Machine Learning
Lab : 6
KMeans Clustering

January 2025

Perform Date: January 13-18, 2025

1 Objective

1.1 Implement K Means Clustering Algorithm on the given dataset

2 Description

K-Means Clustering is an unsupervised learning algorithm that is used to solve clustering problems in machine learning or data science. A cluster refers to a collection of data points aggregated together because of certain similarities. We will define a target number k, which refers to the number of centroids you need in the dataset. A centroid is an imaginary or real location representing the center of the cluster. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares. In other words, the K-means algorithm identifies the k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.

3 Implementation Guidelines

3.1 Part A

[35]:
```
# Imports
from sklearn.datasets import make_blobs
X, _ = make_blobs(n_samples=100, centers=3, n_features=2,
cluster_std=0.2, random_state=0)
```

[1]:
```
# Scatter plot of the data points
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
# Using scikit-learn to perform K-Means clustering
from sklearn.cluster import KMeans
# Specify the number of clusters (3) and fit the data X
```

[ ]:
```
# Get the cluster centroids
```

[ ]:
```
# Get the cluster labels
```

[4]:
```
# Plotting the cluster centers and the data points on a 2D plane
```

[6]:
```
# Calculate silhouette_score
```

[8]:
```
# Import the KElbowVisualizer method

# Instantiate a scikit-learn K-Means model

# Instantiate the KElbowVisualizer with the number of clusters and the metric
# Fit the data and visualize
```

3.2 Part B

Hand Written Digit Recognition

[44]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

from sklearn.cluster import KMeans
from sklearn.datasets import load_digits
#digits dataset from scikit learn consists of 8x8 pixel images of digits

#Data plotting and visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns

from scipy.stats import mode
from sklearn.metrics import accuracy_score, confusion_matrix
```

[45]:
```
digits = load_digits() #load the dataset in digits
```

[45]:
```
(1797, 64)
```

[29]:
```
digits.keys() #Dataset loaded is a dictionary
# data : flattened arrays/tensors used for clustering
# target : label associated with flattened array
#print(digits.target)
```

[29]:
```
dict_keys(['data', 'target', 'frame', 'feature_names', 'target_names', 'images',
'DESCR'])
```

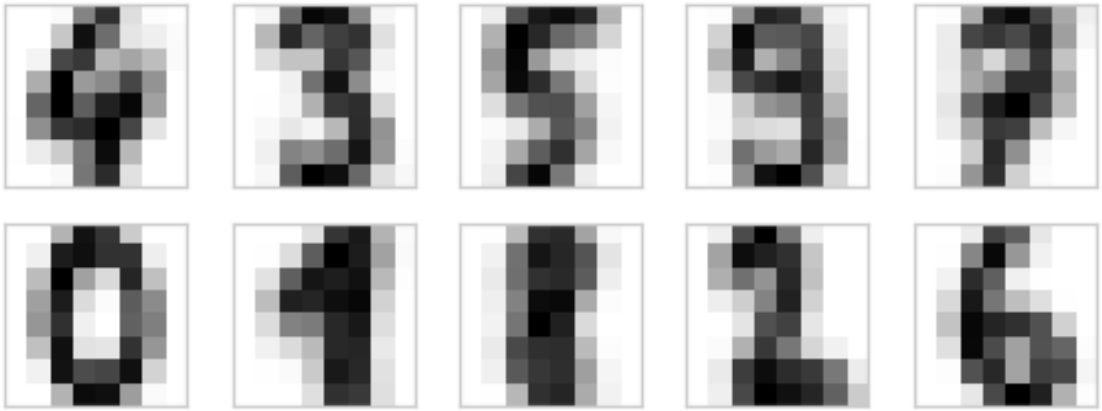
[30]:
```
digits.data[0:3]  #flattened data for 3 images of the dataset
```

[30]:
```
array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
        15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
        12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
         0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
        10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.],
       [ 0.,  0.,  0., 12., 13.,  5.,  0.,  0.,  0.,  0.,  0., 11., 16.,
         9.,  0.,  0.,  0.,  0.,  3., 15., 16.,  6.,  0.,  0.,  0.,  7.,
        15., 16., 16.,  2.,  0.,  0.,  0.,  0.,  1., 16., 16.,  3.,  0.,
         0.,  0.,  0.,  1., 16., 16.,  6.,  0.,  0.,  0.,  0.,  1., 16.,
        16.,  6.,  0.,  0.,  0.,  0.,  0., 11., 16., 10.,  0.,  0.],
       [ 0.,  0.,  0.,  4., 15., 12.,  0.,  0.,  0.,  0.,  3., 16., 15.,
        14.,  0.,  0.,  0.,  0.,  8., 13.,  8., 16.,  0.,  0.,  0.,  0.,
         1.,  6., 15., 11.,  0.,  0.,  0.,  1.,  8., 13., 15.,  1.,  0.,
         0.,  0.,  9., 16., 16.,  5.,  0.,  0.,  0.,  0.,  3., 13., 16.,
        16., 11.,  5.,  0.,  0.,  0.,  0.,  3., 11., 16.,  9.,  0.]])
```

[31]:
```
#run KMeans clustering on digits.data for 1797 records and 64 features
```

[31]:
```
(10, 64)
```

[32]:
```
fig, ax = plt.subplots(2, 5, figsize = (8,3)) #Create a figure and a set of␣
 ↪subplots( 2 rows and 5 columns)
centers = k_means.cluster_centers_.reshape(10,8,8)
#flattened image can't be viewed, re-transform/reshape/inverse transform it to␣
 ↪original form to view matrix shaped image
#reshape 10 rows of clusters (k_means.cluster_centers_ = 10,64) and 64 to 8 * 8␣
 ↪matrix
for axi, center in zip(ax.flat, centers): #ax.flat:flattening the image &␣
 ↪plotting relevant centers
    axi.set(xticks = [], yticks = [])
    axi.imshow(center, interpolation='nearest',cmap = plt.cm.binary)␣
 ↪#imshow(matplotlib method) to render the image in notebook
```



[33]:
```
labels = np.zeros_like(clusters) # blank labels
print(f"The labels are : {labels}")
print(f"\nThe size of labels is : {labels.shape}")
print("The mask values are : ")
for i in range(10):
    mask = (clusters == i)
    #if a specific digit belongs to/equivalent a specific cluster then its True␣
 ↪else False
    print(mask)
    labels[mask] = mode(digits.target[mask])[0]
```

```
The labels are : [0 0 0 ... 0 0 0]

The size of labels is : (1797,)
The mask values are :
[False False False ... False False False]
[False False False ... False False False]
[False False False ... False False False]
[False False False ... False  True  True]
[False False False ... False False False]
[ True False False ... False False False]
[False False False ... False False False]
[False  True  True ...  True False False]
[False False False ... False False False]
[False False False ... False False False]
```

[10]:
```
#if a specific digit belongs to/equivalent a specific cluster then acuracy is 1␣
 ↪else 0
```

4 Exercise

1. What is the accuracy, precision, and recall of the model trained on Hand Written Digit Recognition dataset?
2. Describe the dataset used in this lab exercise.
3. Give insights into the model trained for Hand Written Digit Recognition dataset.
4. Use the k-means algorithm in python to cluster the following 8 examples into 3 clusters: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).
   (a) Suppose that the centers of each cluster are A1, A4 and A7. Run the k-means algorithm for 3 epochs only. At the end of this epoch show:
      i. The new clusters (i.e. the examples belonging to each cluster)(mention the appropriate attribute used to identify the clusters in sklearn)
      ii. The centers of the new clusters (mention the appropriate attribute used to identify the cluster centers in sklearn)

5. Apply Elbow Method on Part A and Part B. How many clusters will you choose according to elbow method?
6. Write the silhouette score for model trained in Part A and Part B using sklearn.

5 Reference

1. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
2. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
3. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html