HW-7 Report

We first do the imports,

Text

Description automatically generated

We copy the update\_centroids and memberships from the labs for the initial step.

I modified the update\_centroids so that it takes the pre defined means from the csv file instead of random assigning.

Graphical user interface, text, application

Description automatically generated

We import the data.

In the score\_func, we return the multivariate normal probability for the given point, mean and covariance and I multiply this with the prior value.

Text

Description automatically generated with medium confidence

I also have my implementation but I don’t use it.

Chart, scatter chart

Description automatically generated

We plot data points.

For the initial step,

Text

Description automatically generated

What we do is first assign members to closest of the means for each cluster. I also update the priors, covariances with np.cov and then update centroids once.

In the main iteration part,

Text

Description automatically generated

Here, I first update the memberships based on the maximum score for each of the clusters.

First, in the E part.

Centroids[0], covariances[0] etc. is the features of first cluster and for each score, I return the argmax that shows which cluster that is. Than in the for loop that iterates over all of the points, I store the values for each points and their posterior probability in the hik[i, k] where k is the cluster respectively.

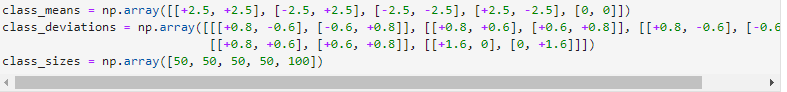
In the M part, we calculate means, priors and covariances as well.

Means, that is shown by centroids is the matrix multiplaction of the data matrix 300,2 with the hik matrix we have. Here, more posterior probability weighs more for the each class and each data point. Unlike the k-cluster algorithm where these were binary rather than continous.

For the priors, I just count the memberships than divide them to total. I also tried the;

[np.sum(hik[:, i]) for i in range(K)] / np.sum(hik) but this doesn’t produce good results for some reason.

For the covariances, I find them with the np.cov for each class in the closed for loop.



I defined the real values here for comparison purpose.

Lastly,

Text

Description automatically generated

I print the results as well as the real pdf dist for the points.

Diagram

Description automatically generated