PAGE NO.

DATE:

Lab Program-8

→ Apply EM algorithm to cluster a set of data stored in .csv file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. Add Python ML library classes/API in the program.

Theory

Algorithm k-means (K, D)-

1- Identify the K-data points as the initial centroids
2 Repeat step 1
3 For each data point x ∈ D do
4 Compute the distance from x to the centroid
5 Assign x to the closest centroid
6 Re-compute the centroids using the current cluster memberships until the stopping criterion is met.

EM algorithm for Gaussian Mixtures.

1 (Estimation E): Calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij}, assuming that the current hypothesis $h = \langle \mu_1, \mu_2 \ldots, \mu_k \rangle$ holds.

2 (Maximization M): Calculate a new maximum likelihood hypothesis $h'' = \langle \mu_1'', \mu_2'', \ldots \mu_k'' \rangle$, assuming the value taken on by each hidden variable z_{ij} is its expected

Teacher's Remarks

85

Teacher's Signature

PAGE NO.

DATE:

value $E[z_{ij}]$ calculated in step1. Then replace the hypothesis $h = \langle \mu_1, \mu_2 \ldots \mu_k \rangle$ by the new hypothesis $h' = \langle \mu_1'', \mu_2'', \ldots \mu_k'' \rangle$ and iterate.

Program-

```
from sklearn.cluster import kMeans
from sklearn.mixture import GaussianMixture.
import numpy as np
import matplotlib.pyplot as plt
X = np.genfromtxt('clusterdata.csv', delimiter=',',
                  skip_header=1)
X = X[:, 1:]
f1 = X[:, 0]
f2 = X[:, 1]
colors = ['b', 'g', 'r']
markers = ['o', 'v', 's']
# data
plt.subplot(511)
plt.xlim([0, 100])
plt.ylim([0, 50])
plt.title('Dataset')
plt.ylabel('speeding_feature')
plt.xlabel('distance_feature')
plt.scatter(f1, f2)
```

PAGE NO.

DATE:

```
#kmeans
plt.subplot(5,1,3)
plt.xlim([0,100])
plt.ylim([0,50])
model = KMeans(n_clusters=3)
model.fit(X)
plt.title('Kmeans')
plt.ylabel('speeding-feature')
plt.xlabel('distance-feature')
labels = model.predict(X)
for i,l in enumerate(labels):
    plt.plot(f1[i], f2[i], color=colors[l],
             marker=markers[l])

#gaussian
plt.subplot(5,1,5)
plt.xlim([0,100])
plt.ylim([0,50])
model = GaussianMixture(n_components=3)
model.fit(X)
plt.title('Gaussian Mixture')
plt.ylabel('speeding-feature')
plt.xlabel('distance-feature')
labels = model.predict(X)
for i,l in enumerate(labels):
    plt.plot(f1[i], f2[i], color=colors[l],
             marker=markers[l])
plt.show()
```

Teacher's Remarks

89

Teacher's Signature

Shubhransh Gupta : 1BG17CS094