Sigurður Ágúst Jakobsson

# Project: Assignment 11 – K-means Clustering and Gaussian Mixture Models

## Section 1.6

Output 1\_6\_1.png from one fitting:

Chart, line chart

Description automatically generated

## Section 1.7

Output 1\_7\_1.png from one fitting:

Chart, line chart

Description automatically generated

## Section 1.8

Answer the following questions:

* According to the results from section 1.7, what is the best value of *k* according to the objective function?
* If we have *n* samples and we set *k*=*n*, what would happen?
* Is it a good strategy to set *k*=*n*? Why/Why not?

According to the result from section 1.7, the best value for k is 10 since this gives the lowest objective function.

If we have *n* samples and we set *k*=*n* we get one mean to calculate for every datapoint. This can be seen in this code snippet:

nn = sk.utils.shuffle(range(X\_standard.shape[0]))

    Mu = X\_standard[nn[0: k], :]

This would probably drive the objective function to zero which might seem good but isn’t really. This is severe overfitting and doesn’t help us solve any relevant tasks on the data later. Its like a lookup table on the data. It is not a good strategy.

Tested this and it actually does drive the objective function to zero.

## Section 1.10

Using the actual class labels of the samples and the predictions made by k\_means determine the accuracy and confusion matrix of the prediction.

You can use \_iris\_kmeans\_accuracy for this.

Output from one fitting:

Accuracy: 0.9533333333333334,

Confusion matrix:

[[50, 0, 0]

[ 0, 45, 5]

[ 0, 2, 48]]

Definition of columns/rows according to sklearn documentation since confusion matrices are not fully standardized as far as I remember: “By definition a confusion matrix C is such that C\_i,j is equal to the number of observations known to be in group i and predicted to be in group j.“

## Section 2.1

Plots 2\_1\_1.png-2\_1\_4.png in descending order:

Chart

Description automatically generated

2\_1\_1: num\_clusters = 2

Chart

Description automatically generated

2\_1\_2: num\_clusters = 5

Chart

Description automatically generated

2\_1\_3: num\_clusters = 10

Chart

Description automatically generated

2\_1\_4: num\_clusters = 20

## Section 3.1

Set up a [sklearn Gaussian mixture model](https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html" \o "https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html) (sklearn.mixture.Gaussian) and estimate:

* the mixing coefficients
* mean vectors
* covariance matrices

These values can be accessed on the Gaussian class through the properties .means\_, .covariances\_ and .weights\_.

for the Iris database. Choose *K*=3 components. What is the purpose of the mixing coefficients?

You can use \_gmm\_info for this.

Output from one run:

Mixing coefficients: [0.36548058 0.33333333 0.30118609]

Mean vectors:

[[6.54632887 2.94943079 5.4834877 1.98716063]

[5.006 3.428 1.462 0.246 ]

[5.91697517 2.77803998 4.20523542 1.29841561]]

Covariance matrices:

[[[0.38741443 0.09223101 0.30244612 0.06089936]

[0.09223101 0.11040631 0.08386768 0.0557538 ]

[0.30244612 0.08386768 0.32595958 0.07283247]

[0.06089936 0.0557538 0.07283247 0.08488025]]

[[0.121765 0.097232 0.016028 0.010124 ]

[0.097232 0.140817 0.011464 0.009112 ]

[0.016028 0.011464 0.029557 0.005948 ]

[0.010124 0.009112 0.005948 0.010885 ]]

[[0.27550587 0.09663458 0.18542939 0.05476915]

[0.09663458 0.09255531 0.09103836 0.04299877]

[0.18542939 0.09103836 0.20227635 0.0616792 ]

[0.05476915 0.04299877 0.0616792 0.03232217]]]

The purpose of the mixing coefficients is to indicate what proportion of data points are associated to each of the K clusters.

## Section 3.2

tools.plot\_gmm\_results() is a function that plots all samples and overlays the distribution of each gaussian in a GMM mixture. Try calling it with the Iris data. Read the documentation in the function for help.

Using \_plot\_gmm as a base you have to:

1. Fit the GMM to the data
2. Make a prediction on the data
3. Call the tools.plot\_gmm\_results() function.

Plot 3\_2\_1.png:

Chart, bubble chart

Description automatically generated

## Independent Section

We have focused mostly on clustering in this assignment. We tried changing the value of *K* but we mostly left num\_its fixed. We only ran a single experiment on the same photo. We only tried it on a single dataset. Maybe there are other types of clustering techniques.

Make an independent experiment of your choice and present your results and all added code with plots and figures is applicable.