UNIVERSITETET I BERGEN

INFO 284, Spring 2018,
Second Obligatory Group Assignment
Gaussian Mixture Models for Clustering

Overview

The task was to check the performance of the different clustering algorithms. We were consistent to use all 210 entries and use a 25% test and 75% train-split in every algorithm. This was done to assure consistency and to have a better overview over the output and results.
However, one of our main mistakes was that we used a random split every single time, creating varying results across the board.

k-Means clustering algorithm

kMeans provided with a steady and plain base of all the entrances based on the nature of the 210 entries we were provided with. The actual graph provided us with a straight and plain oversight without garnering any factual insight upon the task in progress. However, it did provide us with the most base functionality and allowed us to have an overview while fact-checking the provided text of the Train/Test results. The numbers of the evaluation itself are, by design, in the lower accuracy. This is mainly because of the number of entries and the lack of an overall common factor. The entries, while sharing a general connection based on their different classes, have very varying results. As such, the r2-accuracy is of low accuracy.

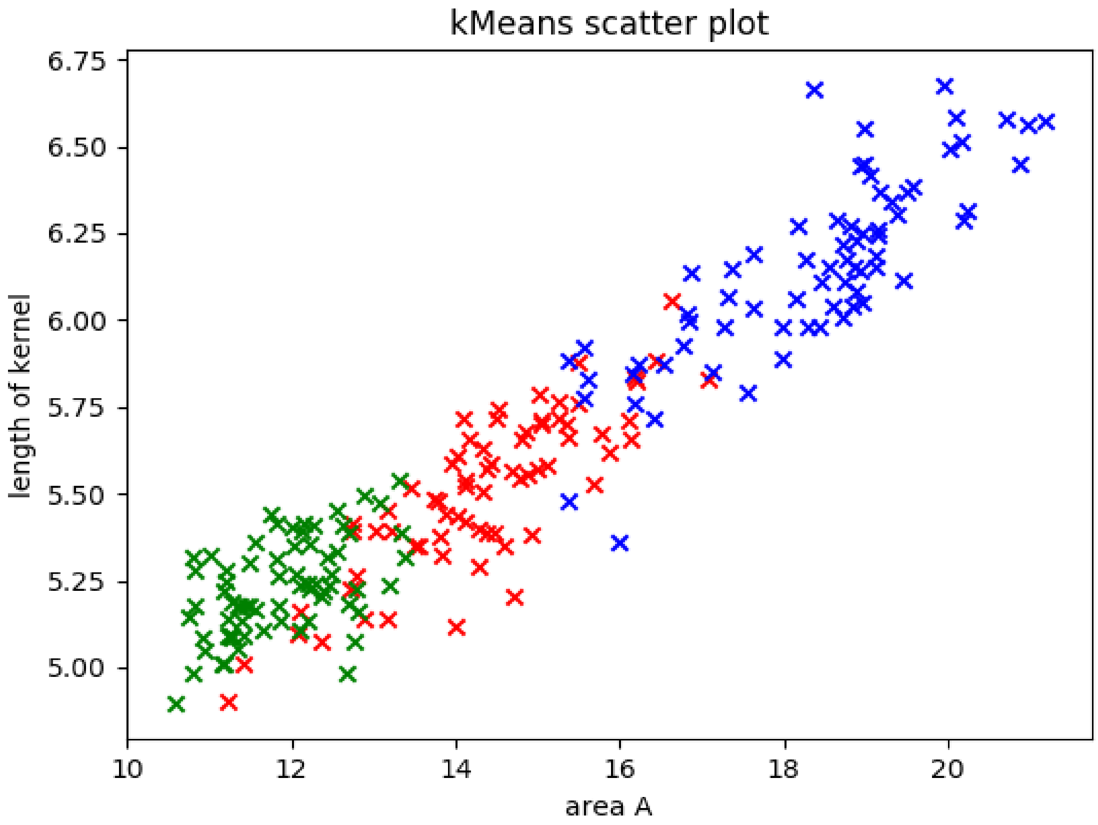

Figure 2 kMeans clusters:: visualisation of the clustered data in a 2D scatter plot

This has mainly to do with the use of all 210 entries. A smaller amount of entries may have resulted in higher accuracy (as it assumes higher probability for differentiation between the different results). While taking smaller numbers would have been possible, it was decided against it. These are non-altered results based on what we were provided with.

Gaussian Mixture Models Clustering algorithm

We covered four covariance types for Gaussian Mixture Models: 'full', 'tied', 'diag' and 'spherical'.

The figure below presents two features: 0 ie. ‘area A’ and 3, ie. ‘length of kernel’ for every covariance type.

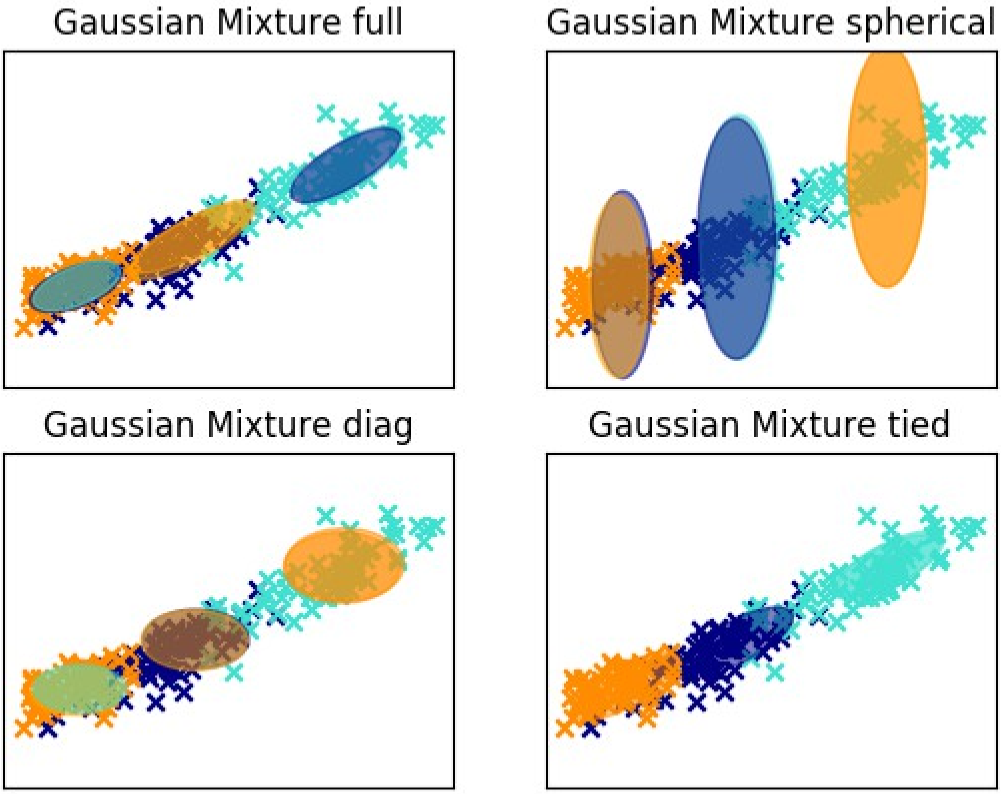

Figure 1 Gaussian Mixture Models

GMM-methods varied a bit more. The four entries were generally providing a similar answer.
In order to not change the input gained from kMeans, all 210 entries were used once more with the same seperation of Test/Train.

In the GMM we decided to keep a random seed.(np.random.seed(0))

Again, the entries were graphically shown correctly (or, at least, consistently compared to kMeans and each other) and could be changed for comparison easily.
However, a problem occurred, and accuracy dropped immensely. While we were able to change the size of the graphical assessment, the learning curve based on the data provided was poor.

Clustering algorithms comparative performance

The overall conclusion of this is that we used too many entries with a poor and inconsistent split. By creating a test-array (one of each possibility across the board) as part of our algorithm. However, this taught a valuable lesson about the necessity of choosing a train/test method and size.
kMeans gave a generic overview suited for any kind of test-result. While basic, it was well-suited for the immediate changes, and showed graphical accuracy.
GMM gave more detail in graphics, but was messy to work with and make it function. The results, partially due to faulty input, were all across the board. But it did allow us to create a more functional basis at this point to learn from failure.

Remarks:

Code runs under Python V3.6.x and Spyder V3.2.8