Project Report 7

NAME: DHIVAKAR.R

COURSE: AI and ML

Project Based On Spherical K-Means: Pattern Discovery in Textures

Question:

Generate a dummy dataset using Scikit-Learn having high dimensionality (number of features >10) and total 4 classes. For this dataset, first implement K-Means clustering and then use the clusters for classification purpose. Now using the same dataset, implement spherical clustering and then check accuracy for classification. Notice the change in accuracy. You may also plot the obtained clusters from both the methods using t-SNE plots or by projecting data into two dimensions using PCA.

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

Importing library and dataset:

```
import

[2]: from sklearn.datasets import make_blobs
     from sklearn.cluster import KMeans
     from sklearn.manifold import TSNE
     from sklearn.decomposition import PCA
     from coclust.clustering import SphericalKmeans
     import numpy as np
     import matplotlib.pyplot as plt
```

Dataset

```
dataset
```

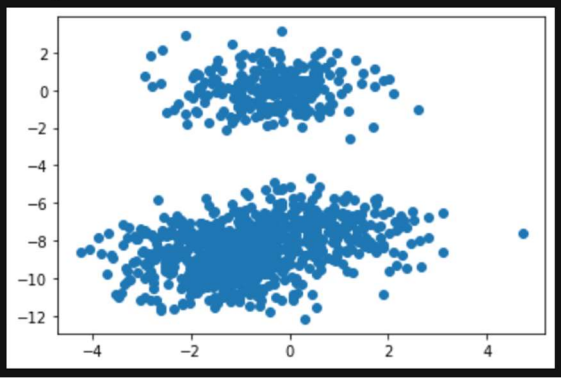
```
[3]: X, y = make_blobs(n_samples=1000, n_features=11, centers=4, random_state=2)
```

```
[4]: X.shape
```

```
[4]: (1000, 11)
```

```
[5]: plt.scatter(X[:, 0], X[:, 1])
```

```
[5]: <matplotlib.collections.PathCollection at 0x1afa907b788>
```



K-Means

```
K Means
```

```
k = KMeans(n_clusters=4, max_iter=100, random_state=62)
```

```
k.fit(X)
```

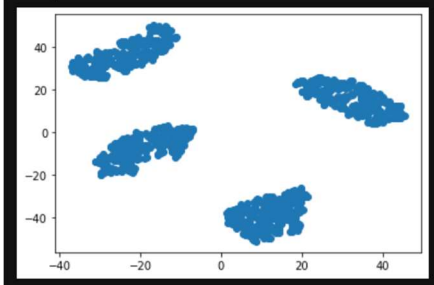
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=100,
       n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=62, tol=0.0001, verbose=0)
```

```
np.count_nonzero((k.predict(X) == y) == True) / len(y)
```

```
0.25
```

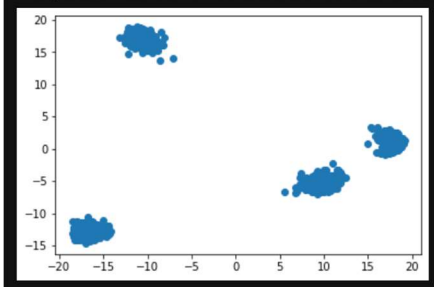
```
X_em = TSNE(n_components=2).fit_transform(k.transform(X))
plt.scatter(X_em[:, 0], X_em[:, 1])
```

```
<matplotlib.collections.PathCollection at 0x1afaa62dbc8>
```




```
X_gm = PCA(n_components=2).fit_transform(k.transform(X))
plt.scatter(X_gm[:, 0], X_gm[:, 1])
```

```
<matplotlib.collections.PathCollection at 0x1afaa69a2c8>
```



SphericalKmeans

Spherical Kmeans

SphericalKmeans

[11]:
```
s = SphericalKmeans(n_clusters=4, max_iter=100, random_state=69, weighting=True)
```

[12]:
```
s.fit(X)
```

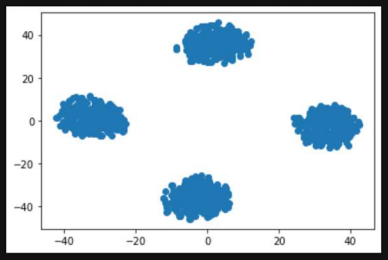
```
 == New init ==
iteration: 0
608.2550533838042
iteration: 1
900.4134546898862
iteration: 2
939.6889312641603
iteration: 3
942.6096521785471
iteration: 4
```

[13]:
```
np.count_nonzero((s.labels_ == y) == True) / len(y)
```

[13]:
```
0.5
```

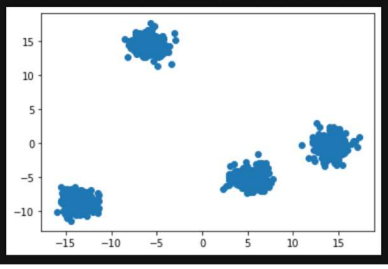
[14]:
```
X_em = TSNE(n_components=2).fit_transform(X)
plt.scatter(X_em[:, 0], X_em[:, 1])
```

[14]:
```
<matplotlib.collections.PathCollection at 0x1afaa76bb48>
```



[15]:
```
X_gm = PCA(n_components=2).fit_transform(X)
plt.scatter(X_gm[:, 0], X_gm[:, 1])
```

[15]:
```
<matplotlib.collections.PathCollection at 0x1afaa7ecf48>
```



[16]:
```
X_em.shape
```

[16]:
```
(1000, 2)
```