CS342 Coursework

Gaussian Mixture Modelling for Clustering

1903188

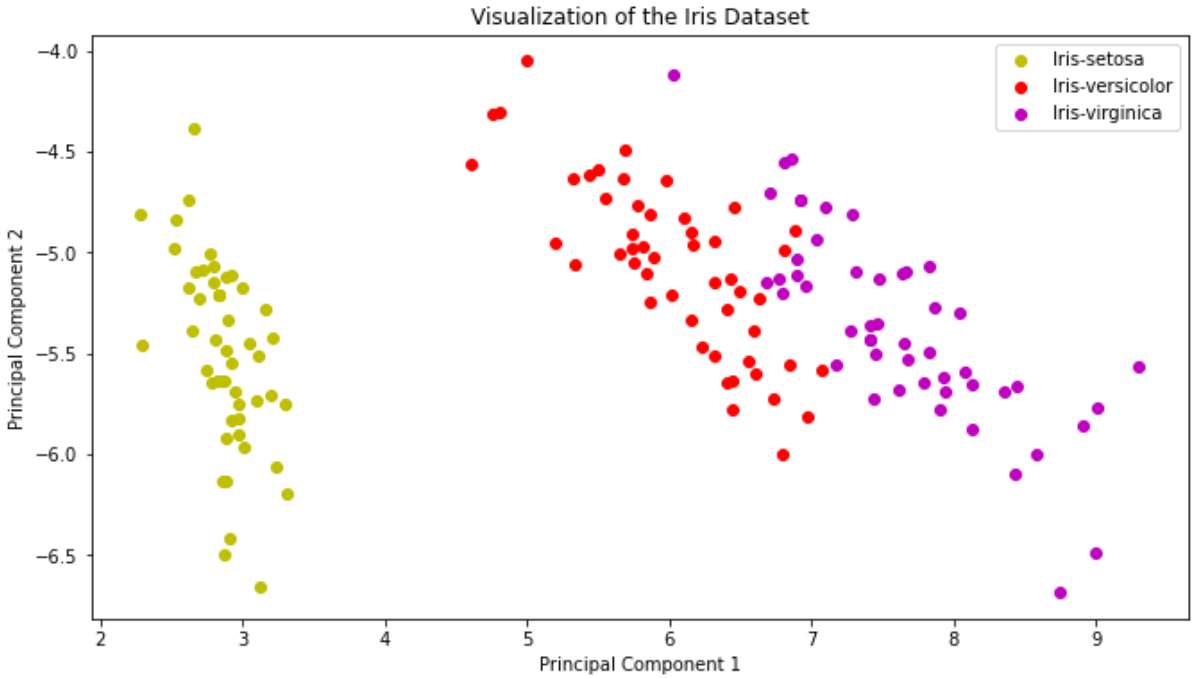
December 6, 2021

A. Data

In this coursework, we are looking at the Iris Dataset from the UCI Machine Learning Repository: `https://archive.ics.uci.edu/ml/datasets/iris`.

B. Data Visualization

We use PCA to project the data onto its top two principal components. This produces the scatter graph below, with the x axis as principal component 1, the y axis as principal component 2, and the points coloured according to their class.

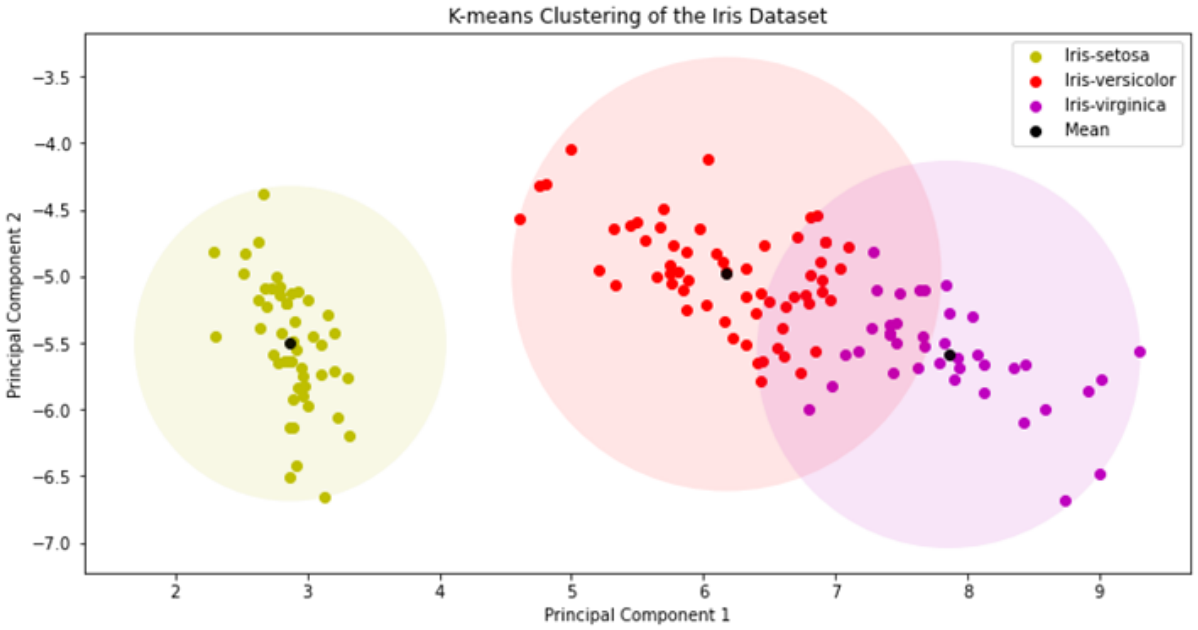


We can clearly see that there are three groups, with 'Iris-setosa' on the left being the most clearly defined cluster.

C. Clustering Using GMM and the EM Algorithm

We now use k-means to group the data into three clusters. The graph below shows the means computed and the shape of the k-means clusters.

We can see that the shape of the cluster generated is a circle, this is because k-means does not account for the variance of the points, only the mean. There are many points that have been incorrectly classified, especially in the overlapping region between 'Iris-versicolor' and 'Iris-virginica'. Using the completeness score, we have an accuracy of 88.67%. If we compute the mean squared error between the k-means prediction and the true labels we get 0.1133 (to 4 decimal places).



Now we can use the computed k-means prediction to get the initial values for the EM algorithm. The formulas used are as follows:

1. The cluster probabilities based on the n data points assigned to each cluster c

$$\pi_c^0 = \frac{1}{n}|x_i \in c|$$

2. The cluster centroid based on the means of the n data points assigned to each cluster c

$$\mu_c^0 = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

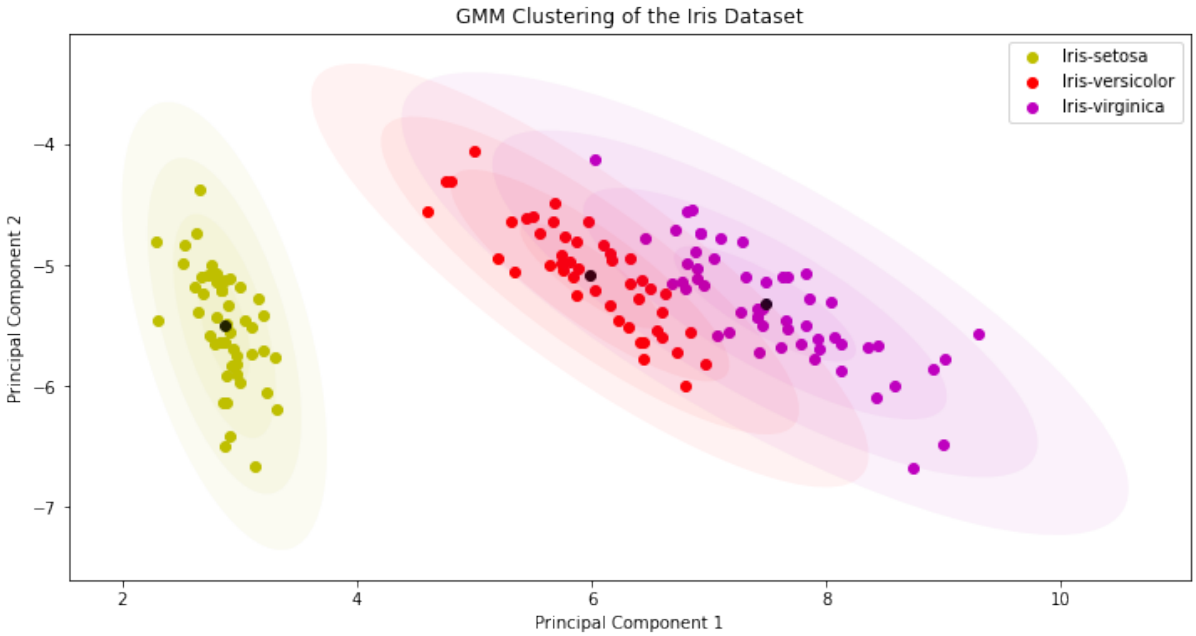
3. The cluster covariances based on the covariances of the n data-points assigned to each cluster c

$$\Sigma_c^0 = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mu_c^0)(\mathbf{x}_i - \mu_c^0)^T$$

These values are then used to calculate the initial responsibilities for the EM algorithm.

The EM algorithm converges when $\Theta^N = (\pi^N, \mu^N, \Sigma^N) = \Theta^{N-1}$. In python I computed convergence to 3 decimal places by comparing the values of the mean, covariance and pi in the M step to the last iteration and seeing if the values had changed. I found that after 75 iterations, the values stop changing (indicating convergence) and hence the algorithm performs 75 iterations.

Once the algorithm has finished running, the graph below demonstrates the result for the assignments, means and covariances. This gives a cluster shape that is an ellipse, not a circle as in k-means which more accurately depicts the true shape of the cluster.



We now have that the accuracy of this clustering is 97.33%. Computing the mean squared error between the GMM prediction and the true labels we get 0.0267 (to 4 decimal places) which is significantly less than that of k-means. Hence we can conclude that the EM-algorithm gives a more accurate result.