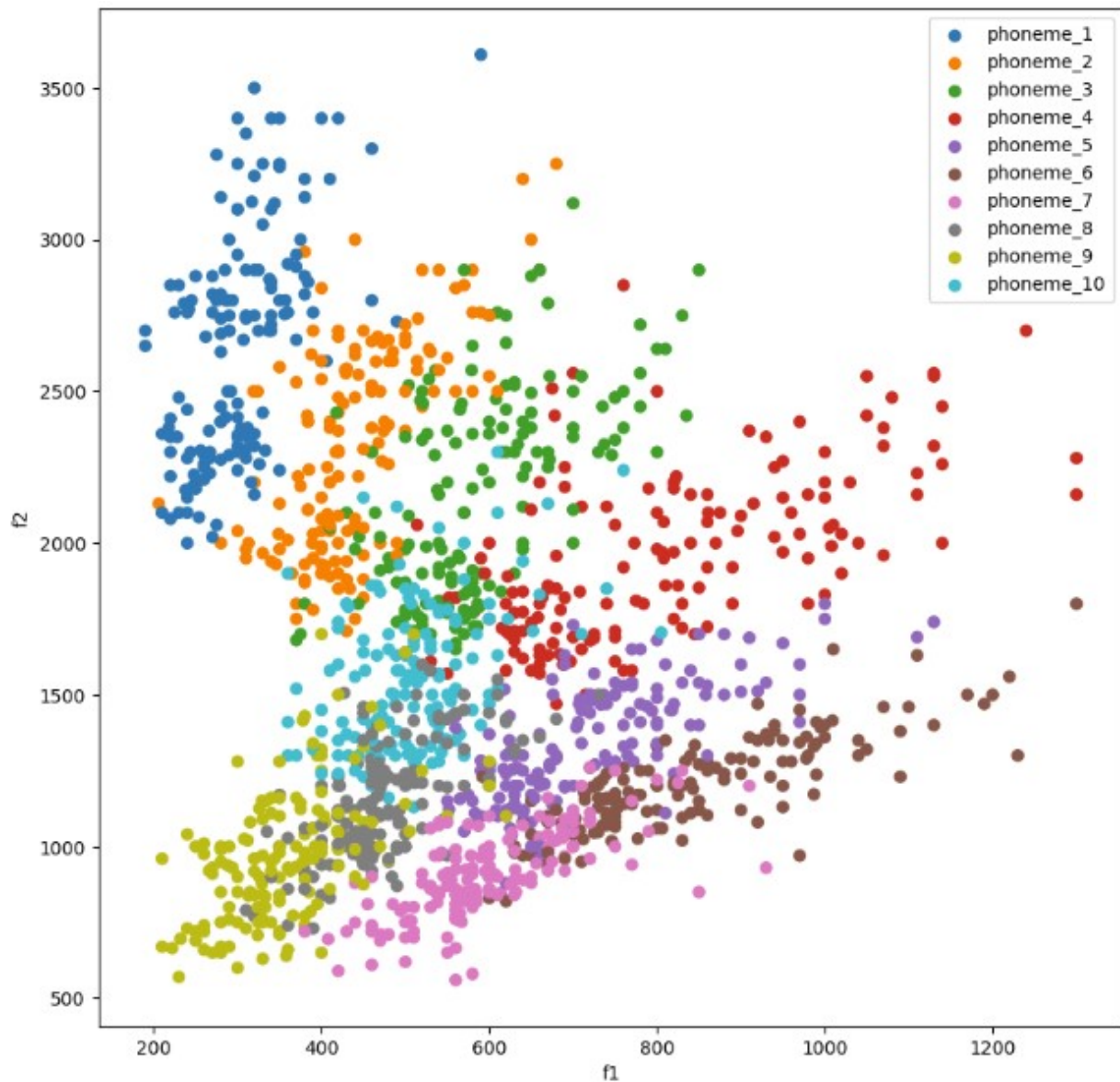


# **ECS708P/ECS708U Machine Learning**

## **Assignment 2**

### **Unsupervised Learning**

**Q1. Produce a plot of  $F1$  against  $F2$ .**



Looking the clustered data is not linearly separable. Also, there are some regions where the data is more dense than others.

Therefore, for such dense data we apply density estimation method for more clarity.

As observed the `phoneme 1` and `phoneme 2` are easy to separate. However, other phonemes are more compact and difficult to separate from the given dataset.

The `phoneme 8` is very close and hard to classify.

**Q2. Run the code multiple times for  $K = 3$ , what do you observe? Use figures and the printed MoG parameters to support your arguments**

Mixtures of Gaussians (MoG) is a probabilistic model that assumes that the data is generated from a mixture of  $K$  Gaussian distributions. The probability density function of the MoG is given by:

$$p(\mathbf{x}) = \sum_{k=1}^K \frac{p(c_k)}{(2\pi)^{D/2} \det \Sigma_k^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right)$$

where:

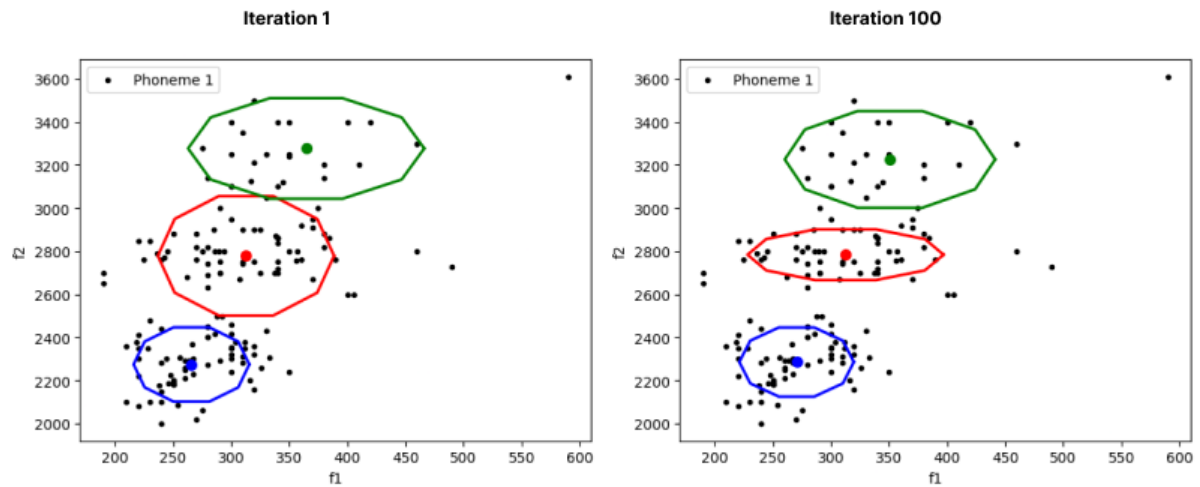
- $\boldsymbol{\mu}_k \in \mathbb{R}^D$
- $\Sigma_k \in \mathbb{R}^{D \times D}$
- $p(c_k) = \pi_k \in \mathbb{R}$

By using the EM algorithm, we can estimate the parameters of the MoG model. The EM algorithm is an iterative algorithm that alternates between two steps:

1. E-step: compute the posterior probability of each data point belonging to each Gaussian component.
2. M-step: update the parameters of the Gaussian components.

We need to calculate the  $\sigma$  and  $\mu$  for each phoneme, based on which we can calculate the probability of each phoneme.

## First Run



## Mu

```
[[ 323.24017 2903.5637 ]
 [ 244.30171 2231.018  ]
 [ 302.26315 2338.855  ]]
```

## Sigma

```
[[[ 4002.5369687  0.          ]
 [  0.          67340.65021549]]

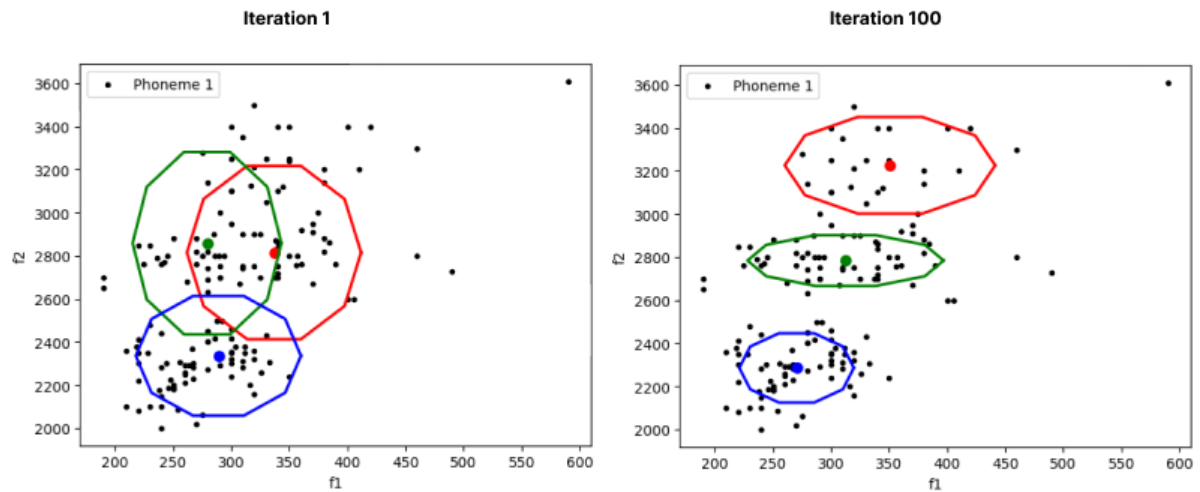
 [[ 356.12422529  0.          ]
 [  0.          13577.9242552  ]]

 [[ 365.77190917  0.          ]
 [  0.          6568.98351071]]]
```

Here, we use `k=3` to fit the data. We can clearly distinguish `Iteration 1` and `Iteration 100` as the cluster being separated.

However, the data is separated based on the density and still large number of data are out of the boundary.

## Second Run



## Mu

```
[[ 393.45273 1993.081 ]
 [ 586.56024 2814.1687 ]
 [ 449.67453 2519.6829 ]]
```

## Sigma

```
[[[ 2454.91694131 0. ]
 [ 0. 15263.41834733]]

 [[ 2111.61766908 0. ]
 [ 0. 49424.42134748]]

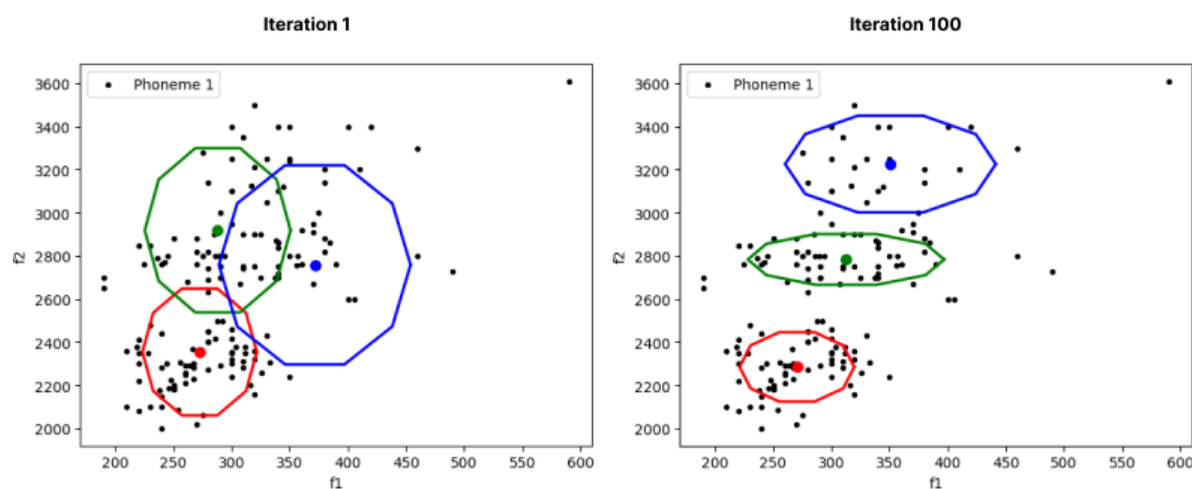
 [[ 2809.22467288 0. ]
 [ 0. 29720.65443662]]]
```

First run displays, we can observe that every data is overlapped, finally at the 100th iteration, we can observe that the data is separated based on the density.

From the values of  $\mu$  and  $\sigma$ , we can say that  $\mu$  and  $\sigma$  are different from the first run.

Therefore, the EM algorithm is not deterministic, and the result will be different every time we run the algorithm.

## Third Run



## Mu

```
[[ 585.5591 2810.4568 ]
 [ 393.40866 1992.6349 ]
 [ 449.19864 2518.5508 ]]
```

## Sigma

```
[[[ 2142.26272133 0. ]
 [ 0. 49412.61169736]]

 [[ 2457.28546025 0. ]
 [ 0. 15177.78395113]]

 [[ 2776.02551222 0. ]
 [ 0. 29823.92779232]]]
```

Comparing the previous two runs, we can see that the  $\mu$  and  $\sigma$  are the same with different order but the value is same. This means that the EM algorithm is able to cluster the data based on the density.

**Q5. Use the 2 MoGs ( $K = 3$ ) learnt in tasks 2 & 3 to build a classifier to discriminate between phonemes 1 and 2, and explain the process in the report**

**Step A.**

We saved the  $\mu$  and  $\sigma$  for each phoneme in the previous task in the `.npy` file, which is kind of like a dictionary. We can use those values just like pickle using `np.load()`.

**Step B.**

The code snippet given above does the following steps:

1. Initially, copy relevant sample ( $p\_id = 1$  or  $2$ ) to  $X$
2. Get the size of the dataset ( $N$ )
3. Initialize  $K$  (change  $k$  here for the whole program)
4. Loads the GMM 1 parameters to  $\mu\_1$ ,  $s\_1$ , and  $p\_1$
5. Gets the prediction and stores it in  $z\_1$
6. Sums the rows of the prediction to get the total probability for the data point and stores in  $sum\_1$
7. Repeat the points 4 to 6 for GMM 2 and gets  $sum\_2$
8. Compares  $sum\_1$  and  $sum\_2$ . The  $p\_id = 1$  if  $sum\_1 > sum\_2$  else  $p\_id = 2$
9. Calculate the accuracy by comparing prediction array with  $y$  array.

With Maximum Likelihood Estimation(MLE) we calculate the probability of each phoneme, if the probability of phoneme 1 is higher than phoneme 2, then we classify the data as phoneme 1, otherwise, we classify the data as phoneme 2. The results obtained are given below:

**Step C**

**Calculate the accuracy**

Accuracy by using GMM with 3 components = 96.38157894736842

Accuracy by using GMM with 6 components = 95.39473684210526

**Q6. Repeat for  $K = 6$  and compare the results in terms of accuracy.**

We are using `k=6` to fit the data, and applied the same method as the previous task.

**Calculated Accuracy**

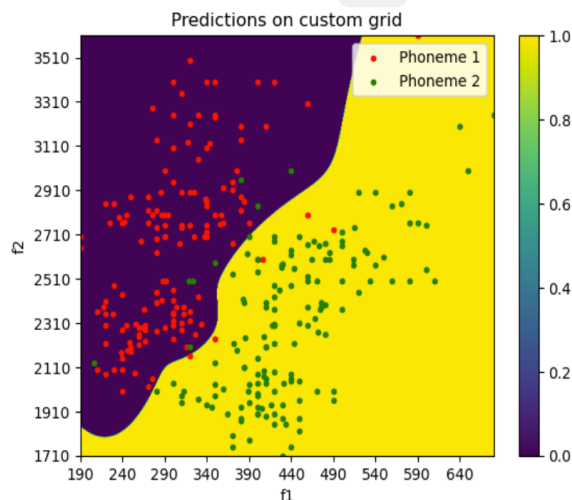
k	accuracy
3	96.38%
6	95.39%

The accuracy of GMM with `k=3` components is higher than GMM with `k= 6` components.

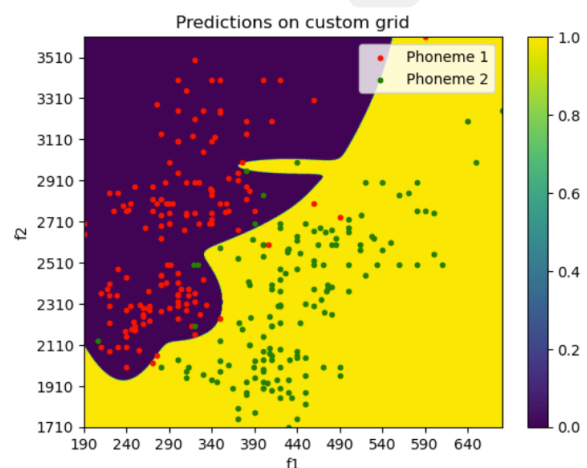
With this observation we can state, when `k= 6` GMM component is prone to overfitting the data points. Hence, the accuracy of the dataset will decrease over time.

**Q7. Display a "classification matrix" assigning labels to a grid of all combinations of the  $F1$  and  $F2$  features for the  $K = 3$  classifiers from above. Next, repeat this step for  $K = 6$  and compare the two.**

Classification Matrix for `k=3`



Classification Matrix for `k=6`



In the above graph, we implemented both `k=3` and `k=6` to classify the data, but the custom grid is created based on the  $N_{f1} * N_{f2}$  shape of the data. As illustrated, the data for the both graph is well separated, and the data is not overlapped.

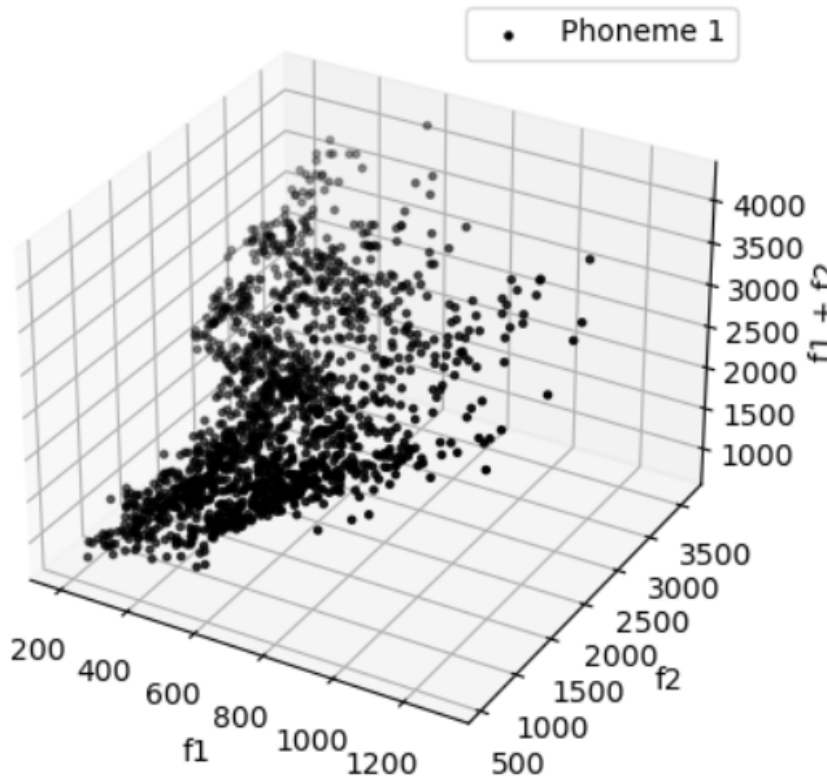


**Q8. Try to fit a MoG model to the new data. What is the problem that you observe? Explain why it occurs**

Create the new column by combining the `F1` and `F2` column.

$$X = [F1, F2, F1 + F2]$$

`ValueError: Input contains infinity or a value too large for dtype('float64').`



**Singularity issues in Gaussian mixture model**

1. Singularity in a likelihood function is when a component collapse on a data point. It is a form of over fitting.
2. The variance becomes zero which in this case leads to singular covariance matrix. When the variance is zero, the likelihood of the component becomes infinity and hence the model over fits.

We are try to use the following formula to calculate the probability of each Gaussian component.

$$\frac{p_i}{\sqrt{2\pi^D |\Sigma_i|}} \exp -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)$$

The reason we get this error is determinant of the covariance matrix is zero, and we are dividing by zero, the covariance matrix is not invertible, and the determinant is zero.

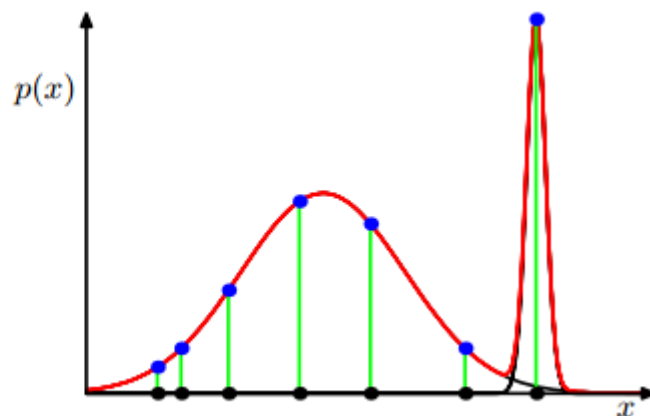
**Q9. Suggest ways of overcoming the singularity problem and implement one of them. Show any training outputs in the report and discuss. [3 marks]**

**Singularity Problem**

If we want to fit a Gaussian to a single data point using maximum likelihood, we will get a very spiky Gaussian that "collapses" to that point. The variance is zero when there's only one point, which in the multi-variate Gaussian case, leads to a singular covariance matrix, so it's called the singularity problem.

When the variance gets to zero, the likelihood of the Gaussian component (formula 9.15) goes to infinity and the model becomes overfitted. This doesn't occur when we fit only one Gaussian to a number of points since the variance can not be zero. But it can happen when we have a mixture of Gaussians, as illustrated on the same page of PRML.

Illustration of how singularities in the likelihood function arise with mixtures of Gaussians. This should be compared with the case of a single Gaussian shown in Figure 1.14 for which no singularities arise.



Reference: <https://stats.stackexchange.com/a/219358>