Gaussian Mixture Model

Subham Roy
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260

1 Overview

The Gaussian Mixture Model or Mixture of Gaussian as it is called is not such a model as it is a Probability Distribution. It is a universally used model for clustering. It is also called Expectation-Maximization Clustering or EM Clustering and is based on the optimization strategy. Gaussian Mixture models are used for representing Normally Distributed subpopulations within an overall population. Let us do a deeper dive into the concept of Gaussian Mixture Models.

2 About Gaussian Mixture Model

The Gaussian mixture model can also be thought of as a prototype method, similar in spirit to K-means. Each cluster is described in terms of a Gaussian density, which has a centroid (as in K-means), and a covariance matrix. The comparison becomes crisper if we restrict the component Gaussians to have a scalar covariance matrix. The two steps of the alternating EM clustering are very similar to the two steps in K-means:

- In the E-step, each observation is assigned a responsibility or weight for each cluster, based on the likelihood of each of the corresponding Gaussians. Observations close to the center of a cluster will most likely get weight 1 for that cluster, and weight 0 for every other cluster. Observations halfway between two clusters divide their weight accordingly.
- In the M-step, each observation contributes to the weighted means (and covariances) for every cluster.

Therefore, the Gaussian mixture model is often referred to as a soft clustering method, while K-means is hard.

Similarly, when Gaussian mixture models are used to represent the feature density in each class, it produces smooth posterior probabilities $\hat{p}(x) = \{\hat{p}_1(x), \ldots, \hat{p}_K(x)\}$ for classifying x. Often this is interpreted as a soft classification, while in fact, the classification rule is $\hat{G}(x) = \arg\max_k \hat{p}_k(x)$.

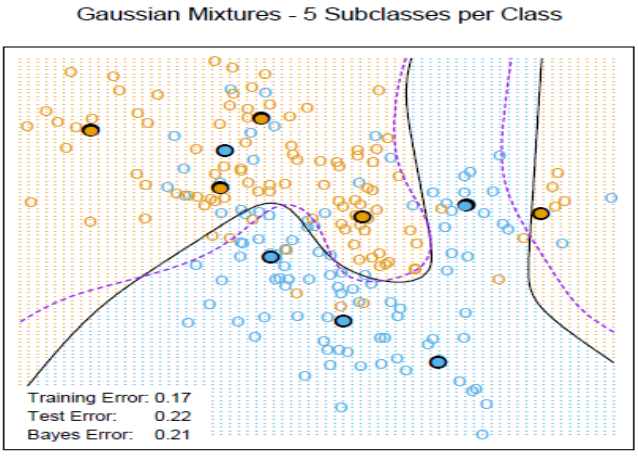

Fig: A gaussian mixture model with a common covariance for all component Gaussians.

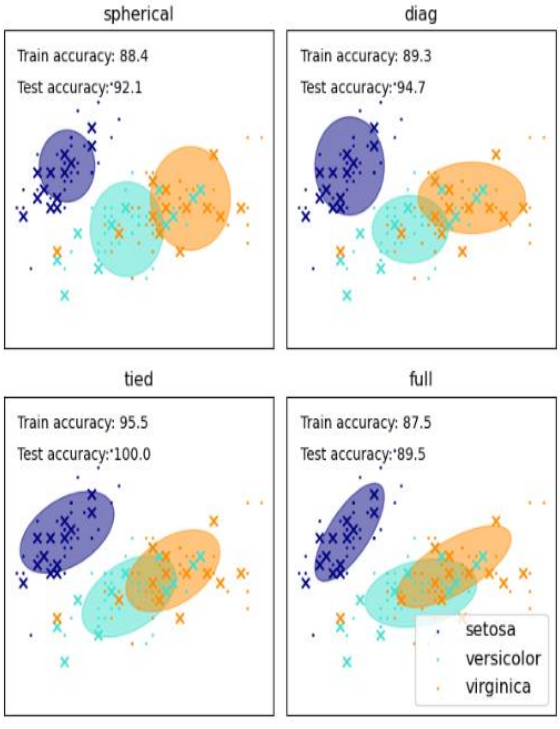

Fig: Types of GMM Options

3 Experiment

In this experiment, I used the IRIS dataset. Initially, we start with loading the Dataset and checking the dimensions and the number of variables within the Dataset. In this case, the dataset has 5 features and 150 rows.

Dataset description –

	Sepal.Length	Sepal.Width	Petal_Length	Petal_Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Fig: Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Sepal.Length  150 non-null    float64
 1   Sepal.Width   150 non-null    float64
 2   Petal_Length  150 non-null    float64
 3   Petal_Width   150 non-null    float64
 4   Species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Fig: Dataset Summary – Column Datatypes

The next step is we perform data preprocessing. We perform the following steps before we can be certain that we can apply the classifier model. This is almost similar to all the techniques we apply. First, we check for missing data, then check for the spread of the data means where the Mean, Median of the data, are there any outliers, we take care of that.

In this case study, the dataset had no Null values or outliers.

```
1 data["Species"].value_counts()
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

We took a count of all records and grouped them w.r.t. Species. We see an even distribution of the count i.e.; each Species has 50 records.

.

Now, we plot a pair-wise plot to observe the correlations among the features. We find that the species are nearly linearly separable with petal size, but sepal sizes are more mixed.

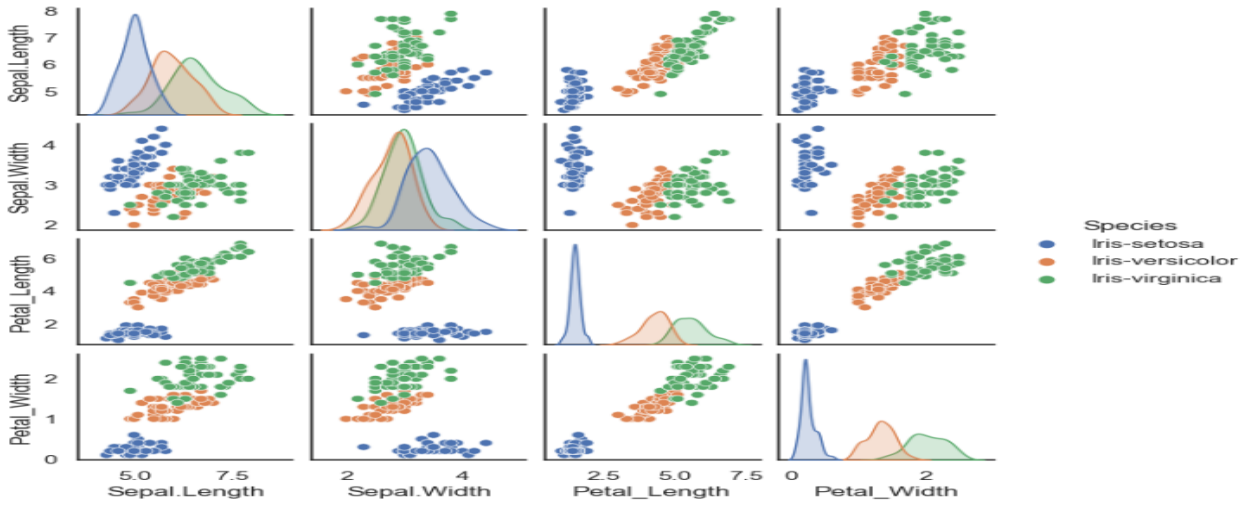

Fig: Pair Plot for all features

After this, we split the data, with the Species column as the response variable and the rest columns as the features. We use StandardScalar to scale the data to avoid in complications and implement the first Principal Component Analysis (PCA) to identify principal components from the dataset. My objective to perform PCA was to reduce the dimensions of the dataset to 2D out of the 4 features provided.

	PC1	PC2
131	2.316082	2.626184
11	-2.327378	0.158587
112	1.884252	0.414333
116	1.471280	0.253192
6	-2.445711	0.074563

Fig: PCA results

In this case, we are not losing much information due to feature reduction as we are reducing mainly the Dimensions to 2 from 4. Here the PCA mashes all information from 4 features to PC1 and PC2.

After this, we implement the GaussianMixture model from sklearn.mixture library on the dataset.

4 Analysis

We find out firstly, the model accuracy for on Test dataset is 90%, which is good considering the data had 2 Species data that were overlapping with each other. General observation for GMM is that it handles ellipsoidal distributions, and makes soft assignments to clusters, but is slower than K-means for large datasets.

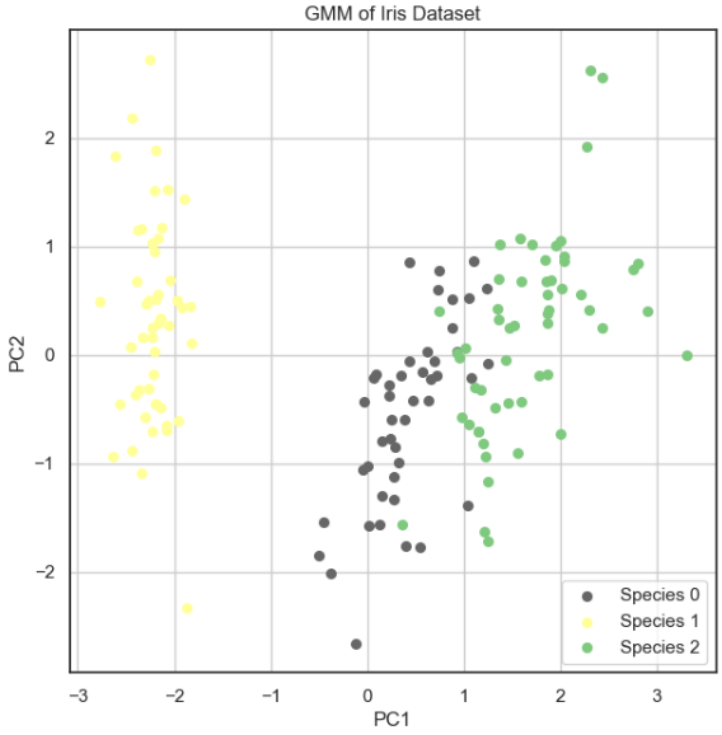

Fig: Gaussian Mixture Cluster

References

[1] Gareth James [at], Daniela Witten [aut], Trevor Hastie [aut, cre], Rob Tibshirani [aut], Balasubramanian Narasimhan [ctb], *Introduction to Statistical Learning, Second Edition*.

[2] Tom M. Mitchell, *Machine Learning: A multistrategy approach, 1997 Edition*.

[3] Iris Data: URL - http://archive.ics.uci.edu/ml/datasets/Iris