

# Programming Assignment 3

## Manthan Thakar

*Note: More information about the code can be found in README.md file*

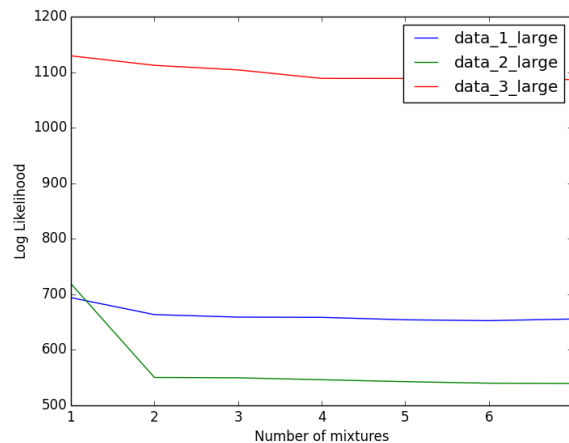
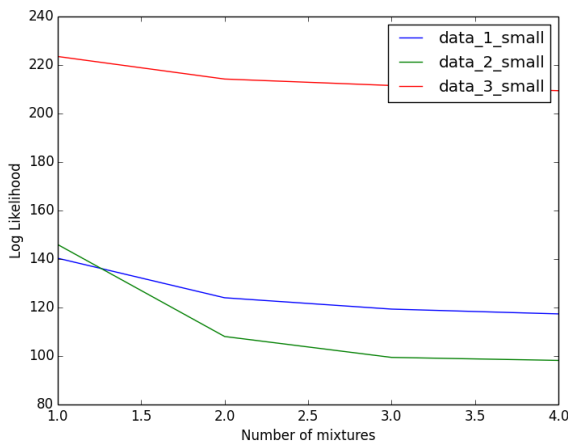
### 1. EM Algorithm

The implementation of EM algorithm for Gaussian Mixture Models can be found in *models.py* file.

#### Varying number of mixtures:

Following figures show the effect on log likelihood as the number of mixtures are varied on both small and big datasets. Note that the likelihood numbers are positive because it makes for better graph, but in calculations the likelihood values are negative (i.e. positive log likelihood function used). Moreover, for both plots the likelihood of the training dataset is plotted.

As we can see from the following plots, the likelihood drops steeply as the number of mixtures is increased to 2 from 1, but after that there's not a significant change in the likelihood. This is expected, since the using only one mixture to fit the data wouldn't be able to model the kind of data we have at hand.



#### Initial mixture parameters:

Initially the parameters for each mixture are set as following:

- **mu:** The value for mixture means are selected randomly
- **sigma:** The covariance matrix is chosen to be an identity matrix in the beginning
- **pi:** The weights for each mixture is assigned a uniform value =  $1/m$ , where  $m$  is the number of mixtures

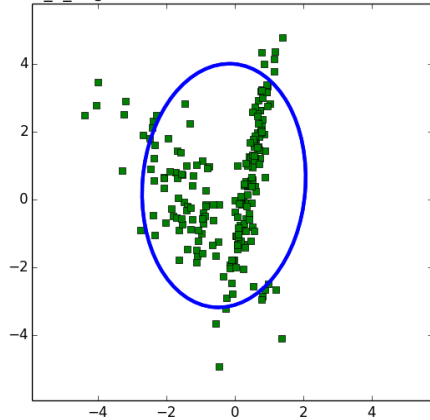
After multiple runs it was observed that when the value of  $\pi$  is chosen as random the algorithm is more likely to run into the problem of having infinity values in mean and covariance vectors. After assigning uniform values for  $\pi$ , the algorithm is less likely to run into that error.

#### Convergence parameter:

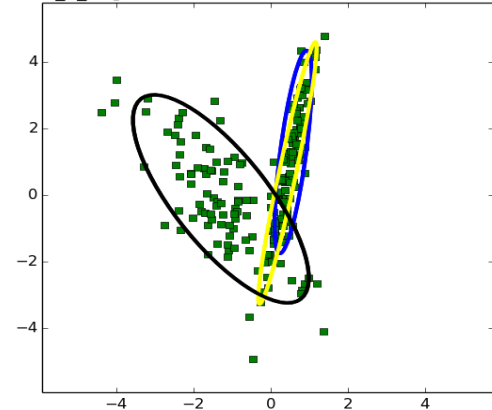
Two different convergence parameters are used in this implementation of EM. They are as follows:

- **Likelihood change threshold:** Likelihood change threshold is used to check the difference between previous and current likelihood and if it is less than the given threshold, the algorithm terminates. Using this measure, it was observed that most of the training sessions converged between 30 to 80 iterations.
- **Maximum Iterations:** If the number of iterations exceeds maximum iterations specified, the algorithm terminates. By default, this value is kept 100, since most of the datasets converge with respect to the likelihood values under 100 iterations.

data\_2\_large Likelihood=-719.194427072 mixtures=1



data\_2\_large Likelihood=-547.801425021 mixtures=3



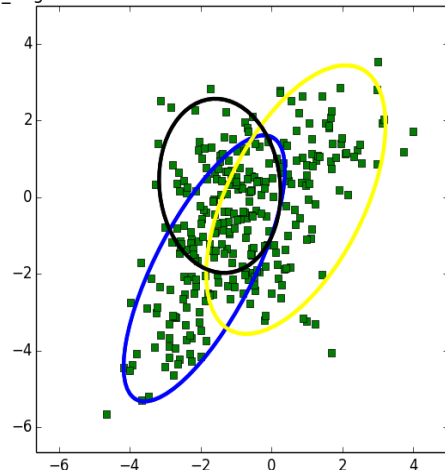
Plots above show the Gaussian mixtures fit by the EM algorithm implementation for data\_2\_large dataset with number of mixtures 1 and 3. It can be seen that the log likelihood for mixtures = 1 (-719.19) is smaller than mixtures = 3 log likelihood (-547.80).

## 2. Variations

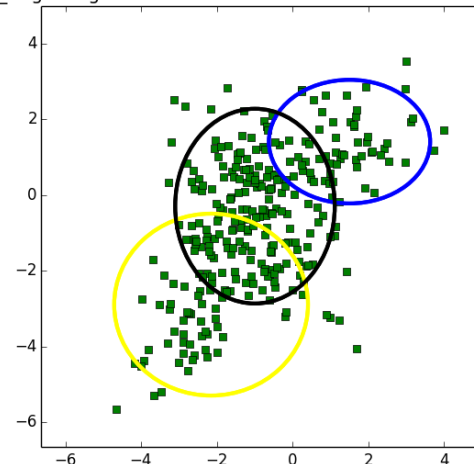
### 2.1 Diagonal covariance matrix

In this variant, only the diagonal values of the covariance matrices for Gaussians in the mixture are updated and the off-diagonal values are zero. Since, the diagonal of the covariance matrix corresponds to the variance of the features, the Gaussians fit by this variant only scale in the direction of the features i.e. horizontal and vertical in the case of 2D features. Whereas, the Gaussians with full covariance matrix fit Gaussian that are tilted. The difference can be seen visually in the plots below for the data\_3\_large and number of mixtures = 3

data\_3\_large full-covariance Likelihood=-1104.95958871 mixtures=3



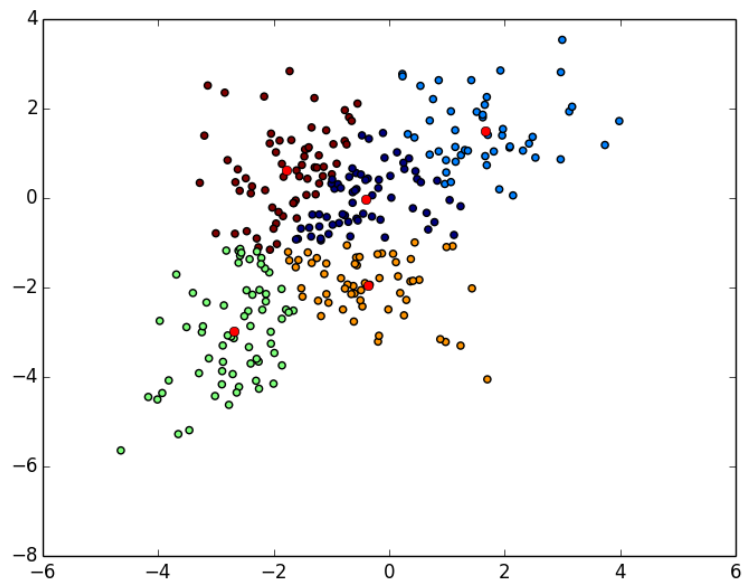
data\_3\_large diag-covariance Likelihood=-1113.05558966 mixtures=3



The plots above show the difference between full-covariance GMM (left) and diagonal-covariance GMM (right). The diagonal-covariance Gaussians are only stretched along x and y axis but they don't scale in both x and y direction together.

### 2.2 K-Means

The k-means algorithm is implemented and the final centroids obtained using k-means are used as initial means for mixtures in the Gaussian mixture model. The following plot shows the means obtained by running k-means algorithm on data\_3\_large.



**Changing number of components:**

### **3. Model Selection**

### **4. Predictions**