Project report

Gaussian Mixture Models: Bag of Words Representation

NAME: HARI KRISHNAN

COURSE: AI and ML

Question:

Using a gaussian mixture model, perform a simple clustering on the given 2D Dataset. Try to find the optimal number of clusters using python (you may use any module to implement this). Now implement the same from scratch using python and a dummy dataset generated using scikit learn dataset generating functions such as make blob.

Prerequisites

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-asan-internal-or-externalcommand/ . Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6

Dataset Link: Clustering_GMM

https://cdn.analyticsvidhya.com/wp-content/uploads/2019/10/Clustering_gmm.csv

Method used :

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. It attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset allowing the model to learn automatically, i.e. in an unsupervised manner. The bag-of-words model is a way of representing text data when modelling text with machine learning algorithms which can be combined with GMM to get a useful model representation.

Load all the required libraries

Launcher | GMM Dhivakar.ipynb

Markdown

Gaussian Mixture Models: Bag of Words Representation

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.mixture import GaussianMixture
from sklearn.cluster import KMeans
```

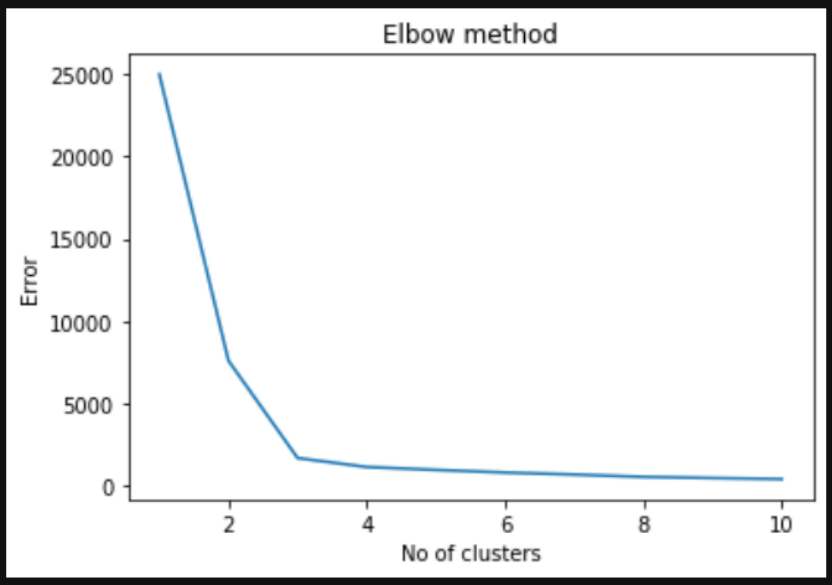
[3]:
```
data = pd.read_csv('Clustering_gmm.csv')
data.head()
```

[3]:

	Weight	Height
0	67.062924	176.086355
1	68.804094	178.388669
2	60.930863	170.284496
3	59.733843	168.691992
4	65.431230	173.763679

Elbow Method:

[4]:
```
Error =[]
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i).fit(data)
    kmeans.fit(data)
    Error.append(kmeans.inertia_)
import matplotlib.pyplot as plt
plt.plot(range(1, 11), Error)
plt.title('Elbow method')
plt.xlabel('No of clusters')
plt.ylabel('Error')
plt.show()
```



Model Building Using GMM on Clustering Gmm dataset

[5]:
```
gm = GaussianMixture(n_components=2, random_state=0).fit(data)
```

[6]:
```
gm.means_
```

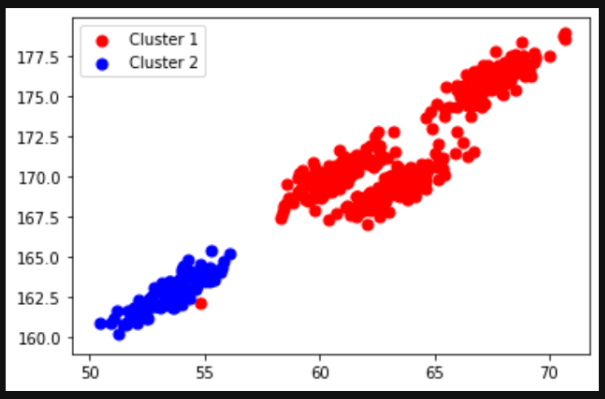
[6]:
```
array([[ 63.77281821, 171.71722858],
       [ 53.57474006, 162.74626605]])
```

[7]:
```
y = gm.predict(data)
```

[8]:
```
x = np.array(data)
plt.scatter(x[y== 0, 0], x[y == 0, 1], s = 50, c = 'red', label = 'Cluster 1')
plt.scatter(x[y == 1, 0], x[y == 1, 1], s = 50, c = 'blue', label = 'Cluster 2')


plt.legend()
```

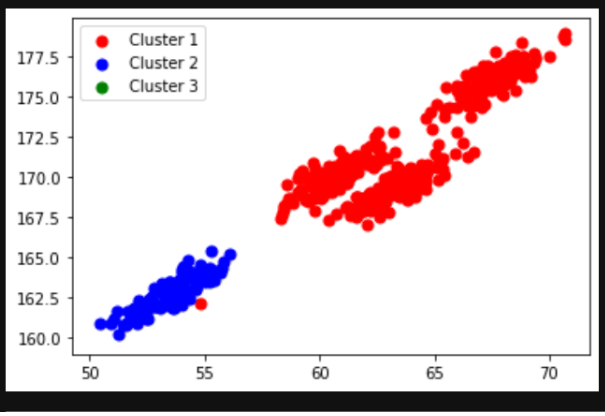
[8]:
```
<matplotlib.legend.Legend at 0x19de03889c8>
```



[9]:
```
gm = GaussianMixture(n_components=2, random_state=0).fit(data)
y1 = gm.predict(data)
x = np.array(data)
plt.scatter(x[y1== 0, 0], x[y1 == 0, 1], s = 50, c = 'red', label = 'Cluster 1')
plt.scatter(x[y1 == 1, 0], x[y1 == 1, 1], s = 50, c = 'blue', label = 'Cluster 2')
plt.scatter(x[y1 == 2, 0], x[y1 == 2, 1], s = 50, c = 'green', label = 'Cluster 3')

plt.legend()
```

[9]:
```
<matplotlib.legend.Legend at 0x19de0405c88>
```

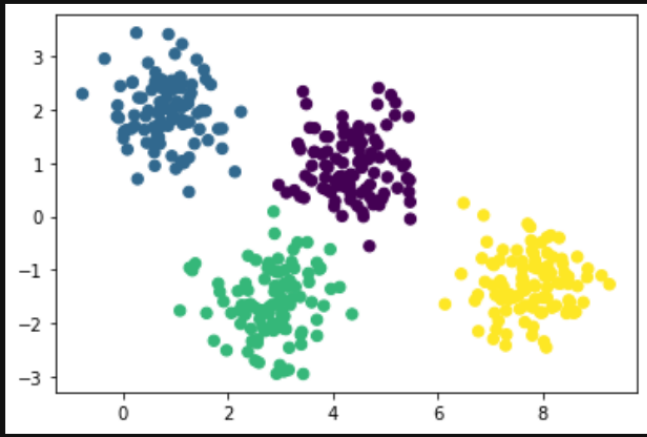


Importing Libraries and Making dummy dataset

[10]:
```
from sklearn.datasets import make_blobs
```

[11]:
```
X, y_true = make_blobs(n_samples=400, centers=4,
                       cluster_std=0.60, random_state=0)
X = X[:, ::-1] # flip axes for better plotting
```

[12]:
```
plt.scatter(X[:, 0], X[:, 1], c=y_true, s=40, cmap='viridis')
plt.show()
```



[13]:
```
from sklearn.mixture import GaussianMixture as GMM
gmm = GMM(n_components=4).fit(X)
labels = gmm.predict(X)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=40, cmap='viridis')
plt.show()
```