FTML practical session 7

20 avril 2025

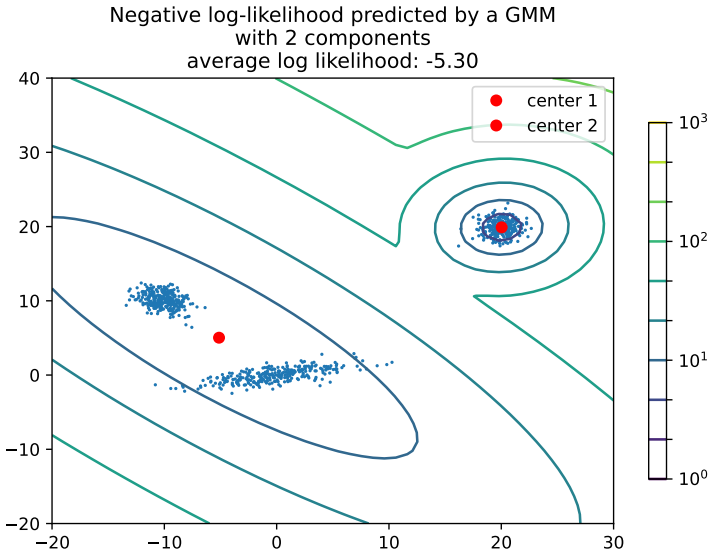

TABLE DES MATIÈRES

1 DENSITY ESTIMATION WITH GAUSSIAN MIXTURES

In this exercise, we will learn a probability distribution from a dataset (density estimation). We use Gaussian mixtures (GMM) as parametric models.

1.1 Gaussian mixtures

Gaussian mixtures model the considered data by a number p of Gaussian distributions. Each one of the p subdistributions is called a "component".

https://fr.wikipedia.org/wiki/Mod%C3%A8le_de_m%C3%A9lange_gaussien

https://scikit-learn.org/stable/modules/mixture.html

You can read and run the file **example_2D.py**, that generates a dataset in 2D, and learns Gaussian mixtures models for different numbers of components. The results are displayed in figures 1, 2, 3, 4, and 5.

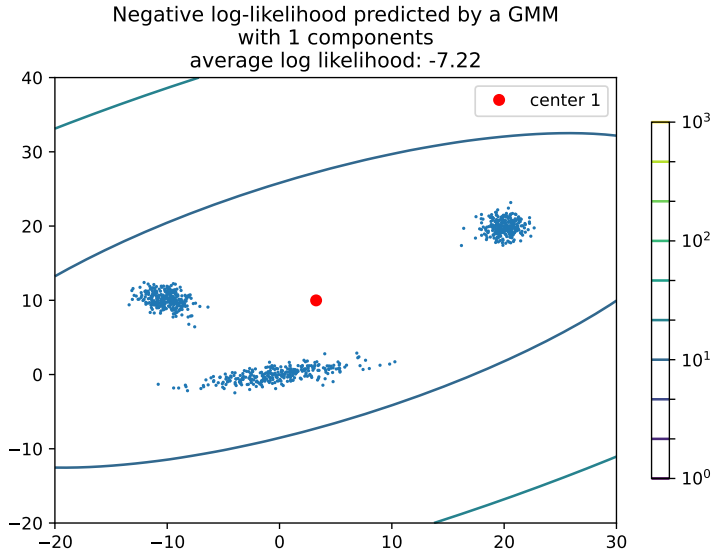

FIGURE 1 – Gaussian mixture with 1 component.

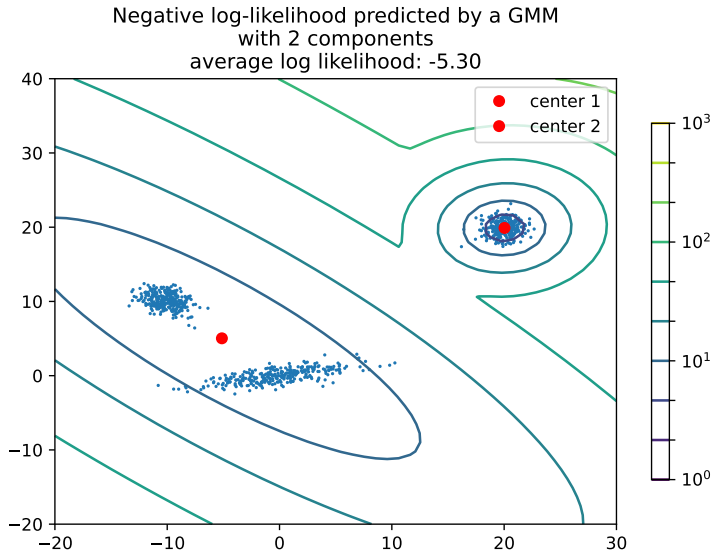

FIGURE 2 – Gaussian mixture with 2 component.

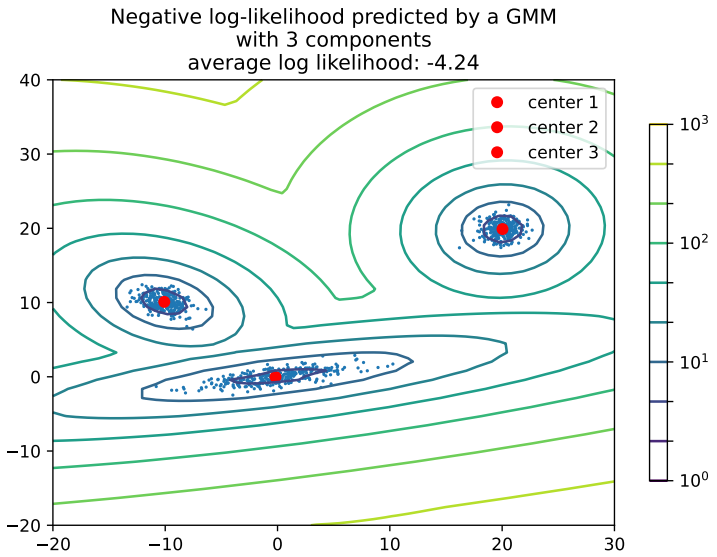

FIGURE 3 – Gaussian mixture with 3 component.

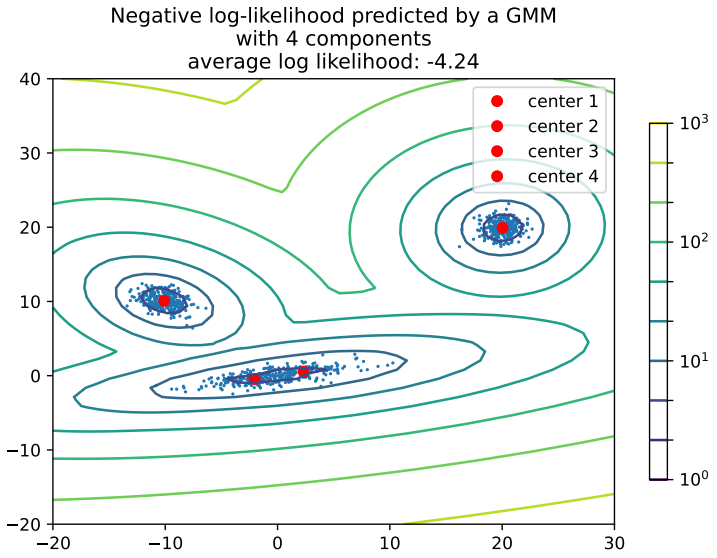

FIGURE 4 – Gaussian mixture with 4 component.

The main parameter of a Gaussian mixture is its number of components. Intuitively, it can be seen as an analogous to the number of centroids in a k-means clustering algorithm. In the previous images, we note that the average log-likelihood does not improve as much when we increase the number of components from 3 to 4, as it does when increasing it from 2 to 3. Take a few minutes to read some of the documentation on the scikit-learn page.

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

1.2 Digits dataset

Gaussian distributions are correctly defined in any finite dimension d. Hence, we can also learn a distribution over high dimensional datasets. We will work with the digits dataset from scikit.

Estimate a Gaussian mixture distribution from the dataset, choosing the number of components that minimizes the Akaike information criterion (AIC) (in order to

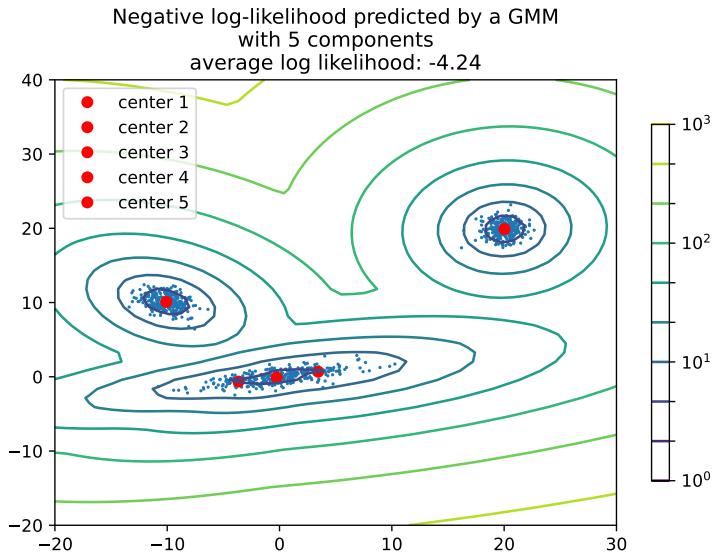

FIGURE 5 – Gaussian mixture with 5 component.

save computation time, you might try stepped numbers of components, below 200), like in figure 6. Plot the mean of each component. (see figures 7 and 8)

https://fr.wikipedia.org/wiki/Crit%C3%A8re_d%27information_d%27Akaike

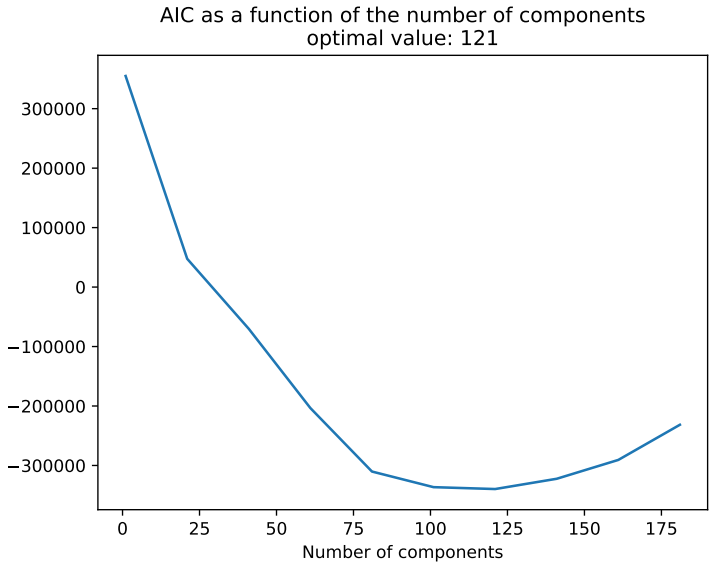

FIGURE 6 – Akaike information criterion.

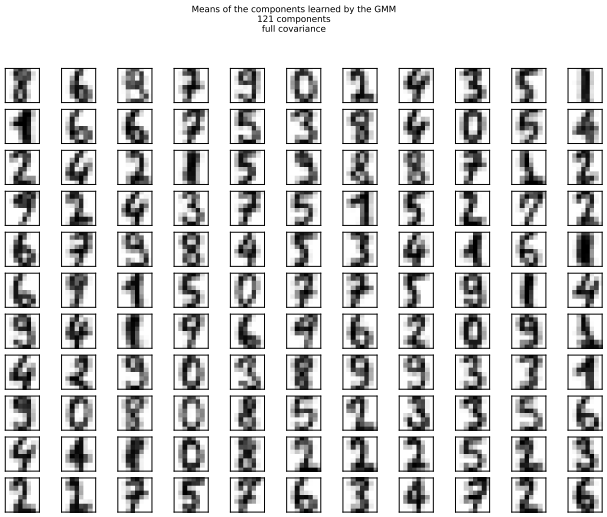

FIGURE 7 – Means of the components learned by the GMM, with 121 components and "full"" covariance (see the scikit documentation).

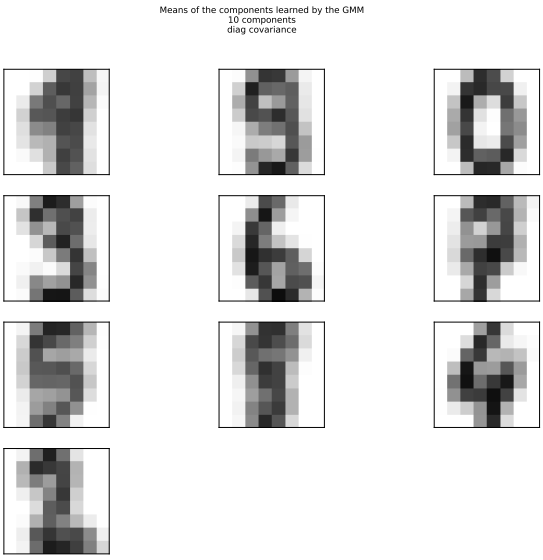

FIGURE 8 – Means of the components learned by the GMM, with 10 components and diagonal covariance (see the scikit documentation).

1.3 Generation of digits

One interesting aspect of having fitted a probabilistic model to a dataset, is that we can sample this model!

Use your previously learned Gaussian mixture model in order to generate images and compare them to images from the dataset. Explore the influence of hyperparameters (like the number of components, the covariance type) on the generated images.

An example output is shown in figure 9.

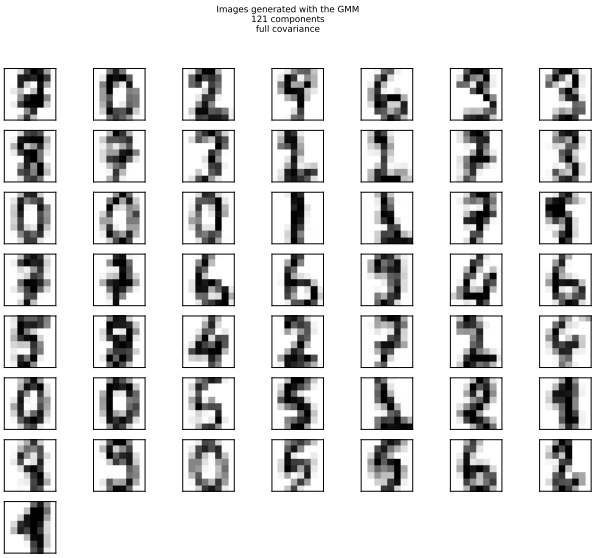

FIGURE 9 – Images generated by sampling a GMM trained on the dataset

1.4 Using Gaussian mixtures for prediction

1.4.1 *Method*

We consider a supervised learning problem, where we need to predict some output y as a function of an input x. Let us admit that we perform a density estimation on the joint variable (x, y), and obtain a distribution p.

How could we build a prediction function with p ?

1.4.2 *Application to prediction*

We will apply this approach to the Old faithful geyser dataset.
`https://www.stat.cmu.edu/~larry/all-of-statistics/=data/faithful.dat`

The data are stored in a txt file and plotted on figure 10. By looking at the image, it seems that it should be possible to fit a GMM reasonably well to the data.

Fit a GMM to the dataset, by first finding the optimal number of components, this time according to the Bayesian information criterion (for this dataset, we expect that a number of components of 2 should be reasonable).
`https://scikit-learn.org/stable/modules/linear_model.html#aic-bic`

Then, build a predictor with this GMM and plot the prediction of the time between two eruptions as a function of the eruption time. You should observe results

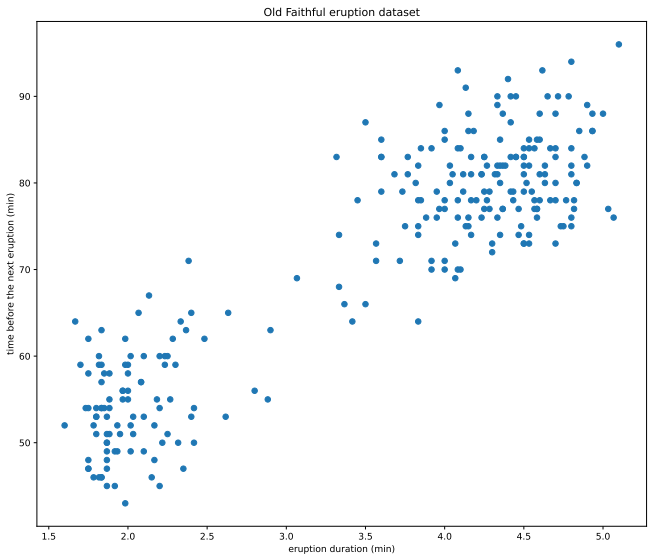

FIGURE 10 – Old faithful Geyser dataset

like figures 11 and 12.

https://en.wikipedia.org/wiki/Riemann_sum

https://www.nps.gov/yell/learn/photosmultimedia/indepth-predictingoldfaithful.htm

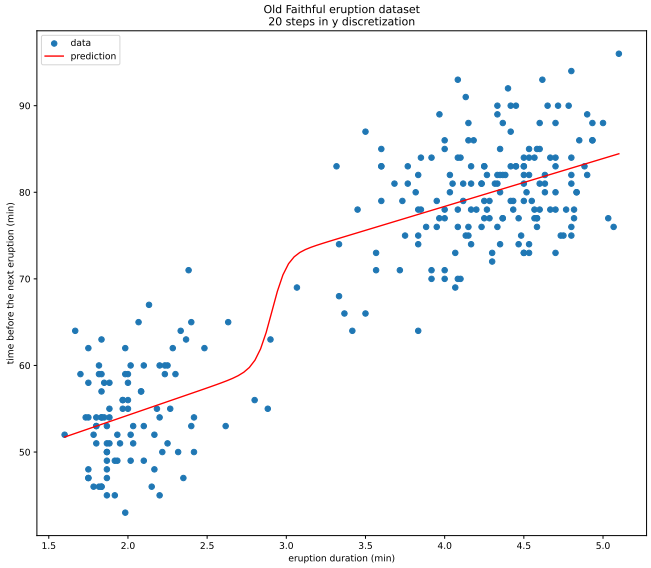

FIGURE 11 – Prediction computed from the learned GMM. In this case, 20 samples in the y space were used in order to approximate the integrals required to compute the conditional expectation of y|x.

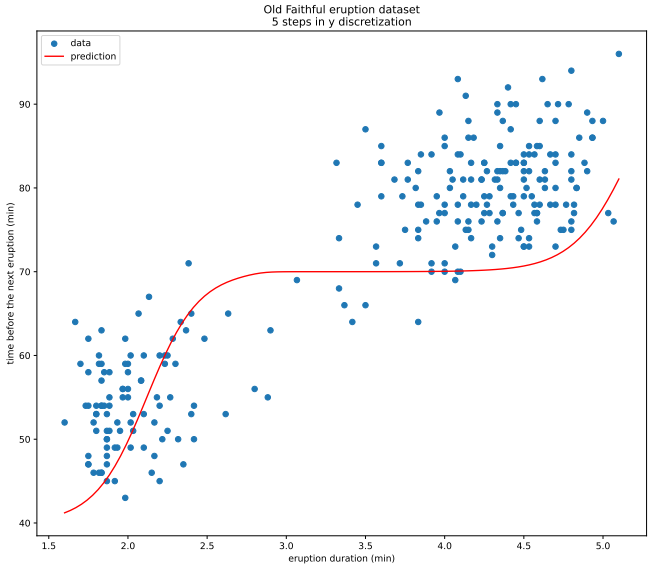

FIGURE 12 – Prediction computed from the learned GMM. In this case, 5 samples in the y space were used.

2 APPLICATION OF UNSUPERVISED LEARNING TO CLASSIFICATION

In this section, we build classifiers based on an unsupervised preprocessing of the data. We hence combine supervised and unsupervised learning in order to solve a problem.

2.1 Meteorological data : dimensionality reduction and visualization

A meteorological station has gathered 1600 data samples in dimension 6, thanks to 6 sensors, that represent various physical measurements (such as wind speed, humidity, temperature, etc). The operators of the station would like to predict the risk of a tempest the next day, but first, they need to reduce the dimensionality of the data, in order to apply a supervised learning algorithm on the reduced data.

The data are stored in the **exercice_3_dimensionality_reduction/data/** folder.

Find a dimensionality reduction method and a dimension (2 or 3), that seems to allow to predict the label based on the projected components only, first by making scatter plots of the projected data, and by coloring the data according to their label. Verify this by training a classifier that learns to predict the labels based on the projections only, with a very good accuracy.

Template files in the folder :

— **main.py**

`https://scikit-learn.org/stable/modules/unsupervised_reduction.html`

Later in the course, we will also study nonlinear dimensionality reduction methods. Note that for this particular problem, it was also possible to solve it without the first unsupervised learning stage.