Implementation

The code is implemented entirely as a Jupyter notebook. The dependencies required are *numpy*, *six*, *matplotlib*.

The GMM code was implemented using a two-step EM algorithm, as described in lectures:

1. First, initialize every data point to be from a random cluster
2. Calculate the means and covariances of all clusters
3. Assign each data point to the cluster that has the highest probability of the data point
4. Repeat steps 2-3 for desired number of iterations.

The probabilities in step 3 are calculated using the usual multivariate Gaussian PDF formula. The formula requires dividing by the square root of the determinant of the covariance matrix. If the covariance matrix is found to have zero determinant, we add a small diagonal matrix to the covariance matrix repeatedly until the determinant is non-zero.

We also added a feature to detect when the model has converged. If the cluster assignments do not change after an iteration, then the model is considered to have converged, and the algorithm is stopped early.

Visual Analysis of the Data

For the data that is provided, we note that there are three species: Setosa, Virginica, Versicolor. From both the cross-section and the PCA plots, we are able to see that the Setosa species forms a very distinct cluster, whereas Virginica and Versicolor are much closer together.

Further observation of the Virginica and Versicolor clusters shows that both of these clusters do not resemble a Gaussian distribution. In other words, the centers of these clusters are not significantly denser than the edges of the clusters.

At the ending section of the notebook, there is a section on comparing Versicolor vs Virginica. We plot the scatterplot of all the data points in a single color, using both the cross-section projection and the PCA projection. Observing both of these scatterplots, we note that there is no clear distinction between Versicolor and Virginica. In fact, these scatterplots appear to reflect 2 clusters (Setosa and Virginica+Versicolor) rather than 3.

This suggests that a GMM may not be the best model to use for this task. Furthermore, GMMs simply perform clustering on unlabeled data. Without the labels, it is very difficult to distinguish the clear boundary between Virginica and Versicolor.

3 clusters

The first attempt gave very bad results, as can be seen from the graphs in the notebook. The cluster boundaries do not correspond at all to the actual boundaries between the real classes. In fact the shapes of the clusters were not visually observable.

To resolve this issue, we tried re-seeding the algorithm with different seeds in the second attempt. This changes the outcome of the algorithm as it affects the initial random assignment of the data points to different classes, which will change the overall clusters generated by the algorithm. After changing the seed a few times, we found a seed that gave satisfactory results that were similar to the correct answers.

Note that there was some confusion within the model regarding the Versicolor+Virginica cluster. This is understandable and explained in the earlier section, “Visual Analysis of the Data”.

Other Observations

We tried repeating the experiment with different numbers of clusters:

- With 2 clusters, the model kept Setosa as a distinct cluster, and grouped Versicolor and Virginica. This was no surprise.
- With 5 clusters and 10 clusters, the model’s output becomes very difficult to comprehend, and it does not seem to give a reasonable clustering of the data. Hence it still makes the most sense to use the correct number of clusters (3).

Another interesting observation is that the model converges much faster than expected, usually on the order of 10 iterations. It is likely that 150 data points across 4 dimensions is not a large enough dataset to justify a large number of iterations.