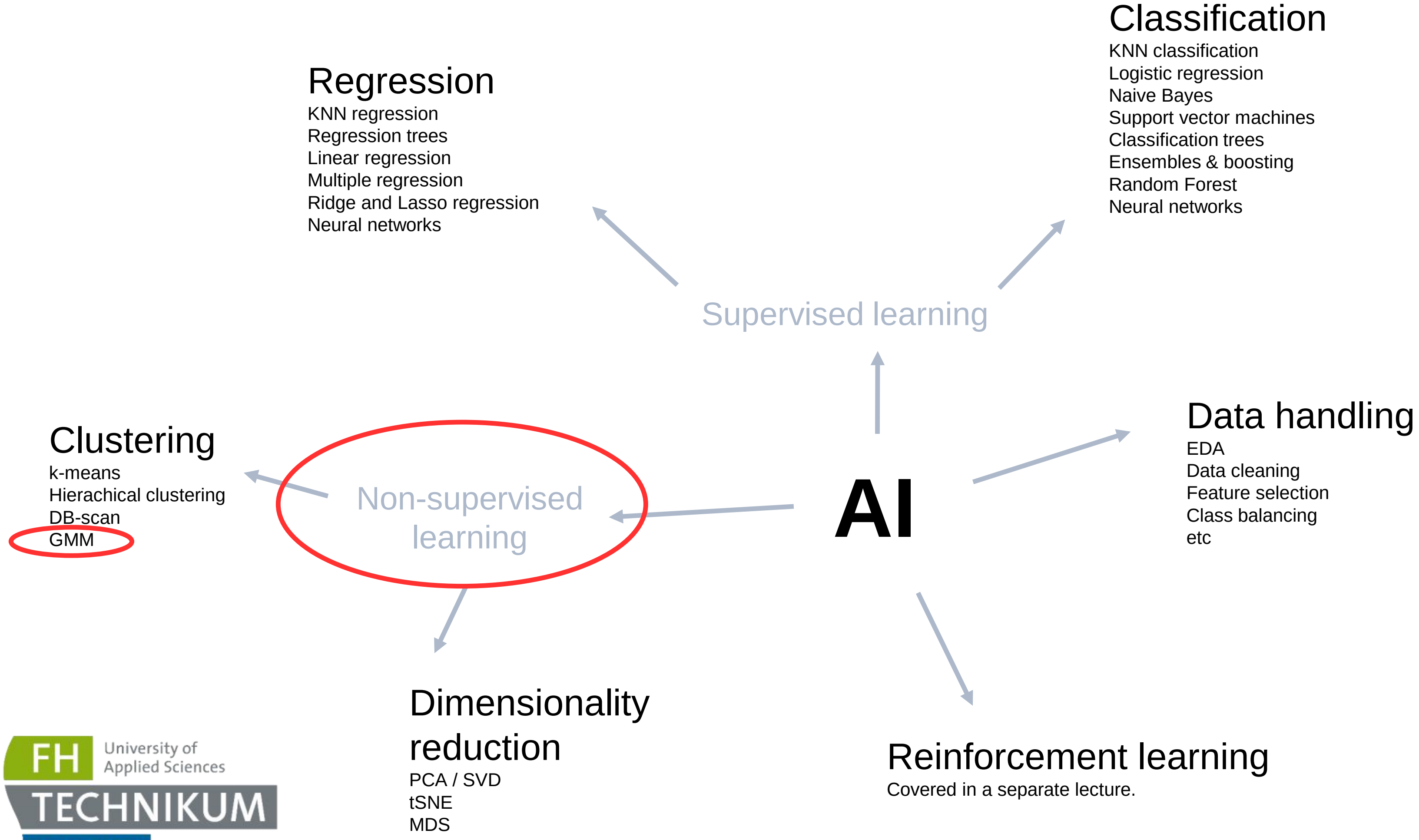
ML2: AI Concepts and Algorithms (SS2025)

Faculty of Computer Science and Applied Mathematics

University of Applied Sciences Technikum Wien

Lecturer: Rosana Gomes

Authors: S. Rezagholi, R.O. Gomes



AI

Supervised learning

Regression

KNN regression
Regression trees
Linear regression
Multiple regression
Ridge and Lasso regression
Neural networks

Classification

KNN classification
Logistic regression
Naive Bayes
Support vector machines
Classification trees
Ensembles & boosting
Random Forest
Neural networks

Data handling

EDA
Data cleaning
Feature selection
Class balancing
etc

Non-supervised learning

Clustering

k-means
Hierachical clustering
DB-scan
GMM

Dimensionality reduction

PCA / SVD
tSNE
MDS

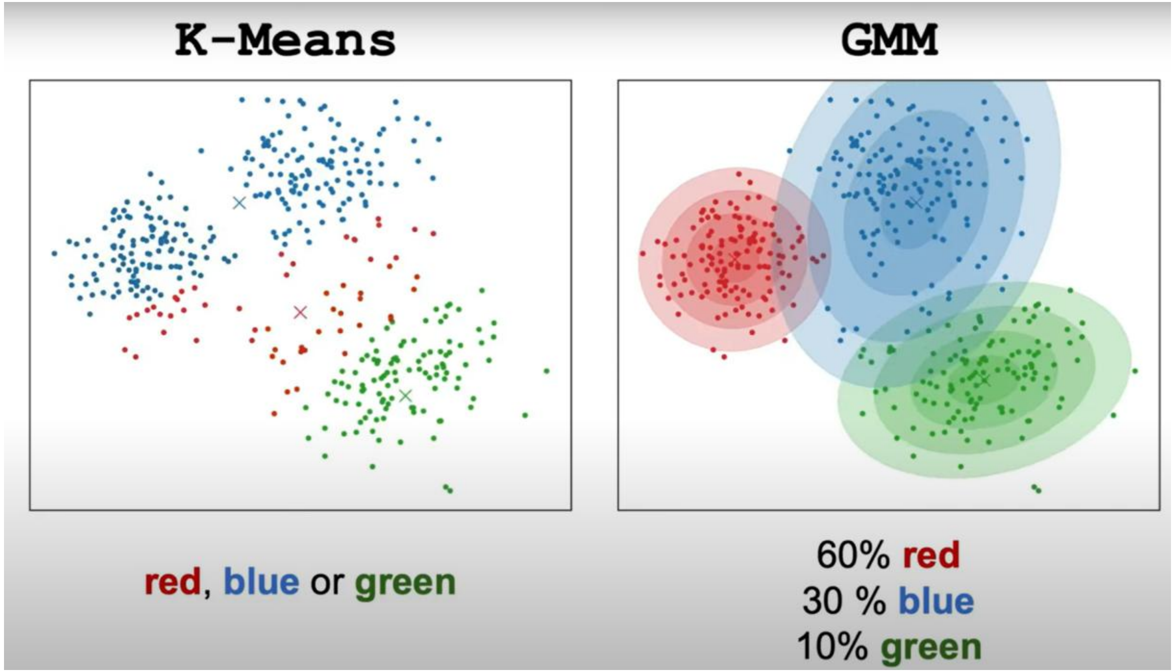
Reinforcement learning

Covered in a separate lecture.

Gaussian Mixture Models: Motivation

K-means puts a hard clustering label on each data point.

GMM assign a softer clustering label, telling the probability of a point belonging to each cluster.



GMM fits a Gaussian on top of each cluster.

Multidimensional Gaussian Distributions

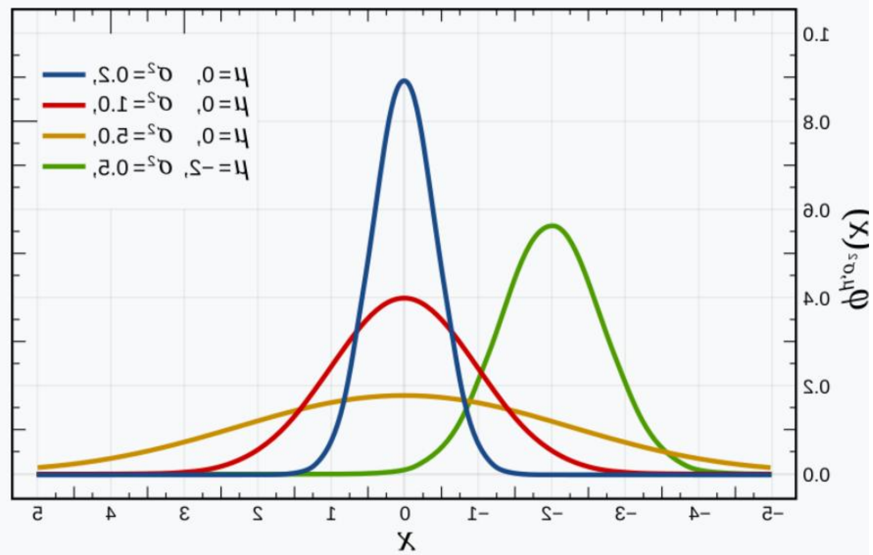
In probability theory and statistics, a normal distribution or Gaussian distribution is a type of continuous probability distribution for a real-valued random variable.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$f(x)$ = probability density function

σ = standard deviation

μ = mean

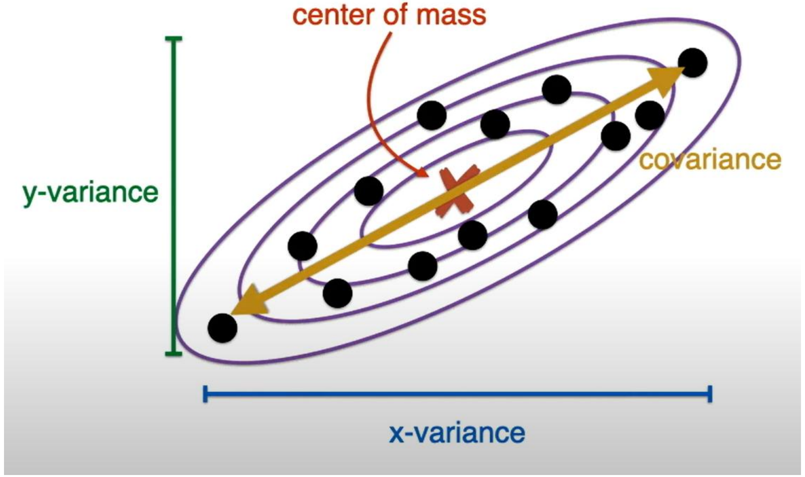


Covariance Matrix

Covariance measures how much two variables change together.

Cov >0: variables tend to move in the same direction.

Cov<0: variables tend to move in opposite directions.



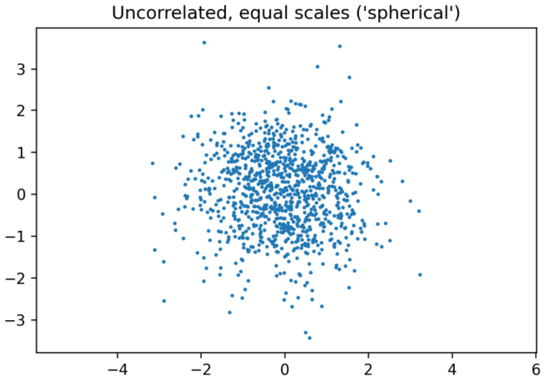
- **Diagonal elements:** The diagonal elements of the matrix represent the variance of each individual variable.
- **Off-diagonal elements:** The off-diagonal elements represent the covariance between different pairs of variables.

$$\mu = Average$$

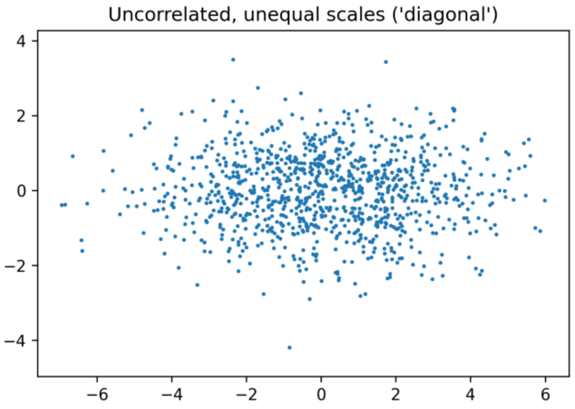
$$\Sigma = \begin{pmatrix} Var(x) & Cov(x,y) \\ Cov(x,y) & Var(y) \end{pmatrix}$$

FH University of Applied Sciences TECHNIKUM WIEN

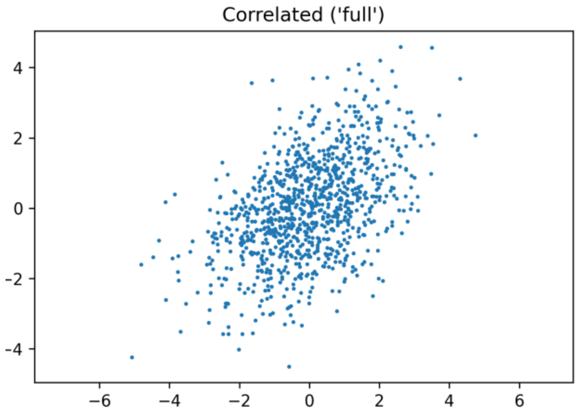
Examples


Uncorrelated, equal scales ('spherical')

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$


Uncorrelated, unequal scales ('diagonal')

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$$


Correlated ('full')

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

Gaussian Mixture Models (GMMs)

The Gaussian mixture distribution can be written as a linear superposition of Gaussians in the form

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

parameters

where the parameters $\{\pi_k\}$ must satisfy

$$0 \leqslant \pi_k \leqslant 1$$

together with

$$\sum_{k=1}^{K} \pi_k = 1$$

Mixture model follow two-stage process for data generation:

1. Choose the Gaussian to sample (define distribution parameters)
2. Sample from the respective distribution: x ~ fi

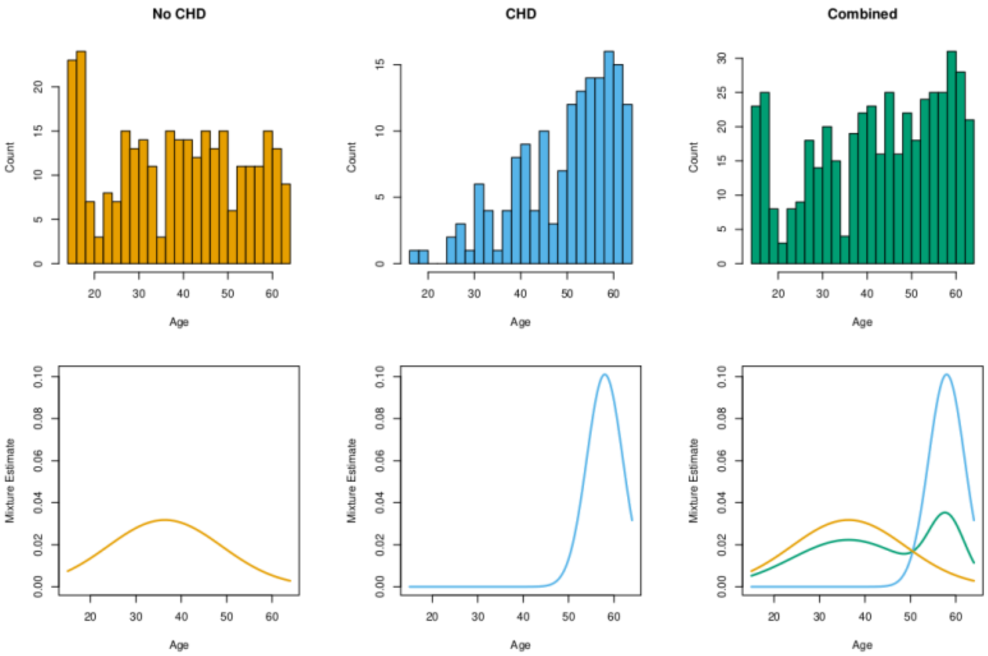
Gaussian Mixture Models (GMMs)

Responsibility of k for x:
The probability that observation x comes from component k

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

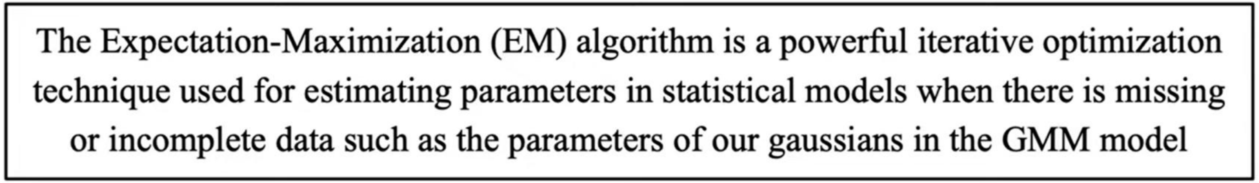
Clustering: assign the datapoint x to the component with the highest responsibility for the point.

Example: Binary Problem



[Source: Hastie et al. (2017): The Elements of Statistical Learning.]

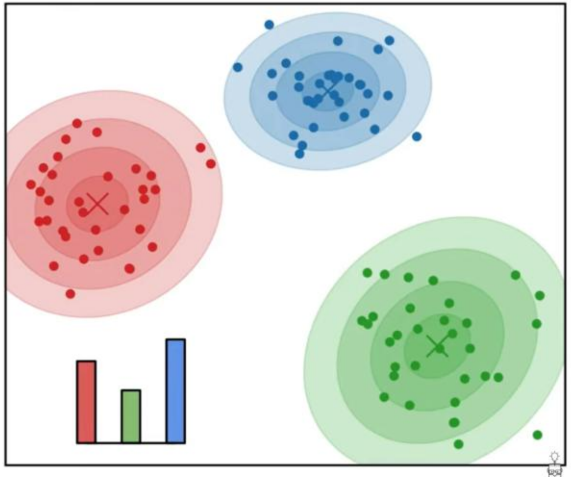
The Expectation-Maximization Algorithm (EM)



The Expectation-Maximization (EM) algorithm is a powerful iterative optimization technique used for estimating parameters in statistical models when there is missing or incomplete data such as the parameters of our gaussians in the GMM model

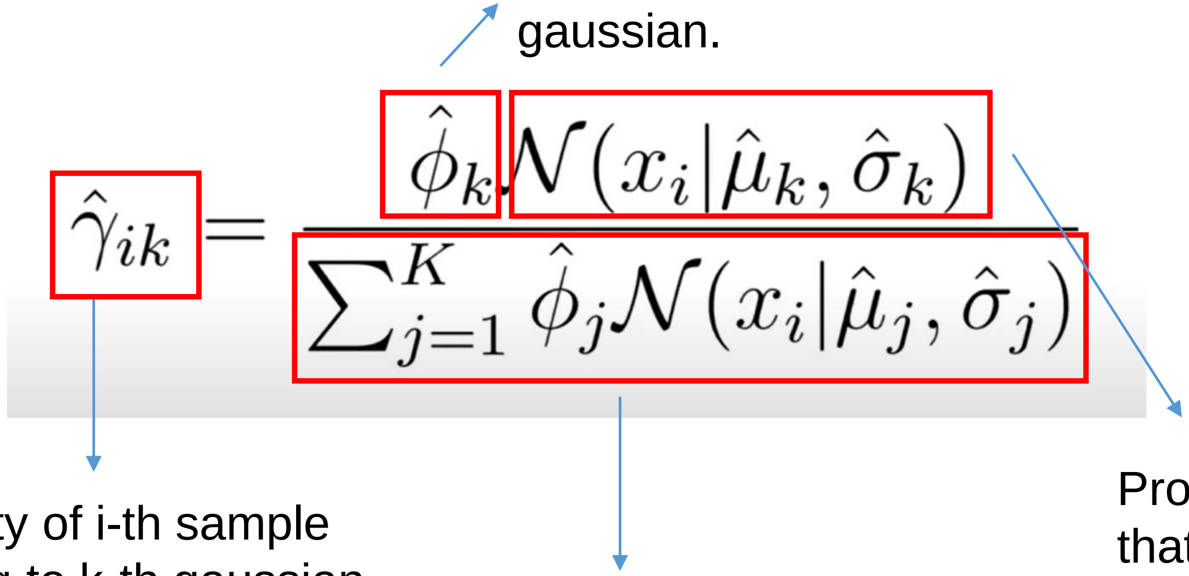
Expectation Step (E-step): calculate the expected value of the log-likelihood function given the current parameter estimates.

Maximization Step (S-step): update the parameter estimates to maximize the expected log-likelihood calculated in the E-step.



Parameters: mean, covariance and weights.

Expectation step (E-step)



$$\boxed{\hat{\gamma}_{ik}} = \frac{\boxed{\hat{\phi}_k}\boxed{\mathcal{N}(x_i|\hat{\mu}_k,\hat{\sigma}_k)}}{\boxed{\sum_{j=1}^{K} \hat{\phi}_j \mathcal{N}(x_i|\hat{\mu}_j,\hat{\sigma}_j)}}$$

Weight of k-th gaussian.

Probability of i-th sample belonging to k-th gaussian.

Weighted sum over the probability that the i-th sample was generated by each j-th gaussian.

Probability that the i-th sample was generated by the k-th gaussian.

Maximization Step (M-step)

$$\boxed{\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}}$$

The new k-th weight becomes the average of the probabilities that a point belongs to that gaussian.

$$\boxed{\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}}$$

The new k-th mean becomes the weighted average of all points.

$$\boxed{\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}}$$

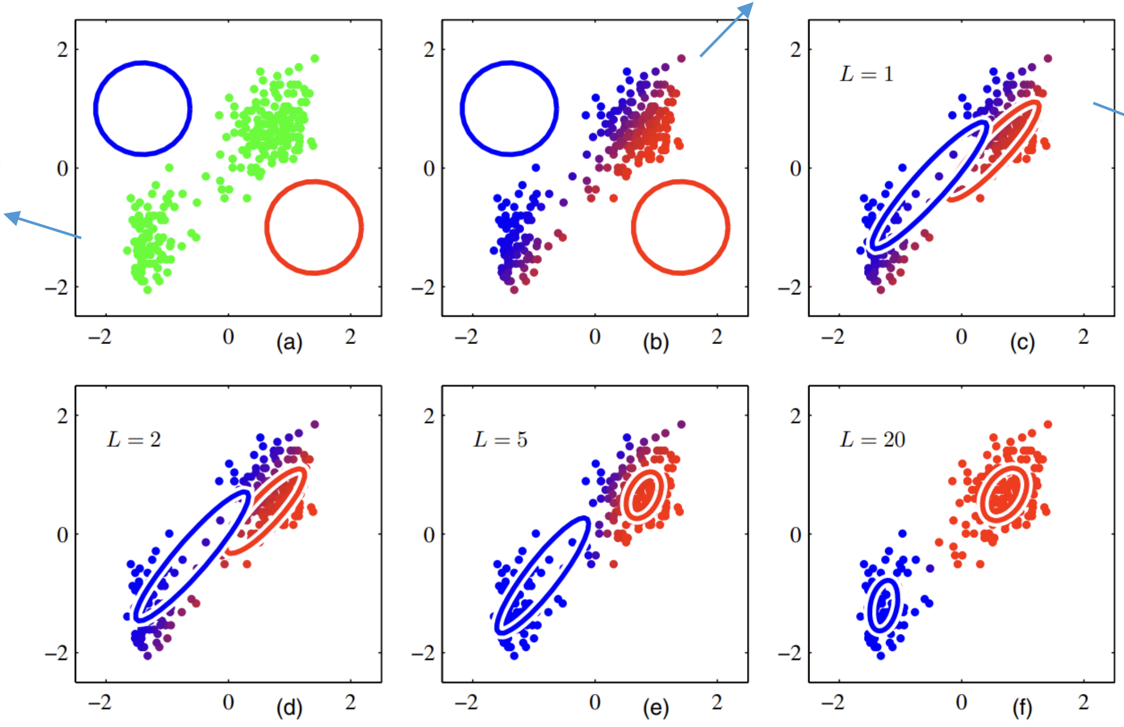
The new k-th variance becomes the weighted variance of all points.

GMMs in action: Clustering

1. Start with random Gaussians (initialization)

2. Assign points to each distribution (E-step)

3. Recalculate parameters for the Gaussians (M-step)



Continue EM algorithm until convergence.

[Source: Bishop (2006), Pattern Recognition and Machine Learning]

GMMs for Anomaly Detection

Check how likely the datapoint is under the GMM-density: The (new) datapoint x is considered anomalous if

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad < f_{\text{min}}.$$

GMMs are sensitive to two types of anomalies:

- Clusters that are very small compared to others (via the mixture probabilities π_k),
- and points that do not belong to any cluster (via the GMM-density).

GMMs for Anomaly Detection + Clustering

GMMs can easily be used as clustering models that are able to say: ‘I do not know what this is!’ This ‘out-of-training-set usability’ of GMMs is very useful, especially for sensitive applications.

- Step 1: If the GMM-density of the (new) datapoint is too low, do not assign it to a component.
- Step 2: Otherwise assign the new datapoint to a component using the responsibility. (One may also introduce an additional responsibility condition.)

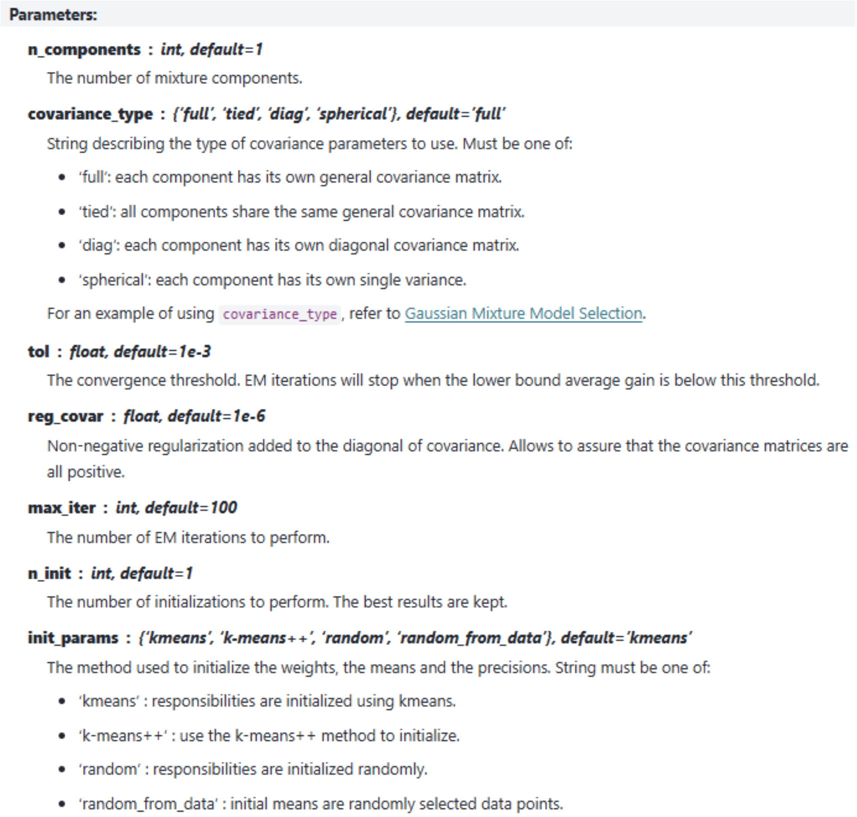
FH University of Applied Sciences TECHNIKUM WIEN

Hyperparameter tuning

https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture

Implementation

(Gaussian) mixture models:
`sklearn.mixture.GaussianMixture`
Also useful:
`scipy.stats.multivariate_normal`
`numpy.random.multivariate_normal`



Parameters:

n_components : *int, default=1*

The number of mixture components.

covariance_type : *{'full', 'tied', 'diag', 'spherical'}, default='full'*

String describing the type of covariance parameters to use. Must be one of:

- 'full': each component has its own general covariance matrix.
- 'tied': all components share the same general covariance matrix.
- 'diag': each component has its own diagonal covariance matrix.
- 'spherical': each component has its own single variance.

For an example of using `covariance_type`, refer to Gaussian Mixture Model Selection.

tol : *float, default=1e-3*

The convergence threshold. EM iterations will stop when the lower bound average gain is below this threshold.

reg_covar : *float, default=1e-6*

Non-negative regularization added to the diagonal of covariance. Allows to assure that the covariance matrices are all positive.

max_iter : *int, default=100*

The number of EM iterations to perform.

n_init : *int, default=1*

The number of initializations to perform. The best results are kept.

init_params : *{'kmeans', 'k-means++', 'random', 'random_from_data'}, default='kmeans'*

The method used to initialize the weights, the means and the precisions. String must be one of:

- 'kmeans' : responsibilities are initialized using kmeans.
- 'k-means++' : use the k-means++ method to initialize.
- 'random' : responsibilities are initialized randomly.
- 'random_from_data' : initial means are randomly selected data points.

How to choose the number of components?

Unsupervised: Akaike information criterion (AIC).
Semi-supervised: (Cross-)Validation.

FH University of Applied Sciences TECHNIKUM WIEN

Pros and Cons

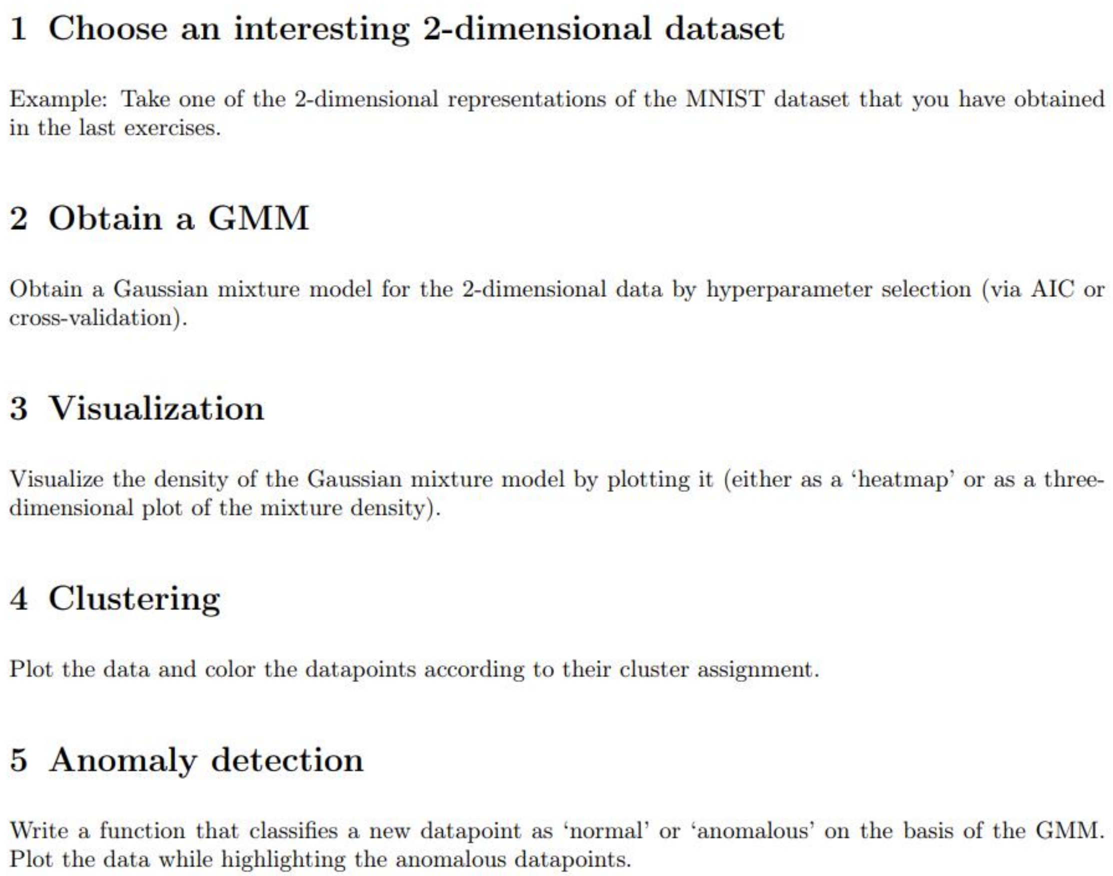
Pros

Large expressive power (for density modeling, for clustering ellipticity of the components distributions is a restriction)
Robustness against noise
Fast to fit using EM algorithm
Interpretable
Versatile: Density modeling, clustering, anomaly detection.

Cons

Computational burden large for high-dimensional datasets (number of parameters increases quadratically in the dimension of the data)
Only applicable to ellipsoid clusters (for clustering)
EM algorithm is prone to bad local optima and sensitive to initialization (usually done by k-means)

Assignment: GMMs



1 Choose an interesting 2-dimensional dataset

Example: Take one of the 2-dimensional representations of the MNIST dataset that you have obtained in the last exercises.

2 Obtain a GMM

Obtain a Gaussian mixture model for the 2-dimensional data by hyperparameter selection (via AIC or cross-validation).

3 Visualization

Visualize the density of the Gaussian mixture model by plotting it (either as a 'heatmap' or as a three-dimensional plot of the mixture density).

4 Clustering

Plot the data and color the datapoints according to their cluster assignment.

5 Anomaly detection

Write a function that classifies a new datapoint as 'normal' or 'anomalous' on the basis of the GMM. Plot the data while highlighting the anomalous datapoints.