# Machine learning for clustering

*M. Mougeot*

*ENSIIE, November 2020*

## Contents

# Introduction

## Goal of the practical sessions

- To understand classification machine learning methods, from a methodological and practical point of view.
- To apply models and to tune the appropriate parameters on several data sets using the 'Python' language.
- To interprete 'Python' outputs.

## Warnings and Advices

- The goal of this practical session is not "just to program with Python"" but more specifically to understand the framework of Modeling, to learn how to developp appropriate models for answering to a given operationnal question on a given data set. The MAL course belongs to the **Data Science courses** . For each MAL practical session, you should **first understand** the mathematical and statistical backgrounds, **then write your own program with 'Python'** to practically answer to the questions.

## Instructions

- The practical work must be carried on with a 'group of two students'.
- The MAL project aims to developp a Jupyter notebook to solve a classification problem (the subject will be given soon). Your names have to be written in the first lines of the program file with comments. Please note that without this information, no grade will be attributed to the missing name project.

For this practical session, the following librairies need bo be uploaded in the python environment.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats
from sklearn import mixture
```

# The data

The data are stored in the irm_thorax.txt file and represent a human thorax image.

## Medical thorax image

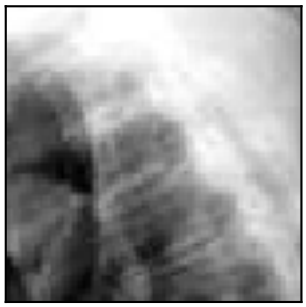

Figure 1: Thorax image

### Image histogram

Display the histogram of the pixel values using the following instructions

```
n=len(tab); X=tab.reshape(n*n);
plt.hist(X, range = (0, 255), bins = 20, color = 'yellow', edgecolor = 'red');
```

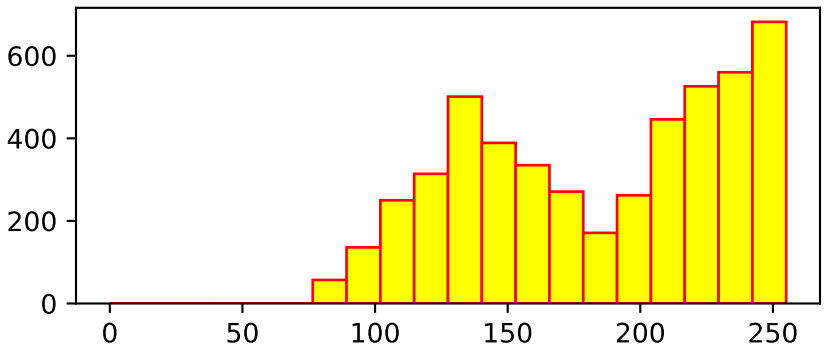


Figure 2: Histogram

What can we observe ?

## Mixture of models and the EM algorithms

The EM algorithm allows to realize a maximum likelihood type approach on mixture models.

### Mixture of Gaussian writing your own Python instructions

- Write your own Python instructions to implement the Expectation-Maximization algorithm with the help of the ML lesson slides.
- Apply your EM code on this dataset to segment in two classes the thorax image based on the pixel values.

#### A bimodal distribution assumption

Display the parameters of the gaussian models (weight, mean and standard deviation) and draw the mixture of gaussian on the empirical density of the pixels (scipy.stats.norm.pdf function) as in the following graph:

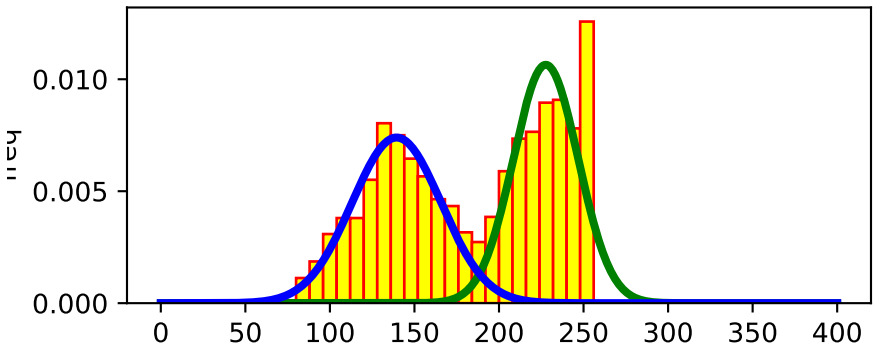


Figure 3: Mixture of gaussian densities. Assumption: K=2

## Gaussian Mixture Models (GMM) using scikit learn

- Study the help of the `mixture.GaussianMixture()` function of the scikitlearn library.
- Use the following instructions to recover the previous segmentation with an assumption 2 classes.

```
from sklearn import mixture
X2 = X.reshape(-1, 1)
modgmm = mixture.GaussianMixture(n_components=2, covariance_type='full');
fitgmm=modgmm.fit(X);predX=fitgmm.fit_predict(X);imgGMMK2=predX.reshape(n,n);

#Display
figure = plt.figure(figsize=(5.5,2));
plt.xticks([], []); plt.yticks([], []);
ax = plt.subplot(1,2, 1); ax.imshow(tab, cmap='gray');
ax.set_title('Original Image')
ax = plt.subplot(1,2, 2); ax.imshow(imgGMMK2, cmap='Paired')
ax.set_title('GMM Image Segmentation')
plt.show();
```

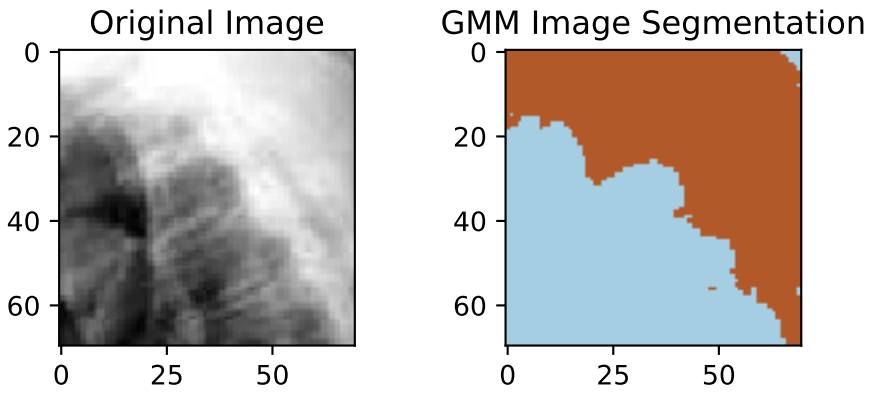


Figure 4: GMM segmentation. Assumption: K=2

### Multimodal distribution assumption

- Study the help of the `mixture.GaussianMixture()` function of the scikitlearn library.
- Compute and dispay a segmentation of the thorax image with 3 and 5 classes.
- Display the parameters of the gaussian models (weight, mean and standard deviation).
- Do the segmentations seem relevant to you?

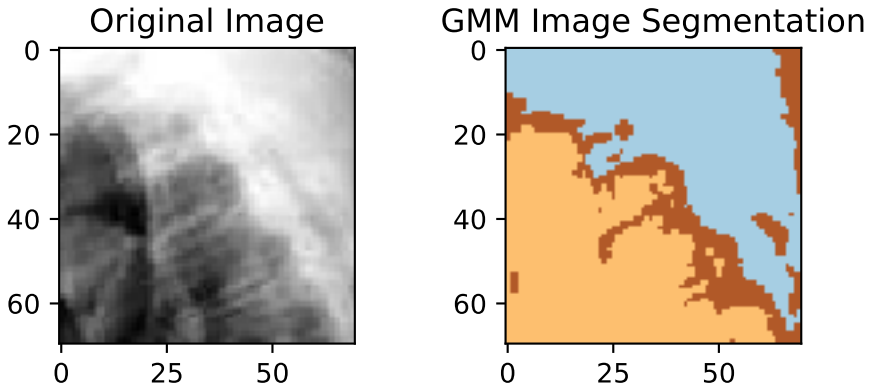


Figure 5: GMM segmentation. Assumption: K=3

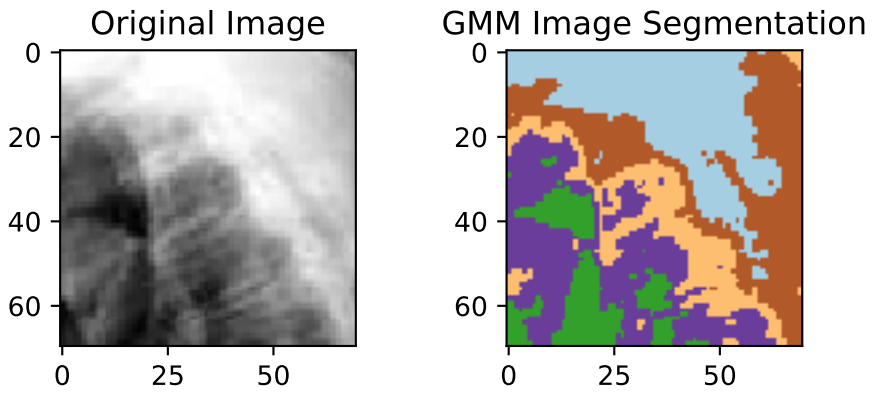


Figure 6: GMM segmentation. Assumption: K=5

## Kmeans

- Study the help of the `Kmeans()` function of the sklearn.cluster library.

```
from sklearn.cluster import KMeans
modkmeans = KMeans(n_clusters=2, random_state=0)
fitkmeans=modkmeans.fit(X)
```

- Compute a segmentation of the thorax image using the Kmeans algorithm with an assumption of k=2 groups.
- Display the values of the K centroids.
- Compare, using your own instructions, the GMM and Kmeans segmentation of the thorax image as illustrated in the following figure.

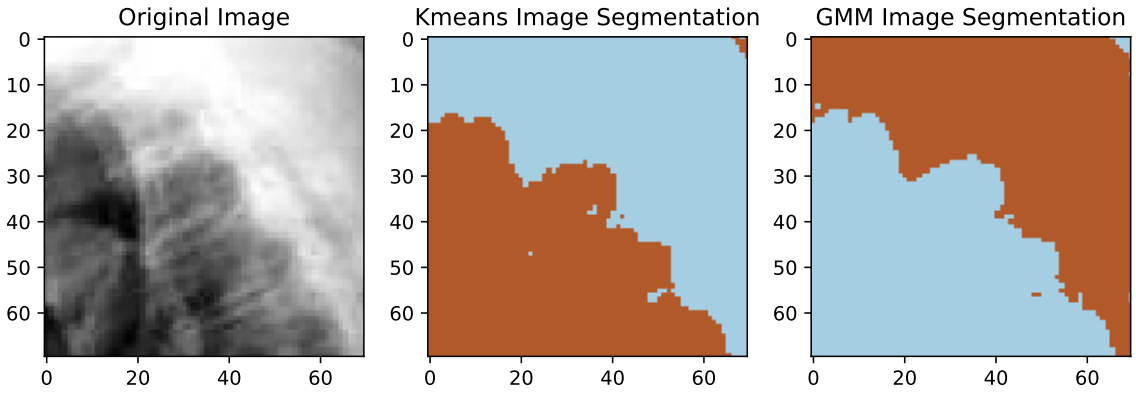


Figure 7: Comparison of Thorax image segmentation with Kmean and GMM algorithms. Assumption: K=2